Representation, Segmentation and Matching of 3D Visual Shapes using Graph Laplacian and Heat-Kernel

Avinash Sharma

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **IMAGERIE, VISION ET ROBOTIQUE**

Arrêté ministériel :

Présentée par

Avinash Sharma

Thèse dirigée par **Prof. Radu Horaud**

préparée au sein **Laboratoire Jean Kuntzmann (LJK) -INRIA Rhône-Alpes**
et de **Mathématiques, Sciences et Technologies de l'Information, Informatique**

Representation, Segmentation and Matching of 3D Visual Shapes using Graph Laplacian and Heat-Kernel

Thèse soutenue publiquement le **29 Octobre 2012**,
devant le jury composé de :

Dr. Edmond Boyer
INRIA Grenoble, France, Président
Dr. Bruno Lévy
INRIA Nancy, France, Rapporteur
Prof. Adrian Hilton
University of Surrey, UK, Rapporteur
Dr. Michael Wand
MPI Saarbruecken, Germany, Examinateur
Dr. Radu Horaud
INRIA Grenoble, France, Examinateur

Abstract

3D shape analysis is an extremely active research topic in both computer graphics and computer vision. In computer vision, 3D shape acquisition and modeling are generally the result of complex data processing and data analysis methods. There are many practical situations where a visual shape is modeled by a point cloud observed with a variety of 2D and 3D sensors. Unlike the graphical data, the sensory data are not, in the general case, uniformly distributed across the surfaces of the observed objects and they are often corrupted by sensor noise, outliers, surface properties (scattering, specularities, color, etc.), self occlusions, varying lighting conditions. Moreover, the same object that is observed by different sensors, from slightly different viewpoints, or at different time instances may yield completely different point distributions, noise levels and, most notably, topological differences, e.g., merging of hands.

In this thesis we outline single and multi-scale representation of articulated 3D shapes and devise new shape analysis methods, keeping in mind the challenges posed by visual shape data. In particular, we discuss in detail the heat diffusion framework for multi-scale shape representation and propose solutions for shape segmentation and dense shape registration using the spectral graph methods and various other machine learning algorithms, namely, the Gaussian Mixture Model (GMM) and the Expectation Maximization (EM).

We first introduce the mathematical background on differential geometry and graph isomorphism followed by the introduction of pose-invariant spectral embedding representation of 3D articulated shapes. Next we present a novel unsupervised method for visual shape segmentation by analyzing the Laplacian eigenvectors. We then outline a semi-supervised solution for shape segmentation based upon a new learn, align and transfer paradigm. Next we extend the shape representation to a multi-scale setup by outlining the heat-kernel framework. Finally, we present a topologically-robust dense shape matching method using the multi-scale heat kernel representation and conclude with a detailed discussion and future direction of work.

Key-Words: Computer Vision, Articulated 3D Shape Analysis, Visual Shapes, Heat Kernel, Dense Matching, Unsupervised Shape Segmentation, Semi-Supervised Shape Segmentation.

Résumé de la thèse

Analyse de la forme 3D est un sujet de recherche extrêmement actif dans les deux l'infographie et vision par ordinateur. Dans la vision par ordinateur, l'acquisition de formes et de modélisation 3D sont généralement le résultat du traitement des données complexes et des méthodes d'analyse de données. Il existe de nombreuses situations concrètes où une forme visuelle est modélisé par un nuage de points observés avec une variété de capteurs 2D et 3D. Contrairement aux données graphiques, les données sensorielles ne sont pas, dans le cas général, uniformément répartie sur toute la surface des objets observés et ils sont souvent corrompus par le bruit du capteur, les valeurs aberrantes, les propriétés de surface (diffusion, spécularités, couleur, etc), l'auto occlusions, les conditions d'éclairage variables. Par ailleurs, le mÃªme objet que l'on observe par différents capteurs, à partir de points de vue légèrement différents, ou à des moments différents cas peuvent donner la répartition des points tout à fait différentes, des niveaux de bruit et, plus particulièrement, les différences topologiques, par exemple, la fusion des mains.

Dans cette thèse, nous présentons une représentation de multi-échelle des formes articulés et concevoir de nouvelles méthodes d'analyse de forme, en gardant à l'esprit les défis posés par les données de forme visuelle. En particulier, nous analysons en détail le cadre de diffusion de chaleur pour représentation multi-échelle de formes 3D et proposer des solutions pour la segmentation et d'enregistrement en utilisant les méthodes spectrales graphique et divers algorithmes d'apprentissage automatique, à savoir, le modèle de mélange gaussien (GMM) et le Espérance-Maximisation (EM).

Nous présentons d'abord l'arrière-plan mathématique sur la géométrie différentielle et l'isomorphisme graphique suivie par l'introduction de la représentation spectrale de formes 3D articulés. Ensuite, nous présentons une nouvelle méthode non supervisée pour la segmentation de la forme 3D par l'analyse des vecteurs propres Laplacien de graphe. Nous décrivons ensuite une solution semi-supervisé pour la segmentation de forme basée sur un nouveau paradigme d'apprendre, d'aligner et de transférer. Ensuite, nous étendre la représentation de forme 3D à une configuration multi-échelle en décrivant le noyau de la chaleur cadre. Enfin, nous présentons une méthode d'appariement dense grâce à la représentation multi-échelle de la chaleur du noyau qui peut gérer les changements topologiques dans des formes visuelles et de conclure par une discussion détaillée et l'orientation future des travaux.

Mots-clés: Vision par ordinateur, Analyse de forme 3D articulé, Formes visuelles, Noyau de la chaleur, Appariement dense, Segmentation non supervisée, Segmentation semi-supervisée.

Publications from Thesis:

- **3D Shape Registration Using Spectral Graph Embedding and Probabilistic Matching**, *Image Processing and Analysis with Graphs: Theory and Practice, CRC Press - 2012. (Editors: Olivier Lézoray and Leo Grady)*
  Avinash Sharma, Radu P. Horaud and Diana Mateus
- **3D Visual-Shape Analysis Using the Heat Kernel**, *International Journal of Computer Vision* **(IJCV)**, *under submission.*
  Radu P. Horaud, Avinash Sharma and Dirk Smeets
- **Topologically-Robust 3D Shape Matching Based on Diffusion Geometry and Seed Growing**, *Computer Vision and Pattern Recognition* **(CVPR)**, *2011, Colorado Springs, USA.*
  Avinash Sharma, Jan Cech, Radu P. Horaud and Edmond Boyer
- **Learning Shape Segmentation Using Constrained Spectral Clustering and Probabilistic Label Transfer**, *European Conference on Computer Vision* **(ECCV)**, *2010, Greece.*
  Avinash Sharma, Etienne Von Lavante and Radu P. Horaud
- **Shape matching based on diffusion embedding and on mutual isometric consistency**, *Non-Rigid Shape Analysis and Deformable Image Alignment* **(CVPR)** , *2010, San Francisco, USA.*
  Avinash Sharma and Radu P. Horaud
- **Mesh Segmentation Using Laplacian Eigenvectors and Gaussian Mixtures**, *AAAI Fall Symposium on Manifold Learning and its Applications, November 2009, Arlington, USA.*
  Avinash Sharma, Radu P. Horaud, David Knossow and Etienne Von Lavante
- **Inexact Matching of Large and Sparse Graphs Using Laplacian Eigenvectors**, *Graph-based Representations in Patterns Recognition, May 2009, Venice, ITALY*
  David Knossow, Avinash Sharma, Diana Mateus and Radu P. Horaud

Contents

CHAPTER 1

Introduction

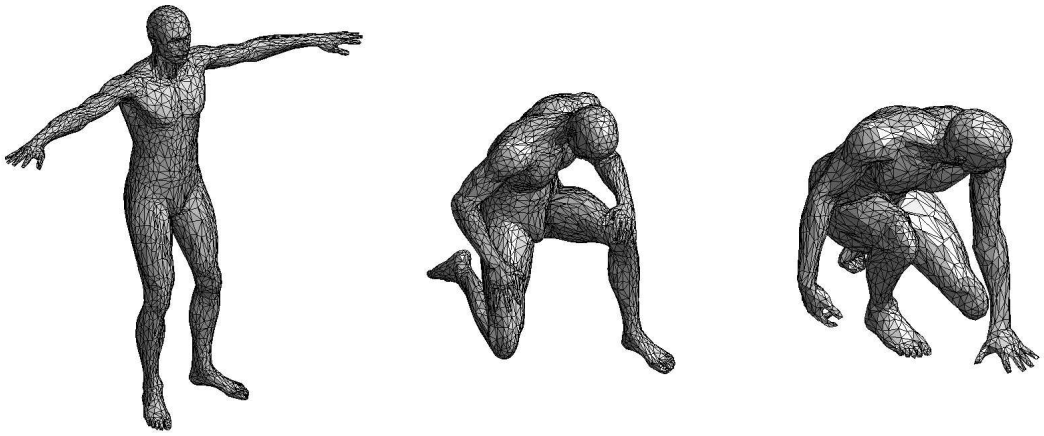
Contents

1.1 Motivation

3D shape analysis has been an extremely active research topic for the last couple of decades. With the increasing progress of 3D capture and display consumer technology, *e.g.*, gaming devices like Kinnect, 3D-television and films, medical scanners, etc. Thus, 3D shape analysis is getting more relevant to the real world applications.

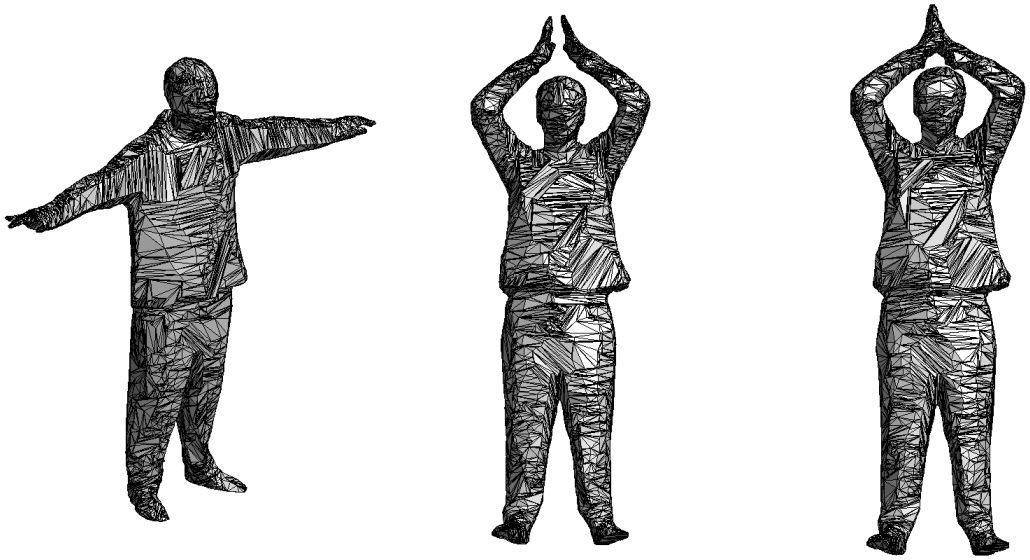
A widely used 3D shape representation is based on *polygonal meshes*: Each mesh vertex corresponds to a 3D point on shape surface and each mesh edge encodes the local connectivity between two vertices. In domains such as computer graphics or geometric design, it is common to assume that such a mesh corresponds to a discrete representation of a smooth, continuous, and closed surface – a Riemannian manifold, as shown in the Figure 1.1. In this case, *i.e.*, *graphical shapes*, the *Delaunay triangulation* seems to be the representation of choice for its nice mathematical properties. In particular, this representation allows the construction of Laplace operators on meshes, which correspond to various discretizations of the Laplace-Beltrami operator on Riemannian manifolds. Mesh Laplacians are extremely powerful mathematical tools because their spectral properties allow the characterization of both the local and global geometric properties of the underlying shape from *within*.

(a) High-resolution Delaunay triangulation of closed (compact and without boundaries) surfaces

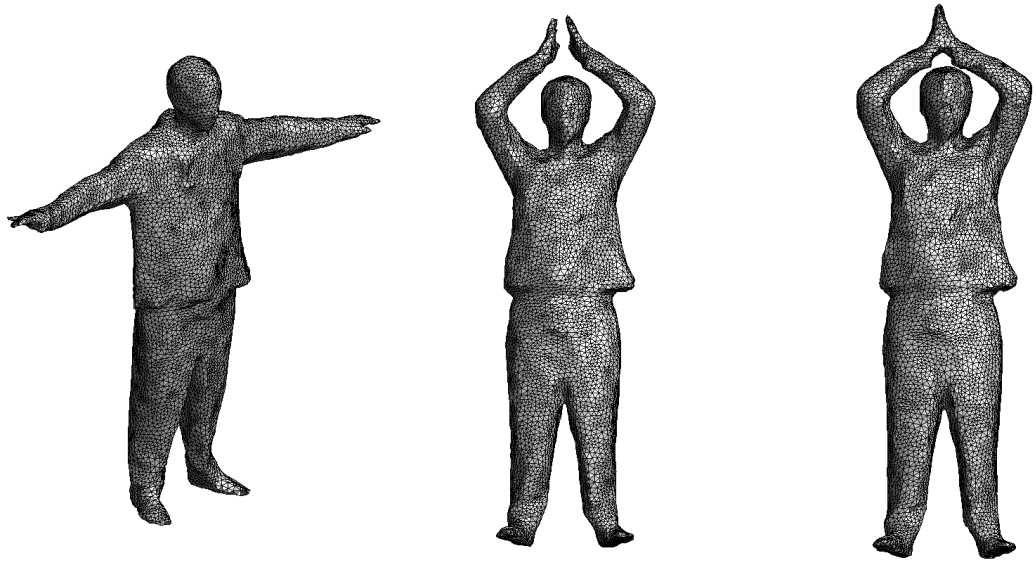


(b) Low-resolution Delaunay triangulation of the same surfaces

Figure 1.1: Triangulated meshes of three *graphical shapes*. These are discrete representations of smooth and closed surfaces, i.e, without boundaries, holes, self-intersections, etc.



(a) Low-resolution triangulated meshes gathered with a multiple-camera system.



(b) High-resolution triangulated meshes after mesh processing

Figure 1.2: Triangulated meshes of three *visual shapes*. (a) "Raw" low-resolution meshes obtained from [Franco 2009]. (b) "Processed" high-resolution meshes obtained from [Zaharescu 2011]. Notice that, unlike the graphical meshes of Figure 1.1, there are surface merges, which yield complex topological changes as the observed shapes deform over time.

In computer vision, 3D shape acquisition and modeling are generally the result of complex data processing and data analysis methods. There are many practical situations where a *visual shape* is modeled by a *set of points* observed with a variety of 2D and 3D sensors, *e.g.*, multiple cameras, time-of-flight devices, structured-light/camera systems, laser range-finders, etc. The data sets gathered with these sensors are, in general, not uniformly distributed across the surfaces of the observed objects and they are often corrupted by noise, outliers, self occlusions, and varying lighting conditions. Moreover, the same object that is observed by different sensors, from slightly different viewpoints, or at different time instances may yield completely different topologies, *e.g.*, holes. Figure 1.2 shows visual shapes typically used in computer vision. Figure 1.2a shows three examples of visual shapes captured with a multi-camera setup [Franco 2009]. Notice how an articulated object undergoes important changes over time, which cause large discrepancies in their topological structure. While mesh processing, as seen in Figure 1.2b, smoothes the "raw" mesh and removes self intersections between triangles [Zaharescu 2011], surface merges caused by temporal deformations remain present in the shapes' discrete representations.

Consequently, the task of extracting a triangulated mesh from 3D visual data is far from being an obvious one. Moreover, it is not always possible to guarantee that the meshes gathered with visual sensors satisfy the properties of a Delaunay triangulation [Boissonnat 1998]. Thus, it is not straightforward to adapt and to apply to visual shapes the geometric processing algorithms specifically designed to deal with graphical shapes. There is both a need to relax the theoretical conditions that the input data must satisfy and to devise a shape analysis methodology that can accommodate flaws in the data. This motivates us to design new shape analysis algorithms that can handle challenges posed by visual data.

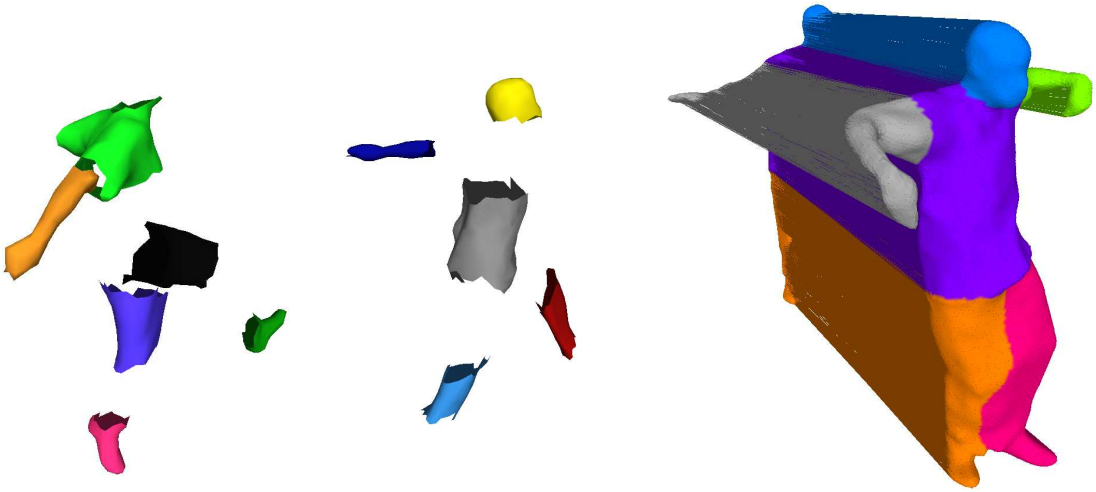
1.2 Thesis Overview

This thesis deals with 3D shape analysis tasks for real articulated 3D shapes also known as the *visual shapes*, captured mainly with multi-camera acquisition systems. In general, we outline single and multi-scale representation for visual shapes and propose new shape analysis methods, keeping in mind the challenges posed by visual shape data. In particular, we discuss in detail the heat diffusion framework for multi-scale shape representation, shedding new light on the relation between scale of analysis and the dimension of shape embedding. We also propose solutions for shape segmentation and dense shape registration using the spectral graph methods and some popular machine learning algorithms, *namely,* Gaussian Mixture Model (GMM) and Expectation Maximization (EM).

1.2.1 3D Shape Analysis

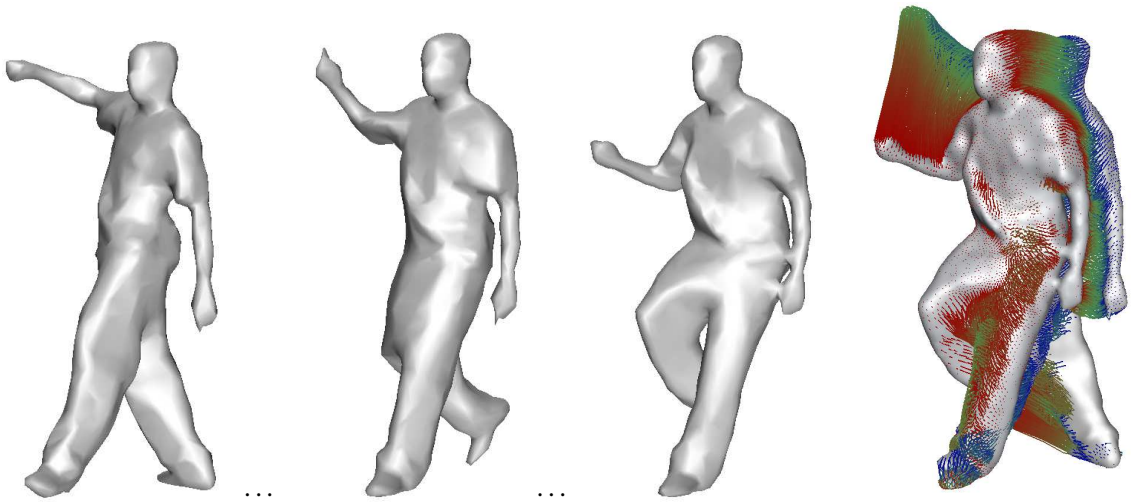
Recent innovations in the field of 3D geometry capture have led to more affordable, manageable and portable acquisition systems. This has generated considerable amount of interest in

3D shape analysis from both the industrial as well as academia. The task of 3D shape analysis involves tracking, segmentation, recognition, registration, animation transfer, motion synthesis, etc. Figure 1.3 illustrates three important shape analysis tasks. In this thesis, we mainly



(a) Segmentation

(b) Registration/Matching



(c) Temporal Tracking [Varanasi 2008]

Figure 1.3: Illustration of three important shape analysis tasks in computer vision. Segmentation and matching results in (a,b) are obtained using the unsupervised segmentation and matching methods proposed in this thesis.

focus on unsupervised and semi-supervised shape segmentation and sparse and dense shape registration/matching.

1.2.2 Input Data

Recent advancement in shape acquisition technology has led to the capture of large amounts of 3D data. Existing real-time multi-camera 3D acquisition methods provide a frame-wise reliable visual-hull or mesh representations for real 3D animation sequences [Franco 2009, Starck 2009, Slabaugh 2001, Seitz 2006, Vlasic 2008, Zaharescu 2011, Stoll 2010]. Analyzing 3D data in a single framework is still a challenging task considering the large variability of the data gathered with different acquisition devices. This variation arises due to: 1) variation in the shape acquisition techniques, 2) local deformations in non-rigid shapes, 3) large acquisition discrepancies (*e.g.*, holes, topology change, surface acquisition noise), 4) local/global scale change.

We have considered 3D data obtained from multiple sources. This includes synthetic data [Bronstein 2010a] and real data captured with multi-camera systems [Franco 2009, Starck 2007b, Stoll 2010] as well as with laser scanner [Vlasic 2008]. The main issue with shape segmentation and matching tasks is the lack of ground-truth for real data. Specifically, the existing ground-truth data for shape matching is too simple to validate the methods that are designed to handle challenging transformations in the real data. In this work, we have also prepared some manual ground-truth data to validate our shape analysis results.

1.2.3 Spectral Representation

In computer vision, graph theory, and machine learning there is a long tradition of representing meshes and graphs using matrices, in particular graph Laplacian. The advantage of such an algebraic approach is that it allows to describe and analyze manifold data, and more generally relational data, using the eigenvalues and eigenvectors of a semi-definite positive symmetric matrix. We embed input 3D shapes that were originally represented as triangulated mesh graphs into a multi-dimensional spectral space spanned by the eigenvectors of the respective graph Laplacian matrix. Thus, the spectral embedding of a 3D shapes is an isometry invariant (multi-dimensional) point-cloud representation, which makes it invariant to articulated poses. Hence, two shapes with different poses should have similar spectral representations, provided the underlying topology of two mesh graphs is similar.

1.2.4 Unsupervised Shape Segmentation

We devise an unsupervised probabilistic method, based on Gaussian mixtures with model selection, which reveals the structure of each eigenvector of graph Laplacian matrix. This enables us to select a subset of eigenvectors among the smallest eigenvectors of a graph, to embed the graph in the space spanned by this selection, and to cluster the graph's vertices using a multidimensional Gaussian Mixture Model (*GMM*) with model selection. When applied to shape graphs corresponding to articulated objects, such as humans, the proposed method segments the latter into perceptually meaningful parts.

1.2.5 Semi-supervised Shape Segmentation

We propose a novel semi-supervised framework for learning shape segmentation. In our two-stage method, first we introduce a new constrained spectral clustering (*CSC*) algorithm which takes as input a shape graph called as 'train-graph' and a sparse set of *'must-link'* and *'cannot- link'* constraints provided by a user. We propose to use the Laplacian embedding and the commute-time distance (*CTD*) to diffuse these sparse pairwise constraints over the train-graph. This lead to a new spectral representation,which is more suitable for clustering and provides the desired segmentation of input shape. Second, we consider shape alignment based on vertex-to-vertex graph matching as a way to probabilistically *transfer* labels from a train-graph in training-set of segmented shapes to a unsegmented shape called as test-graph. This segmentation transfer is carried out via a new probabilistic label transfer (*PLT*) method that computes a point-to-point mapping between the Laplacian embeddings of two graphs. This completely unsupervised matching is based on [Mateus 2008, Horaud 2011] and allows transfer of labels from a segmented shape to an unsegmented one.

1.2.6 Multi-scale Heat-Kernel Representation

We outline a general framework for the representation and analysis of 3D visual shapes based on heat diffusion on undirected weighted graphs. It is well known that the heat diffusion equation has a solution on undirected graphs and that this solution can be made explicit using the eigenvalue/eigenvector pairs of a Laplacian matrix of the graph, together with a *time parameter* that defines a scale space – the *heat kernel*. This allowed us to analyze the heat-kernel matrix within the framework of spectral graph theory [Chung 1997], to construct heat-kernel matrices well suited for 3D shapes, and to represent the latter in the metric space associated with the spectral decomposition of this matrix [Shawe-Taylor 2004].

We capitalize on the fact that the eigenvectors of the combinatorial Laplacian can be interpreted as the directions of maximum variance of the shape embedding. Together with the scale/time parameter, this provides a formal basis for performing dimensionality reduction and, more generally, to characterize the statistical properties of the embedded shape representation at multiple scales. We also study the dimensionality of the embedding, *i.e.*, the number of eigenvectors needed to approximate the heat kernel, as a function of the scale parameter; We show that the multiplicity of the first non-null eigenvalue and associated eigenvector (the Fiedler vector) of the Laplacian matrix plays a crucial role in choosing the dimension. Finally, we propose both a scale-space representation of shapes based on *auto diffusion* and *spectral distances*.

1.2.7 Topologically-Robust Dense Shape Matching

We propose a novel dense 3D shape matching method robust to topological changes in the shape. These topological changes arise in the case of articulated shapes due to complex kine-

matic poses. These poses induce self-occlusions and shadow effects, which cause the topological changes along the sequence, such as merging and splitting. The method starts from *sparse* one-to-one correspondences and produces as output *dense* correspondences. We employ multi-scale heat diffusion descriptors for this task. At small scales these descriptors are fairly local and hence it is robust to changes in topology. It can therefore be used to build a matching score between a point on the first shape and a point of the second shape conditioned by the initial correspondences. This score is then used to iteratively add new point-to-point correspondences based on a novel *seed-growing* method that propagates current correspondences to nearby ones. The final set of dense correspondences is obtained via a point registration method that uses a variant of the EM algorithm. Finally, we show dense matching results on some challenging visual shapes with significantly large topological changes.

1.3 Thesis Structure

This document is structured as follows. We first introduce basic mathematical notations and Riemannian geometry constructs in Chapter 2. This chapter also presents an overview of the existing linear and non-linear dimensionality reduction methods as well as the spectral embedding of a graph. Chapter 3 introduces pose invariant spectral representation for 3D shapes. Unsupervised and semi-supervised shape segmentation methods are presented in Chapter 4 and Chapter 5, respectively. In Chapter 6, we present a detailed outline of the heat-kernel framework for multi-scale shape representation. In the same chapter, we build upon the heat-kernel framework and propose a novel multi-scale heat distance descriptor. This is followed by a new topologically-robust dense shape matching method presented in Chapter 7. Finally, we conclude this thesis with a comprehensive summary and future directions of work in Chapter 8 and extended Appendices A as well as the bibliographic references.

CHAPTER 2

Mathematical Background

Contents

2.1 Introduction

In this chapter, we present some basic mathematical background that is necessary to understand various analytical constructs proposed and discussed in this thesis. We start with introducing various notations and definitions from the field of linear algebra, graph theory and elementary metric geometry in Section 2.2. Next, we present an overview of differential geometry of surfaces focusing on Riemannian geometry, manifold surface mapping functions and the Laplace operator on manifold surfaces in Section 2.3. In the next Section 2.4, we introduce the graph matching problem as a discrete surface mapping problem and discuss spectral graph methods in the context of exact graph matching. We briefly mention the linear and non-linear manifold learning methods for dimensionality reduction in Section 2.5. In Section 2.6, we consider an undirected weighted graph as the discrete manifold representation and combine it with the non-linear dimensionality reduction to form the basis of graph dimensionality reduction using the eigen-decomposition of the discrete Laplace operator on graphs. An intuitive relationship between the continuous and the discrete Laplace operators is discussed in Section 2.7. This is followed by Section 2.8 where we present a brief introduction of two important machine learning algorithms, namely, spectral clustering and expectation maximization, that were frequently used in this document. Finally, we conclude with Section 2.9.

2.2 Definitions and Notations

In this section, we will introduce some basic definitions and notations that will be used throughout the thesis document.

2.2.1 Linear Algebra

We represent scalar values in normal font as: a, A, α, λ, etc. A vector is a set of scalar variables and is represented by a small, bold-face letter, *e.g.*, $\mathbf{u}$ where it is a column vector and $\mathbf{u}^T$ denotes the corresponding row vector format with $\mathbf{u}^T = (u_1, \ldots, u_i, \ldots, u_n)$. $\mathbf{1}$ and $\mathbf{0}$ denotes the column vector of ones and zeros, respectively. A matrix is represented by a capital, bold-face letter, *e.g.*, $\mathbf{U}$ and its transpose as $\mathbf{U}^T$ where

$$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n] = \begin{bmatrix} u_{11}, & \ldots, & u_{n1} \\ \vdots & \vdots & \vdots \\ u_{1n}, & \ldots, & u_{nn} \end{bmatrix}. \tag{2.1}$$

$\mathbf{I}_n$ is the $n \times n$ identity matrix. The dot product between two vectors is represented as $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \sum_k u_{ik} u_{jk} = \mathbf{u}_i^T \mathbf{u}_j$. The vector norm is denoted as $\|\mathbf{u}\|^2 = \langle \mathbf{u}, \mathbf{u} \rangle$. The Euclidean distance between two vectors is computable as $\|\mathbf{u}_i - \mathbf{u}_j\|^2 = \langle \mathbf{u}_i, \mathbf{u}_i \rangle + \langle \mathbf{u}_j, \mathbf{u}_j \rangle - 2\langle \mathbf{u}_i, \mathbf{u}_j \rangle$. The Frobenius Norm of a matrix is written as $\|\mathbf{U}\|_F^2 = \sum_i \sum_j U_{ij} = \mathrm{tr}(\mathbf{A}^T \mathbf{A})$.

2.2.2 Graph Representation

We denote a connected *undirected weighted graph* as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V}(\mathcal{G}) = \{v_1, \ldots, v_n\}$ is the vertex set, $\mathcal{E}(\mathcal{G}) = \{e_{ij}\}$ is the edge set. Let $\mathbf{W}$ be the weighted adjacency matrix of this graph. Each $(i,j)^{\text{th}}$ entry of $\mathbf{W}$ matrix stores, a weight w_{ij} whenever there is an edge $e_{ij} \in \mathcal{E}(\mathcal{G})$ between graph vertices v_i and v_j, and 0 otherwise. All the diagonal elements are set to 0 . We use the following notations: The degree d_i of a graph vertex $d_i = \sum_{i \sim j} w_{ij}$ ($i \sim j$ denotes the set of vertices v_j that are adjacent to v_i), the *degree matrix* $\mathbf{D} = \text{Diag}[d_1 \ldots d_i \ldots d_n]$, the $n \times 1$ *degree vector* $\mathbf{d} = \mathbf{D1}$, and the *graph volume* $\text{Vol}(\mathcal{G}) = \sum_i d_i$.

In spectral graph theory, it is common to use the following expression for the edge weights [Belkin 2003, von Luxburg 2007]:

$$w_{ij} = e^{-\frac{\text{dist}^2(v_i, v_j)}{\sigma^2}}, \tag{2.2}$$

where $\text{dist}(v_i, v_j)$ denotes any distance metric between two vertices and σ is a free parameter. In the case of a *fully connected graph*, matrix $\mathbf{W}$ is also referred to as the *similarity matrix*. The *normalized weighted adjacency matrix* writes $\tilde{\mathbf{W}} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$. The *transition* matrix of the non-symmetric reversible Markov chain associated with the graph is

$$\tilde{\mathbf{W}}_R = \mathbf{D}^{-1}\mathbf{W} = \mathbf{D}^{-1/2}\tilde{\mathbf{W}}\mathbf{D}^{1/2}. \tag{2.3}$$

In this work, unless otherwise stated, we consider $\mathcal{G}$ as a connected graph such that there exist a path from each vertex v_i to every other vertex v_j in $\mathcal{V}$.

2.2.3 Elementary Metric Geometry

Metric geometry is the mathematical construction of the vague idea that classify the relationship between "primitives" as close or far apart, which in turn, is based upon the concept of "near" and "far". Thus, the metric geometry studies the concept of length and geodesic distance in order to obtain an analytical description of geometrical objects. Here we present few basic definitions from the elementary metric geometry as summarized in [Thorstensen 2009] and recommend an excellent monograph [Burago 2001] for a detailed understanding of the metric geometry.

Definition 1 *Let $\mathbb{Y}$ be an arbitrary abstract set. A function $d_{\mathbb{Y}} : \mathbb{Y} \times \mathbb{Y} \mapsto \mathbb{R} \cup \{\infty\}$ is a metric on $\mathbb{Y}$ if the following conditions are met for all $y_1, y_2, y_3 \in \mathbb{Y}$.*

- *Non-negativity: $d_{\mathbb{Y}}(y_1, y_2) \geq 0$ and $d_{\mathbb{Y}}(y_1, y_2) = 0 \Leftrightarrow y_1 = y_2$.*
- *Symmetry: $d_{\mathbb{Y}}(y_1, y_2) = d_{\mathbb{Y}}(y_2, y_1)$.*
- *Triangle inequality: $d_{\mathbb{Y}}(y_1, y_3) \leq d_{\mathbb{Y}}(y_1, y_2) + d_{\mathbb{Y}}(y_2, y_3)$.*

Then the pair $(\mathbb{Y};d_{\mathbb{Y}})$ is a metric space. The elements of $\mathbb{Y}$ are called points of the metric space. The function $d_{\mathbb{Y}}(y_1,y_2)$ returns the distance between two points y_1,y_2. A very well known instance of a metric space is the three-dimensional Euclidean space $\mathbb{R}^3$ with the Euclidean metric. In general, any normed vector space V is a metric space with the metric induced by the norm.

Definition 2 *Let V be a vector space. A function $\|\cdot\| \mapsto \mathbb{R}$ is a norm on V if the following conditions are met for all $\mathbf{v}_1,\mathbf{v}_2 \in V$ and $k \in \mathbb{R}$.*

- *Non-negativity: $\|\mathbf{v}\| > 0$ if $\mathbf{v} \neq 0$.*
- *Linearity: $\|k\mathbf{v}\| = k\|\mathbf{v}\|$.*
- *Triangle inequality: $\|\mathbf{v}_1+\mathbf{v}_2\| \leq \|\mathbf{v}_1\|+\|\mathbf{v}_2\|$*

So a normed vector space is a vector space equipped with a norm. For instance, the Euclidean space $\mathbb{R}^n$ is a normed space with norm $\|(\mathbf{y}^1,\ldots,\mathbf{y}^d)\| = \sqrt{(\mathbf{y}^1)^2+\cdots+(\mathbf{y}^d)^2}$.

Lastly, we notice that a norm is called Euclidean if it is associated with some scalar product.

Definition 3 *Let V be a vector space. A scalar product $\langle\cdot,\cdot\rangle : V \times V \mapsto \mathbb{R}$ on V is a symmetric bi-linear form F whose associated quadratic form is positive definite, i.e., $F(\mathbf{v},\mathbf{v}) > 0$ for all $\mathbf{v} \neq 0$.*

The definition of a bi-linear forms is given as:

Definition 4 *Let V be a vector space. A bi-linear form F on V is a function of two variables $V \times V \mapsto \mathbb{R}$ satisfying the following equations:*

$$
\begin{aligned}
F(\mathbf{v}_1+\mathbf{v}_2,\mathbf{v}_3) &= F(\mathbf{v}_1,\mathbf{v}_3)+F(\mathbf{v}_2,\mathbf{v}_3) && (2.4)\\
F(k\mathbf{v}_1,\mathbf{v}_2) &= kF(\mathbf{v}_1,\mathbf{v}_2) && (2.5)\\
F(\mathbf{v}_1,\mathbf{v}_2+\mathbf{v}_3) &= F(\mathbf{v}_1,\mathbf{v}_2)+F(\mathbf{v}_1,\mathbf{v}_3) && (2.6)\\
F(\mathbf{v}_1,k\mathbf{v}_2) &= kF(\mathbf{v}_1,\mathbf{v}_2) && (2.7)
\end{aligned}
$$

with $\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3 \in V$ and $k \in \mathbb{R}$.

2.3 Differential Geometry of Surfaces

In this section, we will briefly introduce some basic notions of the *Differential Geometry of Surfaces*. These notions, mostly defined for continuous domain, can be easily extended to

discrete domain and will later provide us a strong mathematical framework to address the problem of pose invariant shape representation and analysis. We recommend further reading of two popular books [Do carmo 1992, Gray 2006] for a thorough understanding of the Riemannian geometry.

Differential geometry, in general, is a mathematical discipline that uses the techniques of differential and integral calculus, as well as algebra, to study problems in the geometry. The initial development of differential geometry largely involved the theory of plane, space curves and of surfaces embedded in three-dimensional Euclidean spaces. However, from the late 19th century, differential geometry has evolved into a field concerned more generally with the geometric structures on differentiable manifolds. In particular, we are more interested in the study of *Riemannian* geometry, which mainly deals with the Riemannian manifold - a smooth manifold with a Riemannian metric.

A Riemannian manifold is a general *d-dimensional* manifold with an inner product on the tangent space, *i.e.*, the Riemannian metric. This inner product is defined at each point and vary smoothly from point to point. The Riemannian metric allows one to define the notion of angles, length of the curves, surface area, and volume. We are more concerned with differential geometry of *2-dimensional* manifold surfaces as majority of 3D shapes can be treated as the discrete version of a continuous 2D manifold surface embedded in a 3-dimensional Euclidean space.

An interesting aspect of the Riemannian geometry is that it allows one to study the *intrinsic* properties of a manifold surface that are independent of the ambient three-dimensional Euclidean space in which the surface is immersed. This is in contrast with the previous works in differential geometry that mainly focused on studying the *extrinsic* properties of (curves and) surfaces that depend on the properties of embedded Euclidean space. Hence, this can serve as the basis of design of a intrinsic pose invariant shape representation/analysis algorithm.

2.3.1 Riemannian Geometry of Surfaces

We first introduce the definition of a d-dimensional manifold, a Tangent space, a Riemannian manifold and a Riemannian metric. This is followed by the definition and analysis of a parametric 2D surface patch, which in turn, is a subset of 2-dimensional Riemannian manifold embedded in the three-dimensional Euclidean space. We deduce the local parametrization of this 2D surface patch by defining smooth differentiable functions on it and derive an inner product metric, which in turn, is used to compute the angle and length of the curves on the surface. Finally we summarize two general theorems of the Riemannian geometry, which will facilitate an isometric embedding of a Riemannian manifold into a Euclidean subspace.

2.3.1.1 d-dimensional Manifold

Definition 5 *A d-dimensional manifold $\mathcal{M}$ is a topological space such that each point $x \in \mathcal{M}$ has an ε-neighborhood that is homeomorphic to a disc in the d-dimensional Euclidean space. In case of a manifold with boundary, the disc can be replaced by a d-dimensional half-disk for the boundary points. Figure 2.1 depicts the typical scenario using a surface patch with boundary.*

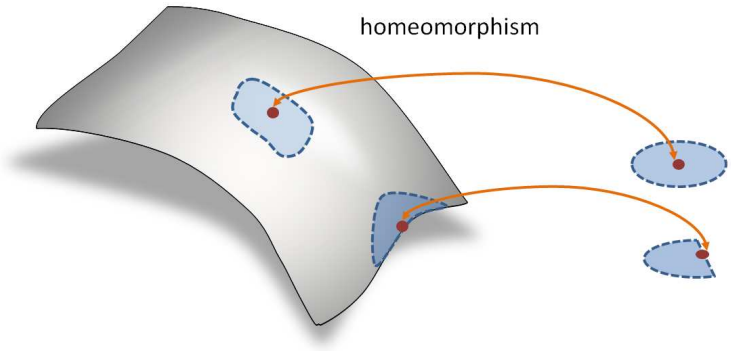


Figure 2.1: Visualization of manifold construction using a 2D manifold surface patch.

2.3.1.2 Tangent Space & Tangent Plane

The concept of tangent space in differential geometry is analogous to the idea of linear approximation of a surface in the vector calculus. For a given manifold $\mathcal{M}$ embedded in $\mathbb{R}^d$, a linear subspace can be associated with each point $x \in \mathcal{M}$. This linear subspace of $\mathbb{R}^d$ is called the tangent space and comprises all the tangent vectors of x. The tangent space is the best linear approximation of the local manifold surface within a small neighborhood around x. In case of a manifold being a parametric $2D$-surface patch embedded in $\mathbb{R}^3$, the tangent space around x is called a tangent plane and is typically depicted as $\mathcal{T}_x\mathcal{M}$. Figure 2.2 depicts the tangent space of a 2D manifold surface embedded in $\mathbb{R}^3$.

2.3.1.3 Riemannian Manifold & Riemannian Metric

Definition 6 *A Riemannian manifold $(\mathcal{M}, g)$ is a smooth manifold $\mathcal{M}$ associated with a family of smoothly varying positive definite inner products $g = g_x$ on $\mathcal{T}_x\mathcal{M}$ for each $x \in \mathcal{M}$. The family g is called Riemannian metric. Note that metric g_x is a bi-linear form on $\mathcal{T}_x\mathcal{M}$, i.e., $g_x : \mathcal{T}_x\mathcal{M} \times \mathcal{T}_x\mathcal{M} \longrightarrow \mathbb{R}$. Thus, for all differential vector fields $X, Y \in \mathcal{T}_x\mathcal{M}$, $x \mapsto g_x(X,Y)$ defines a smooth function $\mathcal{M} \mapsto \mathbb{R}$. More formally, a Riemannian metric g is a symmetric positive definite $(0,2)$-tensor, i.e., $g(X,X) > 0, \ \forall X \neq 0 \in \mathcal{T}\mathcal{M}$.*

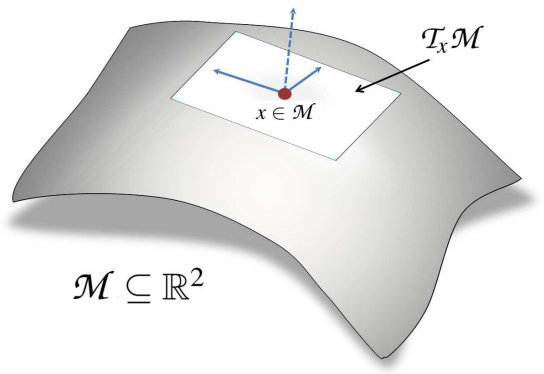


Figure 2.2: Depiction of tangent space of a 2D manifold surface embedded in the three-dimensional Euclidean space.

Next, we will derive an exact formulation of the Riemannian metric for a parametric surface patch (embedded in the three-dimensional Euclidean space) where it is also known as the first fundamental form.

2.3.1.4 Parametric Surface Patch

A 2-dimensional parametric surface patch is a surface in the Euclidean space $\mathbb{R}^n$ defined by a parametric equation of two parameters. Let $f : \mathbb{R}^2 \supseteq \Omega \mapsto \mathbb{R}^n$ be a smooth differentiable function of two variables (u,v) that maps a point in the domain $\Omega \subseteq \mathbb{R}^2$ to a point in $\mathbb{R}^n$. A parametric surface patch $\mathcal{S}$ can be defined as: $\mathcal{S} = f(\Omega), \mathcal{S} \subseteq \mathbb{R}^n$. If we consider $n = 3$, *i.e.*, a three-dimensional Euclidean space with canonical axes x, y and z, then we can write: $f(u,v) = (x(u,v), y(u,v), z(u,v))$. Figure 2.3 illustrate a surface patch and its domain.

2.3.1.5 Local Parameterization using Canonical Tangents and Surface Normal

A surface patch $\mathcal{S}$ can be locally parametrized at any point $\mathbf{p} = f(u,v) \in \mathcal{S}$ using the canonical tangents and surface normal at that point.

The *canonical tangents* of a surface patch can be written as: $\partial_u f(u,v)$ and $\partial_v f(u,v)$. The *tangent plane* at a point $\mathbf{p}$ consist of all the tangent vectors to $\mathbf{p}$ and can be defined as the linear combination of canonical tangents. The *surface normal* at point $\mathbf{p}$ is a unit vector orthogonal to the tangent plane and is defined as the normalized cross product of canonical tangent vectors as:

$$\mathbf{n}_{uv} = \frac{\partial_u f(u,v) \times \partial_v f(u,v)}{\| \partial_u f(u,v) \times \partial_v f(u,v) \|}. \tag{2.8}$$

Figure 2.4 depict canonical tangent vectors and the surface normal on a parametric surface patch.

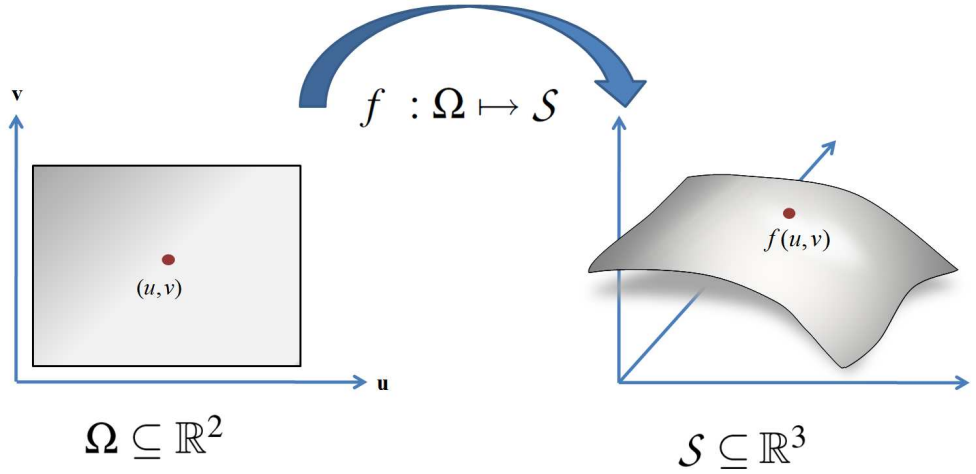


Figure 2.3: Visualization of a parametric surface patch embedded in the three-dimensional Euclidean space.

2.3.1.6 The First Fundamental Form

The *first fundamental form* describes the local parametrization of the surface. It is the inner product on the tangent space of a surface in three-dimensional Euclidean space, which is induced canonically from the dot product in $\mathbb{R}^3$. This inner product allows to measure the distortion of angle and lengths in an ε-neighborhood around a point on the surface, which is induced by the embedding of the surface in the ambient Euclidean space. Hence, it is also called as the *"metric tensor"*.

The metric tensor can be derived as follows. Let $\omega_0 = (u_0, v_0) \in \Omega$ be the origin of the local coordinate system corresponding to a point $\mathbf{p}_0 = f(x(u_0,v_0), y(u_0,v_0), z(u_0,v_0)) \in \mathcal{S}$. The function mapping $f(\omega)$ for a point ω in the neighborhood of ω_0 can be written in terms of the local first order Taylor approximation as:

$$f(\omega) = f(\omega_0) + \nabla f(\omega_0)(\omega - \omega_0). \tag{2.9}$$

Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$ be the two arbitrary vectors as shown in the Figure 2.5. The scalar product of mapping of these two vectors can be simplified using Eq. (2.9) as:

$$\begin{aligned} \langle f(\omega_0 + \mathbf{a}) - f(\omega_0),\, f(\omega_0 + \mathbf{b}) - f(\omega_0) \rangle &\approx \langle \nabla f(\omega_0)\mathbf{a},\, f(\omega_0)\mathbf{b} \rangle \\ &\approx \mathbf{a}^T \underbrace{\left(\nabla f(\omega_0)^T \nabla f(\omega_0)\right)}_{\text{First Fundamental Form}} \mathbf{b}. \end{aligned} \tag{2.10}$$

The first order derivative of f denoted as ∇f is a Jacobian matrix of dimensional 3×2 computed on the Euclidean surface. Whereas, the first fundamental form represents an inner product matrix $\nabla f^T \nabla f$ computed using the dot product of rows of the Jacobian matrix and hence

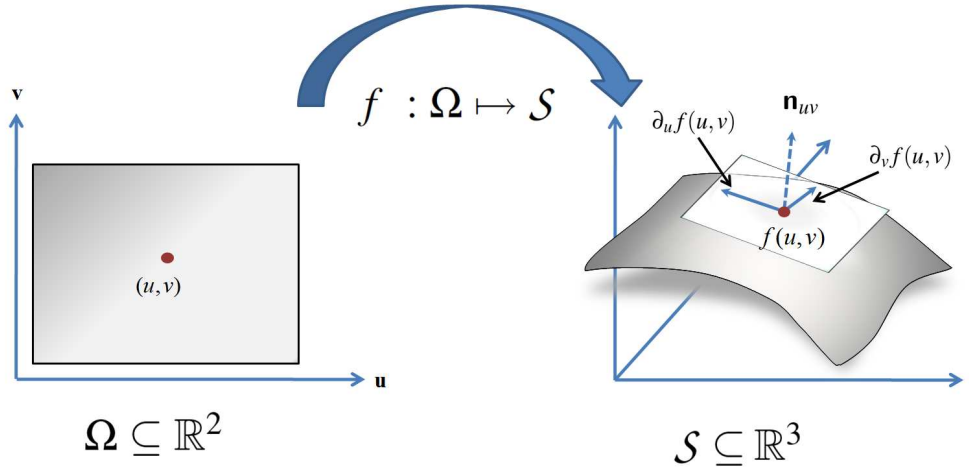


Figure 2.4: Visualization of the canonical tangents and the surface normal on a parametric surface patch embedded in the three-dimensional Euclidean space.

can be written in a 2×2 square matrix as:

$$(\nabla f^T \nabla f) = \begin{pmatrix} \partial_u f.\partial_u f & \partial_u f.\partial_v f \\ \partial_u f.\partial_v f & \partial_v f.\partial_v f \end{pmatrix} =: \begin{pmatrix} E & F \\ F & G \end{pmatrix}. \tag{2.11}$$

This matrix is positive definite for a regular parametrization and positive semi-definite otherwise.

Thus, the first fundamental form defines a generalized scalar product, which in turn measures the length and angles on the surface. The metric tensor is commonly denoted as:

$$\mathbf{I}(\mathbf{a},\mathbf{b}) := \mathbf{a}^T \left(\nabla f^T \nabla f\right) \mathbf{b} \tag{2.12}$$

However, it only captures the change in length, which lead to same parametrization for a cylindrical surface and a planar surface. This is because only the first order approximation of the mapping function f) was used, which is inherently flat. Thus, we need a tool to measure curvature of the surface in order to differentiate between two such surfaces. This will require a second order approximation of f.

2.3.1.7 The Second Fundamental Form

The *second fundamental form*, also known as the *shape operator* or *curvature tensor*, captures the local curvature of the parametric surface. The shape operator can be written in a 2×2 matrix form as:

$$\mathbf{S} = \begin{pmatrix} \partial_{uu} f.\mathbf{n} & \partial_{uv} f.\mathbf{n} \\ \partial_{uv} f.\mathbf{n} & \partial_{vv} f.\mathbf{n} \end{pmatrix} =: \begin{pmatrix} e & f \\ f & g \end{pmatrix}. \tag{2.13}$$

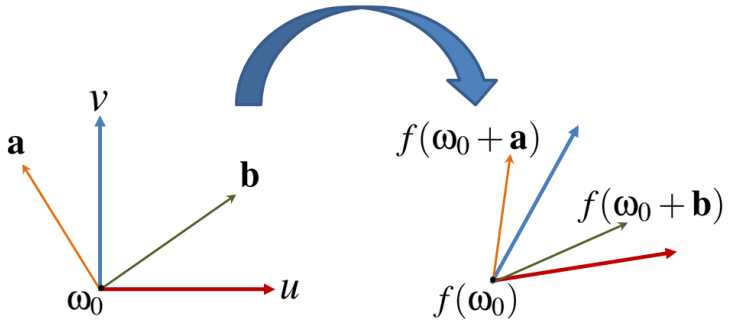


Figure 2.5: Construction of Metric Tensor.

The matrix $\mathbf{S}$ is a symmetric matrix, but not a positive definite one. Here, $\mathbf{n}$ is the surface normal at the point where the shape operator is computed and ∂_{uu}, ∂_{uv} and ∂_{vv} are the second order partial differential operators applied to mapping function f at that point. Hence, the second fundamental form computes the second derivatives of the function mapping f and project them in the normal direction in order to cancel the tangential acceleration, thus capturing the curvature. This can also be seen as the second order Taylor expansion of f with quadratic terms. For two arbitrary vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$, the curvature tensor is applied as: $\mathbf{II}(\mathbf{a}, \mathbf{b}) = \mathbf{a}^T S \mathbf{b}$.

The curvature tensor is useful to compute different types of curvatures listed below.

Principal Curvatures: The eigenvalues of a shape operator defined at a given point on the parametric surface are called the principal curvatures of surface at that point and commonly denoted as κ_1, κ_2. The corresponding eigenvectors form the orthonormal tangent bases and are known as the principal directions of curvature. Since the shape operator matrix is not a positive (semi-)definite, its eigenvalues can be negative, resulting in negative curvature values. Figure 2.6 visualize the planes of the principal curvatures.

Normal Curvature: The *normal curvature* $\kappa(\mathbf{r})$ at a point is the curvature of the surface curve obtained by the intersection of the parametric surface and a plane defined by the normal vector at that point along with a directional vector $\mathbf{r}$ in the respective tangent plane. Hence, the plane of the normal curvature is same as the plane of principal curvature, rotated around the normal vector, in the direction of $\mathbf{r}$ vector. This can be written as:

$$\kappa(\mathbf{r}) = \mathbf{r}^T \begin{pmatrix} e & f \\ f & g \end{pmatrix} \mathbf{r}. \tag{2.14}$$

Thus, the maximum and minimal value of the normal curvature are the values of the principal curvatures in the direction of orthonormal tangent basis vectors defined by the eigenvectors

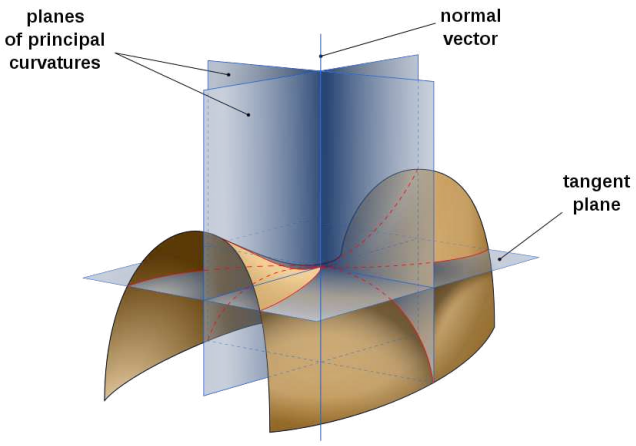


Figure 2.6: Visualization of Principal Curvature. (Source: http://wikipedia.org)

of the shape operator, *i.e.*,

$$\kappa_1 = \max_{\mathbf{r}}(\kappa(\mathbf{r})), \quad \kappa_2 = \min_{\mathbf{r}}(\kappa(\mathbf{r})). \tag{2.15}$$

Mean Curvature: The *mean curvature* at a given point on the surface is defined as the average of the principal curvatures at that point, *i.e.*,

$$H = \frac{\kappa_1 + \kappa_2}{2} = \frac{1}{2}\mathrm{tr}\begin{pmatrix} e & f \\ f & g \end{pmatrix}. \tag{2.16}$$

Gaussian Curvature: The *Gaussian curvature* at a given point on the surface is defined as the multiplication of the principal curvatures at that point, *i.e.*,

$$K = \kappa_1 * \kappa_2 = \det\begin{pmatrix} e & f \\ f & g \end{pmatrix}. \tag{2.17}$$

The Gaussian curvature is very important notion in the Riemannian geometry. The famous theorem *Theorema Egregium* from Gauss states that the Gaussian curvature of a surface can be expressed solely in terms of the first fundamental form and its derivatives. Thus, though computed extrinsically, Gaussian curvature is essentially an intrinsic property of the surface and hence not affected by the embedding of surface in the ambient Euclidean space.

2.3.1.8 Geodesic Curvature of a Curve

Geodesic curvature is another important notion in differential geometry and leads to the concept of *geodesic* metric. This is the intrinsic curvature of a curve lying on the surface and does not depend on how the surface is immersed in the Euclidean space.

Definition 7 *For a given curve lying on the manifold surface and characterized by the arc-length s and unit tangent vector T, the geodesic curvature is the norm of projection of derivative* $\frac{dT}{ds}$ *on the tangent plane of the surface.*

Figure 2.7 depicts a curve lying on the surface and the curvature of its projection on the tangent plane. Thus, any curve that connects two points on a parametric surface and has zero

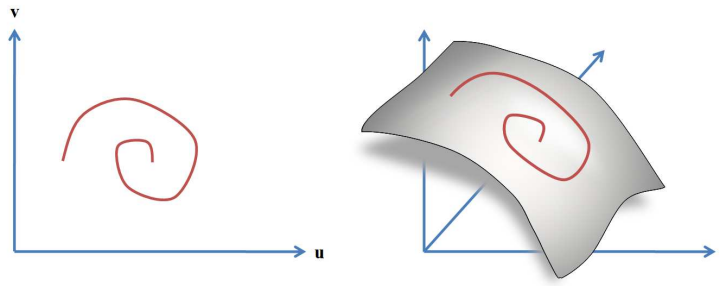


Figure 2.7: Visualization of intrinsic (geodesic) curvature of a curve lying on manifold surface.

geodesic curvature at each point, is called the geodesic curve and corresponding arc-length is called the *geodesic distance* between those two points. This lead us to the notion of minimum distance path on manifold surfaces.

2.3.1.9 General Theorems of Riemannian Geometry

Here we briefly state two important theorems in Riemannian geometry of surfaces. The first one, Gauss-Bonnet theorem incredibly relates the topological characteristic of a manifold with its total curvature. The second is a set of two theorems, also known as the fundamental theorems of Riemannian geometry, lead to an isometric embedding of a Riemannian manifold into n-dimensional Euclidean space. More formally:

Gauss-Bonnet Theorem: The integral of the Gaussian curvature also known as the total curvature of a compact 2-dimensional Riemannian manifold is equal to $2\pi\chi(\mathcal{M})$, where $\chi(\mathcal{M})$ denotes the Euler characteristic of $\mathcal{M}$, which is a topological characterization of a Riemannian manifold.

Nash Embedding Theorems: The theorems state that every Riemannian manifold can be isometrically embedded in a Euclidean space $\mathbb{R}^n$. The first theorem is for continuously differentiable (C^1) embeddings and the second for analytic embeddings or embeddings that are smooth of class $C^k, 3 \leq k \leq \infty$.

These theorems will be useful for finding a pose invariant isometric embedding of an articulated 3D shape typically represented as the discrete counterpart of a compact Riemannian manifold (see Chapter 3).

2.3.2 Mapping between Surfaces

Surface mapping is an interesting problem in differential geometry. The idea is to seek a (bijective) mapping function that allows one-to-one correspondence between two manifold surfaces which are embedded differently in the Euclidean space. One classical example of surface mapping problem is to find a flat map of the earth's surface. An important result from Gauss proved that it is impossible to do so without admitting deformations. This significant theorem from Gauss, known as the *Theorema Egregium*, laid the foundation of modern differential geometry which can deal with intrinsic surface mappings. The surface mapping has many important applications, *e.g.*, texture mapping, dense registration, animation transfer, MRI analysis, etc. Figure 2.8 illustrates surface mapping scenario in $\mathbb{R}^3$.

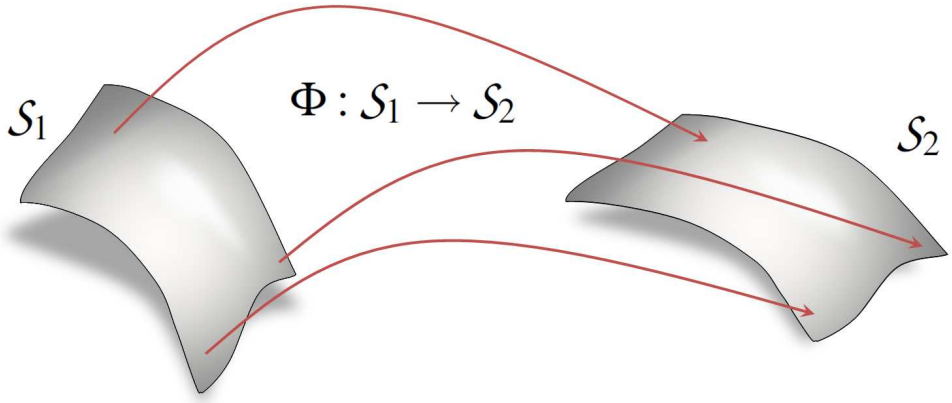


Figure 2.8: Mapping between manifold surfaces.

We have already introduced the Riemannian manifolds that are equipped with the Riemannian (dot product) metric. This helps to define various mapping functions between two manifold surfaces that can preserve certain properties of these surfaces. The three important types of surface mapping are:

- Isometric Mapping - preserves angles, distances and area.
- Conformal Mapping - preserves only angles.
- Equi-areal Mapping - preserves only area.

Given a mapping function $f : \mathbb{R}^2 \supseteq \Omega \mapsto \mathcal{S} \subset \mathbb{R}^3$ (see Figure 2.3) that maps a 2D plane (domain) to a 3D surface patch, we can compare these mapping based on the respective Jacobian matrix (∇f) and metric tensor introduced in Section 2.3.1.6.

The isometric mapping preserves the distances on surfaces by choosing an orthogonal rotation matrix (**R**) as the Jacobian matrix. This causes the metric tensor to become an identity matrix (**I**). Thus, in an isometric transformation, a pure rotation is applied to the local tangent planes, thereby preserving the lengths, angles and area. In case of a conformal mapping, the Jacobian is a scaled orthogonal rotation matrix ($\eta\mathbf{R}$) and metric tensor becomes $\eta\mathbf{I}$. This scaling enables an arbitrary stretching or shrinking of tangent planes, resulting in change in lengths and area. Hence, only the angles are preserved in the conformal mapping. The equi-areal mapping constraints the Jacobian matrix to have a determinant value equal to one. This causes the local tangent plane to stretch in one direction and at the same time shrink in the other direction in order to keep the determinant value one, thus preserving the area. We will discuss in detail only the isometric mapping as this thesis primarily uses isometric transformations to model non-rigid deformations in 3D articulated shapes. Please refer to notes in [Belk 2011] for a brief mathematical overview of isometries.

2.3.2.1 Isometric Mapping

An isometric mapping is a distance, angle and area preserving mapping between two metric spaces. A smooth injective map between two surfaces is called a *local* isometric map if it preserves the length of curves. A local isometric mapping is called an *isometry* if it is *bijective*.

Let $\mathcal{S}_1$, $\mathcal{S}_2$ be two metric spaces associated with metrics $d_{\mathcal{S}_1}$, $d_{\mathcal{S}_2}$ and tangent spaces $\mathcal{T}\mathcal{S}_1$, $\mathcal{T}\mathcal{S}_2$, respectively. A map $\Phi : \mathcal{S}_1 \to \mathcal{S}_2$ is called local isometry if for any $a, b \in \mathcal{S}_1$ one has

$$d_{\mathcal{S}_2}(\Phi(a), \Phi(b)) = d_{\mathcal{S}_1}(a, b). \tag{2.18}$$

More formally, a smooth map $\Phi : \mathcal{S}_1 \to \mathcal{S}_2$ is a local isometry if and only if the derivative

$$\nabla_{\mathbf{p}}\Phi : \mathcal{T}_{\mathbf{p}}\mathcal{S}_1 \to \mathcal{T}_{\Phi(\mathbf{p})}\mathcal{S}_2 \tag{2.19}$$

at each point $\mathbf{p} \in \mathcal{S}_1$ is an isometric linear transformation. Thus, in case of isometric mapping, for each tangent vector $\mathbf{t} \in \mathcal{T}_{\mathbf{p}}\mathcal{S}_1$, the following is true.

$$\| \nabla_{\mathbf{p}}\Phi(\mathbf{t}) \| = \| \mathbf{t} \| \tag{2.20}$$

If $\mathbf{I}_1$ and $\mathbf{I}_2$ be the first fundamental forms of $\mathcal{S}_1$ and $\mathcal{S}_2$, then we can write:

$$(\nabla\Phi)^T \mathbf{I}_2 (\nabla\Phi) = \mathbf{I}_1. \tag{2.21}$$

When two surfaces are mapped from a common domain, *e.g.*, $f_1 : \Omega \mapsto \mathcal{S}_1$ and $f_2 : \Omega \mapsto \mathcal{S}_2$, the derivative $\nabla\Phi$ is simply a 2×2 identity matrix. This results in

$$\mathbf{I}_1 = \mathbf{I}_2, \tag{2.22}$$

i.e. $\mathcal{S}_1$ and $\mathcal{S}_2$ having the same metric tensor. Thus, an isometric mapping between two surface patches (with shared domain), has same metric tensor and consequently, preserves the Gaussian curvature. Figure 2.9 demonstrates a typical isometric mapping scenario when two surfaces have the same co-domain.

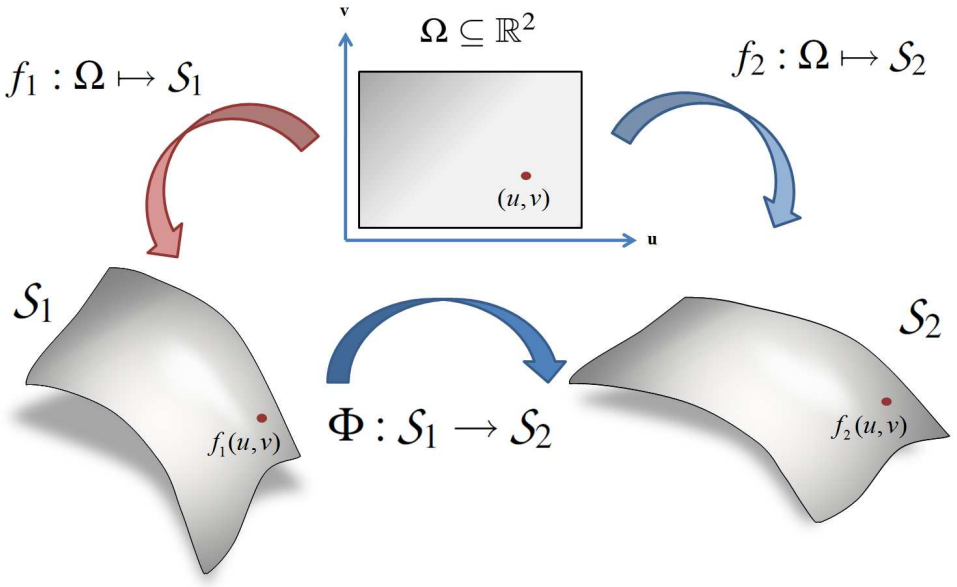


Figure 2.9: Visualization of isometric mapping when two surface patches have same co-domain.

2.3.3 Laplace Operator on Manifold

The Laplace operator is a second order differential operator in the n-dimensional Euclidean space, defined as the divergence $(\nabla\cdot)$ of the gradient (∇f). Thus, if f is a twice-differentiable real-valued function, then the Laplacian of f is defined as:

$$\Delta f = \nabla^2 f = \nabla \cdot \nabla f. \tag{2.23}$$

In general, it maps a k times smoothly differentiable C^k-function to a C^{k-2} function defined on an open set Ω as:

$$\Delta : C^k(\Omega) \rightarrow C^{k-2}(\Omega) \tag{2.24}$$

2.3.3.1 Laplace-Beltrami Operator

The Laplace-Beltrami operator is an extension of the classical Laplace operator on Riemannian manifold (see [Rosenberg 1997] for details).

Let f be a real valued twice differentiable function defined on a Riemannian manifold $\mathcal{M}$. The Laplace-Beltrami operator ∇ is given by:

$$\Delta_{\mathcal{M}} f := div(\nabla_{\mathcal{M}} f) \tag{2.25}$$

with $\Delta_{\mathcal{M}} f$ be the gradient of f and div be the divergence of a vector field on a manifold. The divergence is an operator that measures the magnitude of a vector field's source or sink at a given point. The Laplace-Beltrami operator is a linear differential operator and can be calculated using a local parametrization, *i.e.*, the tangent space. Given the metric tensor g of $\mathcal{M}$, the Laplace-Beltrami operator on $\mathcal{M}$ writes as:

$$\Delta_{\mathcal{M}} f = \frac{1}{\sqrt{\|g\|}} \sum_{i,j} \partial_i (g_{ij}^{-1} \sqrt{\|g\|} \partial_j f). \tag{2.26}$$

$\|g\|$ is the determinant of g and ∂_i, ∂_j are the basis vector of the tangent space. In case where $\mathcal{M} \subset \mathbb{R}^2$, the metric tensor g simplifies to the identity and the Laplace-Beltrami reduces to the Laplace operator in $\mathbb{R}^2$:

$$\Delta f = \frac{\partial^2 f}{\partial (x^1)^2} + \frac{\partial^2 f}{\partial (x^2)^2}. \tag{2.27}$$

The spectrum of the Laplacian is the set of eigenfunctions and associated eigenvalues solving the Helmholtz equation.

$$\Delta_{\mathcal{M}} f = -\lambda f. \tag{2.28}$$

The solution is an infinite number of eigenvalues λ_i and eigenfunctions f_i (with $0 \leq i \leq \infty$).

In the case of a closed surface without boundary, the first eigenvalue λ_0 is always equal to zero and the associated eigenfunction f_0 is a constant function.

Few properties of the Laplace-Beltrami operator as mentioned in the book [Reuter 2006a] are:

- The spectrum depends only on the metric tensor and is invariant under isometric mappings.
- The spectrum of the Laplace-Beltrami operator of d-dimensional manifolds at different scales can be compared by normalizing the eigenvalues appropriately.
- A change of the surface's shape results in continuous change of the spectrum.
- The spectrum does not characterize the shape completely, since some non-isometric manifolds with the same spectrum exist. Nevertheless, these cases appear to be very rare.
- A substantial amount of geometrical and topological information is known to be contained in the spectrum. As a consequence of the high dimensionality of the eigenspectrum, cropping the spectrum is unavoidable and consequently induces a loss of information. But nevertheless, the first few eigenvalues contain important information

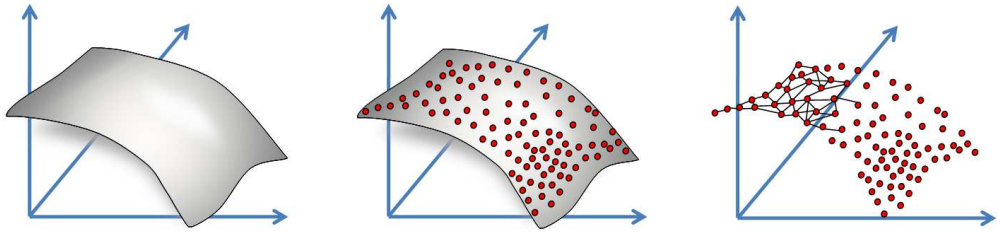


Figure 2.10: Visualization of a continuous manifold surface embedded in R^3 and its discrete counterpart with non-uniform sampling and corresponding sparse graph structure that is typically used to approximate the local neighborhood around each sampled point.

With these properties at hand, the Laplace-Beltrami operator is an important tool to analysis 3D shapes, represented as manifold surfaces. In particular, the first property is useful to find an isometry invariant shape representation and analysis of articulated 3D shapes.

2.4 Surface Mapping as Graph Matching

We have already introduced a discussion on surface mapping in the continuous domain setting in Section 2.3.2. However, this setting is best suited mainly to analyze the theoretical formulations as one can make assumptions about the smoothness and continuity of the domain. On the other hand, our sensing capabilities lead us to a discrete world where making such assumptions is difficult and the best we can do is to approximate the theoretical notions. One related example in this context is the surface acquisition in computer vision. We have already presented some example shape visual shapes in Section 1.1, in order to understand the problems with the discrete approximation of continuous manifold surface. Figure 2.10, reiterates this problem where a manifold surface is visualized in a discrete setup with non-uniform sampling and the corresponding sparse graph structure is also shown, which is typically used to approximate the local neighborhood around each sampled point. easily be violated (due to sensor limitations).

Hence, one can cast the surface mapping as the graph matching problem, by assuming an infinitely dense sampling of the continuous surface. This consequently makes a large number of existing graph matching methods as the candidate for solving the surface mapping problem. The task of graph matching involves finding correspondence between vertices and edges of two graphs, subject to certain constraints. Traditionally, the structural (skeleton) representation of 3D shapes is used in the context of graph matching [Hilaga 2001, Cornea 2005, Siddiqi 2008, Biasotti 2006].

Existing graph matching methods can be largely classified as *Exact* and *Inexact*, based upon whether they find an exact solution by enforcing a strict one-to-one correspondence or an approximate correspondence between graph sub-structures. The exact graph matching is

traditionally known as the *Graph Isomorphism*. The idea is to seek a one-to-one mapping between vertex set of two graph, subject to a strict mapping between corresponding edges. Thus, the graph isomorphism problem reduces to finding a permutation matrix (see Appendix A.1) between two sets of graph vertices. One can relax this to a *sub-graph isomorphism* problem by finding a mapping between vertices of the first graph and the subset of vertices of the second graph. This can further be relaxed to *maximum common sub-graph* problem where one seeks the largest subgraph in each of the graphs, for which an isomorphism exists. However, the complexity of the exact graph isomorphism has not yet been demonstrated and rest of the above problems are NP-complete [Garey 1990]. Therefore, either one finds an exact graph isomorphism or seeks an approximate solution for the graph matching problem [Cour 2006].

Though there exists a vast amount of literature on graph matching, in this thesis we mainly focus on spectral framework for solving graph matching problem. In particular, we first introduce the spectral graph isomorphism in this section and extend it to large graphs by introducing a graph dimensionality reduction technique in the following sections. In later part of the thesis, we discuss in detail the inexact graph matching formulations in the context of dense shape matching in Chapter 7.

2.4.1 Spectral Graph Isomorphism

Let $\mathcal{G}_A$ and $\mathcal{G}_B$ be the two *undirected weighted graphs* with the same number of nodes, n, and let $\mathbf{W}_A$ and $\mathbf{W}_B$ be their adjacency matrices. They are real-symmetric matrices. In the general case, the number r of distinct eigenvalues of these matrices is smaller than n. The standard spectral methods only apply to those graphs whose adjacency matrices have n distinct eigenvalues (each eigenvalue has multiplicity one), which implies that the eigenvalues can be ordered.

Graph isomorphism [Godsil 2001] can be written as the following minimization problem:

$$\mathbf{P}^{\star} = \arg\min_{\mathbf{P}} \|\mathbf{W}_A - \mathbf{P}\mathbf{W}_B\mathbf{P}^{\top}\|_F^2 \tag{2.29}$$

where $\mathbf{P}$ is an $n \times n$ permutation matrix (see Appendix A.1) with $\mathbf{P}^{\star}$ as the desired vertex-to-vertex permutation matrix and $\|\bullet\|_F$ is the Frobenius norm defined by (see Appendix A.2):

$$\|\mathbf{W}\|_F^2 = \langle \mathbf{W}, \mathbf{W} \rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2 = \mathrm{tr}(\mathbf{W}^{\top}\mathbf{W}) \tag{2.30}$$

Let:

$$\mathbf{W}_A = \mathbf{U}_A\mathbf{\Lambda}_A\mathbf{U}_A^{\top} \tag{2.31}$$
$$\mathbf{W}_B = \mathbf{U}_B\mathbf{\Lambda}_B\mathbf{U}_B^{\top} \tag{2.32}$$

be the eigen-decompositions of the two matrices with n eigenvalues $\mathbf{\Lambda}_A = \mathrm{Diag}[\alpha_i]$ and $\mathbf{\Lambda}_B = \mathrm{Diag}[\beta_i]$ and n orthonormal eigenvectors, the column vectors of $\mathbf{U}_A$ and $\mathbf{U}_B$.

2.4.2 An Exact Spectral Solution

If there exists a vertex-to-vertex correspondence that makes (2.29) equal to 0, we have:

$$\mathbf{W}_A = \mathbf{P}^\star \mathbf{W}_B \mathbf{P}^{\star\top}. \tag{2.33}$$

This implies that the adjacency matrices of the two graphs should have the same eigenvalues. Moreover, if the eigenvalues are non null and, the matrices $\mathbf{U}_A$ and $\mathbf{U}_B$ have full rank and are uniquely defined by their n orthonormal column vectors (which are the eigenvectors of $\mathbf{W}_A$ and $\mathbf{W}_B$), then $\alpha_i = \beta_i, \forall i,\ 1 \leq i \leq n$ and $\mathbf{\Lambda}_A = \mathbf{\Lambda}_B$. From (2.33) and using the eigen-decompositions of the two graph matrices we obtain:

$$\mathbf{\Lambda}_A = \mathbf{U}_A^\top \mathbf{P}^\star \breve{\mathbf{U}}_B \mathbf{\Lambda}_B \breve{\mathbf{U}}_B^\top \mathbf{P}^{\star\top} \mathbf{U}_A = \mathbf{\Lambda}_B, \tag{2.34}$$

where the matrix $\breve{\mathbf{U}}_B$ is defined by:

$$\breve{\mathbf{U}}_B = \mathbf{U}_B \mathbf{S}. \tag{2.35}$$

Matrix $\mathbf{S} = \text{Diag}[s_i]$, with $s_i = \pm 1$, is referred to as a sign matrix with the property $\mathbf{S}^2 = \mathbf{I}$. Post multiplication of $\mathbf{U}_B$ with a sign matrix takes into account the fact that the eigenvectors (the column vectors of $\mathbf{U}_B$) are only defined up to a sign. Finally we obtain the following permutation matrix:

$$\mathbf{P}^\star = \mathbf{U}_B \mathbf{S} \mathbf{U}_A^\top. \tag{2.36}$$

Therefore, one may notice that there are as many solutions as the cardinality of the set of matrices $\mathbf{S}_n$, *i.e.*, $|\mathbf{S}_n| = 2^n$, and that *not all of these solutions correspond to a permutation matrix*. This means that there exist some matrices $\mathbf{S}^\star$ that exactly make $\mathbf{P}^\star$ a permutation matrix. Hence, all those permutation matrices that satisfy (2.36) are solutions of the exact graph isomorphism problem. Notice that once the permutation has been estimated, one can write that the rows of $\mathbf{U}_B$ can be aligned with the rows of $\mathbf{U}_A$:

$$\mathbf{U}_A = \mathbf{P}^\star \mathbf{U}_B \mathbf{S}^\star. \tag{2.37}$$

The rows of $\mathbf{U}_A$ and of $\mathbf{U}_B$ can be interpreted as isometric embeddings of the two graph vertices: A vertex v_i of $\mathcal{G}_A$ has as coordinates the i^{th} row of $\mathbf{U}_A$. This means that the spectral graph isomorphism problem becomes a point registration problem, where graph vertices are represented by points in $\mathbb{R}^n$. To conclude, the exact graph isomorphism problem has a spectral solution based on: (i) the eigen-decomposition of the two graph matrices, (ii) the ordering of their eigenvalues, and (iii) the choice of a sign for each eigenvector.

2.4.3 The Hoffman-Wielandt Theorem

The Hoffman-Wielandt theorem [Hoffman 1953, Wilkinson 1965] is the fundamental building block of spectral graph isomorphism. The theorem holds for normal matrices; Here, we restrict the analysis to real symmetric matrices, although the generalization to Hermitian matrices is straightforward:

Theorem 1 *(Hoffman and Wielandt) If* $\mathbf{W}_A$ *and* $\mathbf{W}_B$ *are real-symmetric matrices, and if* α_i *and* β_i *are their eigenvalues arranged in increasing order,* $\alpha_1 \leq \ldots \leq \alpha_i \leq \ldots \leq \alpha_n$ *and* $\beta_1 \leq \ldots \leq \beta_i \leq \ldots \leq \beta_n$*, then*

$$\sum_{i=1}^{n}(\alpha_i - \beta_i)^2 \leq \|\mathbf{W}_A - \mathbf{W}_B\|_F^2. \tag{2.38}$$

Proof: The proof is derived from [Wilkinson 1970, Horn 1994]. Consider the eigen-decompositions of matrices $\mathbf{W}_A$ and $\mathbf{W}_B$, (2.31), (2.32). Notice that for the time being we are free to prescribe the ordering of the eigenvalues α_i and β_i and hence the ordering of the column vectors of matrices $\mathbf{U}_A$ and $\mathbf{U}_B$. By combining (2.31) and (2.32) we write:

$$\mathbf{U}_A\mathbf{\Lambda}_A\mathbf{U}_A^\top - \mathbf{U}_B\mathbf{\Lambda}_B\mathbf{U}_B^\top = \mathbf{W}_A - \mathbf{W}_B \tag{2.39}$$

or, equivalently:

$$\mathbf{\Lambda}_A\mathbf{U}_A^\top\mathbf{U}_B - \mathbf{U}_A^\top\mathbf{U}_B\mathbf{\Lambda}_B = \mathbf{U}_A^\top(\mathbf{W}_A - \mathbf{W}_B)\mathbf{U}_B. \tag{2.40}$$

By the unitary-invariance of the Frobenius norm (see Appendix A.2) and with the notation $\mathbf{Z} = \mathbf{U}_A^\top\mathbf{U}_B$ we obtain:

$$\|\mathbf{\Lambda}_A\mathbf{Z} - \mathbf{Z}\mathbf{\Lambda}_B\|_F^2 = \|\mathbf{W}_A - \mathbf{W}_B\|_F^2, \tag{2.41}$$

which is equivalent to:

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \beta_j)^2 z_{ij}^2 = \|\mathbf{W}_A - \mathbf{W}_B\|_F^2. \tag{2.42}$$

The coefficients $x_{ij} = z_{ij}^2$ can be viewed as the entries of a doubly-stochastic matrix $\mathbf{X}$: $x_{ij} \geq 0, \sum_{i=1}^n x_{ij} = 1, \sum_{j=1}^n x_{ij} = 1$. Using these properties, we obtain:

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i - \beta_j)^2 z_{ij}^2 &= \sum_{i=1}^{n}\alpha_i^2 + \sum_{j=1}^{n}\beta_j^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}^2\alpha_i\beta_j \\ &\geq \sum_{i=1}^{n}\alpha_i^2 + \sum_{j=1}^{n}\beta_j^2 - 2\max_{Z}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}^2\alpha_i\beta_j\right\}.\end{aligned} \tag{2.43}$$

Hence, the minimization of (2.42) is equivalent to the maximization of the last term in (2.43). We can modify our maximization problem to admit all the doubly-stochastic matrices. In this way, we seek an extremum over a convex compact set. The maximum over this compact set is larger than or equal to our maximum:

$$\max_{Z\in O_n}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} z_{ij}^2\alpha_i\beta_j\right\} \leq \max_{X\in\mathcal{D}_n}\left\{\sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}\alpha_i\beta_j\right\} \tag{2.44}$$

where O_n is the set of orthogonal matrices and $\mathcal{D}_n$ is the set of doubly stochastic matrices (see Appendix A.1). Let $c_{ij} = \alpha_i\beta_j$ and hence one can write that the right term in the equation above as the dot-product of two matrices:

$$\langle\mathbf{X}, \mathbf{C}\rangle = \mathrm{tr}(\mathbf{X}\mathbf{C}) = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij}c_{ij}. \tag{2.45}$$

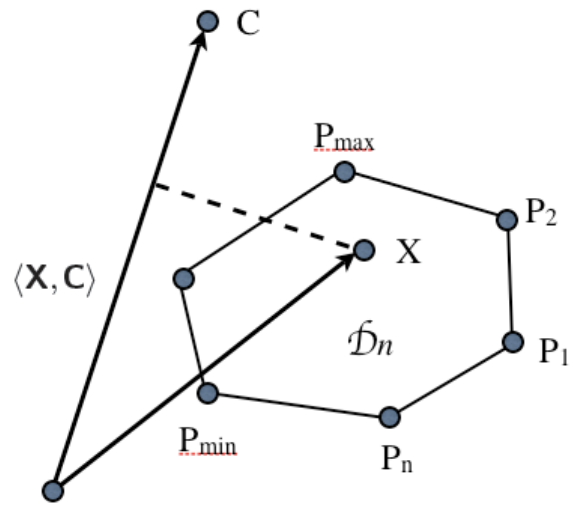


Figure 2.11: This figure illustrates the maximization of the dot-product $\langle \mathbf{X}, \mathbf{C} \rangle$. The two matrices can be viewed as vectors of dimension n^2. Matrix $\mathbf{X}$ belongs to a compact convex set whose extreme points are the permutation matrices $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_n$. Therefore, the projection of this set (*i.e.*, $\mathcal{D}_n$) onto $\mathbf{C}$ has projected permutation matrices at its extremes, namely $\langle \mathbf{P}_{\min}, \mathbf{X} \rangle$ and $\langle \mathbf{P}_{\max}, \mathbf{X} \rangle$ in this example.

Therefore, this expression can be interpreted as the projection of $\mathbf{X}$ onto $\mathbf{C}$, see Figure 2.11. The Birkhoff theorem (Appendix A.1) tells us that the set $\mathcal{D}_n$ of doubly stochastic matrices is a compact convex set. We obtain that the extrema (minimum and maximum) of the projection of $\mathbf{X}$ onto $\mathbf{C}$ occur at the projections of one of the extreme points of this convex set, which correspond to permutation matrices. Hence, the maximum of $\langle \mathbf{X}, \mathbf{C} \rangle$ is $\langle \mathbf{P}_{\max}, \mathbf{X} \rangle$ and we obtain:

$$\max_{X \in \mathcal{D}_n} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{n} x_{ij} \alpha_i \beta_j \right\} = \sum_{i=1}^{n} \alpha_i \beta_{\pi(i)}. \tag{2.46}$$

By substitution in (2.43) we obtain:

$$\sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i - \beta_j)^2 z_{ij}^2 \geq \sum_{i=1}^{n} (\alpha_i - \beta_{\pi(i)})^2. \tag{2.47}$$

If the eigenvalues are in increasing order then the permutation that satisfies theorem 2.38 is the identity matrix, *i.e.*, $\pi(i) = i$. Indeed, let's assume that for some indices k and $k+1$ we have: $\pi(k) = k+1$ and $\pi(k+1) = k$. Since $\alpha_k \leq \alpha_{k+1}$ and $\beta_k \leq \beta_{k+1}$, the following inequality holds:

$$(\alpha_k - \beta_k)^2 + (\alpha_{k+1} - \beta_{k+1})^2 \leq (\alpha_k - \beta_{k+1})^2 + (\alpha_{k+1} - \beta_k)^2 \tag{2.48}$$

and hence (2.38) holds. ■

Corollary 1.1 *The inequality (2.38) becomes an equality when the eigenvectors of* $\mathbf{W}_A$ *are aligned with the eigenvectors of* $\mathbf{W}_B$ *up to a sign ambiguity:*

$$\mathbf{U}_B = \mathbf{U}_A\mathbf{S}. \tag{2.49}$$

Proof: Since the minimum of (2.42) is achieved for $\mathbf{X} = \mathbf{I}$ and since the entries of $\mathbf{X}$ are z_{ij}^2, we have that $z_{ii} = \pm 1$, which corresponds to $\mathbf{Z} = \mathbf{S}$. ■

Corollary 1.2 *If* $\mathbf{Q}$ *is an orthogonal matrix, then*

$$\sum_{i=1}^{n}(\alpha_i - \beta_i)^2 \leq \|\mathbf{W}_A - \mathbf{Q}\mathbf{W}_B\mathbf{Q}^\top\|_F^2. \tag{2.50}$$

Proof: Since the eigen-decomposition of matrix $\mathbf{Q}\mathbf{W}_B\mathbf{Q}^\top$ is $(\mathbf{Q}\mathbf{U}_B)\mathbf{\Lambda}_B(\mathbf{Q}\mathbf{U}_B)^\top$ and since it has the same eigenvalues as $\mathbf{W}_B$, the inequality (2.50) holds and hence corollary 1.2. ■

These corollaries will be useful in the case of spectral graph matching methods presented below.

2.4.4 Umeyama's Method

The exact spectral matching solution presented in Section 2.4.2 finds a permutation matrix satisfying (2.36). This requires an exhaustive search over the space of all possible 2^n matrices. Umeyama's method presented in [Umeyama 1988] proposes a relaxed solution to this problem as outlined below.

Umeyama [Umeyama 1988] addresses the problem of *weighted graph matching* within the framework of spectral graph theory. He proposes two methods, the first for *undirected weighted graphs* and the second for *directed weighted graphs*. The adjacency matrix is used in both cases. Let us consider the case of undirected graphs. The eigenvalues are (possibly with multiplicities):

$$\mathbf{W}_A: \quad \alpha_1 \leq \ldots \leq \alpha_i \leq \ldots \leq \alpha_n \tag{2.51}$$

$$\mathbf{W}_B: \quad \beta_1 \leq \ldots \leq \beta_i \leq \ldots \leq \beta_n. \tag{2.52}$$

Theorem 2 *(Umeyama) If* $\mathbf{W}_A$ *and* $\mathbf{W}_B$ *are real-symmetric matrices with n distinct eigenvalues (that can be ordered),* $\alpha_1 < \ldots < \alpha_i < \ldots < \alpha_n$ *and* $\beta_1 < \ldots < \beta_i < \ldots < \beta_n$*, the minimum of :*

$$J(\mathbf{Q}) = \|\mathbf{W}_A - \mathbf{Q}\mathbf{W}_B\mathbf{Q}^\top\|_F^2 \tag{2.53}$$

is achieved for:

$$\mathbf{Q}^\star = \mathbf{U}_A\mathbf{S}\mathbf{U}_B^\top \tag{2.54}$$

and hence (2.50) becomes an equality:

$$\sum_{i=1}^{n}(\alpha_i-\beta_i)^2=\|\mathbf{W}_A-\mathbf{Q}^\star\mathbf{W}_B\mathbf{Q}^{\star\top}\|_F^2. \tag{2.55}$$

Proof: The proof is straightforward. By corollary 1.2, the Hoffman-Wielandt theorem applies to matrices $\mathbf{W}_A$ and $\mathbf{Q}\mathbf{W}_B\mathbf{Q}^\top$. By corollary 1.1, the equality (2.55) is achieved for:

$$\mathbf{Z}=\mathbf{U}_A^\top\mathbf{Q}^\star\mathbf{U}_B=\mathbf{S} \tag{2.56}$$

and hence (2.54) holds. ■

Notice that (2.54) can be written as:

$$\mathbf{U}_A=\mathbf{Q}^\star\mathbf{U}_B\mathbf{S} \tag{2.57}$$

which is a *relaxed* version of (2.37): The permutation matrix in the exact isomorphism case is replaced by an orthogonal matrix.

A Heuristic for Spectral Graph Matching: Let us consider again the exact solution outlined in Section 2.4.2. Umeyama suggests a heuristic in order to avoid exhaustive search over all possible 2^n matrices that satisfy (2.36). One may easily notice that:

$$\|\mathbf{P}-\mathbf{U}_A\mathbf{S}\mathbf{U}_B^\top\|_F^2=2n-2\mathrm{tr}(\mathbf{U}_A\mathbf{S}(\mathbf{P}\mathbf{U}_B)^\top). \tag{2.58}$$

Using Umeyama's notations, $\bar{\mathbf{U}}_A=[|u_{ij}|],\bar{\mathbf{U}}_B=[|v_{ij}|]$ (the entries of $\bar{\mathbf{U}}_A$ are the absolute values of the entries of $\mathbf{U}_A$), one may further notice that:

$$\mathrm{tr}(\mathbf{U}_A\mathbf{S}(\mathbf{P}\mathbf{U}_B)^\top)=\sum_{i=1}^{n}\sum_{j=1}^{n}s_ju_{ij}v_{\pi(i)j}\le\sum_{i=1}^{n}\sum_{j=1}^{n}|u_{ij}||v_{\pi(i)j}|=\mathrm{tr}(\bar{\mathbf{U}}_A\bar{\mathbf{U}}_B^\top\mathbf{P}^\top). \tag{2.59}$$

The minimization of (2.58) is equivalent to the maximization of (2.59) and the maximal value that can be attained by the latter is n. Using the fact that both $\mathbf{U}_A$ and $\mathbf{U}_B$ are orthogonal matrices, one can easily conclude that:

$$\mathrm{tr}(\bar{\mathbf{U}}_A\bar{\mathbf{U}}_B^\top\mathbf{P}^\top)\le n. \tag{2.60}$$

Umeyama concludes that when the two graphs are isomorphic, the optimum permutation matrix maximizes $\mathrm{tr}(\bar{\mathbf{U}}_A\bar{\mathbf{U}}_B^\top\mathbf{P}^\top)$ and this can be solved by the Hungarian algorithm [Burkard 2009].

When the two graphs are not exactly isomorphic, theorem 1 and theorem 2 allow us to relax the permutation matrices to the group of orthogonal matrices. Therefore with similar arguments as above we obtain:

$$\mathrm{tr}(\mathbf{U}_A\mathbf{S}\mathbf{U}_B^\top\mathbf{Q}^\top)\le\mathrm{tr}(\bar{\mathbf{U}}_A\bar{\mathbf{U}}_B^\top\mathbf{Q}^\top)\le n. \tag{2.61}$$

The permutation matrix obtained with the Hungarian algorithm can be used as an initial solution that can then be improved by some hill-climbing or relaxation technique [Umeyama 1988].

The spectral matching solution presented in this section is not directly applicable to large graphs. In the following sections, we introduce the notion of dimensionality reduction for graphs, which can lead to a tractable graph matching solution.

2.5 Manifold Learning

In this section, we introduce the notion of *Manifold Learning* for dimensionality reduction, providing a platform for a method (presented in the next section) to perform the dimensionality reduction of graphs.

Data mining techniques are commonly used in computer vision for analyzing the large amount of data while performing recognition and classification tasks. However, the majority of this data consist of different types of datum (or their feature representation) that typically exists in a high dimensional space, *e.g.*, image data sets with millions of images where each image can have a feature representation of dimension 10^3 to 10^4. This is known as the *the curse of dimensionality* and is an important bottleneck for majority of the existing data mining techniques. The intuitive solution is to find a way to reduce the dimensionality of the input data while keeping the maximum information intact. The idea is to adapt a generative approach and find a set of latent (hidden) variables by fitting a mathematical model for data generation. These latent variables are then used for a reduced dimensional data representation. Among the large number of approaches for dimensionality reduction, we will focus on methods with assumption that the observed high dimensional data is often much simpler and typically sampled from a lower dimensional manifold. Thus, a dimensionality reduction task can be achieved by discovering the underlying lower dimensional manifold embedded in the higher dimensional Euclidean space. This is also called the *Manifold Learning*.

Given a set of n data points $\mathbf{X}_{D\times n} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^D$, we assume that they approximately lie on a smooth manifold $\mathcal{X}$. The intrinsic dimension of $\mathcal{X}$ is $d = dim(\mathcal{X})$ with $d \ll D$. Let the desired output (latent) space be a d dimensional space $\mathbb{R}^d$ with set of data points represented as $\mathbf{Y}_{d\times n} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^d$. In other words, we hypothesize that $\mathcal{X}$ is the image of some latent variable domain $\mathcal{Y} \subset \mathbb{R}^d$ under a smooth mapping $\Phi : \mathcal{Y} \mapsto \mathbb{R}^D$. The idea of manifold learning is to find the low dimensional coordinates $\mathbf{y}_i \in \mathcal{Y}$ for the corresponding high dimensional points $\mathbf{x}_i \in \mathcal{X}$.

Manifold Learning can be largely classified as linear and non-linear methods. The linear methods try to find a reduced dimensional subspace $\mathbb{R}^d \subset \mathbb{R}^D$. On the other hand, the non-linear methods seek a global parametrization of a manifold $\mathcal{Y} \subset \mathbb{R}^d$. The non-linear methods can further be divided into purely global methods and methods recovering the global manifold structure from local information. Here, we present an overview of the existing manifold learning techniques.

2.5.1 Linear Dimensionality Reduction

In this section, we will mainly present two linear dimensionality reduction methods, namely, the principal component analysis and multi-dimensional scaling.

2.5.1.1 Principal Component Analysis

The Principal Component Analysis (PCA) is a standard dimensionality reduction technique in multivariate statistics. The idea is to consider as input a vectorial data and output a new orthogonal coordinate system representation while capturing most of the variance of the data set. In particular, PCA can be formulated as the problem of finding a d-dimensional subspace fitting that approximates the input $\mathbb{R}^D$ data in the least square sense. This d-dimensional subspace uses as its basis, the directions of maximum variance; which boils down to computing the principal eigenvectors of the covariance matrix of the data. The PCA algorithm from maximum variance perspective is outlined in Algorithm 1.

Algorithm 1 *Principal Component Analysis* (PCA)

Input: : $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ with $\mathbf{x}_i \in \mathbb{R}^D$.

Output: : A set of orthonormal basis vectors $\mathbf{U}_{D\times d} = [\mathbf{u}_1, \ldots, \mathbf{u}_d]$ for a d-dimensional subspace.

1: Let $\mathbf{u} \in \mathbb{R}^D$ be the 1-dimensional subspace (with $\mathbf{u}^T\mathbf{u} = 1$), that maximizes the variance of data $\mathbf{X}$.

2: Project each data point $\mathbf{x}_i$ to a scalar value $\mathbf{u}^T\mathbf{x}_i$ and represent the mean of the projected data as $\mathbf{u}^T\mathbf{x}_0$.

3: The variance of the projected data can be written as: $\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{u}^T\mathbf{x}_i - \mathbf{u}^T\mathbf{x}_0\right)^2 = \mathbf{u}^T\Sigma_{\mathbf{X}}\mathbf{u}$, where $\Sigma_{\mathbf{X}} = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{x}_i - \mathbf{x}_0\right)^2$ is the covariance matrix of input data.

4: Thus, the problem of maximum variance can be stated in terms of Rayleigh quotient (see Section A.4) as: $\mathbf{u}^* = \arg\max_{\mathbf{u}} \mathbf{u}^T\Sigma_{\mathbf{X}}\mathbf{u}$.

5: The solution is the set of eigenvectors of $\Sigma_{\mathbf{X}}$, *i.e.*, $\{\mathbf{u}_1, \ldots, \mathbf{u}_d\}$ corresponding to the first d largest eigenvalues $\{\lambda_1 \geq \cdots \geq \lambda_d\}$.

2.5.1.2 Multi-Dimensional Scaling

Multidimensional Scaling (MDS) is a class of linear dimensionality reduction techniques that are very similar to PCA except that they can act as a coordinate-free techniques that can also operate on non-vector spaces. The classical MDS uses a matrix of squared distance between pairs of points $(\mathbf{x}_i, \mathbf{x}_j)$ in the data set instead of a covariance matrix that was used in the PCA. The pairwise distance matrix is a $n \times n$ matrix. The algorithm seeks a d-dimensional Euclidean coordinates for each data point so that the pairwise distance of their Euclidean coordinates match the original pairwise distance as closely as possible, thus formulating it as an optimization problem. This can be generalized to use a *Gram* matrix, a dis-similarity matrix, instead of the pairwise distance matrix, thereby replacing the squared distances with the dot products and allowing to operate on no-metric spaces.. Existing literature has large number of several cost functions and different minimization algorithms, proposed in the context of desired optimization problem, which we will not discuss in detail in this text. However, one

common optimal solution derives the eigenvectors of the Gram matrix weighted with corresponding eigenvalues as the d-dimensional Euclidean coordinates of data points. Please refer to [Borg 2005] for a detail discussion on MDS.

One interesting generalization of MDS is known as the *generalized multidimensional scaling* (GMDS), in which the target space is an arbitrary smooth non-Euclidean space. In case when the dis-similarities are distances on a surface and the target space is another surface, GMDS allows finding the minimum-distortion embedding of one surface into another. This was proposed in the context of finding a surface mapping solution in [Bronstein 2006].

2.5.2 Non-Linear Dimensionality Reduction

The linear dimensionality reduction methods fails when data is sampled from a non-linear manifold. There exists a class of non-linear methods that assume that data is sampled from a d-dimensional sub-manifold $\mathcal{Y}$ embedded in $\mathbb{R}^D$. Majority of these methods rely on adjacency graphs that are locally defined on data points. The idea is to define the notion of geodesic distance between data points as they are assumed to lie on a manifold surface.

The Kernel Trick: The majority of existing non-linear dimensionality reduction methods boil down to computing the eigen analysis of positive definite matrices that are constructed by measuring the pairwise dis-similarities/distances between data points. The *kernel trick* facilitates the analysis of input data originally sampled from a non-linear space into a high dimensional (possibly) linear space. The idea is to seek a mapping of data points from non-linear space into a high dimensional (inner product) space where hopefully, the data is linearly distributed. However, finding an explicit mapping function can be a tedious task. Instead, the kernel trick allows one to reinterpret the each entry of a dis-similarity matrix as a dot product between a pair of data points in the mapped high dimensional (linear) space by the means of a kernel (function) without actually computing the explicit mapping function. The kernel function acts as a (non-linear) symmetric real valued function over the pairs of data points, *i.e.*, given a space $\mathcal{X}$, the kernel function is denoted as:

$$\mathcal{K} : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}, \quad \text{such that,} \quad (\mathbf{x}_i, \mathbf{x}_j) \mapsto \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \quad \forall \mathbf{x}_i, \mathbf{x}_j \in \mathbf{X} \subseteq \mathcal{X}. \tag{2.62}$$

We will introduce a kernel function in Chapter 6 while introducing the heat kernel based multi-scale analysis of 3D shapes. Please refer to [Bishop 2006] for details on kernel trick.

Popular Methods: Here, we briefly mention some popular non-linear dimensionality reduction methods in manifold learning.

- **Kernel PCA**
  The kernel PCA [Schölkopf 1998b] is an extension of linear PCA method where the covariance matrix is kernelized with some non-linear kernel function. This allows one

to interpret the entries of a kernel matrix as the covariance matrix of data points that are mapped in a high dimensional space where it is assumed to be linearly distributed. This allows one to perform the non-linear dimensionality reduction of data without explicitly mapping it to a higher dimensional space, using the kernel trick.

- **Isomap**
  This method was first presented in [Tenenbaum 2000] and is a generalization of the classical MDS where the pairwise distances were replaced with the geodesic distances that are commonly computed as the shortest distances on graphs.

- **Locally Linear Embedding (LLE)**
  First introduced in [Roweis 2000], *LLE* aims at recovering the low dimensional geometry of the data by preserving the local structure of data points. LLE assumes that the small neighborhood can be approximated by the linear manifolds where the position of each point can be reconstructed from the weighted linear combination of its nearest neighbors. This is followed by seeking a low dimensional space that best preserves the reconstructing weights.

- **Laplacian Eigenmaps**
  Introduced by Belkin [Belkin 2003], the Laplacian Eigenmaps embed the input data $\mathbf{X}$ in a d-dimensional space such that the small neighborhood edges in an adjacency graphs are preserved. We will discuss this method in detail in the next Chapter 3 where we introduce the notion of pose-invariant shape representation.

- **Hessian Eigenmaps**
  This method is also known as the Hessian Locally Linear Embedding (*HLLE*) and was proposed in [Donoho 2003]. HLLE can be viewed as a practical modification of the Locally Linear Embedding and in theoretical framework as the modification of the Laplacian Eigenmaps where a quadratic form based on the Hessian is substituted in place of one based on the Laplacian.

- **Diffusion Maps**
  Diffusion maps, presented in [Coifman 2006], propose to build upon random walk analysis of similarity graphs and use the random walk matrix as the Gram matrix. This random walk matrix can be parametrized by a scale parameter relating to length of paths in a random walk over the graph. Thus, the diffusion maps provides a scale dependent embedding of input data in terms of random walk on the similarity graph.

- **Heat Kernel Embedding (HKE)**
  The Heat Kernel Embedding builds upon the classical framework of heat diffusion on graphs in order to provide a scale parametrized representation of data. The Gram matrix used here is the heat matrix, which is the fundamental solution of the (heat)-diffusion equation on graphs. This has been discussed in the context of shape analysis in [Sun 2009, Ovsjanikov 2010]. We will discuss this method in detail in Chapter 6 while outlining the heat kernel framework for multi-scale shape representation.

- **Local Tangent Space Alignment (LTSA)**
  LTSA, first proposed in [Zhang 2005], is a non-linear dimensionality reduction method, which is based on the intuition that when a manifold is correctly unfolded, all of the tangent hyper-planes to the manifold will become aligned. The method first computes the tangent space at every point using the d-dimensional principal component analysis in the local neighborhood and then optimizes to find an embedding that aligns these tangent spaces.

2.6 Spectral Graph Embedding

The *Spectral Graph Embedding* is an important concept in the spectral graph theory that allows one to perform a dimensionality reduction of graphs using the discrete Laplace operator defined on graphs. This is similar to the embedding of the Riemannian manifold surfaces using the spectrum of the Laplace-Beltrami operator presented in Section 2.3.3.1. The idea is quite similar to the non-linear dimensionality reduction in manifold learning (Section 2.5.2) where a graph can be assumed to be a discretized manifold lying in a lower dimensional subspace.

The dimensionality reduction of graphs is eminent in certain scenarios, *e.g.*, solving the problem of graph matching (Section 2.4) where one needs to compute the eigen-decomposition of graph adjacency matrix. For *large* and *sparse* graphs, the results of Section 2.4.1 and Umeyama's method (Section 2.4.4) hold only *weakly*. Indeed, one cannot guarantee that all the eigenvalues have multiplicity equal to one: the presence of symmetries causes some of eigenvalues to have an algebraic multiplicity greater than one. Under these circumstances and due to numerical approximations, it might not be possible to properly order the eigenvalues. Moreover, for very large graphs with thousands of vertices, it is not practical to compute all its eigenvalue-eigenvector pairs. This means that one has to devise a method that is able to perform graph isomorphism using a small set of eigenvalues and eigenvectors. Spectral dimensionality reduction techniques in spectral graph theory provides such a solution where one seeks the intrinsic dimension of underlying manifold. The eigen-decomposition of the graph Laplacian matrices is a popular choice while performing the dimensionality reduction of graphs [Belkin 2003].

In this section, first we will introduce various graph Laplacian matrices and their spectral analysis. Next, we will study the spectral properties of the graph Laplacian matrices and derive a principal component analysis analogy for the Laplacian eigenvectors, yielding a spectral embedding of the graph. Finally, we will present a heuristic for choosing the size of the graph embedding.

2.6.1 Graph Laplacian Matrix

We can now build the concept of the *graph Laplacian operator*. We consider the following variants of the Laplacian matrix [Chung 1997, von Luxburg 2007, Grady 2010]:

- The *unnormalized Laplacian*, which is also referred to as the *combinatorial Laplacian* $\mathbf{L}$,
- the *normalized Laplacian* $\tilde{\mathbf{L}}$, and
- the *random-walk Laplacian* $\tilde{\mathbf{L}}_R$, also referred to as the *discrete Laplace operator*.

In more detail, we have:

$$\begin{aligned} \mathbf{L} &= \mathbf{D} - \mathbf{W} && (2.63)\\ \tilde{\mathbf{L}} &= \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2} = \mathbf{I} - \tilde{\mathbf{W}} && (2.64)\\ \tilde{\mathbf{L}}_R &= \mathbf{D}^{-1}(\mathbf{D} - \mathbf{W}) = \mathbf{I} - \tilde{\mathbf{W}}_R && (2.65) \end{aligned}$$

with the following relations between these matrices:

$$\begin{aligned} \mathbf{L} &= \mathbf{D}^{1/2}\tilde{\mathbf{L}}\mathbf{D}^{1/2} = \mathbf{D}\tilde{\mathbf{L}}_R && (2.66)\\ \tilde{\mathbf{L}} &= \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} = \mathbf{D}^{1/2}\tilde{\mathbf{L}}_R\mathbf{D}^{-1/2} && (2.67)\\ \tilde{\mathbf{L}}_R &= \mathbf{D}^{-1/2}\tilde{\mathbf{L}}\mathbf{D}^{1/2} = \mathbf{D}^{-1}\mathbf{L}. && (2.68) \end{aligned}$$

2.6.2 The Spectral Theorem

The spectral theorem states the condition under which an operator or a matrix can be diagonalized. The theorem states that a linear transformation $\mathbf{T}$ applied over vector $\mathbf{f}$ can be written as the linear combination of the eigenvectors of $\mathbf{T}$, each multiplied with corresponding eigenvalue and scalar product of eigenvector with $\mathbf{f}$. The theorem is valid for all self-adjoint linear transforms (*i.e.*, transforms given by real symmetric and Hermitian matrices) and for more general class of (complex) normal matrices. Formally, we can write:

$$\mathbf{T}(\mathbf{f}) = \sum_{i=1:n} \lambda_i \langle \mathbf{u}_i, \mathbf{f} \rangle \mathbf{u}_i, \tag{2.69}$$

where $\{\mathbf{u}_i\}_{i=1:n}$ and $\{\lambda_i\}_{i=1:n}$ denotes the eigenvectors and eigenvalues of the linear operator or matrix $\mathbf{T}$.

This can be used to diagonalize the Laplacian matrices $\mathbf{L}, \tilde{\mathbf{L}}$ and $\tilde{\mathbf{L}}_R$. Let us consider the case of combinatorial Laplacian matrix $\mathbf{L}$. The spectral theorem suggests that matrix $\mathbf{L}$ can be diagonalized to yield a complete set of orthonormal basis functions $\{\mathbf{u}_i\}_{i=1:n}$ such that

$$\begin{aligned} \langle \mathbf{u}_i, \mathbf{u}_j \rangle &= 0 \quad \forall i \neq j && (2.70)\\ \langle \mathbf{u}_i, \mathbf{u}_j \rangle &= 1 \quad \forall i = j. && (2.71) \end{aligned}$$

Using these orthonormal basis functions $\mathbf{L}$ can be reconstructed as follows:

$$\mathbf{L} = \sum_{i=1:n} \lambda_i \mathbf{u}_i \mathbf{u}_i^T = \mathbf{U}\Lambda\mathbf{U}^T, \tag{2.72}$$

where $\mathbf{U}$ represents a matrix of eigenvectors (set as columns), *i.e.*, $\mathbf{U} = [\mathbf{u}_1|\mathbf{u}_2|\dots|\mathbf{u}_n]$ and Λ represents a sparse diagonal matrix with eigenvalues as its diagonal entries, *i.e.*, $\Lambda = \text{Diag}(\lambda_1, \dots, \lambda_n)$.

The same analysis is also applicable to the normalized Laplacian matrix $\tilde{\mathbf{L}}$. In the case of random-walk Laplacian, unlike other the two Laplacian matrices, $\tilde{\mathbf{L}}_R$ is not a symmetric matrix. Hence, the spectral theorem is not directly applicable. In the next section, we will see how the close relation between spectral analysis of these Laplacian matrices can be used to compute the eigenvalues and the eigenvectors of $\tilde{\mathbf{L}}_R$ matrix.

2.6.3 Spectral Analysis of the Laplacian Matrices

The *spectral analysis* finds the eigenvalues for a general square $(n \times n)$ matrix $\mathbf{M}$ by solving the *characteristic equation*:

$$\det(\mathbf{M} - \lambda\mathbf{I}) = 0, \tag{2.73}$$

where 'det' represents matrix determinant and $\mathbf{I}$ is a $(n \times n)$ identity matrix. The computation of the determinant yields the *characteristic polynomial* of the matrix. Thus, Eq. (2.73) becomes a system of linear equations and its solution is the roots of the characteristic polynomial. These are also called the eigenvalues of matrix $\mathbf{M}$, typically denoted as λ_i. For each λ, there exist a vectors $\mathbf{u} \neq 0$ such that:

$$(\mathbf{M} - \lambda\mathbf{I})\mathbf{u} = 0. \tag{2.74}$$

The set of vectors corresponding to eigenvalues are called eigenvectors of matrix $\mathbf{M}$. Thus, reorganizing Eq. (2.74), the eigen-decomposition of matrix $\mathbf{M}$ can be obtained by solving the following system of linear equation:

$$\mathbf{M}\mathbf{u} = \lambda\mathbf{u}. \tag{2.75}$$

Hence, the spectral analysis of the combinatorial and normalized Laplacian matrices, derived in Eq. (2.63) and Eq. (2.64), can be written as follows:

$$\mathbf{L}\mathbf{u} = (\mathbf{D} - \mathbf{W})\mathbf{u} = \lambda\mathbf{u} \quad (\textit{combinatorial Laplacian}), \tag{2.76}$$

$$\tilde{\mathbf{L}}\mathbf{u} = \mathbf{D}^{-1/2}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-1/2}\mathbf{u} = \lambda\mathbf{u} \quad (\textit{normalized Laplacian}). \tag{2.77}$$

In the case of non-symmetric random-walk Laplacian in Eq. (2.65), using the substitution $\mathbf{v} = \mathbf{D}^{-1/2}\mathbf{u}$ in Eq. (2.77), we can write:

$$\tilde{\mathbf{L}}_R\mathbf{v} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda\mathbf{v} \quad (\textit{random walk Laplacian}). \tag{2.78}$$

The linear system can be transformed to a generalized eigenvalue problem as follows:

$$(\mathbf{D} - \mathbf{W})\mathbf{v} = \lambda\mathbf{D}\mathbf{v} \quad (\textit{generalized normalized Laplacian}). \tag{2.79}$$

These different eigen-decompositions have been applied for solving different problems in image processing and data clustering. For example, the normalized Laplacian in Eq. (2.77)

was thoroughly used in [Chung 1997] for deriving theoretical results and also employed in *normalized cut* based image segmentation algorithm proposed in [Shi 2000] as well as in the spectral clustering algorithm presented in [Ng 2001]. The expression in Eq. (2.78) was used in [Meila 2001, Qiu 2007] in the context of understanding random walk on graphs. A Laplacian eigenmap algorithm for dimensionality reduction was proposed in [Belkin 2003] using the generalized Laplacian eigen-decomposition shown in Eq. (2.79).

The explicit relationship among these Laplacians as shown in Eqs. (2.66-2.68) as well as the intrinsic relationship in their eigen-decompositions in Eqs. (2.76-2.78) is very important and implies that spectral analysis serve as the basis for developing spectral graph methods for various graph analysis tasks such as the segmentation, clustering and dimensionality reduction. Another important fact about these equivalence in the formulation is that they allow us to use more stable, accurate and efficient eigen-decomposition system, *e.g.*, a more stabilized generalized Laplacian can be used to compute the eigen-decomposition of the normalized Laplacian, which in turn, can be used to compute the eigen-decomposition of non-symmetric random-walk Laplacian. Nevertheless, there can be need for different algorithms for the eigen-decomposition of different Laplacian matrices depending on their size, sparseness and desired size of spectrum. It is also important to practice caution when inverting degree matrix when a graph can have either completely isolated vertex or a very low degree vertex. Please refer to [Golub 1989] for details on various eigen-decomposition algorithms.

2.6.4 Spectral Properties of Graph Laplacian

The spectral properties of the Laplacian matrices introduced in Section 2.6.1 have been thoroughly studied. The three graph Laplacians definitions presented in 2.6.1 are *positive semi-definite* matrices yielding *non-negative eigenvalues*. Spectral properties of these variants are summarized in Table 2.1. We derive some subtle properties of the combinatorial Laplacian

Laplacian	Null space	Eigenvalues	Eigenvectors
$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$	$\mathbf{u}_1 = \mathbf{1}$	$0 = \lambda_1 < \lambda_2 \leq \ldots \leq \lambda_n$	$\mathbf{u}_{i>1}^\top \mathbf{1} = 0, \mathbf{u}_i^\top \mathbf{u}_j = \delta_{ij}$
$\tilde{\mathbf{L}} = \tilde{\mathbf{U}}\mathbf{\Gamma}\tilde{\mathbf{U}}^\top$	$\tilde{\mathbf{u}}_1 = \mathbf{D}^{1/2}\mathbf{1}$	$0 = \gamma_1 < \gamma_2 \leq \ldots \leq \gamma_n$	$\tilde{\mathbf{u}}_{i>1}^\top \mathbf{D}^{1/2}\mathbf{1} = 0, \tilde{\mathbf{u}}_i^\top \tilde{\mathbf{u}}_j = \delta_{ij}$
$\tilde{\mathbf{L}}_R = \mathbf{T}\mathbf{\Gamma}\mathbf{T}^{-1}, \mathbf{T} = \mathbf{D}^{-1/2}\tilde{\mathbf{U}}$	$\mathbf{t}_1 = \mathbf{1}$	$0 = \gamma_1 < \gamma_2 \leq \ldots \leq \gamma_n$	$\mathbf{t}_{i>1}^\top \mathbf{D}\mathbf{1} = 0, \mathbf{t}_i^\top \mathbf{D}\mathbf{t}_j = \delta_{ij}$

Table 2.1: Summary of the spectral properties of the Laplacian matrices. Assuming a connected graph, the null eigenvalue (λ_1, γ_1) has multiplicity one. The first non null eigenvalue (λ_2, γ_2) is known as the Fiedler value and its multiplicity is, in general, equal to one. The associated eigenvector is denoted as the Fiedler vector [Chung 1997].

which will be useful for the task of graph isomorphism. In particular, we show that the eigenvectors of the combinatorial Laplacian can be interpreted as the directions of maximum variance (*i.e.*, the principal components) of the associated embedded graph representation. We

note that the embeddings of the normalized and random-walk Laplacians have different spectral properties which make them less interesting for graph isomorphism, *i.e.*, Appendix A.3.

The combinatorial Laplacian. Let $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ be the spectral decomposition of the combinatorial Laplacian with $\mathbf{U}\mathbf{U}^\top = \mathbf{I}$. Let $\mathbf{U}$ be written as:

$$\mathbf{U} = \begin{bmatrix} u_{11} & \dots & u_{1k} & \dots & u_{1n} \\ \vdots & & \vdots & & \vdots \\ u_{n1} & \dots & u_{nk} & \dots & u_{nn} \end{bmatrix} \tag{2.80}$$

Each column of $\mathbf{U}$, $\mathbf{u}_k = (u_{1k} \dots u_{ik} \dots u_{nk})^\top$ is an eigenvector associated with the eigenvalue λ_k. From the definition of $\mathbf{L}$ in Eq. (2.63) (see [Belkin 2003]), one can easily see that $\lambda_1 = 0$ and that $\mathbf{u}_1 = \mathbf{1}$ (a constant vector). Hence, $\mathbf{u}_{k\geq 2}^\top \mathbf{1} = 0$ and by combining this with $\mathbf{u}_k^\top \mathbf{u}_k = 1$, we derive the following proposition:

Proposition 1 *The components of the non-constant eigenvectors of the combinatorial Laplacian satisfy the following constraints:*

$$\sum_{i=1}^{n} u_{ik} = 0, \quad \forall k, 2 \leq k \leq n \tag{2.81}$$

$$-1 < u_{ik} < 1, \quad \forall i,k, 1 \leq i \leq n, 2 \leq k \leq n. \tag{2.82}$$

Assuming a connected graph, λ_1 has multiplicity equal to one [von Luxburg 2007]. Let us organize the eigenvalues of $\mathbf{L}$ in increasing order: $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$. We prove the following proposition [Chung 1997]:

Proposition 2 *For all $k \leq n$, we have $\lambda_k \leq 2\max_i(d_i)$, where d_i is the degree of vertex i.*

Proof: The largest eigenvalue of $\mathbf{L}$ corresponds to the maximization of the Rayleigh quotient, *i.e.*,

$$\lambda_n = \max_{\mathbf{u}} \frac{\mathbf{u}^\top \mathbf{L}\mathbf{u}}{\mathbf{u}^\top \mathbf{u}}. \tag{2.83}$$

We have $\mathbf{u}^\top \mathbf{L}\mathbf{u} = \sum_{e_{ij}} w_{ij}(u_i - u_j)^2$. From the inequality $(a-b)^2 \leq 2(a^2+b^2)$ we obtain:

$$\lambda_n \leq \frac{2\sum_{e_{ij}} w_{ij}(u_i^2 + u_j^2)}{\sum_i u_i^2} = \frac{2\sum_i d_i u_i^2}{\sum_i u_i^2} \leq 2\max_i(d_i). \quad \blacksquare \tag{2.84}$$

This ensures an upper limit on the eigenvalues of $\mathbf{L}$. By omitting the zero eigenvalue and associated eigenvector, we can rewrite $\mathbf{L}$ as:

$$\mathbf{L} = \sum_{k=2}^{n} \lambda_k \mathbf{u}_k \mathbf{u}_k^\top. \tag{2.85}$$

Each entry u_{ik} of an eigenvector $\mathbf{u}_k$ can be interpreted as a real-valued function that projects a graph vertex v_i onto that vector. The mean and variance of the set $\{u_{ik}\}_{i=1}^{n}$ are therefore a measure of how the graph *spreads* when projected onto the k-th eigenvector. This is clarified by the following result:

Proposition 3 *For each eigen vector* $\mathbf{u}_k$*, the mean is* $\bar{u}_k$ *and the variance is* σ_{u_k}*, then, for* $2 \leq k \leq n$*, and* $1 \leq i \leq n$ *we have:*

$$\bar{u}_k = \sum_{i=1}^{n} u_{ik} = 0 \tag{2.86}$$

$$\sigma_{u_k} = \frac{1}{n}\sum_{i=1}^{n}(u_{ik} - \bar{u}_k)^2 = \frac{1}{n} \tag{2.87}$$

Proof: These results can be easily derived from $\mathbf{u}_{k\geq 2}^{\top}\mathbf{1} = 0$ and $\mathbf{u}_k^{\top}\mathbf{u}_k = 1$. ■

2.6.5 Principal Component Analysis of a Graph Embedding

The Moore-Penrose pseudo-inverse of the Laplacian can be written as:

$$\begin{aligned} \mathbf{L}^{\dagger} &= \mathbf{U}\mathbf{\Lambda}^{-1}\mathbf{U}^{\top} \\ &= (\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^{\top})^{\top}(\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^{\top}) \\ &= \mathbf{X}^{\top}\mathbf{X} \end{aligned} \tag{2.88}$$

where $\mathbf{\Lambda}^{-1} = \mathrm{Diag}(0, 1/\lambda_2, \ldots, 1/\lambda_n)$.

The symmetric semi-definite positive matrix $\mathbf{L}^{\dagger}$ is a *Gram* matrix with the same eigenvectors as those of the graph Laplacian. When omitting the null eigenvalue and associated constant eigenvector, $\mathbf{X}$ becomes a $(n-1)\times n$ matrix whose columns are the coordinates of the graph's vertices in an *embedded (or feature) space*, *i.e.*, $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_j \ldots \mathbf{x}_n]$. It is interesting to note that the entries of $\mathbf{L}^{\dagger}$ may be viewed as *kernel* dot-products, or a Gram matrix [Ham 2004]. The Gram matrix representation allows us to embed the graph in an Euclidean feature-space where each vertex v_j of the graph is a feature point represented as $\mathbf{x}_j$.

The left pseudo-inverse operator of the Laplacian $\mathbf{L}$, satisfying $\mathbf{L}^{\dagger}\mathbf{L}\mathbf{u} = \mathbf{u}$ for any $\mathbf{u} \perp \mathrm{null}(\mathbf{L})$, is also called the *Green function* of the heat equation. Under the assumption that the graph is connected and thus $\mathbf{L}$ has an eigenvalue $\lambda_1 = 0$ with multiplicity 1, we obtain:

$$\mathbf{L}^{\dagger} = \sum_{k=2}^{n} \frac{1}{\lambda_k}\mathbf{u}_k\mathbf{u}_k^{\top}. \tag{2.89}$$

The Green function is intimately related to random walks on graphs, and can be interpreted probabilistically as follows.

2.6.5.1 Commute Time Distance (CTD)

Given a Markov chain such that each graph vertex is the state, and the transition from vertex v_i is possible to any adjacent vertex $v_j \sim v_i$ with probability w_{ij}/d_i, the expected number of steps required to reach vertex v_j from v_i, called the *access* or *hitting time* $O(v_i, v_j)$. The expected number of steps in a round trip from v_i to v_j is called the *commute-time distance*: $\text{CTD}^2(v_i, v_j) = O(v_i, v_j) + O(v_j, v_i)$. The commute-time distance [Qiu 2007] can be expressed in terms of the entries of $\mathbf{L}^\dagger$:

$$\begin{aligned}
\text{CTD}^2(v_i, v_j) &= \text{Vol}(\mathcal{G})(\mathbf{L}^\dagger(i,i) + \mathbf{L}^\dagger(j,j) - 2\mathbf{L}^\dagger(i,j)) \\
&= \text{Vol}(\mathcal{G}) \left(\sum_{k=2}^{n} \frac{1}{\lambda_k} \mathbf{u}_{ik}^2 + \sum_{k=2}^{n} \frac{1}{\lambda_k} \mathbf{u}_{jk}^2 - 2 \sum_{k=2}^{n} \frac{1}{\lambda_k} \mathbf{u}_{ik}\mathbf{u}_{jk} \right) \\
&= \text{Vol}(\mathcal{G}) \sum_{k=2}^{n} \left(\lambda_k^{-1/2} (\mathbf{u}_{ik} - \mathbf{u}_{jk}) \right)^2 \\
&= \text{Vol}(\mathcal{G}) \|\mathbf{x}_i - \mathbf{x}_j\|^2, \qquad (2.90)
\end{aligned}$$

where the volume of the graph, $\text{Vol}(\mathcal{G})$ is the sum of the degrees of all the graph vertices. The CTD function is positive-definite and sub-additive, thus defining a *metric* between the graph vertices, referred to as *commute-time* (or *resistance*) *distance* [Grinstead 1998]. The CTD is inversely related to the number and length of paths connecting two vertices. Unlike the shortest-path (geodesic) distance, CTD captures the connectivity structure of the graph volume rather than a single path between the two vertices. The great advantage of the commute-time distance over the shortest geodesic path is that it is robust to topological changes and therefore is well suited for characterizing complex graphs. Since the volume is a graph constant, we obtain:

$$\text{CTD}^2(v_i, v_j) \propto \|\mathbf{x}_i - \mathbf{x}_j\|^2. \qquad (2.91)$$

Hence, the Euclidean distance between any two feature points $\mathbf{x}_i$ and $\mathbf{x}_j$ is the commute time distance between the graph vertex v_i and v_j.

2.6.5.2 Commute Time Graph Embedding

Using the first K non-null eigenvalue-eigenvector pairs of the Laplacian $\mathbf{L}$, the *commute-time embedding* [Qiu 2007] of the graph's nodes corresponds to the column vectors of the $K \times n$ matrix $\mathbf{X}$:

$$\mathbf{X}_{K\times n} = \mathbf{\Lambda}_K^{-1/2} (\mathbf{U}_{n\times K})^\top = [\mathbf{x}_1 \ldots \mathbf{x}_j \ldots \mathbf{x}_n]. \qquad (2.92)$$

From (2.82) and (2.92) one can easily infer lower and upper bounds for the i-th coordinate of $\mathbf{x}_j$:

$$-\lambda_i^{-1/2} < x_{ji} < \lambda_i^{-1/2}. \qquad (2.93)$$

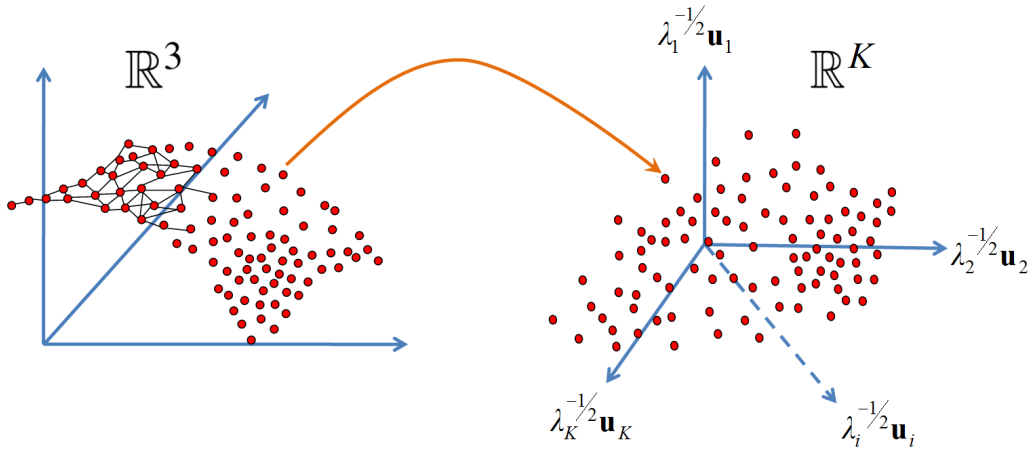


Figure 2.12: Visualization of the spectral graph embedding of a discrete manifold surface, originally embedded in $\mathbb{R}^3$, using the commute-time graph embedding.

The last equation implies that the graph embedding stretches along the eigenvectors with a factor that is inversely proportional to the square root of the eigenvalues. Theorem 3 below, characterizes the smallest non-null K eigenvalue-eigenvector pairs of $\mathbf{L}$ as the directions of maximum variance (the principal components) of the commute-time embedding.

Theorem 3 *The largest eigenvalue-eigenvector pairs of the pseudo-inverse of the combinatorial Laplacian matrix are the principal components of the commute-time embedding, i.e., the points* $\mathbf{X}$ *are zero-centered and have a diagonal covariance matrix.*

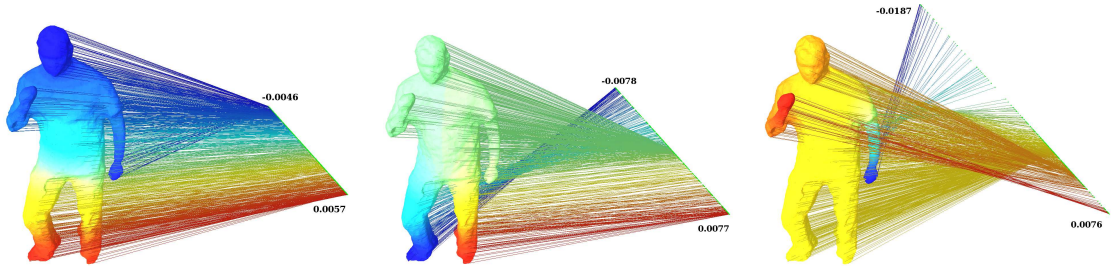
Proof: Indeed, from (2.86) we obtain a zero-mean while from (2.92) we obtain a diagonal covariance matrix:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \frac{1}{n}\mathbf{\Lambda}^{-\frac{1}{2}}\begin{pmatrix}\sum_{i=1}^{n}\mathbf{u}_{i2}\\ \vdots \\ \sum_{i=1}^{n}\mathbf{u}_{ik+1}\end{pmatrix} = \begin{pmatrix}0\\ \vdots \\ 0\end{pmatrix} \tag{2.94}$$

$$\mathbf{\Sigma}_X = \frac{1}{n}\mathbf{X}\mathbf{X}^\top = \frac{1}{n}\mathbf{\Lambda}^{-\frac{1}{2}}\mathbf{U}^\top\mathbf{U}\mathbf{\Lambda}^{-\frac{1}{2}} = \frac{1}{n}\mathbf{\Lambda}^{-1} \tag{2.95}$$

■.

Figure 2.12 visualizes the spectral graph embedding of a discrete surface in $\mathbb{R}^3$ to an Euclidean space $\mathbb{R}^K$, using the commute time graph embedding. Figure 2.13 shows the projection of the graph (in this case 3D shape represented as meshes) vertices on eigenvectors.



(a) (b) (c)

Figure 2.13: This is an illustration of the concept of the PCA of a graph embedding. Here 3D shape is represented as a (undirected weighted) shape graph and its vertices are projected onto the second, third and fourth eigenvectors of the graph Laplacian matrix. These eigenvectors can be viewed as the principal directions of the shape. (see Chapter 3 for details on shape graph.)

2.6.6 Choosing the Dimension of the Embedding

A direct consequence of theorem 3 is that the embedded graph representation is centered and the eigenvectors of the combinatorial Laplacian are the directions of maximum variance. The *principal* eigenvectors correspond to the eigenvectors associated with the K *largest* eigenvalues of the $\mathbf{L}^\dagger$, *i.e.*, $\lambda_2^{-1} \geq \lambda_3^{-1} \geq \ldots \geq \lambda_K^{-1}$. The variance along vector $\mathbf{u}_k$ is λ_k^{-1}/n. Therefore, the total variance can be computed from the trace of the $\mathbf{L}^\dagger$ matrix :

$$\mathrm{tr}(\mathbf{\Sigma}_X) = \frac{1}{n}\mathrm{tr}(\mathbf{L}^\dagger). \tag{2.96}$$

A standard way of choosing the principal components is to use the *scree diagram* [Berthold 1999]:

$$\theta(K) = \frac{\sum_{k=2}^{K+1}\lambda_k^{-1}}{\sum_{k=2}^{n}\lambda_k^{-1}}. \tag{2.97}$$

The selection of the first K principal eigenvectors, therefore depends on the spectral fall-off of the inverses of the eigenvalues. In spectral graph theory, the dimension K is chosen on the basis of the existence of an eigengap such that $\lambda_{K+2} - \lambda_{K+1} > t$ with $t > 0$. In practice, it is extremely difficult to find such an eigengap, in particular, in the case of sparse graphs that correspond to a discretized manifold. Instead, we propose to select the dimension of the embedding in the following way. Notice that Eq. (2.97) can be written as $\theta(K) = A/(A+B)$ with $A = \sum_{k=2}^{K+1}\lambda_k^{-1}$ and $B = \sum_{k=K+2}^{n}\lambda_k^{-1}$. Moreover, from the fact that the λ_k's are arranged in increasing order, we obtain $B \leq (n-K-1)\lambda_{K+1}^{-1}$. Hence:

$$\theta_{\min} \leq \theta(K) \leq 1, \tag{2.98}$$

with

$$\theta_{\min} = \frac{\sum_{k=2}^{K+1}\lambda_k^{-1}}{\sum_{k=2}^{K}\lambda_k^{-1} + (n-K)\lambda_{K+1}^{-1}}. \tag{2.99}$$

This lower bound can be computed from the K smallest non null eigenvalues of the combinatorial Laplacian matrix. Hence, one can choose K such that the sum of the first K eigenvalues of the $\mathbf{L}^{\dagger}$ matrix is a good approximation of the total variance, *e.g.*, $\theta_{\min} = 0.95$.

2.7 Discrete and Continuous Laplace Operator

We have already introduced the definition of the continuous Laplace operator on n-dimensional Euclidean space as well as the Laplace-Beltrami operator on Riemannian manifold in Section 2.3.3.1. In this section, we will investigate the relationship between these continuous operators and the discrete Laplacian matrix operator on graphs presented in Section 2.6.1.

In Section 2.4, we cast the problem of surface mapping as graph matching in a discrete setup by assuming the discrete sampling of the continuous manifold surface to be infinitely dense. With this notion of discrete sampling of the continuous domain, the relationship between the continuous Laplace operator on n-dimensional Euclidean space and the discrete Laplace (matrix) operator is more intuitive. As shown in [Belkin 2003], one can apply the discrete (combinatorial) Laplacian $\mathbf{L}$ in Eq. (2.63) to a discrete function $\mathbf{f} \in \mathbb{R}^n$ defined on a graph $\mathcal{G}$ as:

$$[\mathbf{L}\mathbf{f}](i) = \sum_{(i,j)\in\mathcal{E}(\mathcal{G})} w_{ij}\left(\mathbf{f}(i) - \mathbf{f}(j)\right), \tag{2.100}$$

where w_{ij} are the edge weights in the graph's adjacency matrix and $\mathbf{f}(i)$ are the elements of $\mathbf{f}$. Similarly, we can write the quadratic form $\mathbf{f}^T\mathbf{L}\mathbf{f}$ as:

$$\mathbf{f}^T\mathbf{L}\mathbf{f} = \langle \mathbf{f}, \mathbf{L}\mathbf{f}\rangle = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}(\mathcal{G})} w_{ij}\left(\mathbf{f}(i) - \mathbf{f}(j)\right)^2. \tag{2.101}$$

Since, in a similarity graph structure $w_{ij} \propto \frac{1}{dist(v_i,v_j)} \approx \frac{1}{\varepsilon_{ij}^2}$, we can write,

$$\mathbf{f}^T\mathbf{L}\mathbf{f} \approx \frac{1}{2}\sum_{(i,j)\in\mathcal{E}(\mathcal{G})} \left(\frac{\mathbf{f}(i) - \mathbf{f}(j)}{\varepsilon_{ij}^2}\right)^2. \tag{2.102}$$

Thus, Eq. (2.102) can be interpreted as the discrete version of quadratic form associated to the continuous Laplace operator on $\mathbb{R}^n$ given in Eq. (2.23).

In the case of continuous Laplace-Beltrami on Riemannian manifold surface, *i.e.*, Eq. (2.25), the relationship with the discrete Laplace operator is more complex and is an active subject of study. Some recent work on convergence of various discretizations of the graph Laplacian to a continuous Laplace-Beltrami operator have produced formal demonstrations that convergence is likely when the graph sampling grows to infinity [von Luxburg 2007, Belkin 2003].

2.8 Machine Learning Tools

In this section, we introduce two important machine learning tools that we have extensively used in this document.

2.8.1 Spectral Clustering (SC)

One important problem in machine learning is to classify or cluster a set of data points. This is an important notion and is strongly related to the shape segmentation task while considering a 3D shape as point-cloud. Traditionally, this is achieved by *k-means* [Lloyd 1982] type of algorithm that tries to group the data based on their coordinate representation in the original space. However, this is difficult to achieve when data lies on a non-linear manifold. The non-linear dimensionality reduction (kernel trick) techniques are typically used to obtain an embedding of data points where groups or clusters are linearly separable. This idea is directly related to the graph dimensionality reduction presented in Section 2.6.

Spectral clustering is one such technique where first the data represented as a similarity graph is embedded in a d-dimensional space using the Laplacian eigenmap method [Belkin 2003] and then the traditional K-means algorithm is applied to obtain clustering results. Typically, d is chosen to be equal to k, *i.e.*, the desired number of clusters in the data. This technique is strongly related to the notion of *graph cut* and *random walk* in graph theory. [von Luxburg 2007] presents a comprehensive discussion of spectral clustering technique. Although, SC has many variants, we briefly outline a sample method in Algorithm 2.

Algorithm 2 *Spectral Clustering* (SC)

Input: : Data points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^D$, number k of clusters to construct.
Output: : Clusters $C_1, \ldots, C_k$ where $\forall (1 \leq i \leq k)\ C_i \subset \mathbf{X}$ and $C_i \cap C_j = NULL$.

1: Construct a similarity graph for $\mathbf{X}$.
2: Compute the eigenvectors $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\} \in \mathbb{R}^n$ of the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ of similarity graph (see Section 2.6.3).
3: Let $\mathbf{U}_{n \times k} = [\mathbf{u}_1, \ldots, \mathbf{u}_k]$ as the embedding of $\mathbf{X}$.
4: Each data point $\mathbf{x}_i$ can be represented as $\mathbf{y}_i \in \mathbb{R}^k$ with its coordinates as i^{th} row of $\mathbf{U}$ matrix.
5: Apply *K-means* clustering on points $\{\mathbf{y}_i\}_{i=1,\ldots,n} \in \mathbb{R}^k$ in order to obtain a set of k clusters $\{C_1, \ldots, C_K\}$.

2.8.2 Expectation Maximization (EM)

The Expectation Maximization (EM) algorithm is an important machine learning tool that was first proposed in [Dempster 1977]. It enables an efficient iterative computation of the *Maximum Likelihood* (ML) estimate in the presence of missing or hidden data. The basic idea is to fit a generative model to a given data-set and iteratively compute the model parameter(s). ML estimation allows one to estimate the model parameter(s) for which the observed data is the most likely. EM algorithm is an iterative two step method where in the first step, *i.e.*, E-step (or expectation), the missing data is estimated given the observed data and the current estimate of the model parameters. This is achieved using the conditional expectation, explaining the choice of terminology. In the second step, *i.e.*, M-step, the likelihood function is maximized under the assumption that the missing data is known. The estimate of the missing data from the

E-step is used in place of the actual missing data. EM has assured convergence property since the algorithm is guaranteed to increase the likelihood at each iteration. However, one critical issue is the initialization of model parameters in the very first iteration. If the initialization is too bad then it could lead to a sub-optimal solution as the iterative algorithm might get stuck in a local maxima.

Please refer to [Borman 2004] for an informative tutorial on the EM algorithm. We will use a variant of EM algorithm for d-dimensional point-cloud registration presented in [Mateus 2008, Horaud 2011], while performing the probabilistic dense shape matching as an intermediate step for segmentation transfer in Chapter 5.

2.9 Conclusion

In this chapter, we have presented ideas from graph theory, differential geometry and combined them to form the basis of spectral embedding of a graph. In the next chapter, we will discuss our choice of graph Laplacian embedding for a pose invariant representation of articulated 3D shapes. In Chapter 6, we will extend the spectral representation in a multi-scale setup using the notion of heat diffusion on discrete surfaces. In both single and multi-scale representation and analysis, we will extensively use the mathematical constructs presented in this chapter.

We would like to mention that the content present in this chapter is only to get an overview and to build a platform for rest of the document. These mathematical constructs were originated from different well established areas in the research community and hence were largely borrowed from many popular books and research publications, duly mentioned in the text, as well as the publicly available knowledge-bases like *Wikipedia*. We recommend a further study of the cited publications for a detail understanding of these constructs.

CHAPTER 3

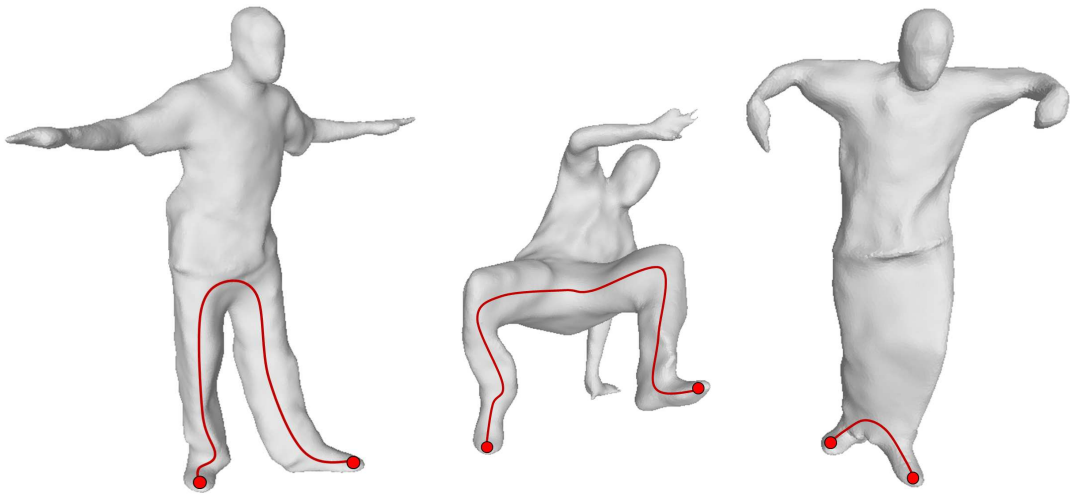
Spectral Representation for 3D Articulated Shapes

Contents

3.1 Introduction

We have motivated our thesis with a brief discussion in Chapter 1 where we highlighted the challenges associated with the visual shape data that are commonly used to represent the real-world articulated 3D shapes and are typically captured with the multi-camera acquisition systems. As mentioned in the thesis structure, we mainly focus on developing methods for shape analysis tasks, namely, segmentation and matching that can deal with the challenging visual shape data. This would require building an appropriate model or representation for these shapes.

As pointed out in Section 1.1, the computer graphics and geometry processing community commonly assume that a 3D shape represented by a polygonal mesh corresponds to a discrete representation of a smooth, continuous, and closed 2D Riemannian manifold surface embedded in a three-dimensional Euclidean space. This enables one to analyze a mesh using the geometry processing tools. In particular, it enable one to locally characterize a 3D shape, *e.g.*, using the first and second fundamental form of the respective parametric representation. This, for example, is useful while computing an isometric mapping between two shapes by analyzing their first fundamental form or the Gaussian curvature (see Section 2.3.2.1). However, the situation is more challenging in the domain of computer vision where visual shapes do not



(a) (b) (c)

Figure 3.1: Visualization of geodesic distances on articulated visual shapes in different poses and with topological changes.

agree with the Riemannian manifold assumption. In this domain, it is common to use the discrete counterpart of the Riemannian geometry tools, namely, the graph Laplacian to study the pose-invariant properties of an articulated 3D shape typically represented as the mesh graph.

The Laplace-Beltrami operator on continuous manifold, introduced in the Section 2.3.3.1, is a key operator in this regard as the theoretical studies in literature suggests that the spectral analysis of this operator reveals some important information about the shape (see in [Reuter 2006a]). In addition to this, there exists a direct analogy between the Laplace-Beltrami and Fourier operator on manifold, which has been used in creating geometry-aware basis functions useful for shape compression tasks [Lévy 2006, Vallet 2008]. The geometric properties of the discrete eigenfunctions of the Laplace operator have been extensively used in the computer graphics for mesh editing and processing [Floater 2005, Sorkine 2006]. These eigenfunctions are also very popular for shape representation and analysis [Jain 2007, Rustamov 2007, Mateus 2008, Cuzzolin 2008, Elad 2003].

In the same spirit, we propose to use the spectral embedding representation, a widely used shape modeling technique for pose-invariant shape representation of articulated 3D shapes [Mateus 2008, Jain 2007]. Such a representation will be useful to find the dense matching as well as a consistent segmentation of the articulated shapes.

The basic idea of a pose-invariant representation is based upon the observation that the articulated poses of a 3D shape can be obtained by locally deforming the surface at certain joints in the object. Hence, a local deformation of an articulated shape should preserve the intrinsic structure or topology of the object, thereby preserving certain properties like distances on shape surface. The geodesic distance is a commonly used metric for the measurement of

distances on 3D shapes (see Section 2.3.1.8). In case of discrete mesh representation, the geodesic distance are typically computed as the shortest path on the underlying graph structure.

Thus, a visual shape obtained by locally deforming the another visual shape should have a same distribution of pairwise geodesic distances. However, this is not true when the shape topology is deformed, *e.g.*, if some parts of the shape are merged together. Figure 3.1 visualizes the geodesic distance on visual shapes. In Figure 3.1(a,b), the geodesic distances are preserved since the deformations are local and the overall topology is the same. However, in Figure 3.1(c), both legs are merged together causing a topology change and hence the geodesic distances are not preserved.

In general, the larger parts of shape surface should not be affected by the local deformations around certain surface patches that are induced by the topology preserving articulated poses. One can exploit this property by locally characterizing each point on the surface using a local neighborhood around it. Hence an input shapes represented by the discrete meshes structure can be treated as a undirected weighted graphs.This graph representation provides us the notion of local adjacency relationship between neighboring points on the shape, thus capturing an intrinsic distance measure that is invariant to the pose of the object.

One direct consequence of graph representation is that the shape can be finally mapped to a higher dimensional intrinsic Hilbert space using the spectral embedding method presented in Section 2.6. An interesting property of the spectral embedding method is that it preserves the local distance measure on graphs, as much as possible, while removing the pose specific information and yielding a pose invariant representation [Mateus 2009].

In this chapter, we summarize the method to obtain a pose invariant 3D shape representation. First, in Section 3.2, we briefly present the shape graphs representation of a 3D shape. Second, we outline the spectral embedding method for mapping a shape graph into a pose-invariant space. Finally, we conclude with a discussion on strengths and weaknesses of the outlined spectral representation.

3.2 Shape Graphs

Similarity graph is a popular graph based representation for multi-dimensional data in machine learning. It is an undirected weighted where the original data points that are originally represented as a d-dimensional point-cloud are taken as the graph vertices and the similarity score between each pair of data points is encoded with the corresponding (positive) edge weight.

A discrete 3D shape is commonly represented as a point-cloud in three-dimensional space. In computer graphics, this can be obtained by controlled sampling of parametric surface while in computer vision it is the typical output of acquisition systems that uses geometric triangulation using the multiple camera images or any depth sensor, *e.g.*, laser sensor. This point-cloud representation is subsequently processed to obtain a polygonal mesh representation, which is a common way to represent shapes in the geometry processing. A polygonal mesh is a graph

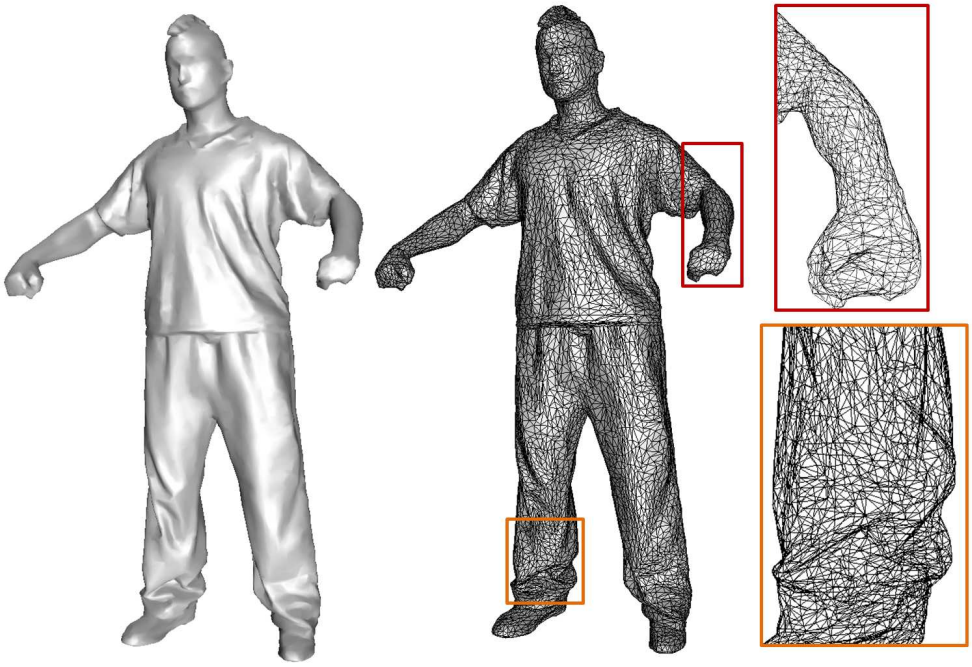


Figure 3.2: Visualization of a shape graph. On the left, we show a 3D shape and on the right a sparse graph representation obtained by the mesh processing toolbox proposed in [Zaharescu 2011]

representation where each 3D point on the shape surface corresponds to a vertex of the graph and an edge encodes connectivity between two vertices. In a polygonal mesh, an edge is typically binary weighted, storing only the connectivity information between a pair of 3D points. It is common to extend this representation by choosing a custom weighting scheme, which can encode certain characterization of the local surface, which was originally sampled during the shape acquisition.

We call such an extended representation as the *shape graph*. The shape graph is considerably different from the similarity graph structure commonly used in the machine learning community. A similarity graph has densely connected data points that are globally distributed in the space as opposed to a shape graph where the point cloud is obtained by sampling a 2D manifold surface in three-dimensional Euclidean space. Hence, for a shape graph, the connectivity is very sparse and the notion of neighborhood is fairly local. This local neighborhood around a vertex typically approximates the tangent space of the corresponding point on the continuous surface (see Section 2.3.1.2).

There are two key steps while constructing a shape graph from an input 3D point-cloud. The first step involves defining a connectivity structure in the local neighborhood around each sampled point. The K-Nearest-Neighbors (KNN) graph and ε-graph are two primitive connectivity structures that can be locally defined on a point-cloud. Another choice is to define

a voxel based representation, which however is a volumetric parametrization of a 3D shape. However, these primitive structures do not guarantee that the output graph will be a discrete approximation of the shape surface. Nevertheless, there exists a vast amount of literature on mesh processing tools that uses advanced techniques to obtain a polygonal mesh representation from these primitive structures. [Botsch 2010] provides a detailed account on existing mesh processing tools in the geometry processing. We obtain the initial volumetric shape representation using the reconstruction method proposed in [Franco 2009] and subsequently use a mesh processing toolbox proposed in [Zaharescu 2011] to obtain a triangulated mesh representation for a visual shape.

Figure 3.2 shows an example shape graph. In the second step, we choose the appropriate weighting scheme for a shape graph. We will defer the discussion on the weighting scheme until when we introduce the spectral embedding method in the next section.

3.3 Spectral Embedding of Shape Graphs

In Chapter 2, we have already introduced the notion of spectral graph embedding in the context of graph dimensionality reduction (see Section 2.6). In this section, we will understand the spectral embedding method in the context of mapping the shape graphs into a pose invariant subspace.

The Riemannian geometry of surfaces (introduced in Section 2.3.1) and the isometric mapping between manifold surfaces (described in Section 2.3.2.1), together imply that any articulated shape assumed to be a compact Riemannian manifold (without boundary, *e.g.*, holes and open surfaces) can be isometrically mapped to any other similar shape of the same topology if the distances on two shape surfaces are preserved. This consequently suggests that an articulated deformation can be modeled as an isometric transform between surfaces if there is no stretching involved. Additionally, an interesting set of theorems called Nash Embedding Theorems (briefly stated in Section 2.3.1.9) allows one to isometrically embed a Riemannian manifold into a n-dimensional Euclidean space. Thus, two articulated shapes can be mapped to a common pose invariant space and consequently matched to each other.

Such a mapping of an articulated shape to an isometric embedding space can be achieved by defining smoothly varying functions that maps nearby points on the original manifold surface to nearby points in the mapped space, thereby preserving the neighboring distances and hence canceling the effect of articulated poses. However, formalizing the smoothness constraint is not straightforward when dealing with a function defined over manifold surfaces. Nevertheless, Riemannian geometry equipped with construction of tangent space provides a solution. The Laplace-Beltrami operator introduced in Section 2.3.3.1 is a key operator that is traditionally used to map a compact Riemannian manifold surface to an isometry invariant space. Such a space is spanned by the smoothly varying eigenfunctions of the Laplace-Beltrami operator (see [Reuter 2006a]).

3.3.1 Manifold Embedding using Laplace-Beltrami Operator

The Laplace-Beltrami operator in Eq. (2.25), is an extension of the classical Laplace operator on Riemannian manifold, and can also be interpreted as a smoothness constraint in the following setup. Let $f : \mathcal{M} \mapsto \mathbb{R}$ be a real valued function defined on compact Riemannian manifold $\mathcal{M}$. The function f should be a local geometry preserving function if it maps nearby points on $\mathcal{M}$ to nearby values in $\mathbb{R}$, *i.e.*, f should be a smoothly varying function on $\mathcal{M}$. One possible way to enforced the smoothness of f is by making sure that it should have a small gradient $\nabla_{\mathcal{M}} f$. Interestingly, the Laplace-Beltrami operator $\Delta_{\mathcal{M}} f$ is defined as the divergence of gradient, *i.e.*, the flux of the gradient field or the rate of change of gradient. Thus, the ideal solution should be:

$$\Delta_{\mathcal{M}} f = 0, \tag{3.1}$$

i.e., zero change in the gradient field. This however, can be interpreted as the solution to the Helmholtz equation given in Eq. (2.28) with $\lambda = 0$ and $f(x) = \delta, \ \forall x \in \mathcal{M}$, *i.e.*, f being a constant eigenfunction of $\Delta_{\mathcal{M}} f$ associated to a zero eigenvalue.

Hence, the subsequent eigenvectors of the Laplace-Beltrami operator corresponding to the (non-zero) increasing eigenvalues can be seen as the family of smoothly varying orthogonal functions on $\mathcal{M}$. This interpretation will lead to a mapping of manifold surface to an infinite dimensional space spanned by the orthogonal eigenfunctions of the Laplace-Beltrami operator as its basis vectors. This space is, by definition, a pose invariant space as the smoothness constraint enforced by the Laplace-Beltrami regularization does not seek to preserve the original embedding of the manifold surface in the ambient three-dimensional Euclidean space.

Nevertheless, such an isometric mapping would preserve the distance on manifold surface as the smoothness of mapping is enforced. Thus, the geodesic distance $d_{geodesic}(x_i, x_j)$ between two points $x_i, x_j \in \mathcal{M}$ on an articulated shape representing a compact Riemannian manifold $\mathcal{M} \subset \mathbb{R}^2$ should be equivalent to the Euclidean distance between their mapped images $(y_i, y_j) \in \mathcal{Y}$ in the finite dimensional (Euclidean) embedding space $\mathcal{Y} \subset \mathbb{R}^n$.

Figure 3.3 shows an example of isometric embedding of a visual shape. Such an embedding would (approximately) preserve the geodesic distances on shape surface. The plotted embedding is indeed a discrete Laplacian embedding of the corresponding shape graph, which was obtained using the method outlined in the next section.

3.3.2 Shape Graph Embedding using Discrete Laplace Operator

The analysis presented above is applicable to continuous manifold surfaces. The discussion presented in Section 2.7 concludes that under certain assumptions, the discrete Laplace operator (introduced in Section 2.6) can be considered as the discrete version of the continuous Laplace-Beltrami operator on manifold surfaces.

Thus, we can straightforwardly conclude that the eigen-decomposition of a discrete Laplace operator on graphs can be used to isometrically embed an articulated shape represented by a

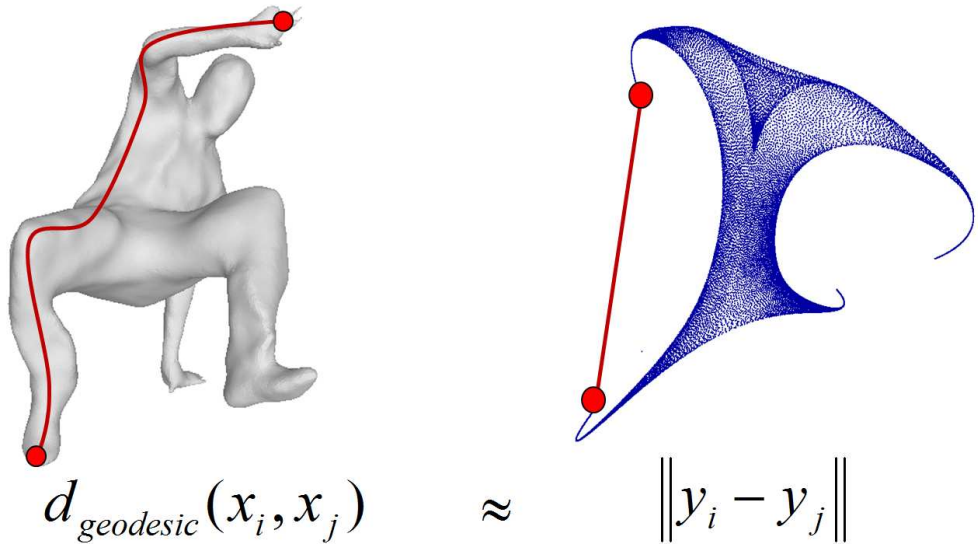


Figure 3.3: Distance preserving isometric embedding of a visual shape. The geodesic distance $d_{geodesic}(x_i, x_j)$ between two points $x_i, x_j \in \mathcal{M}$ on visual shape is equivalent to the Euclidean distance between their mapped images $(y_i, y_j) \in \mathcal{Y}$ in the embedding space $\mathcal{Y} \subset \mathbb{R}^n$

shape graph into a pose invariant n-dimensional Euclidean space. However, finding such a mapping is computationally not feasible for very large shape graphs as it requires the computation of all the eigenvectors of the graph Laplacian matrix. Nevertheless, we can adapt the graph dimensionality reduction framework derived in Section 2.6.5 to obtain a d-dimensional Euclidean embedding of the shape graphs where d is significantly less than n. It is relatively easy to compute such a reduced dimensional embedding as the graph Laplacian matrices are very sparse by their definition and there exists efficient algorithms to compute smaller set of eigenvectors for such matrices. Figure 3.4, shows the sparse structure of a typical graph Laplacian matrix of a visual shape.

3.3.2.1 The Laplacian Embedding Method

A shape graph represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ can be mapped to a n-dimensional Euclidean space using an injecting mapping function $\Phi : \mathcal{V} \mapsto \mathcal{Y} \subset \mathbb{R}^n$. The mapping function Φ should project the set of graph vertices $\mathcal{V}$ to points in a n-dimensional Euclidean (embedding) space $\mathcal{Y}$ such that the geodesic distances on $\mathcal{G}$ are preserved. In other words, the mapping should be isometric. This can be achieved by preserving the local neighborhood structure around each vertex of $\mathcal{G}$ by finding a smooth map that projects nearby points together. In the discrete setup, let Φ be a set of n discrete functions $\Phi = [f_1, \ldots, f_n]$, each per dimension in the embedding space such that $f : \mathcal{V} \mapsto \mathbb{R}$.

For a single dimensional embedding, following quadratic minimization criteria can be

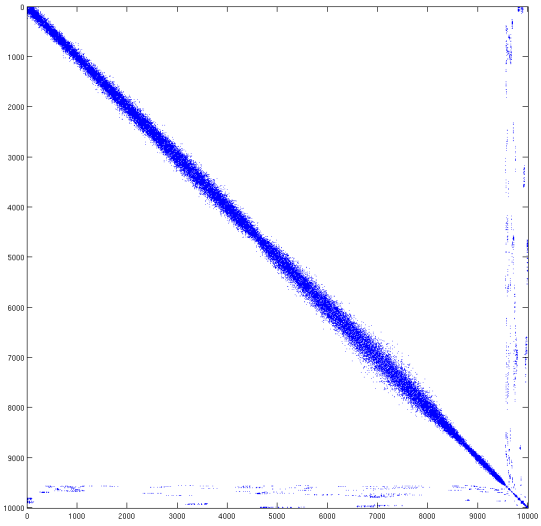


Figure 3.4: Sparse Laplacian Matrix for shape graph.

formulated for preserving the local neighborhood structure:

$$f = \arg\min_{f} \sum_{(i,j)\in\mathcal{E}(\mathcal{G})} w_{ij}\,(f(i)-f(j))^2 \quad i,j \in \mathcal{V} \quad w_{ij} \in \mathbf{W} \tag{3.2}$$

This minimization can be formulated as a least-square optimization task:

$$\begin{aligned} \underset{f}{\text{minimize}} \quad & \frac{1}{2}\arg\min_{f} \sum_{(i,j)\in\mathcal{E}(\mathcal{G})} w_{ij}\,(f(i)-f(j))^2 = \arg\min_{f} f^T\mathbf{L}f, \\ \text{subject to} \quad & f^T\mathbf{D}f = 1, \\ & f^T\mathbf{D}\mathbf{1} = 0. \end{aligned} \tag{3.3}$$

where $\mathbf{L}$ is the discrete Laplacian matrix and $\mathbf{D}$ is the degree matrix of $\mathcal{G}$ (see Section 2.6.1). The first constraint in Eq. (3.3) ensures the removal of arbitrary scaling of f by enforcing the orthogonality. The second constraint ensures the translation invariance property.

This formulation can be extended to a higher dimensional mapping function Φ as:

$$\Phi = \frac{1}{2}\arg\min_{\Phi} \sum_{(i,j)\in\mathcal{E}(\mathcal{G})} w_{ij}\|\Phi(i)-\Phi(j)\|^2 \tag{3.4}$$

$$= \arg\min_{\Phi} \operatorname{tr}(\Phi^T\mathbf{L}\Phi) \tag{3.5}$$

The least-square optimization problem in Eq. (3.4) can be minimized by solving for the smallest eigenvectors of the graph Laplacian matrix $\mathbf{L}$ (see appendix A.4). Hence, the eigenvectors of the graph Laplacian can be used as mapping functions to individually map each shape graph vertex $v_i \in \mathcal{V}$ to a real value in $\mathbb{R}$.

Thus, the mapping function Φ project shape graph vertices to a space spanned by the n orthogonal eigenvectors of graph Laplacian matrix. This is known as the Laplacian embedding method [Belkin 2003] and summarized in Algorithm 3. It is common to consider a

Algorithm 3 *The Laplacian Embedding Method*

Input: : A sampled compact Riemannian manifold $\mathbb{R}^2 \supset \mathcal{M} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ represented as the point-cloud in three-dimensional Euclidean space with $\mathbf{x}_i \in \mathbb{R}^3$.

Output: : A Laplacian embedding $\mathbb{R}^n \supset \mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ of $\mathcal{M}$.

1: Build a shape graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ (see Section 3.2).
2: Define the combinatorial graph Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$.
3: Solve the eigensystem $\mathbf{L}\mathbf{u} = \lambda\mathbf{u}$.
4: Define a mapping function as $\Phi = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ where each eigenvector of $\mathbf{L}$ acts as a discrete function on graph vertices, *i.e.*, $\mathcal{V}$ $\mathbf{u}_i : \mathcal{V} \mapsto \mathbb{R}$.
5: The mapping function Φ maps each vertex $v_i \in \mathcal{V}$ to a point $\mathbf{y}_i$ in n-dimensional Euclidean space such that $\mathbf{y}_i = [u_1^i, \ldots, u_n^i]^T$ where notation u_j^i denotes the i-th element of $\mathbf{u}_j$ vector. Thus, we can obtain a n-dimensional Laplacian embedding of $\mathcal{M}$ as $\mathbf{Y} = \Phi^T$.

d-dimensional embedding with $d \ll n$ for very large shape graphs due to computation efficiency, thereby, interpreting the Laplacian embedding method as a dimensionality reduction tool for shape graphs.

Figure 3.5 shows the one dimensional projection of shape graphs for individual eigenvectors of graph Laplacian (*i.e.*, $f : \mathcal{V} \mapsto \mathbb{R}$). The pose invariance property of graph Laplacian is clearly visible as two shape graphs representing two articulated shapes in different poses have similar projection on corresponding eigenvectors.

Embedding Alignment: We have already seen that an eigenvector of the graph Laplacian acts as a discrete function and map graph vertices to real values. Thus, in theory two similar graphs should have their eigenvectors aligned in the sense that they should project respective graph vertices in a similar fashion. For example, Figure 3.5 shows the projection of two sets of eigenvectors that are well aligned but there is no theoretical guarantee to achieve such an alignment. In practice, two sets of eigenvectors are similar only in the sense that the resulting embedding preserves the geodesic distances on their respective shape graphs. This is because the Laplacian eigenvectors of the shape graphs might not be reliably ordered and can have sign ambiguity. The problem of sign ambiguity is inherent to any eigen-decomposition since the eigen-solvers can compute eigenvectors only up to their sign. Thus, the problem of sign flip is inevitable in any eigen-solver. The main cause of unreliable ordering of eigenvectors is two fold. First, the shape graphs corresponding to the visual shapes are not strictly isomorphic and hence two visual shapes captured using the same articulated object can have different ordering of eigenvectors due to change in the principal directions of the corresponding shape graphs. Second, due to inherent symmetry in the structure of a 3D shape, there can be few Laplacian eigenvectors (*i.e.* the principal directions of the corresponding shape graph) with

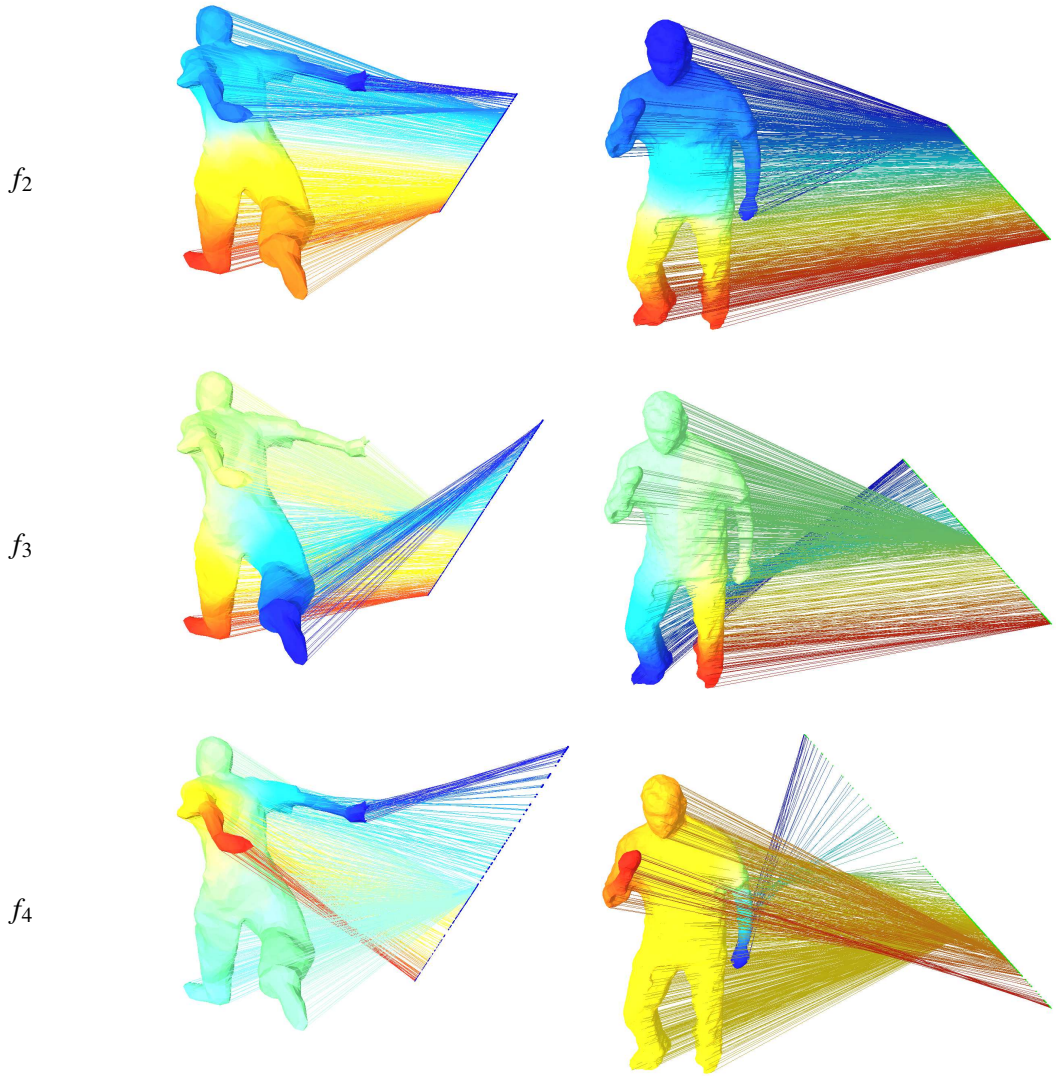


Figure 3.5: 1D projection of the shape graphs on individual Laplacian eigenvectors visualizing mapping of the two shape graph to a real line. It is clearly visible that even though the two shape graphs corresponds to two visual shapes in different poses (and captured by two different multi-camera acquisition systems), their projection of real line is quite similar. The colors represent variation in eigenvector elements from negative (blue) to positive (red) values.

almost the same eigenvalues, thus causing an ordering ambiguity. This can also happen due to the computational approximation done by the eigen-solvers. Figure 3.6 demonstrates a scenario where the Laplacian eigenvectors of the two shape graphs are not aligned due to the existence of both sign flip and ordering problem.

Figure 3.7 visualizes the alignment problem in three-dimensional embeddings of the two shape graphs. The first three non-null eigenvectors of respective graph Laplacian were used while plotting these embeddings. It is clearly visible that two embeddings are not aligned due to sign-flip and ordering problem discussed in the previous paragraph. Here, we would like to mention that existing spectral shape matching methods cast shape matching as d-dimensional point-cloud registration problem in the embedding space and hence use different heuristics to overcome the problem of embedding alignment. For example, the dense probabilistic shape registration method proposed in [Mateus 2008, Sharma 2012] uses histogram matching to find an alignment of eigenvectors.

3.3.3 Choice of Laplacian Weighting Scheme

We have introduced the shape graph construction in Section 3.2. The second key step in the construction of shape graphs is to fill the entries of the weighted adjacency matrix $\mathbf{W}$ of a shape graph with appropriate edge weights. The choice of weighting scheme becomes crucial while embedding a shape graph using the eigenvectors and eigenvalues of the combinatorial graph Laplacian matrix $\mathbf{L}$, which in turn is derived using the weighted adjacency matrix as shown in Eq. (2.63). Thus, choosing a weighting function that has a relevant interpretation on continuous manifold surface can lead to a better discretization of the continuous Laplace-Beltrami operator.

The only constraint while constructing a Laplacian matrix is that it should be a positive semi-definite symmetric matrix. Hence, the edge weights of adjacency matrix must be non-negative (see Section 2.6.3). The *Binary* weighting is one simple weighting scheme where w_{ij} is set to value 1 if there exists an edge between vertices v_i and v_j and set a zero value if there is no such edge. The *Gaussian weighs* and *cotangent weights* are two popular weighting schemes in the literature. In this section, we will briefly introduce these two weighting schemes and provide an abstract discussion on the choice of weighting scheme, supported by the empirical results computed over a small set of geometrically varying 3D shapes.

3.3.3.1 Cotangent Weights

We have already mentioned that a triangulated mesh may well be viewed as the discretization of a continuous Riemannian 2D surface embedded in $\mathbb{R}^3$. Several discretizations of the Laplace-Beltrami operator were proposed in the literature [Wardetzky 2007]. The *cotangent weights* [Pinkall 1993, Meyer 2003] is one of the most popular weighing scheme for graphical shapes. Let v_i and v_j be two vertices connected by an edge e_{ij}. The cotangent weight of e_{ij} is

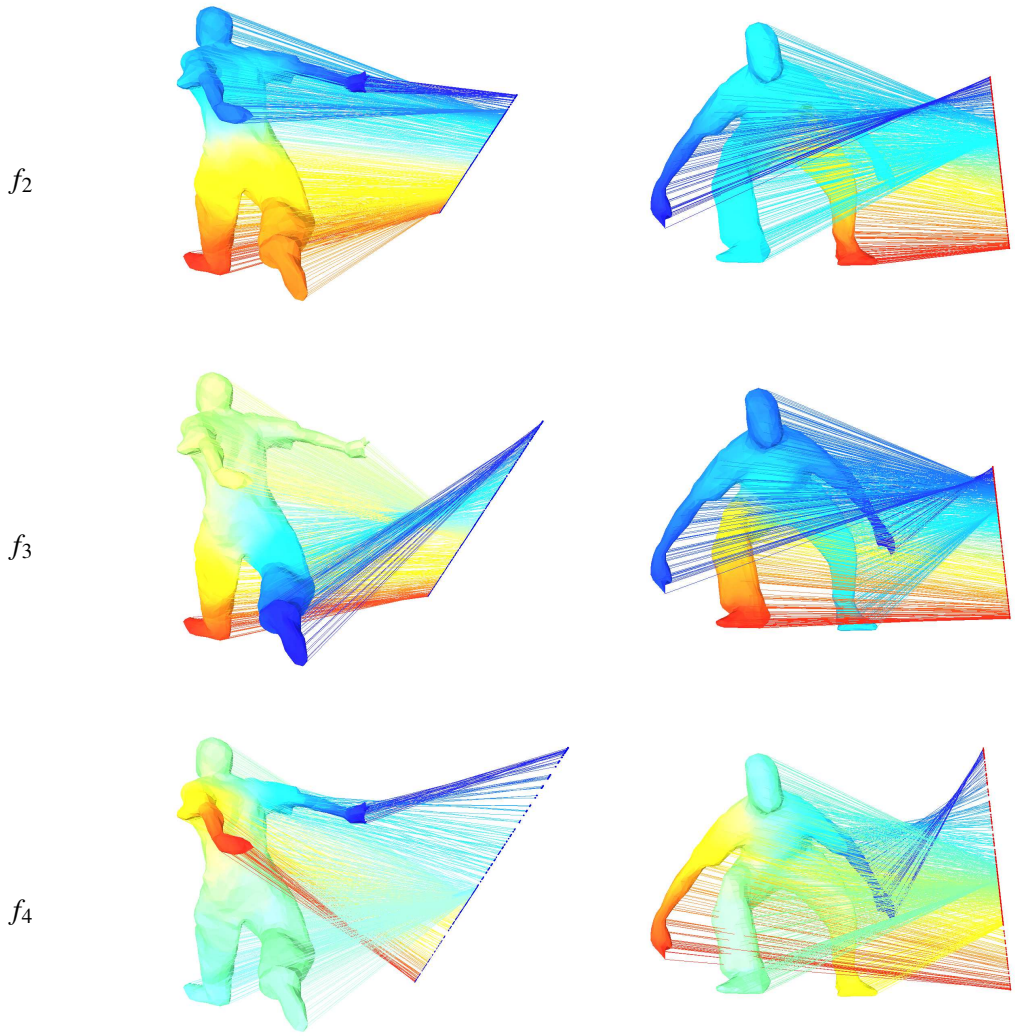
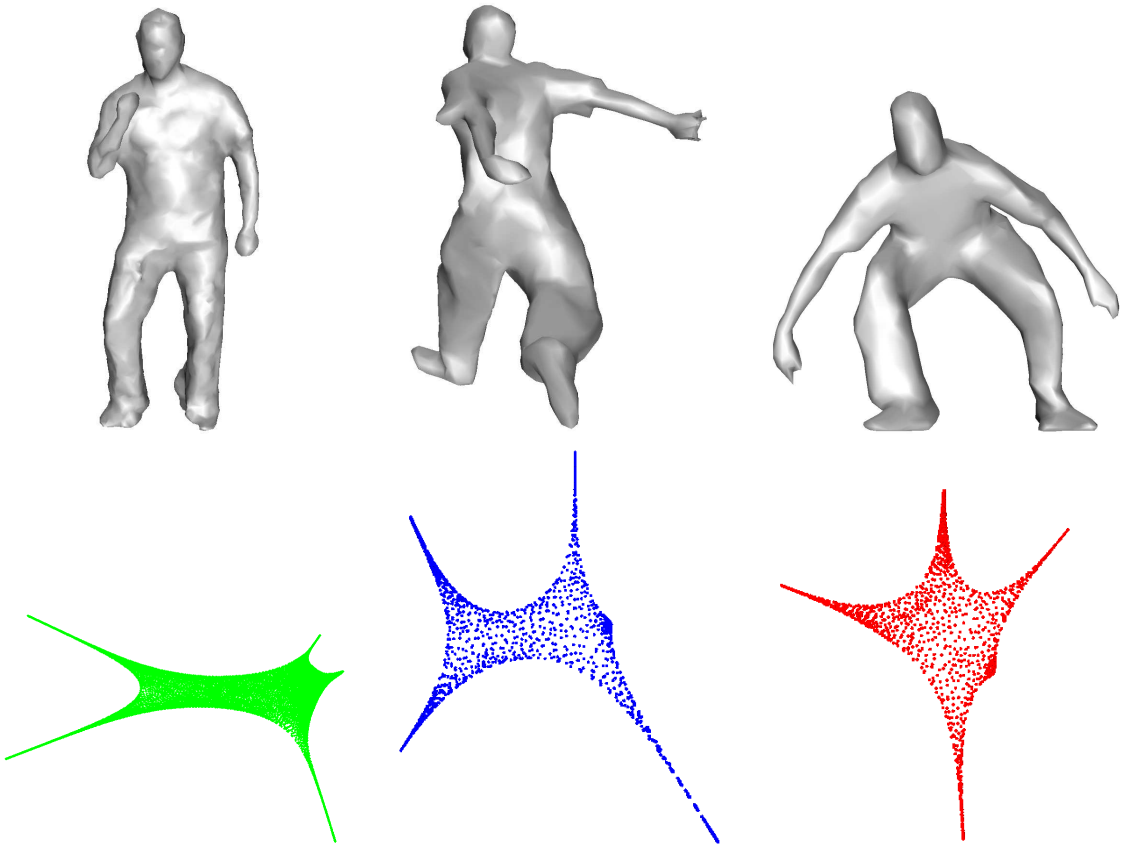


Figure 3.6: 1D projection of the shape graphs on individual Laplacian eigenvectors visualizing mapping of the two shape graph to a real line. The sign flip and ordering problem of eigenvectors is clearly visible even though two shapes are from the same sequence. This however does not affect the pose invariant property of the Laplacian embeddings.



(a)Ben510 [Franco 2009] with 10000 vertices.

(b)Flashkick55 [Starck 2007b] with 1500 vertices.

(c)Flashkick87 [Starck 2007b] with 1500 vertices.

Figure 3.7: Three dimensional embedding visualization using the first three non-null Laplacian eigenvectors of the shape graphs used in Figure 3.5 and 3.6. One can see that three embeddings are similar (though not aligned) irrespective of the difference in the pose, sampling and data set of the shapes.

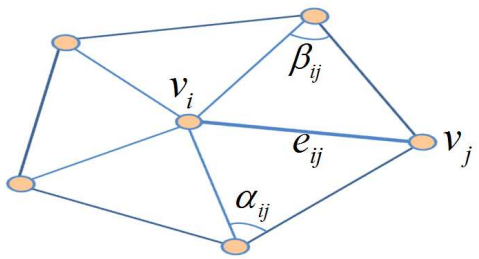


Figure 3.8: Construction of cotangent weights.

defined by

$$w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}, \tag{3.6}$$

where α_{ij} and β_{ij} are the two angles on opposite side of the edge e_{ij}. In [Bronstein 2010c] it is proposed to combine cotangent weights with vertex degrees taken to be proportional to the sum of the areas of the triangles sharing the vertex v_i. It has been argued that such a geometry-dependent discretization preserves many important properties of the continuous Laplace-Beltrami operator and is numerically consistent, *i.e.*, it yields consistent approximations across various triangulations [Wardetzky 2008]. The drawback of this representation is that it is well suited only for the Delaunay triangulation. In other cases they are problematic because the weights either become negative for flat triangles or become infinitely large for very short edges.

3.3.3.2 Gaussian Weights

In computer vision a weight function that ensures the locality of the operator when $v_i \approx v_j$ is often preferred, where $v_i, v_j \in \mathcal{V}$ are the shape graph vertices in $\mathbb{R}^3$. The most popular choice is the Gaussian function typically characterized by a bandwidth parameter σ, namely

$$\omega_{ij} = \exp(-\|v_i - v_j\|^2 / 2\sigma^2). \tag{3.7}$$

This choice has been thoroughly justified in the framework of non-linear dimensionality reduction based on Laplacian eigenmaps [Belkin 2003] and of spectral clustering [von Luxburg 2007]. Variations of these proximity-dependent weights were proved to be valid both for meshed surfaces [Belkin 2008] and for point-clouds [Belkin 2009]. We note that Gaussian weights have been successfully used to compute gradients on discrete manifolds in a robust manner [Luo 2009, Mukherjee 2010]. The ability to estimate gradients on meshed surfaces is extremely useful in the context of surface feature detection and description [Zaharescu 2009]. Graph Laplacians based on Gaussian weights were also used for spectral shape matching [Jain 2007, Mateus 2008], shape segmentation [Sharma 2010b] and recognition [Mahmoudi 2009].

3.3.3.3 Gaussian v/s Cotangent edge weighting schemes

In this section, we perform an empirical analysis of eigenvalues associated with these weighting schemes. We consider a variety of visual and graphical 3D shapes (shown in Figure 3.9) while analyzing the behavior of Laplacian eigenvalues with respect to these two weighting schemes. This includes: (a) A synthetic high resolution human mesh (SHR); (b) A synthetic low resolution human mesh (SLR); (c) A real unprocessed multi-camera human visual-hull (RVH); (d) A real processed multi-camera human mesh (RP); (e) A real processed human mesh with topological changes (RT); (f) A synthetic low resolution wolf mesh (SW); (g) A synthetic sphere mesh (SS); (h) A synthetic face mesh (SF), which is an open surface.

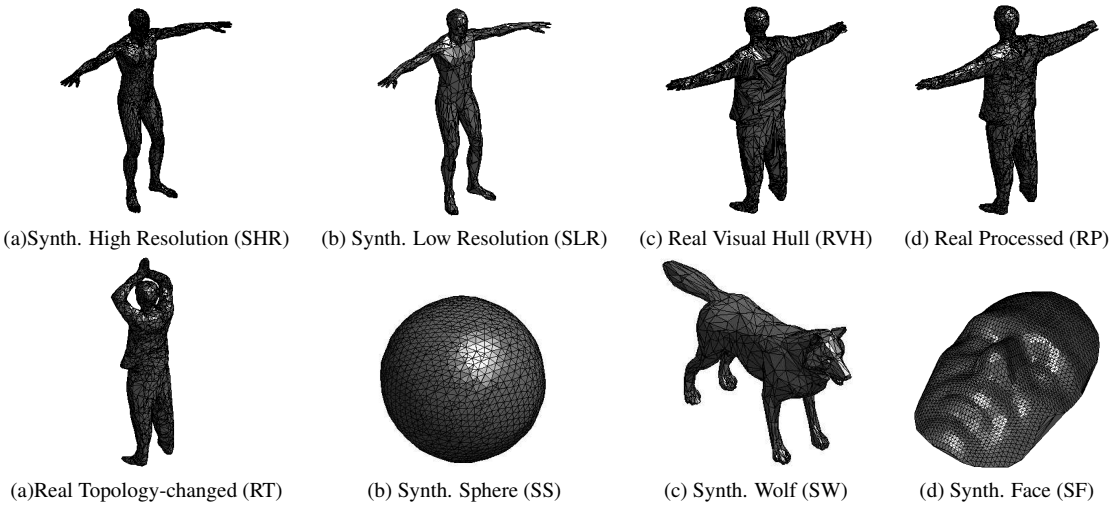


Figure 3.9: A collection of graphical and visual 3D shapes with varying geometrical characteristics. These shapes were used to analyze the eigenvalue distribution for the graph Laplacian computing using the Gaussian and cotangent weighting schemes.

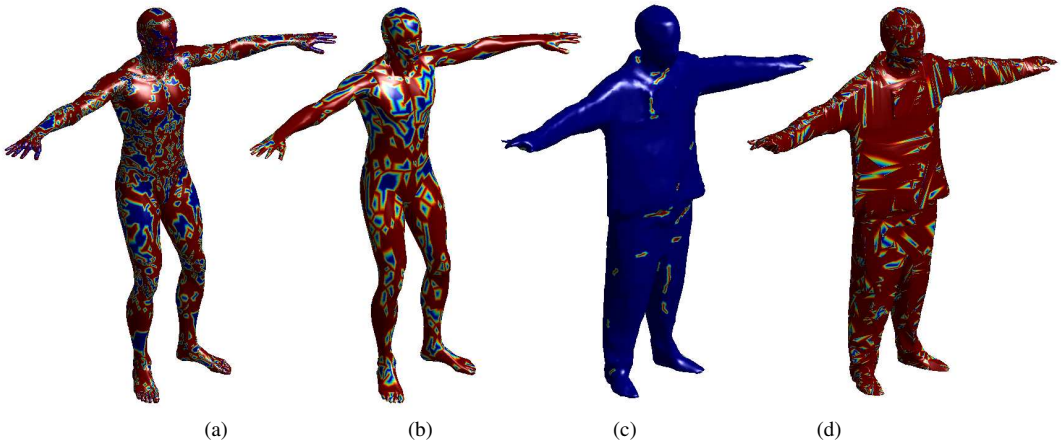


Figure 3.10: A visualization of negative cotangent edge weights. A facet is drawn in red if any of the vertices involved in that facet is also a part of an edge with negative cotangent weight. While the unprocessed visual shape (Real Visual Hull) shown in (d) has a large number of negative weights as a consequence of bad meshing, the graphical shapes in (a,b) also posses many negative cotangent weights due to the presence of flat triangles.

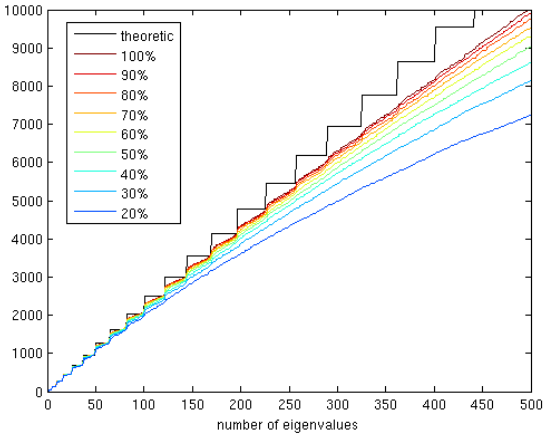


Figure 3.11: The eigenvalues of many discrete triangulated meshes of a sphere with decreasing resolution are compared with the theoretical eigenvalues using the Gaussian weighting.

In an attempt to quantify the approximation error of both the weighting schemes, the discrete approximation of corresponding eigenvalues are compared with the continuous eigenvalues for the synthetic sphere shape (shown in the Figure 3.9(b)) that are analytically computable (see [Courant 1962] for details). Figure 3.11 and Figure 3.12 show the plot of Laplacian eigenvalues of many discrete triangulated meshes of a sphere with decreasing resolution, compared with the theoretical eigenvalues using the Gaussian and cotangent weighting, respectively. From these figures, it appears that the discrete Laplacian using the cotangent weighting gives a slightly better approximation for the continuous Laplacian than the discrete Laplacian using the Gaussian weighting.

However, the cotangent weighting scheme is susceptible to the surface reconstruction noise in the visual data and also to the triangulation in the synthetic data. We compute cotangent weights for the shapes shown in Figure 3.9(a)-(d) and plot them with blue and red color in Figure 3.10 such that a facet is drawn in red if any of the vertices involved in that facet is also a part of an edge with negative cotangent weight and blue otherwise. In Figure 3.10(d), we can see that the cotangent weights get negative values for large number of edges due to raw triangulation, which is not a desirable property for edge weights. For a processed visual shape with uniform triangulation, we still get some negative weights as shown in the Figure 3.10(c). It is important to note that the cotangent edge weights can also be largely negative for graphical shapes, as shown in the Figure 3.10(a,b), due to the presence of large number of flat triangles.

Next, we analyze the distribution of eigenvalues of different shapes together, using the Gaussian and cotangent weighting schemes. When the meshing is consistent over all the shapes, the average size of the neighborhood $|\mathcal{N}|$ for all the shape graphs is approximately constant (being 5.960 ± 0.016 for the eight shapes in Figure 3.9). Since the average neighbor-

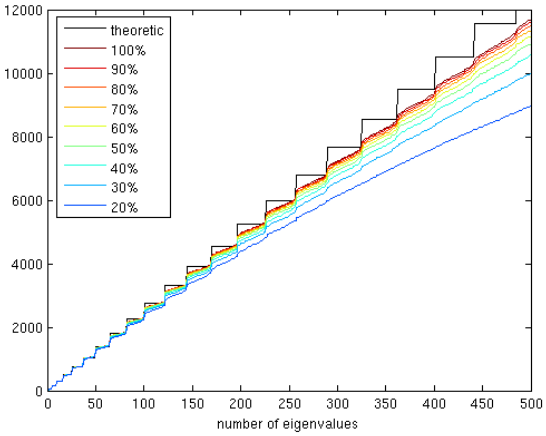


Figure 3.12: The eigenvalues of many discrete triangulated meshes of a sphere with decreasing resolution are compared with the theoretical eigenvalues using the cotangent weighting.

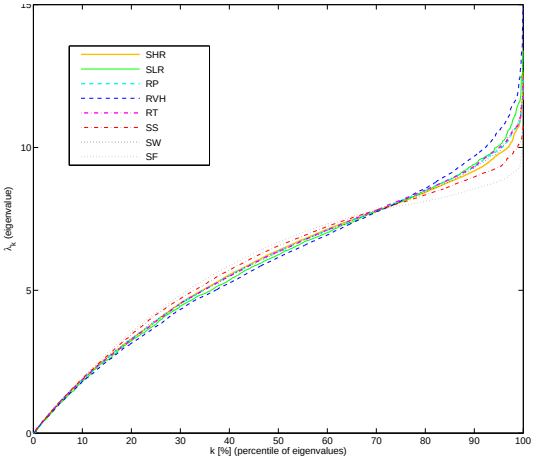


Figure 3.13: The distribution of the eigenvalues of the unnormalized Laplacian with Gaussian weighting for different shapes.

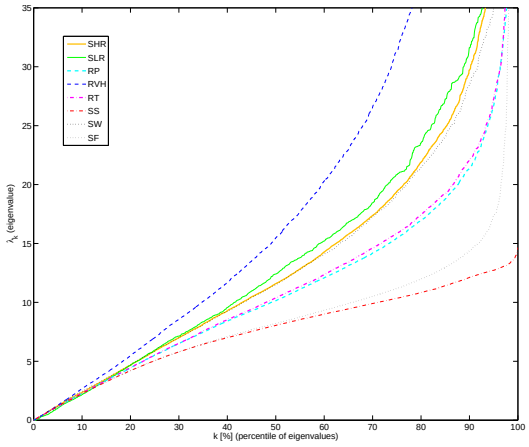


Figure 3.14: The distribution of the eigenvalues of the unnormalized Laplacian with cotangent weighting for different shapes.

hood size is equal to the average eigenvalue of $\mathbf{L}$,

$$|\mathcal{N}| = \frac{\text{tr}(\mathbf{L})}{n} = \frac{\sum_{k=1}^{n} \lambda_k}{n}, \tag{3.8}$$

the average eigenvalue is also stable for different geometries as shown in Figure 3.13.

The eigenvalue distribution is, however, not similar for different shapes when the cotangent weighting is used to compute the combinatorial Laplacian (see Figure 3.14). This is because, in case of cotangent weights, the average neighborhood size $|\mathcal{N}| = 127.6 \pm 325.9$ for the eight shapes has larger variance.

An important conclusion of this analysis is that the Gaussian weighting scheme is more appropriate for shape matching, while the cotangent weighting allows shape recognition, based on the Laplacian spectrum.

3.4 Discussion

In this chapter, we have outlined the pose invariant representation for visual shapes using the spectral decomposition of corresponding graph Laplacian eigenvectors. We further suggest the study of [Bruno 2010, Mateus 2009] for building a good understanding of spectral geometry processing. In this section, we will discuss various important aspects of the spectral representation outlined in this chapter.

The first important aspect of spectral embedding representation of a shape graph is the convergence of the discrete Laplace operator on graphs to a continuous Laplace-Beltrami operator on compact manifold. As mentioned in Section 2.7, this is an active area of study focusing

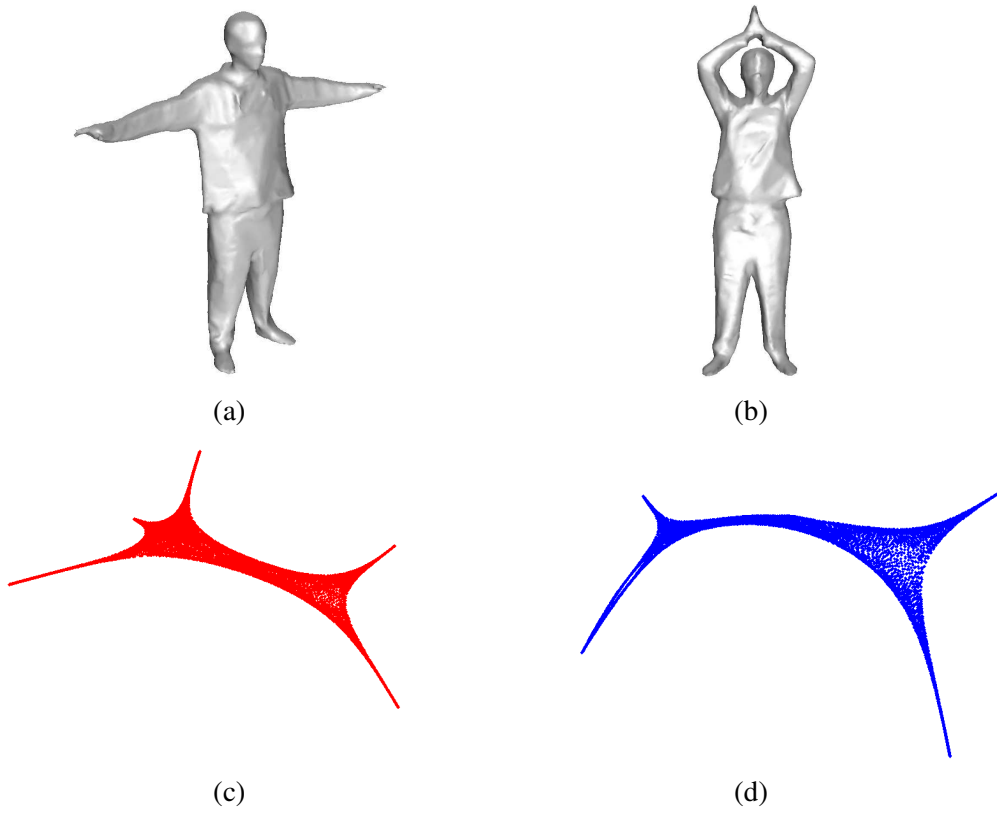


Figure 3.15: Illustration of deformation in spectral representation due to topological changes. Visual shapes with and without topological changes (a,b) and corresponding Laplacian embedding (c,d) plotted using first three non-null eigenvectors. Two embeddings are no more similar due to merging of hands.

mainly on problem of graph construction and weighting schemes. The graph construction problem is easy to handle in case of a shape graph as opposed to the similarity graphs for machine learning data due to manifold assumption and existence of rich mesh processing literature. However, the weighting scheme issue is more complicated given the trade-off among several weighting schemes, claiming a better discretization of the continuous Laplace-Beltrami operator [Wardetzky 2007]. It is more challenging to deal with these issues in case of visual shapes where majority of theoretical analysis is not directly applicable.

The second important aspect is the choice of embedding dimension K. As discussed in Section 2.6.6, the eigen-gap does not exist in case of regularly connected graphs, *e.g.*, shape graphs, due to absence of strongly and weekly connected cliques in such graphs. This can be seen in Figure 3.13 and 3.14 where eigenvalues for different geometric shape graphs are continuously increasing. However, using the PCA interpretation of Laplacian eigenvectors, we have proposed a scree diagram based method using Eq. (2.99) for the selection of K. The majority of existing shape analysis methods experimentally choose a value for embedding di-

mension. In particular, for 3D shape segmentation methods it is commonly assumed to choose a K equal to the number of desired segments based on perturbation assumption in spectral clustering [Ng 2001, von Luxburg 2007]. However, there is no theoretical guarantee that this will yield good segmentation results since the shape graphs are sparse regularly connected graphs and perturbation analysis of spectral clustering is not applicable for such graphs.

We have proposed an heuristic method to automatically select important eigenvectors for unsupervised shape segmentation in Chapter 4, thereby bypassing the need to choose K. We have also extended the scree diagram method to related dimensionality of Laplacian embedding and scale parameter while discussing a multi-scale shape representation in Chapter 6.

Finally, the pose invariance aspect of the spectral representation (discussed in Section 3.3) has some major limitations. The assumption that articulated shapes are (quasi-)isometric transformations with no stretching is little too restrictive and is valid mainly for graphical shapes. Visual shapes are more challenging due to the existence of surface noise (causing arbitrary surface stretching) as well as the topological changes like merging and splitting of the shape parts and incomplete/partial shapes. Hence, the spectral representation of the visual shapes is not a perfect solution for pose invariance. Figure 3.15 shows how the spectral embedding of visual shape deforms due existence of topological changes.

Nevertheless, we have proposed new probabilistic methods, using the machine learning tools like Gaussian Mixture Model and Expectation Maximization, for shape segmentation and registration in Chapter 4 and Chapter 5. These methods allows us to handle arbitrary surface stretching. We have also outlined a heat-kernel framework in Chapter 6 that allows us to describe and analyze a shape multiple scales. Thus, equipped with heat-kernel framework, we have proposed a method to analyze the visual shapes at smaller scale in order to address the problem of topological changes in Chapter 7.

CHAPTER 4

Unsupervised 3D Shape Segmentation

Contents

4.1 Introduction

In this chapter, we address the problem of segmenting 3D shapes into their constituting parts with emphasis onto complex articulated visual shapes. These shapes are difficult to describe in terms of their parts, *e.g.*, body parts of humans, because there is a large variability within the same class of perceptually similar shapes. The reasons for this are numerous: changes in pose due to large kinematic motions, local deformations, topological changes, etc. We have already obtained a shape graph representation for articulated 3D shapes in Chapter 3, which can be viewed as both a 2D discrete Riemannian manifold and a sparse graph. Therefore, the shape segmentation task can be cast as a graph partitioning problem for which the spectral clustering (SC) algorithm(s) 2.8.1 provides tractable solution.

We use the geometric properties of the Laplacian eigenvectors introduced in Section 2.6.4 and devise a new spectral clustering algorithm, well suited for the graphs with uniform connectivity, such as the shape graphs. More precisely, we attempt to characterize the one-dimensional (1D) *projections* of a graph onto its eigenvectors based upon the *nodal domain* theory [Biyikoglu 2007] and to build a practical algorithm using the *principal component* interpretation of the Laplacian eigenvectors presented in Section 2.6.5. The novel unsupervised clustering algorithm is applied over a shape graph in order to obtain an unsupervised segmentation of the corresponding visual shape.

The organization of this chapter is as follows. The literature survey is presented in Section 4.1.1 followed by the summary of contributions in Section 4.1.2. Section 4.2 briefly recalls few mathematical properties of the graph Laplacian eigenvectors and introduce the nodal domain theorem along with a strategy to characterize the Laplacian eigenvectors using these properties. In Section 4.3, we propose a novel clustering algorithm for shape segmentation. Section 4.4 presents shape segmentation results before concluding the chapter in Section 4.5.

4.1.1 Related Works

Unsupervised segmentation of the articulated shapes is a well investigated problem. Some popular (non-spectral) unsupervised shape segmentation methods includes: [Shlafman 2002]-a K-means algorithm applied on mesh facets; [Attene 2006]-a hierarchical clustering algorithm based on fitting primitives like sphere and cylinder; [LAI 2008]-an algorithm with random walk on the dual shape graph; [Golovinskiy 2008]-two hierarchical clustering algorithms using the normalized cut and randomize cut on the shape graph; [Shapira 2008]-a shape diameter function based algorithm. One can find a quantitative comparison of these methods in [Chen 2009]. These non-spectral methods explicitly analyze the connectivity of the graph structure in order to find clusters/segments.

On the other hand, the spectral methods are natural choice for pose-invariant segmentation as they exploit the underlying manifold structure of the shape graphs by embedding the shape in an isometry invariant space. Spectral clustering methods use the shape graph representations of the data and solve for graph partitioning within the context of the spectral graph theory. Early spectral approaches recursively compute the normalized cut [Shi 2000] over the graph using the first non-null Laplacian eigenvector (also known as the Fiedler vector [Chung 1997]) and are referred to as *spectral bi-partitioning (SB) methods*. It has been noticed that this does not guarantee *good* clusters as the normalized cut is computed recursively, irrespective of the global structure of the data (see [Belkin 2003]).

Recent spectral methods use the k smallest non-null eigenvectors of the Laplacian matrix (or one of its variants) to optimally embed the graph onto a k dimensional subspace [Belkin 2003], and to cluster the embedded graph nodes into k groups. Various grouping strategies can be used, such as a direct extensions of SB to multiple eigenvectors, *i.e.*, greedy ordering heuristic [Alpert 1999] or a K-means algorithm in the embedding space [Ng 2001, Belkin 2003, Yu 2003]. In [Zelnik-manor 2004, Nadler 2006], a number of limitations of the spectral clustering method are analyzed, thus focusing on the problems of noise, density and scale variations in the data. A spectral clustering based temporally consistent shape segmentation method was proposed in [Cuzzolin 2008].

However, the use of these spectral clustering algorithms cannot be generalized to any type of graphs. In the case of *sparse* graphs with *uniform* connectivity, there is no obvious optimal graph partitioning solution, namely, the extraction of a number of strongly connected components that are only weakly interconnected. Indeed, the Laplacian matrices of such graphs

cannot be viewed as *slightly perturbed* matrices of the ideal case (between-cluster similarity is exactly 0) because of the lack of a meaningful eigengap [von Luxburg 2007]. As a consequence, the estimation of the dimension of the embedding (and hence of the number of clusters) based on eigenvalue analysis [Chung 1997, Ng 2001, von Luxburg 2007] has several drawbacks when one deals with sparse graphs whose vertex connectivity is almost uniform across the graph. First, there is no eigengap in such cases and therefore it is not straightforward to estimate the dimension of the spectral embedding in a completely unsupervised way. Second, the eigenvalues of any large semi-definite positive symmetric matrix are estimated only approximately; this means that it is not easy to study the eigenvalues' multiplicities (which play a crucial role in the analysis of the Laplacian eigenvectors [Biyikoglu 2007]) and that ordering the eigenvectors based on these estimated eigenvalues is not reliable 3.3.2.1. This has dramatic consequences if one seeks some form of repeatability when clustering similar but not identical sets of data.

Interestingly, there exists a theoretical framework that extends the Fiedler's theorem to the other eigenvectors of a graph, namely, the *discrete nodal domain theorem* [Davies 2001]. The discrete nodal domain theorem is an extension to the Courant's nodal domain theorem, which originally applies to the eigenfunctions of the Laplace-Beltrami operator on continuous manifolds. Thus, the discrete extension applies to the eigenvectors of the discrete Laplacian operator on graphs. Nodal domains (for details see [Biyikoglu 2007] and Section 4.2 below) provide families of graph segmentations, where there is one segmentation (partitioning) for each eigenvector of the adjacency matrix [Powers 1988] or of the Laplacian matrix [Davies 2001]. This provides a framework for *individual-eigenvector* analysis.

In practice, each one of the graph's cluster is present not only in one such segmentation, but in several segmentations. This means that there is no direct association between clusters and eigenvectors. Therefore, a clustering algorithm based on these geometric properties should combine only those eigenvectors that best reveal the graph's clusters. Empirically, it has been observed by us and by others [Reuter 2009], that the perceptually prominent clusters of a graph are revealed by the nodal domains of certain eigenvectors among the ones associated with the smallest eigenvalues. The nodal domains have been recently discussed in the context of shape parametrization [Lévy 2006] and of shape segmentation [Reuter 2009]. Although these authors suggest the idea of selecting a few significant eigenvectors, they do not actually propose an algorithm for that.

We have empirically observed that more than one perceptually prominent cluster of a graph is projected onto a single eigenvector, contrary to the assumption of the ideal case in the standard spectral clustering, where a single eigenvector represents an indicator vector for an individual cluster. Therefore, a clustering algorithm based on these geometric properties, should combine only those eigenvectors that best reveal the graph's clusters.

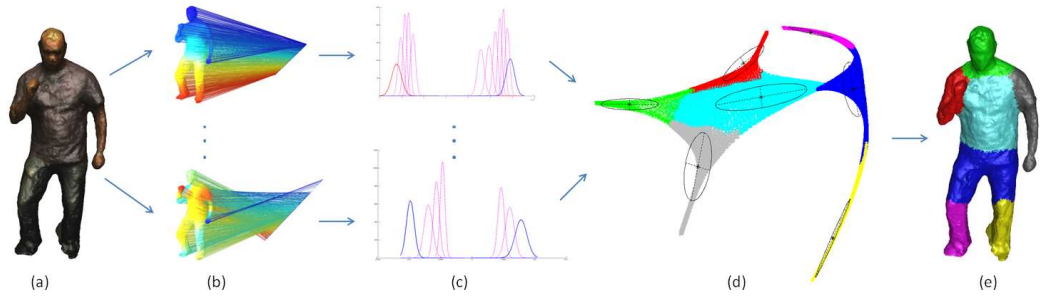


Figure 4.1: A shape graph with approximately 17,000 vertices and with an average of six edges per vertex is (a) projected onto ten eigenvectors (b) corresponding to the ten smallest eigenvalues of the normalized Laplacian. Once the vertices with zero eigenfunction values are removed (see Section 4.3 below) we fit 1D Gaussian mixtures with an optimal BIC criterion (c). The eigenvector values associated with the left most (negative values) and right most (positive values) components of these Gaussian mixtures are selected as potential cluster seeds on the extrema of the shape graph. Whenever such a cluster corresponds to a connected component, the corresponding eigenvector is selected and considered for the embedding. In this example, the method selected 5 eigenvectors. A 5-dimensional Gaussian mixture with 7 components is fitted to this embedding (d) thus segmenting the shape into 7 clusters (e).

4.1.2 Contributions

In this chapter, we devise a novel unsupervised probabilistic segmentation method, based on 1D Gaussian mixtures with model selection, in order to reveal the structure of the graph projections on its eigenvectors. Based on this, we show that we are able to select a subset of the set of eigenvectors corresponding the the smallest eigenvalues of the Laplacian. These selected eigenvectors are then used to embed the graph. We show how this eigenvector-by-eigenvector analysis allows to initialize the clustering that is carried out either with a non-parametric method (hierarchical clustering, K-means) or with Gaussian mixtures. The advantage of the latter is that it can be combined with the Bayesian information criterion (BIC) [Fraley 2002] to estimate the optimal number of clusters, when this number cannot be provided by eigenvalue analysis, *i.e.*, the existence of an eigengap. Figure 4.1 presents a step-by-step outline of the proposed shape segmentation method.

We apply the proposed segmentation method to the shape graphs representing complex shapes such as articulated bodies with several protrusions. We seek natural segmentations of these shapes such that clusters correspond to body parts. We observe that the proposed algorithm provides perceptually meaningful clustering and that finer body-part details (hands, legs, thighs, hips, torso, head, etc.) correspond to finer segmentations, i.e. a simple increase in the number of clusters.

In practice, unlike traditional discrete geometry approaches to mesh processing (which use the cotangent weights [Pinkall 1993, Lévy 2006, Reuter 2009]), we use the Gaussian weights

while describing the shape graphs for the visual shapes (for details see Section 3.3.3).

4.2 Analysis of Laplacian Eigenvectors

We have already discussed in detail various properties of the graph Laplacian eigenvectors in Section 2.6.4. In this section, we present the nodal domain analysis of these eigenvectors and introduce a heuristic eigenvector selection method for finding a subset of interesting eigenvectors that are for shape segmentation method proposed in the next section.

We can recall few interesting properties of the graph Laplacian eigenvectors from Chapter 2. Assuming the combinatorial Laplacian eigen-decomposition $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ for a given shape graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ where $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ and using the proposition 1, we can deduce:

$$\sum_{i=1}^{n} u_{ik} = 0, \quad \forall k, 2 \leq k \leq n \tag{4.1}$$

$$-1 < u_{ik} < 1, \quad \forall i,k, 1 \leq i \leq n, 2 \leq k \leq n. \tag{4.2}$$

While, the notation u_{ik} represents the i-th element of the k-th eigenvector and can also be written as $\mathbf{u}_k(v_i)$. It emphasizes the fact that an eigenvector is an eigenfunction of $\mathbf{L}$ that maps the vertices of the graph onto an open interval on the real line:

$$\mathbf{u}_k : \mathcal{V} \mapsto \ ]-1;+1[. \tag{4.3}$$

Another important property of the graph Laplacian used by the proposed algorithm is the principal component analysis (PCA) of a graph (see Section 2.6.5). An important implication of Theorem 3 is that the Laplacian eigenvectors can be interpreted as the principal component of the respective shape graph or in other words the eigenvectors of the Laplacian matrix $\mathbf{L}$ associated with its smallest non-null eigenvalues can be interpreted as the axes of maximal variance of the embedded graph.

4.2.1 Nodal Sets and Nodal Domains

Some important properties of the Laplacian eigenvectors stem from the discrete version of the *nodal domain theorem* for elliptic operators on manifolds (see [Biyikoglu 2007], Chapter 3). The *nodal set* of an eigenvector $\mathbf{u}$ is the set of graph vertices such that $\{v_j | \mathbf{u}(v_j) = 0\}$. The *nodal domains* refer to the set of *connected components* of the complement of the nodal set, *i.e.*, the components such that $\{v_j | \mathbf{u}(v_j) \neq 0\}$ and bounded by the nodal sets. However, on a discrete domain such as a graph representation, an eigenvector can change from positive to negative values without passing through zero. Therefore one needs to extend the nodal domain definition to include both *strong* nodal domains ($\mathbf{u}(v_j) < 0$ or $\mathbf{u}(v_j) > 0$) as well as *weak* nodal domains ($\mathbf{u}(v_j) \leq 0$ or $\mathbf{u}(v_j) \geq 0$). The number of strong and weak nodal domains of an eigenvector is bounded by the following theorem (adapted from [Davies 2001]):

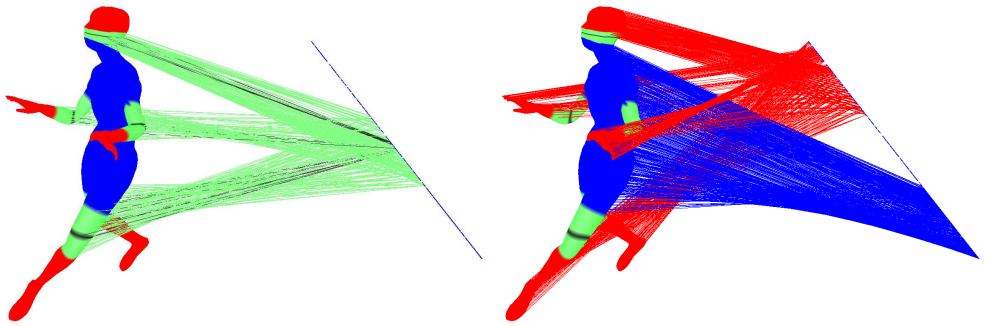


Figure 4.2: Nodal sets and the nodal domains associated with the sixth Laplacian eigenvector of a 7,000 vertex shape graph. Eigenvector $\mathbf{u}_6$ is shown with a blue line. The nodal sets (left) are shown in green. There are six weak nodal domains in this case (right).

Discrete nodal domain theorem. *Let* $\mathbf{L}$ *be the Laplacian matrix of a connected graph with* n *vertices. Then any eigenvector* $\mathbf{u}_i$ *corresponding to the* i*-th eigenvalue* λ_i *with multiplicity* r *has at most* i *weak nodal domains and* $i+r-1$ *strong nodal domains.*

We observe that the properties of the Fiedler vector, *i.e.*, $\mathbf{u}_2$ associated with eigenvalue $\lambda_2 > 0$ can be viewed as a corollary of this theorem: The multiplicity of λ_2 is equal to one and the Fiedler vector has two weak nodal domains, a positive one and a negative one. This property has already been used in the past by spectral bi-partitioning. Nevertheless, the design of an algorithm based on the above theorem, i.e, properties associated with the nodal domains of the other eigenvectors is not straightforward. Figure 4.2 shows the nodal set (left) and the nodal domains (right) of eigenvector $\mathbf{u_6}$ of a graph Laplacian corresponding to a shape graph with 7,000 vertices. The nodal set (left) appears on the head as well as on the two arms and two thighs. In this example, there six nodal domains (right), namely, one positive domain (the main body) and five negative domains (head, left and right hand, left and right legs). Clearly, the nodal domains/sets do not provide a perceptually meaningful segmentation.

4.2.2 A Heuristic for Eigenvector Selection

The intuition behind the PCA of a graph is that two different connected components that are farther away from each other should project onto the positive and negative extremities of one of the eigenvectors. Given the nodal domains of individual eigenvectors, it is intuitive to select only those eigenvectors that have a 1D cluster of vertices either at its positive or at its negative extremity. Notice that not *all* of the smallest eigenvectors of a graph feature significant clusters at their extremities. This suggests that the spectral clustering may include some form of eigenvector selection based on the availability of 1D clusters at their extremities. This heuristic is also consistent with the fact that the nodal domains have high values at their centers and low values on their borders (*i.e.*, the nodal sets).

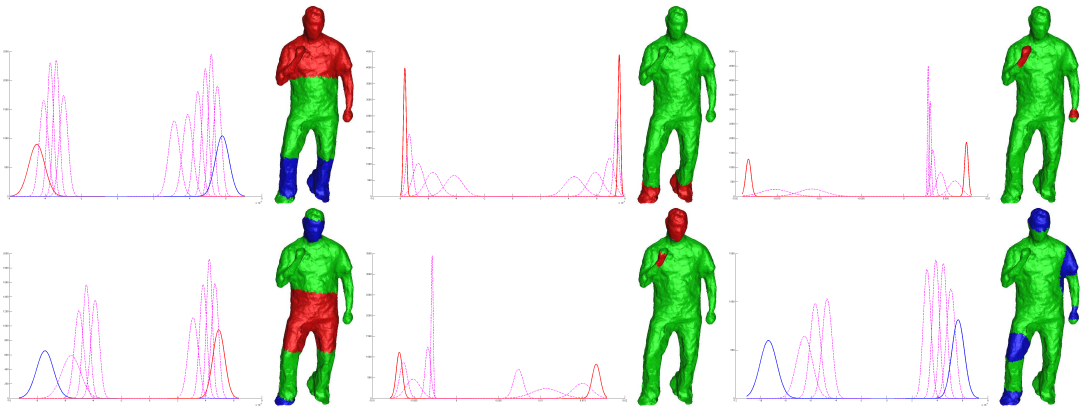


Figure 4.3: 1D Gaussian clusters along the left- and right-side of Laplacian eigenvectors. The leftmost and the rightmost clusters of each eigenvector are either represented in red, if the corresponding vertex set belongs to a single connected component, or in blue, if the corresponding vertex set belongs to several connected components.

4.3 The Proposed Clustering Algorithm

The PCA interpretation of the Laplacian embedding suggest that the projection of a graph onto each one of its eigenvectors could provide interesting information about the structure of the graph associated with the data, as discussed above. Indeed, connected components project as one-dimensional clusters along the eigenvectors; Moreover, these connected components are more likely to project towards the extremities of the eigenvectors rather than towards their centers, *i.e.*, at the lower and upper bounds of the open interval $]-1;+1[$, see Eq. (4.3). This suggests that one can detect 1D clusters along the eigenvectors, select the leftmost and rightmost ones, and associate them with the central regions of the nodal domains of a graph and use them for initialization of the graph clustering, as shown in the Figure 4.3.

Traditionally, the latter is performed by the K-means algorithm, which uses as many clusters as the number of smallest eigenvectors. In the proposed algorithm, we select a set of q eigenvectors that have identifiable 1D clusters at their extremities and we embed the graph in the isometric space spanned by these q eigenvectors. Then we perform clustering in this q-dimensional space. An open issue with the proposed algorithm is how to choose the number of clusters. For this purpose we combine GMM clustering with an optimality criterion for model selection, *i.e.*, the Bayesian information criterion (BIC). For practical reasons, we remove the vertices with nearly zero eigenfunction value, while performing 1-D Gaussian analysis. This allows GMM to capture small scale clusters along the extremities of an eigenvector.

Consequently, we suggest an algorithm that first performs 1D clustering along the p smallest eigenvectors. Second, it detects clusters found at the extremities of these vectors. Third, it performs a simple connectivity analysis in order to determine whether each one of these extremity-clusters belong to a single connected component, *i.e.*, belongs to a single nodal do-

main. This allows us to select a subset of $q \leq p$ eigenvectors that are well suited to embed the graph. Finally we fit a q-dimensional Gaussian mixture to the embedded data and we apply an optimality criterion for model selection. This yields the algorithm outlined below.

Clustering algorithm:

1. Compute the first p non null eigenvectors of the graph Laplacian, $[\mathbf{u}_2, \ldots, \mathbf{u}_{p+1}]$.
2. For each eigenvector $\mathbf{u}_i$:
   (a) Remove the vertices with eigenfunction value close to zero.
   (b) Perform 1D clustering using GMM with optimal model selection.
   (c) Choose the outer clusters, *i.e.*, the leftmost one and the rightmost ones.
3. Perform connectivity analysis of all the vertex sets associated with the outer clusters, thus providing an embedding of size $q \leq p$.
4. Embed the graph in the space spanned by the selected eigenvectors and fit a q-dimensional GMM. Select the number of clusters based on BIC.

The main differences between the proposed algorithm and the standard spectral clustering are followings. The size of the embedding is not governed any more by the detection of an eigengap. Instead, 1D GMM allows a completely unsupervised selection of a set of eigenvectors well suited to embed the graph. The K-means clustering is replaced with fitting the GMM with model selection. Thus, instead of the Euclidean distance metric used in the K-means, we use the Mahalanobis distance metric while fitting GMM to data.

4.4 3D Shape Segmentation Results

We illustrate the segmentation results obtained with the unsupervised method developed in this chapter and applied to the shape graphs representing 3D articulated objects. Such shape graphs are by definition, very sparse and with regular local connectivity. Hence, it is desired that the clusters should correspond to the object parts, protrusions (hands, legs, head) as well as the torso and hips.

We obtain results on the shape graphs representing the visual shapes that are more challenging as compared to most of the graphical shapes. We compare the results of proposed method with the standard spectral clustering approach presented in [Ng 2001]. However, the proposed algorithm was unable to find an optimal BIC value for deciding the number of components, *i.e.*, k in selected subspace. This is because the BIC curve was observed to be continuously increasing (see Figure 4.4), thereby suggesting the non-existence of cluster structure in the regularly connected shape graphs. We tried different values for k and here we show results with $k = 5, 7, 9, 11, 13$.

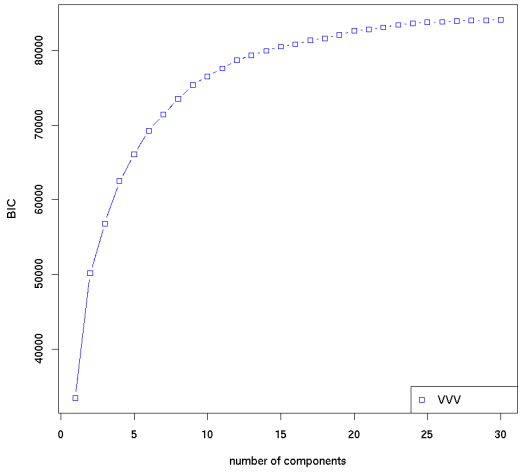


Figure 4.4: BIC plot as a function of number of components (k) for the shape shown in Figure 4.5.

(a) k=5 (b) k=7 (c) k=9 (d) k=11 (e) k=13 (f) k=15

Figure 4.5: Segmentation results obtained with the graphs of an articulated shape and with different values of k. Colors encode different clusters. But the coloring is not the same for all the segmentations. The top row shows the segmentation obtained with the standard spectral clustering. The bottom row shows the results obtained with the proposed algorithm.

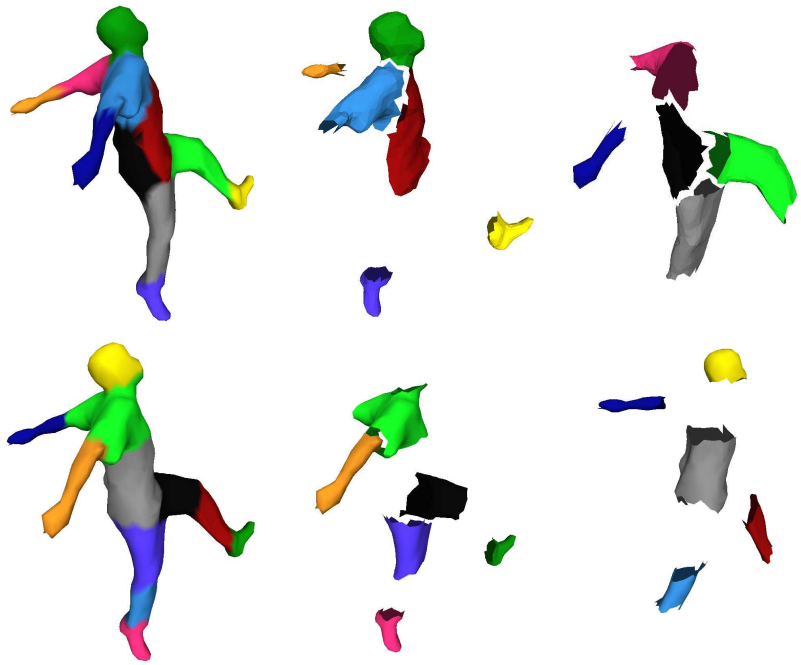
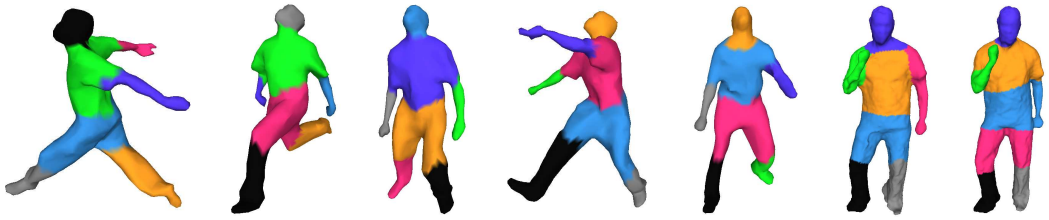
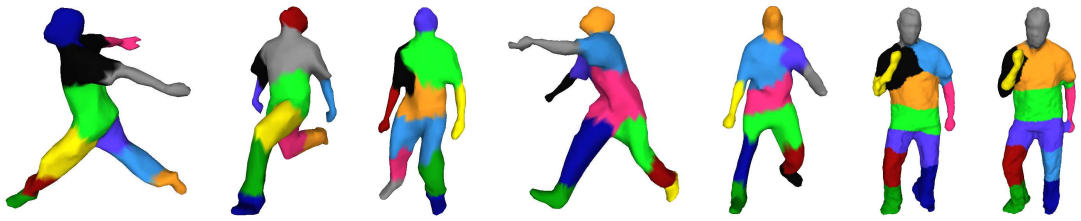


Figure 4.6: The figure shows how standard and proposed clustering method splits an articulated visual. In the first row, the segmentation obtained by the standard approach segments the torso vertically while the segmentation obtained by the proposed method, shown in the bottom row, segments the shape into perceptually meaningful body parts. Here we choose $k = 11$.



(a) (b) (c) (d) (e) (f) (g)

Figure 4.7: Segmentation results obtain with the proposed method over different articulated visual shapes with $k = 7$. Here the different colors are used to represent different parts of the shape. But this coloring is not part-wise consistent over different poses and subjects.



(a) (b) (c) (d) (e) (f) (g)

Figure 4.8: Segmentation results obtain with the proposed method over different visual shapes with $k = 11$. Notice that as the number of clusters is increased from 7 to 11, we still obtain a segmentation in terms of semantically and perceptually meaningful parts.

In case of standard spectral clustering implementation, we use the k-dimensional subspace while finding k segments. In the proposed method, we consider a set of $p = 10$ eigenvectors for selecting a subset of eigenvectors, independent of the value of k. The average dimension of our selected subspace was 5. In the examples shown here, the coloring of identified segments is not consistent over the individual images. Nevertheless, we can still easily observe the consistency of our results *w.r.t.* segment boundaries.

Figure 4.5 compares our results with the standard approach. Each column of the figure represents segmentation for an human articulated 3D shape with with different values of k. The first row of the figure corresponds to segmentations obtained by the standard spectral clustering approach, which applies K-means to the embedding defined by the first k non-null eigenvectors. The second row presents the results of the proposed method. In column Figure 4.5(a),(b) the standard approach completely failed in assigning distinct parts with same label color, while in columns (c-d) the segmentation results by the proposed method are more consistent and perceptually meaningful.

Figure 4.6 shows a comparison of the segmentation results by the standard and proposed method at $k = 11$. Here, we can see that the proposed method segments the shape graph into perceptually meaningful parts (working with 5-dimensional embeddings in most of the examples shown here) as compared to the standard approach (which uses k dimensional space).

One possible explanation is that the standard SC method uses Euclidean distance metric in K-means, while the proposed method uses Gaussian mixture modeling, which in turn, uses the Mahalanobis distance metric. Although the Gaussian assumption for the embedded shape graph coordinates is not completely true, still the GMM equipped with the Mahalanobis distance is more appropriate to model elongated shapes in embedded space.

In Figure 4.7, we present the segmentation results with the proposed method on visual shapes with different articulated poses, with subject variation, with no vertex correspondences and with $k = 7$. In the last column 4.7(g) we have shown a result on a difficult topology: a visual shape having topological change, *i.e.*, in this example the merging of hand with torso. Although the proposed method, which uses the Laplacian eigenvectors is not completely ro-

bust to such large topological changes (see Section 3.4), we are still able to get consistent segmentation results, except locally for the regions that are directly involved in the topology change.

In Figure 4.8, we present the segmentation results on different articulated visual shapes with $k = 11$. The first five examples involve shapes taken from the flashkick sequence [Starck 2007b], with an average of 1500 vertices. The last pair of results correspond to the shapes of the Ben data sequence [Franco 2009] with (approx.) 17000 vertices.

We use the implementation provided by MCLUST library in R [Fraley 2006] for the purpose of both 1D and multidimensional Gaussian fitting.

4.5 Conclusion

A novel spectral clustering algorithm based on a detailed analysis of geometric properties of the Laplacian eigenvectors has been proposed in this chapter. More specifically, we devised an unsupervised probabilistic method, based on Gaussian mixtures with model selection, which reveals the nodal-domain structure of each eigenvector. This enables us to select a subset of eigenvectors among the smallest eigenvectors of a graph, to embed the graph in the space spanned by this selection, and to cluster the graph's vertices using a multidimensional GMM with model selection. When applied to shape graphs corresponding to articulated objects, such as humans, our method segments the latter into perceptually meaningful parts.

Nevertheless, the unsupervised segmentation is a difficult task since the segmentation itself is inherently a subjective task. Though, here we characterize individual eigenvectors based upon how they project a shape graph, in order to find a subspace that separately project different shape protrusions, making it suitable for the spectral clustering technique. However, in case of visual shapes, there is no guarantee that proposed algorithm will always provide a coherent and meaningful segmentation on similar shapes. This is due to the data driven nature of the algorithm, which can easily fail to handle the challenges associated with visual shapes, *e.g.*, the topological changes, variability in the sampling and variation in graph construction techniques. To overcome these problems, we propose a new semi-supervised shape segmentation method in the next chapter.

CHAPTER 5

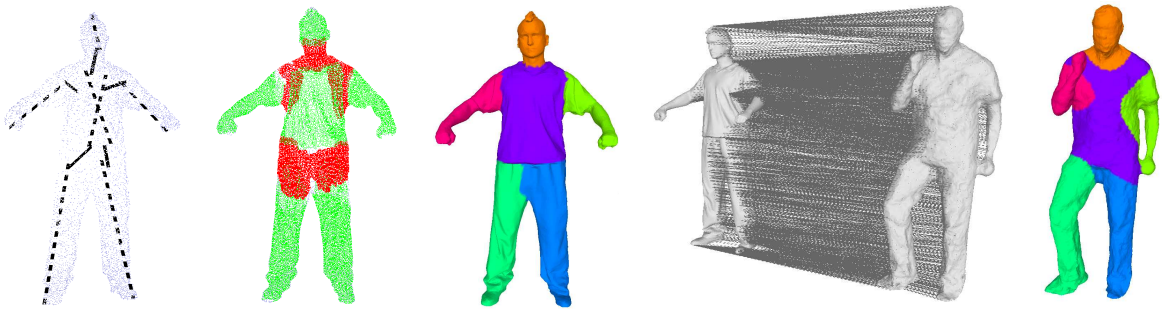
Semi-supervised 3D Shape Segmentation

Contents

5.1 Introduction

We have already introduced an unsupervised 3D shape segmentation method in Chapter 4. However, we realized that the segmentation task is very subjective in nature and a completely unsupervised solution can lead to undesired segmentation results. The spectral clustering based data-driven shape segmentation solution presented in the previous chapter provides some consistent shape segmentation result for a visual shape represented by the shape graph. Nevertheless, the *unsupervised* spectral clustering algorithms will not always yield satisfactory shape segmentation results for the following reasons: Distances between vertices are only locally Euclidean (manifold structure), the graph has bounded connectivity (sparseness), and the number of edges meeting at each vertex is almost the same through the graph (regular connectivity). Manifoldness will exclude methods that need a fully-connected affinity matrix. While the sparseness makes the *shape graphs* a good candidates for the Laplacian embedding [Belkin 2003, Spielman 2007], the usual spectral clustering assumptions do not hold in the case of regular connectivity. First, the Laplacian matrix of a shape graph cannot be viewed as a slightly perturbed version of the ideal case[1], namely, a number of strongly connected

[1] In the ideal case the between-cluster similarity cost is exactly 0.



(a) (b) (c) (d) (e)

Figure 5.1: First stage: *Constrained spectral clustering* (CSC), which takes as input a shape graph together with a sparse set of must-link (dashed lines) and cannot-link (full lines) constraints (a). These constraints are propagated using the commute-time distance (b). Spectral clustering is applied to a modified graph Laplacian (c). Second stage: *Probabilistic label transfer* (PLT). Shape segmentation is performed via a vertex-to-vertex matching (d) and probabilistic label transfer (e).

components that are only weakly interconnected [von Luxburg 2007]. Second, there is no eigengap and hence there is no simple way to determine the number of clusters. Third, the eigenvectors associated with the smallest non-null eigenvalues cannot be viewed as the relaxed indicator vectors [von Luxburg 2007]. Thus, a completely data-driven method can never guarantee meaningful and consistent segmentation/clustering results on visual shapes. Hence, we focus on a semi-supervised setup for shape segmentation task. In this chapter, we propose a learning approach to shape segmentation via a two-stage method, see an overview of the proposed method in Figure 5.1. First we introduce a new constrained spectral clustering (CSC) algorithm, which takes as input a shape graph $\mathcal{G}_{\text{tr}}$ from a *training* set. $\mathcal{G}_{\text{tr}}$ contains unlabeled vertices as well as a set of *must-link* and *cannot-link* constraints between pairs of vertices, Figure 5.1-(a). These constraints are propagated, using the unnormalized Laplacian embedding and the commute-time distance (CTD) such that the edge-weights corresponding to within-cluster connectivity's are strengthened while those corresponding to between-cluster connectivity's are weakened, Figure 5.1-(b). This modified embedding yields improved shape segmentation results than the initial one, *e.g.*, Figure 5.1-(c), because it better fits into the theoretical requirements of spectral clustering [von Luxburg 2007].

Second, we consider shape alignment based on vertex-to-vertex graph matching as a way to probabilistically *transfer* labels from a training-set of segmented shapes to a test-set of unsegmented ones. We consider a shape graph $\mathcal{G}_{\text{test}}$ from a *test* set. The segmentation of $\mathcal{G}_{\text{test}}$ is carried out via a new probabilistic label transfer (PLT) method that computes a point-to-point mapping between the embedding of $\mathcal{G}_{\text{tr}}$ and $\mathcal{G}_{\text{test}}$, *e.g.*, Figure 5.1-(d). This completely unsupervised matching is based on [Mateus 2008, Horaud 2011] and allows to transfer labels from a segmented shape to an unsegmented one. Consequently, the vertices of $\mathcal{G}_{\text{test}}$ can be

classified using the segmentation trained with $\mathcal{G}_{\text{tr}}$, Figure 5.1-(e). While the spectral graph matching is appealing, it adds an extra difficulty because of the ambiguity in the definition of spectral embeddings up to switching between eigenvectors corresponding to eigenvalues with multiplicity and changes in their sign [Mateus 2008, Bronstein 2010b]. This is particularly critical in the presence of symmetric shapes [Ovsjanikov 2008].

The structure of this chapter is as follows. In Section 5.1.1, a brief survey of the related methods in the literature is presented. This is followed by chapter contributions in Section 5.1.2. In the next Section 5.2, we introduce the notion of modified Laplacian embedding. Section 5.3 introduces the novel constraint spectral clustering method, which uses modified Laplacian embedding obtained with constraints propagation. In the following Section 5.4, we outline the shape segmentation method using the output of the CSC and the probabilistic label transfer algorithm. Finally, we present the experimental results and comparison with existing methods in Section 5.5, followed by the concluding remarks in Section 5.6.

5.1.1 Literature Survey

For the reasons already mentioned in the introduction, the results of simple spectral clustering (SC) are unsatisfactory [von Luxburg 2007]. Therefore, more recent methods, such as [Reuter 2010] and [Liu 2007], also take the topological features of a shape in its embedded space into account and can achieve this way impressive segmentation results. However, these methods do not provide intuitive means to include constraints in a semi-supervised framework.

Regarding semi-supervised spectral methods, we distinguish between semi-supervised and constrained spectral clustering methods: With semi-supervised spectral methods, we consider algorithms, which attempt to find good partitions of the data given partial labels. In [Szummer 2002] the labeled data information is propagated to nearby unlabeled data using a probabilistic label diffusion process, which needs an extra time parameter that must be specified in advance [Coifman 2006, Qiu 2007, Bronstein 2010b]. In [Belkin 2004] the labeled data are used to learn a classifier that is then used to sort the unlabeled data. These methods work reasonably well if there are sufficient labeled data or if the data can be naturally split into clusters. Furthermore, these methods were only applied to synthetic "toy" data and their extension to graphs that represent shapes may not be straightforward.

Constrained clustering methods use prior information under the form of pairwise must-link and cannot-link constraints, and were first introduced in conjunction with constrained K-means [Wagstaff 2001]. Subsequently, a number of solutions were proposed that consist in learning a distance metric that takes into account the pairwise relationships; This generally leads to convex optimization [Xing 2002, Bilenko 2004]. Since K-means is a ubiquitous post-processing step with almost any SC technique, it is tempting to replace it with constrained K-means. However, this does not take full advantage of the graph structure of the data where edges naturally encode pairwise relationships. Recently, metric learning has been extended to constrained spectral clustering leading to quadratic programming [Li 2009]. The semi-

supervised kernel K-means method [Kulis 2009] incorporates constraints by adding reward and penalty terms to the cost function to be minimized.

Another way to incorporate constraints into spectral methods is to modify the affinity matrix of a graph using a simple rule: Edges between must-link vertex-pairs are set to 1 and edges between cannot-link pairs are set to 0 [Kamvar 2003]. Despite its simplicity, this method is not easily extendable to our case due to graph sparsity: one has to add new edges (with value 1) and to remove some other edges. This will modify the graph's topology and hence it will be difficult to use the segmentation learned on one shape in order to segment another shape.

All methods described above need a large number of constraints to work well, which is a major drawback, as it is desirable to work with a small set of sparse constraints. We note that the issue of constraint propagation is not well studied: The transitivity property of the must-link relationship has already been explored [Yu 2004] but this cannot be used with the cannot-link relationship, which is not transitive.

5.1.2 Contributions

This chapter has two main contributions: 1) A new constrained spectral clustering method that uses the unnormalized Laplacian embedding to propagate pairwise constraints and a modified Laplacian embedding to cluster the data; 2) A new shape segmentation method based on spectral graph matching and on a novel probabilistic label-transfer process.

We exploit the properties of the *unnormalized graph Laplacian* presented in Section 2.6.4, which embeds the graph into an isometric space armed with a metric, namely the Euclidean *commute-time distance* (CTD) (see Section 2.6.5.1). Unlike the diffusion maps that are parametrized by a discrete time parameter, which acts as a scale, [Coifman 2006], the CTD reflects the average connectivity of two graph vertices: All possible paths of all lengths. We build on the idea of modifying the weighted adjacency matrix of a graph using instance level constraints on vertex-pairs [Kamvar 2003]. We provide an explicit *constraint propagation* method that uses the Euclidean CTD to densify must-link and cannot-link relationships within small volumes lying between constrained data pairs. We show that the modified weighted adjacency matrix thus obtained can be used to construct a *modified Laplacian*. The latter respects the topology of the initial graph but with a distinct geometric structure that have the presence of dense *graph lumps*, which is a direct consequence of the constraint propagation process: This makes it particularly well suited for clustering.

We introduce a shape segmentation method based on a learn, align, transfer, and classify paradigm. This introduces an important innovation, namely that one can perform the training on one data-set and then classify a completely different data-set on the premise that the two sets are approximately isomorphic. Our probabilistic label transfer algorithm is robust to topological noise as we consider dense soft correspondences between two shapes.

We compare our CSC algorithm with several other methods recently proposed in the literature, and we evaluate it against ground-truth segmentations of both graphical and visual

shapes. We note that the existing CSC methods have not been applied to articulated shapes, which are rather complex discrete Riemannian manifolds. Visual shapes gathered with scanners and cameras are very challenging data-set. As already mentioned, these manifold data are very difficult to cluster due to the regularity of the associated graph structure.

5.2 Shape Graph Embeddings

We have already introduced various properties of graph Laplacian in Section 2.6.4. In this section, first we recall some important theoretical results derived in Chapter 2. Second, we introduce the notion of modified Laplacian embedding Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, let edge weights $w_{ij} \in \mathbf{W}$ for two neighboring vertices in $\mathcal{V}$ be the Gaussian weights *i.e*, Eq. (3.7). Thus, without loss of generality, we can assume that $0 < w_{\min} \leq w_{ij} \leq w_{\max} \leq 1$, since the corresponding two vertices $v_i, v_j \in \mathcal{V}$ are typically close neighbors on shape graphs representing a discrete Riemannian surface.

The L-embedding: The commute-time embedding representation introduced in Section 2.6.5.2 allow us to isometrically embed the graph in a K-dimensional Euclidean feature space spanned by the first non-null eigenvectors of the graph Laplacian matrix $\mathbf{L}$. Let a shape graph $\mathcal{G}$ has a commute-time embedding represented as:

$$\mathbf{X}_{K\times n} = \mathbf{\Lambda}_K^{-1/2}(\mathbf{U}_{n\times K})^\top = [\mathbf{x}_1 \dots \mathbf{x}_j \dots \mathbf{x}_n]. \tag{5.1}$$

The Euclidean embedding $\mathbf{X}$ is named after the associated commute-time metric introduced in Section 2.6.5.1, which allows to compute the average connectivity between two graph vertices $v_i, v_j \in \mathcal{V}$ by simply computing the Euclidean distance between their images $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{X}$.

From the orthonormality of the eigenvectors and from Eq. (2.86), we obtain following bound on eigenvector elements:

$$-\lambda_i^{-1/2} < \mathbf{u}_i(v_j) < \lambda_i^{-1/2},\ \forall j, 1 \leq j \leq n \tag{5.2}$$

Using Eq. (5.2) and Eq. (5.1), we can easily derive bounds on embedding coordinates. The lower and upper bounds for the i-th coordinate of $\mathbf{x}_j$ are (reproduced from Eq. (2.93):

$$-\lambda_i^{-1/2} < x_{ji} < \lambda_i^{-1/2}. \tag{5.3}$$

In this chapter, we call the commute-time embedding $\mathbf{X}$ as $\mathbf{L}$-embedding.

The $\widehat{\mathbf{L}}$-embedding: So far we have described the properties of the spectral embeddings, which correspond to the graphs that contain only unlabeled vertices. As it will be explained in the next section, the presence of pairwise constraints could lead to a modified Laplacian

embedding and in this paragraph we describe the rationale of this modified spectral representation. We suppose that pairwise constraints are provided and we consider one such vertex-pair. Two situations can occur: (i) the two vertices are adjacent or, more generally, (ii) the two vertices are connected by one or several graph paths. While the former situation leads to simply modifying the edge weights of the corresponding pairs, the latter is more problematic to implement because it involves some form of constraint propagation and it constitutes the topic of Section 5.3. To summarize, the presence of constraints leads to modifying some of the edge weights in the graph. We denote the *modified adjacency matrix* with $\widehat{\mathbf{W}}$. We also obtain a modified degree matrix $\widehat{\mathbf{D}}$ and a modified unnormalized Laplacian $\widehat{\mathbf{L}}$:

$$\widehat{\mathbf{L}} = \widehat{\mathbf{D}} - \widehat{\mathbf{W}} \tag{5.4}$$

This leads to modified Euclidean coordinates:

$$\widehat{\mathbf{X}} = \widehat{\Lambda}^{-1/2}\widehat{\mathbf{U}}^{\top} = [\widehat{\mathbf{x}}_1 \dots \widehat{\mathbf{x}}_j \dots \widehat{\mathbf{x}}_n] \tag{5.5}$$

The initial graph can therefore be represented with two different embeddings, the exact geometry of the embedded space depending on the edge weights. Notice, that there is a one-to-one correspondence between the columns of $\mathbf{X}$ and of $\widehat{\mathbf{X}}$.

5.3 Propagating Pairwise Constraints

In a constrained clustering task instance-level constraints are available. In practice, it is convenient to be able to cope with a sparse set of constraints. The counterpart is that they are not easily exploitable: propagating these constraints over a manifold (or more generally over a graph) is problematic. In this section, we describe a constraint propagation method that uses the $\mathbf{L}$-embedding and the associated Euclidean *commute-time distance* (CTD). As already mentioned, must-link and cannot-link constraints were successfully incorporated in several variant of the K-means algorithm [Wagstaff 2001, Xing 2002, Bilenko 2004]. However, these methods did not incorporate constraint propagation. Rather than modifying the K-means step of spectral clustering, we incorporate a constraint-propagation process directly into the $\mathbf{L}$-embedding, thus fully exploiting the properties outlined in the previous section.

Consider a subset of the set of graph vertices $\mathcal{S} = \{\bar{v}_i\}, \mathcal{S} \subset \mathcal{V}$ from which we build two sets of constraints: A must-set $\mathcal{M} \subset \mathcal{S} \times \mathcal{S}$ and a cannot-set $\mathcal{C} \subset \mathcal{S} \times \mathcal{S}$. Vertex pairs from the must-set should be assigned to the same cluster while vertex pairs from the cannot-set should be assigned to different clusters. Notice that the cardinality of these sets is independent of the final number of clusters. Also, it is necessary neither to provide must links for all the clusters, nor to provide cannot links across all cluster pairs. A straightforward strategy for enforcing these constraints consists in modifying the weights w_{ij} associated with *adjacent* vertex-pairs that belong either to $\mathcal{M}$ or to $\mathcal{C}$ such that w_{ij} is replaced with $\tilde{w}_{ij} = 1$ if $(\bar{v}_i, \bar{v}_j) \in \mathcal{M}$ and $\tilde{w}_{ij} = \varepsilon$ if $(\bar{v}_i, \bar{v}_j) \in \mathcal{C}$, where ε is a small positive number. We recall that $0 < w_{\min} \leq w_{ij} \leq w_{\max} \leq 1$. Notice that for graphs corresponding to regular meshes, the edge-weight variability is small.

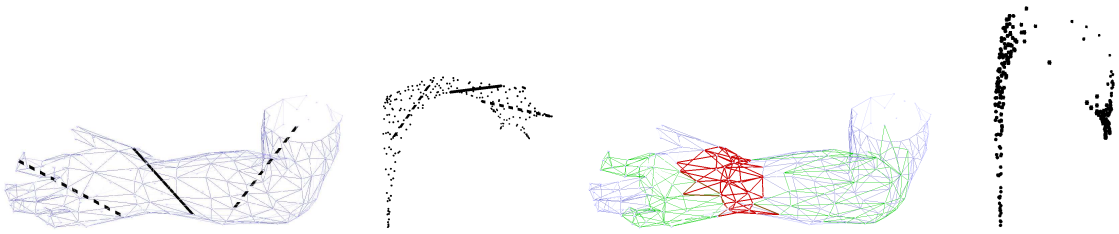


Figure 5.2: Propagating constraints. (a): Constraint placement onto the initial graph, two must-links (dashed lines) and one cannot-link; (b): The $\mathbf{L}$-embedding used for constraint propagation. (c): The propagated constraints are shown on the graph. (d): The new embedding obtained with the modified Laplacian $\widehat{\mathbf{L}}$.

Since the set $\mathcal{S}$ is composed of sparsely distributed vertices, the pairs $(\bar{v}_i, \bar{v}_j)$ do not necessarily correspond to adjacent vertices. Hence, one has to propagate the initial must-link and cannot-link constraints to nearby vertex pairs. We propose to use the commute-time distance (CTD) already mentioned. The CTD introduced in 2.6.5.1 is a well known quantity in Markov chains [Grinstead 1998]. For undirected graphs, it corresponds to the average number of (weighted) edges that it takes, starting at vertex v_i, to randomly reach vertex v_j for the first time and go back. The CTD has the interesting property that it decreases as the number of paths connecting the two nodes increases and when the lengths of the paths decrease. We prefer the CTD to the shortest-path geodesic distance in the graph because it captures the connectivity structure of a small graph volume rather than a single path between two vertices. The CTD is the integral of the diffusion distances over all times. Hence, unlike the latter, the former does not need the free parameter t to be specified [Coifman 2006, Qiu 2007, Bronstein 2010b]. Indeed, the scale parameter introduces an additional difficulty because different vertex-pairs may need to be processed at different scales.

The commute-time distance (Section 2.6.5.1) between two vertices is an Euclidean metric and it can be written in closed form using the $\mathbf{L}$-embedding, *i.e.*, Eq. (5.1):

$$d^2_{\mathrm{CTD}}(v_i, v_j) = \|\mathbf{x}_i - \mathbf{x}_j\|^2 \tag{5.6}$$

The CTD will allow us to propagate must-link and cannot-link constraints within small graph *volumes*, *e.g.*, Figure 5.2.

We briefly describe the **propagation of must-link constraints.** For each pair $(\bar{v}_i, \bar{v}_j) \in \mathcal{M}$ with embedded coordinates $\bar{\mathbf{x}}_i$ and $\bar{\mathbf{x}}_j$: We consider the hypersphere centered at $(\bar{\mathbf{x}}_i + \bar{\mathbf{x}}_j)/2$ with diameter given by (5.6) and we build a subset $\mathbf{X}_s \subset \mathbf{X}$ that contains embedded vertices lying in this hypersphere. We build a subgraph $\mathcal{G}_s \subset \mathcal{G}$ having as vertices the set $\mathcal{V}_s = \{v_i\}_{i=1}^r$ corresponding to $\mathbf{X}_s$. Finally, we modify the weights w_{ij} of the edges of $\mathcal{G}_s$: $\tilde{w}_{ij} = 1$. There is an equivalent procedure for the **propagation of cannot-link constraints**. In order to preserve the topology of the modified graph, in this case the weights are set to a small positive number,

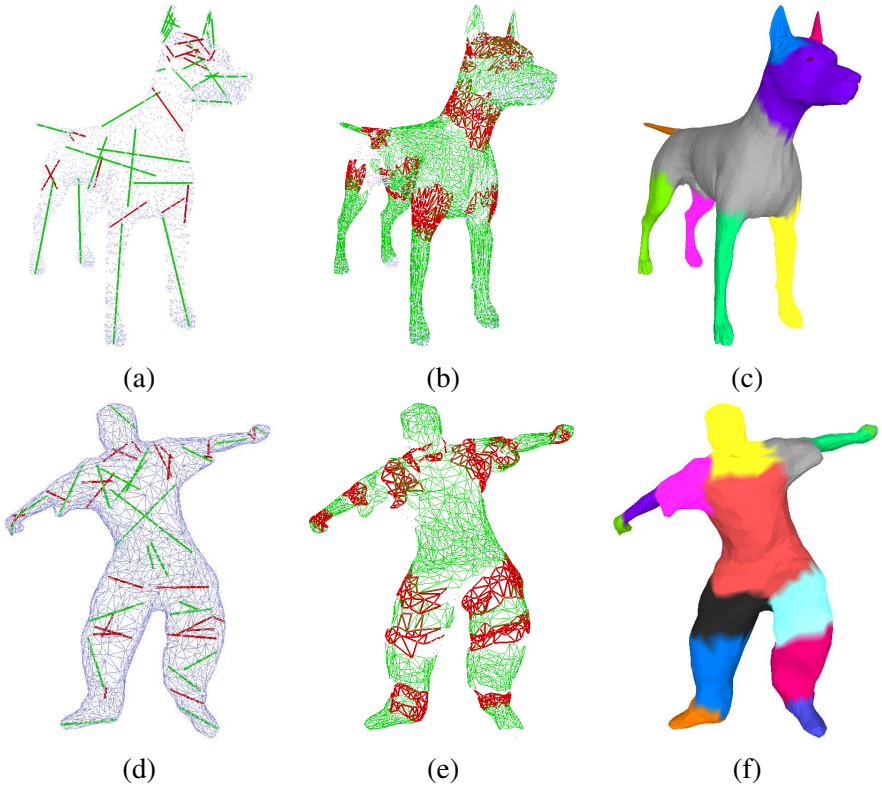


Figure 5.3: The CSC algorithm applied to the the *dog* and to the *flashkick* data (Note: unlike the results in Table 5.1, we seek here for *flashkick* 14 segments). Initial graphs and manually placed constraints (a), (d); Constraint propagation (b), (e); Final clustering results (c), (f).

i.e., the modified weight of a cannot-edge is $\tilde{w}_{ij} = \varepsilon$. Hence the proposed CSC Algorithm 4 and it's illustration in Figure 5.3.

Algorithm 4 *Constrained Spectral Clustering* (CSC)

Input: : Unnormalized Laplacian $\mathbf{L}$ of a shape graph $\mathcal{G}$, a must-link set $\mathcal{M}$, a cannot-link set $\mathcal{C}$, the number of cluster k to construct.

Output: : A set of binary variables $\Delta = \{\delta_{il}\}$ assigning a cluster label l to each graph vertex v_i.

1: Compute the $\mathbf{L}$-embedding of the graph using the p first non-null eigenvalues and eigenvectors of $\mathbf{L}$, $p \geq k$.
2: Propagate the $\mathcal{M}$ and $\mathcal{C}$ constraints, modify the adjacency matrix of $\mathcal{G}$ and build the modified Laplacian $\widehat{\mathbf{L}}$ using Eq. (5.4).
3: Compute the $\widehat{\mathbf{L}}$-embedding using the k first non-null eigenvalues and eigenvectors of $\widehat{\mathbf{L}}$.
4: Assign a cluster label l to each graph vertex v_i by applying K-means to the points $\widehat{\mathbf{X}}$ in Eq. (5.5).

5.4 Shape Segmentation via Probabilistic Label Transfer

The CSC algorithm that we just described is applied to a shape graph $\mathcal{G}_{\text{tr}}$ such that the latter is segmented into k clusters. Given a second shape $\mathcal{G}_{\text{test}}$ we wish to use the segmentation result obtained with $\mathcal{G}_{\text{tr}}$ to segment $\mathcal{G}_{\text{test}}$. Therefore, the segmentation of $\mathcal{G}_{\text{test}}$ can be viewed as an inference problem, where we seek a cluster label for each one of its vertices conditioned by the segmentation of $\mathcal{G}_{\text{tr}}$.

We formulate this label inference problem in a probabilistic framework and adopt a generative approach where we model the conditional probability of assigning a label to a test shape vertex. More formally, let $\mathbf{X}^{\text{tr}}$ and $\mathbf{X}^{\text{test}}$ be the $\mathbf{L}$-embeddings of the two shapes with n and m vertices respectively, *i.e.*, Eq. (5.1). We introduce three sets of hidden variables: $S = \{s_1, \ldots, s_m\}$, which assign each *test-shape* vertex to its cluster, $R = \{r_1, \ldots, r_n\}$, which assign each *train-shape* vertex to its cluster, and $Z = \{z_1, \ldots, z_m\}$, which assign a test-shape vertex to a train-shape vertex. Then the posterior probability of assigning a cluster label $l \in \{1, \ldots, k\}$ to a test-shape vertex $\mathbf{x}_i^{\text{test}} \in \mathbf{X}^{\text{test}}$ can be written as :

$$P(s_i = l|\mathbf{x}_i^{\text{test}}) = \sum_{j=1}^{n} P(r_j = l|\mathbf{x}_j^{\text{tr}}) P(z_i = j|\mathbf{x}_i^{\text{test}}), \tag{5.7}$$

Here, $P(r_j = l|\mathbf{x}_j^{\text{tr}})$ is the posterior probability of assigning a label l to a train-shape vertex $\mathbf{x}_j^{\text{tr}}$, conditioned by the train-shape vertex. Similarly, $P(z_i = j|\mathbf{x}_i^{\text{test}})$ is the posterior probability of assigning train-shape vertex $\mathbf{x}_j^{\text{tr}}$ to test-shape vertex $\mathbf{x}_i^{\text{test}}$ and can be termed as *soft assignment*. We propose to replace the posteriors $P(r_j = l|\mathbf{x}_j^{\text{tr}})$ with hard assignments, namely the output of the CSC algorithm:

$$P(r_j = l|\mathbf{x}_j^{\text{test}}) = \delta_{jl} \tag{5.8}$$

The estimation of the posteriors $P(z_i = j|\mathbf{x}_i^{\text{test}})$ is an instance of graph matching in the spectral domain, which is a difficult problem in its own right, especially in the presence of switches between eigenvectors and changes in their sign. The graph/shape matching task is further complicated when the two graphs are not isomorphic and when they have different numbers of vertices.

We adopted the articulated shape matching method proposed in [Mateus 2008, Horaud 2011] to obtain these *soft assignments*. This method proceeds in two steps. The first step uses the histograms of the k first non-null eigenvectors of the *normalized* Laplacian matrix to find an alignment between the Euclidean embeddings of two shapes. The second step registers the two embeddings using the expectation-maximization (EM) algorithm and selects the best vertex-to-vertex assignment based on the maximum a posteriori probability (MAP) of a vertex from one shape to be assigned to a vertex from the other shape. In order to fit to our methodological framework, we introduce two important modifications to the technique described in [Mateus 2008]:

1. We use the unnormalized Laplacian. This is justified by the properties of the **L**-embeddings, which where described in detail in Section 5.2. In particular, the property in Eq. (5.2) facilitates the task of comparing the histograms of two eigenvectors.

2. We do not attempt to find the best one-to-one assignments based on the MAP criterion. Instead, we keep all the assignments and hence we rely on *soft* rather than *hard* assignments.

The resulting shape matching algorithm will output the desired posterior probabilities $P(z_i = j|\mathbf{x}_i^{\text{test}}) = p_{ij}$, $\forall 1 \leq i \leq m$. From (5.7) and (5.8) we obtain the following expression that probabilistically assigns a vertex of $\mathcal{G}_{\text{test}}$ to a cluster of $\mathcal{G}_{\text{tr}}$:

$$\gamma_{il} = \arg\max_{1 \leq l \leq k} \sum_{j=1}^{n} p_{ij} \delta_{jl} \tag{5.9}$$

This corresponds to the maximum posterior probability of a test-shape vertex to be assigned to a train-shape cluster conditioned by the test-shape vertex and by the train-shape-to-test-shape soft assignments of vertices. The proposed segmentation method is summarized in Algorithm 5. Figure 5.4 illustrates the PLT method on two examples.

5.5 Experiments & Results

We evaluated the performance of our approach on 3D meshes, consisting of both graphical shapes [Bronstein 2010a] as well as visual shapes [Franco 2009, Starck 2007b, Stoll 2010, Vlasic 2008] having a wide range of variability in terms of mesh topology, kinematic poses, noise and scale. Particularly, the data acquired by multi-camera systems are non-uniformly sampled and there are major topological changes in between the various kinematic poses, *e.g.*,

Algorithm 5 *Probabilistic Label Transfer* (PLT)

Input: : $\mathbf{L}$-embeddings $\mathbf{X}^{\text{tr}}$ and $\mathbf{X}^{\text{test}}$ of train and test shape graphs $\mathcal{G}_{\text{tr}}$ and $\mathcal{G}_{\text{test}}$; a set of binary variables $\Delta = \{\delta_{jl}\}$ assigning a cluster label l to each vertex $\mathbf{x}_j^{\text{tr}} \in \mathbf{X}^{\text{tr}}$.

Output: : A set of binary variables $\Gamma = \{\gamma_{il}\}$ assigning a cluster label l to each vertex $\mathbf{x}_i^{\text{test}} \in \mathbf{X}^{\text{test}}$.

1: Align two $\mathbf{L}$-embeddings $\mathbf{X}^{\text{tr}}$ and $\mathbf{X}^{\text{test}}$ using the histogram alignment method [Mateus 2008].

2: Compute the posterior probability p_{ij} of assigning each test graph vertex $\mathbf{x}_i^{\text{test}}$ to every train graph vertex $\mathbf{x}_j^{\text{tr}}$ using the EM based rigid point registration method proposed in [Horaud 2011].

3: Find the cluster label l for each test graph vertex $\mathbf{x}_i^{\text{test}}$ using the Eq. (5.9).

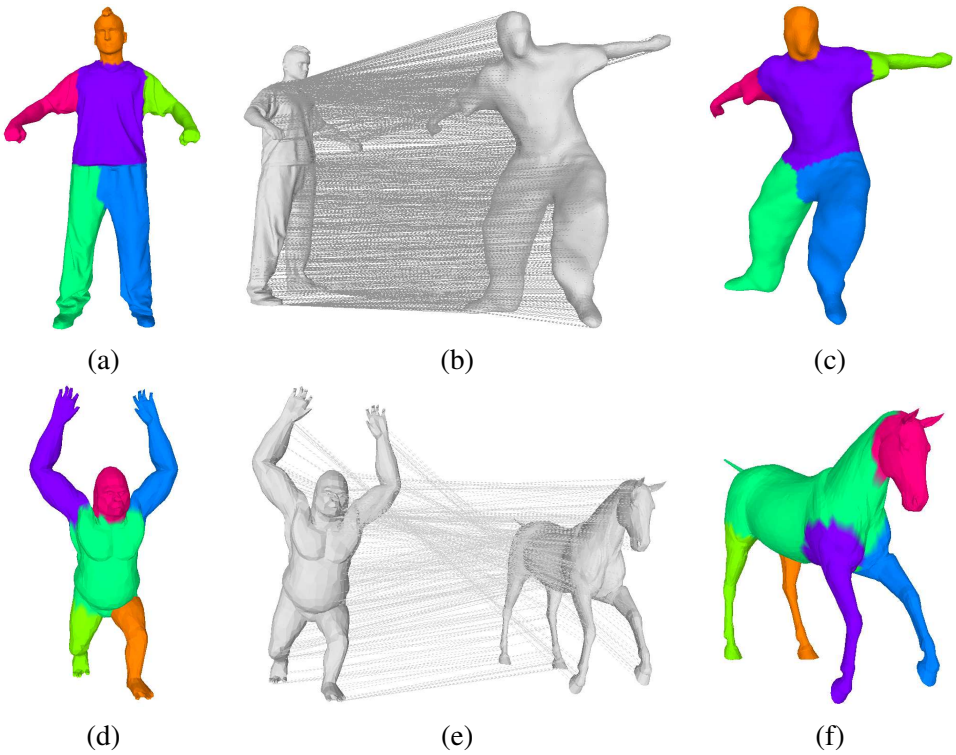


Figure 5.4: Clustering obtained with CSC (a), (d); vertex-to-vertex probabilistic assignment between two shapes (b), (e); The result of segmenting the second shape based on label transfer (c), (f).

					CSC		CCSKL [Li 2009]		SL [Kamvar 2003]		SC	
	$\|\mathcal{V}\|$	k	$\|\mathcal{M}\|$	$\|C\|$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$
dog	3400	9	28	19	0.8876	0.9243	0.5215	0.6239	0.4342	0.5644	0.5879	0.6825
crane	10002	6	9	8	0.9520	0.9761	0.6401	0.7952	0.8673	0.7905	0.7818	0.8526
handstand	10002	6	7	5	0.9659	0.9586	0.6246	0.7691	0.6475	0.7246	0.7584	0.9248
flashkick	1501	6	18	5	0.9279	0.9629	0.5898	0.7539	0.5412	0.5984	0.6207	0.7376
ben	16982	6	7	5	0.9054	0.9563	0.4002	0.5888	0.6434	0.6084	0.5587	0.6494

Table 5.1: Quantitative evaluation of visual shape segmentation using semi-supervised constrained spectral clustering algorithms.

Figure 5.1(c). We have generated manual segmentations of all the employed meshes as a ground truth for the quantitative evaluation of our approach. As a consequence of this, one-to-one correspondences between ground-truth and our results are available. Therefore, the standard statistical error measures like the true positives e_i^{tp}, the false negatives e_i^{fn} and the false positives e_i^{fp} can be easily computed for each segmentation and for each cluster i. From these measures we derive the true positive rate m_i^{tpr} (recall) and positive predictive value m_i^{ppv} (precision) for every cluster: m_i^{tpr} gives for each cluster i the percentage of vertices, which have been correctly identified from the ground truth, and m_i^{tpr} gives for each identified cluster the percentage of vertices, which actually truly belong to this cluster. Using these two measures, we tabulate the overall performance of our segmentation results by computing the mean over all clusters of each shape mesh. We can define recall and precision as:

$$\bar{m}^{tpr} = \sum_{i=1}^{k} \frac{e_i^{tp}}{e_i^{tp} + e_i^{fn}}, \quad \bar{m}^{ppv} = \sum_{i=1}^{k} \frac{e_i^{tp}}{e_i^{tp} + e_i^{fp}}$$

with k being the total number of clusters on the evaluated mesh. To maintain the independence of the ground truth from the test data, the manual segmentation and constraint placement for the tested algorithms were performed by different persons. We performed two sets of experiments. First, we evaluate the segmentation performance of the CSC algorithm described in Section 5.3 against two other constrained spectral clustering algorithms; Second, we evaluate the probabilistic label-transfer method described in Section 5.4.

We compared our CSC algorithm with the *constrained clustering by spectral kernel learning* (CCSKL) method [Li 2009], and with the *spectral learning* (SL) method [Kamvar 2003]. For completeness we also provide a comparison with the spectral clustering algorithm (SC) based on the random-walk graph Laplacian. Our implementations of these methods were duly checked with their respective cited results. With all these constrained spectral clustering methods the same set of constraints was used as well as the same number of clusters (the latter varies from one data set to another). The *normalized* SC algorithm that we implemented corresponds to the second algorithm in [von Luxburg 2007]: it applies K-means to the unnormalized Laplacian embedding, *i.e.*, Eq. (5.1) and it corresponds to steps 3 and 4 of our own CSC algorithm. A summary of these results can be found in Table 5.1 and Figure 5.5. The most surprising result is that, except for the "Crane" data and with SL, both CCSKL and SL could not significantly improve over the unsupervised SC algorithm, despite the side-information available

(a) GT (b) CSC (c) CCSKL (d) SL (e) SC

Figure 5.5: Manual segmentation (a), results obtained with our algorithm (b) and results obtained with three other methods: [Li 2009](c), [Kamvar 2003](d) and [von Luxburg 2007](e).

Results for several meshes (I)						Results for corrupted horse meshes (II)				
$\mathcal{G}_{tr}$	$\mathcal{G}_{test}$	$\lvert\mathcal{V}_{tr}\rvert$	$\lvert\mathcal{V}_{test}\rvert$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	transform	$\lvert\mathcal{V}_{tr}\rvert$	$\lvert\mathcal{V}_{test}\rvert$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$
ben	*handstand*	16982	10002	0.9207	0.9594	*topology*	19248	19248	0.9668	0.9642
handstand	*ben*	10002	16982	0.9672	0.9462	*sampling*	19248	8181	0.8086	0.9286
flashkick 50	*flashkick* 89	1501	1501	0.8991	0.9248	*noise*	19248	19248	1.0	1.0
gorilla	*horse*	2038	3400	0.8212	0.8525	*holes*	19248	21513	0.9644	0.9896

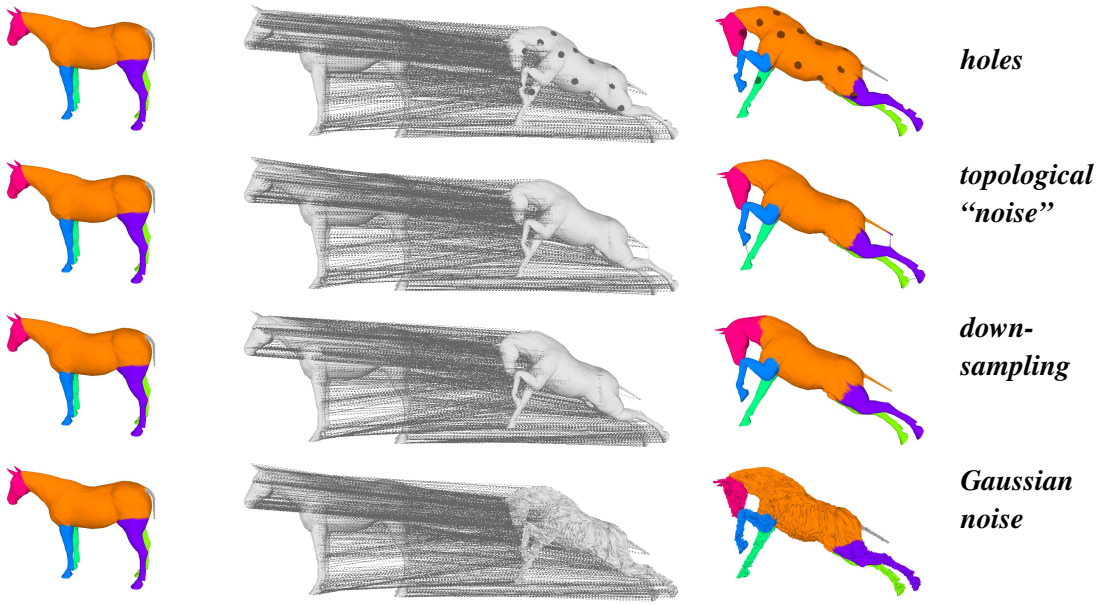
Table 5.2: Summary of the quantitative evaluation of the PLT algorithm.

to guide the segmentation. The CCSKL algorithm fails to improve over SC with our mesh data. Indeed, both assume that there are natural partitions (subgraph) in the data that are only weakly inter connected. Therefore, CCSKL only globally stretches each eigenvector in the embedded space to satisfy the constraints, without any local effect of these constraints on the segmentation. The SL algorithm can barely improve over the SC results as it requires a large number of constraints. With our method the placement of the *cannot-link* constraints is crucial. Although our method needs only a sparse set of constraints, the number of constraints increases (still number of constraints $\ll |\mathcal{V}|$) if the desired segmentation is not consistent with the graph topology, *e.g.*, Figure 5.3(d).

In the second experiment, we evaluate the performance of our probabilistic label transfer (PLT) method. In all these examples, we consider two different shapes, one from the training set and one from the test set. First we apply the CSC algorithm to the train-shape and then we apply the PLT algorithm to the test-shape. Figure 5.1 shows an example of PLT between two different shapes and in the presence of significant topological changes: the right arm of Ben, (e), touches the torso. Figure 5.4 shows the additional results, that are quantified in Table 5.2 (I). We also evaluate the robustness of PLT algorithm with respect to the various mesh corruptive transformations, such as holes, topological noise, etc. Figure 5.6 and Table 5.2 (II) shows the segmentation results obtained by transferring labels from the original horse mesh to its corrupted instances. We obtain zero error if the corruptive transformation does not change the triangulation of the mesh as in the case of Gaussian noise. In Figure 5.7, we show the segmentation obtained with PLT where the test shape in Figure 5.7-(c) significantly differs from the training shape in Figure 5.7-(a) due to large topological change (see the left hand merged with the torso).

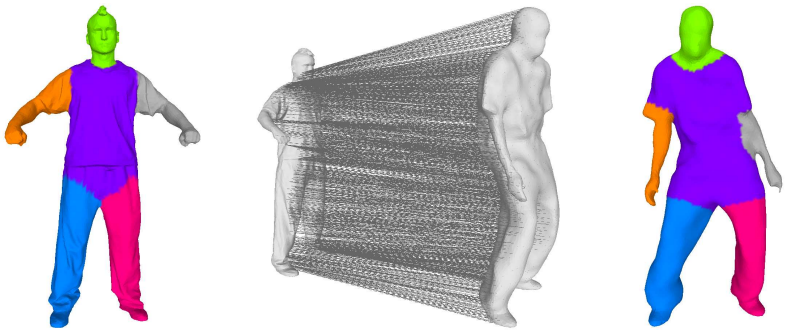
5.6 Conclusion

In this chapter, we proposed a novel framework for learning shape segmentation. We made two contributions: (1) we proposed to use the unnormalized Laplacian embedding and the commute-time distance to diffuse sparse pairwise constraints over a graph and to design a new constrained spectral clustering algorithm, and (2) we proposed a probabilistic label transfer algorithm to segment an unknown test-shape by assigning labels between an already segmented train-shape and a test-shape. We perform extensive testing of both the CSC and the PLT al-



(a) Ground-truth (b) Label transfer (c) Segmentation

Figure 5.6: Segmentation results with synthetic meshes, which have been corrupted in various ways.



(a) (b) (c)

Figure 5.7: Clustering obtained with CSC (a); vertex-to-vertex probabilistic assignment between two shapes (b); The result of segmenting the second shape based on label transfer (c).

gorithms on real and synthetic meshes. We compare our shape segmentation method with recent constrained/semi-supervised spectral clustering methods, which were known to outperform unsupervised SC algorithms. However, we found it difficult to adapt these existing constrained clustering methods to the problem of shape segmentation. This is due to the fact that, unlike the proposed method, they do not explicitly take into account the properties inherently associated with shape graphs, such as sparsity and regular connectivity.

One major limitation of the proposed method is that it heavily rely on dense probabilistic shape matching for label transfer. However, variability in visual shapes is large enough to rely on any existing shape matching method, *e.g.*, the majority of existing methods cannot handle commonly occurring topological changes in the visual shapes. In the following chapters, we develop a multi-scale heat-kernel framework to describe and reliably match visual shapes with large topological changes.

CHAPTER 6

Multi-scale 3D Shape Representation using Heat Diffusion Framework

Contents

6.1 Introduction

In Chapter 3, we have already introduced an isometry invariant spectral representation of 3D visual shapes using the eigen-decomposition of graph Laplacian matrices. However, this is a single scale representation where mainly the global structure of the shape is highlighted. This

behavior is understandable given the analogy with PCA 2.6.5 and Fourier Analysis [Lévy 2006, Vallet 2008]. However, analyzing a shape a multiple scales might help to simultaneously study it's local and global structure, *e.g.*, characterizing a human shape with it's (local) facial geometrical features can be useful along with the global limb structure. A major limitation of single scale representation is that such a global representation fails to handle the topological changes and partial shapes as they focus on global structure (see Section 3.4). The use of a multi-scale analysis can help to overcome this problem by locally analyzing a visual shape.

In this chapter, we focus on multi-scale shape representation and outline a general framework for the representation and analysis of 3D visual shapes based on heat diffusion on undirected weighted graphs. It is well known that the heat diffusion equation has a solution on undirected graphs and that this solution can be made explicit using the eigenvalue/eigenvector pairs of graph Laplacians, together with a *time parameter* that defines a scale space – the *heat kernel*. This will allow us to analyze the heat-kernel matrix within the framework of spectral graph theory [Chung 1997], to construct heat-kernel matrices well suited for 3D shapes, and to represent the latter into the metric space associated with the spectral decomposition of this matrix [Shawe-Taylor 2004]. We will capitalize on the fact that the eigenvectors of the combinatorial Laplacian can be interpreted as the directions of maximum variance of the shape embedding (see Section 2.5.1.1). Together with the scale/time parameter, this will provide a formal basis for performing dimensionality reduction and, more generally, to characterize the statistical properties of the embedded shape representation at multiple scales. We will study the dimensionality of the embedding, *i.e.*, the number of eigenvectors needed to approximate the heat kernel, as a function of the scale parameter; We will show that the multiplicity of the first non-null eigenvalue and associated eigenvector (the Fiedler vector) of the Laplacian matrix plays a crucial role in choosing the dimension. Finally, we propose both a scale-space representation of shapes based on *auto diffusion* and on *spectral distances*.

The organization of this chapter is as follows. First, we outline the heat-kernel framework for multi-scale shape representation. We start with a detailed literature survey in Section 6.1.1. In Section 6.1.2 we list the main contributions of this chapter. Section 6.2 introduce details of heat diffusion on manifold surface and it's discrete counterpart for shape graphs. In Section 6.3, we discuss spectral properties of heat-kernel matrices and devise a scale dependent feature space (embedding) representation for shape graphs. In the next Section 6.4, we discuss the analogy of eigenvectors heat-kernel matrices with PCA and propose a novel analysis relating scale parameter and dimensionality of heat-kernel embedding. Section 6.5 introduce various normalization schemes for heat-kernel embedding. Second, we present a dense descriptor based shape representation in Section 6.6, where we propose a new multi-scale heat distance descriptor for each vertex of the shape graph. Finally, we summarize our conclusions and future directions of work in Section 6.7.

6.1.1 Related work

Multi-scale representation and analysis have long been considered among the most powerful and successful tools for describing images [Burt 1983, Crowley 1984, Hummel 1987, Lindeberg 1994]. Scale-space analysis of scalar functions defined over Euclidean domains, *e.g.*, images, is generally performed based on convolutions with a Gaussian function, the scale parameter being associated with the variance of this function. The rationale of this definition of scale is that the Gaussian function corresponds to the closed-form solution of the heat-diffusion equation on Euclidean domains such as images [Koenderink 1984]. Therefore, one may think of scale-space representations of 3D shapes as an extension of this mathematical framework. This implies the understanding of heat diffusion on non Euclidean continuous domains, such as Riemannian manifolds, or discrete domains, such as graphs.

Multiple-scale shape properties based on heat diffusion are studied in [Sun 2009]: The *heat-kernel signature* (HKS) is studied in detail for this purpose. The HKS at a mesh vertex v_i is simply the i-th diagonal entry of the heat-kernel matrix. The mesh Laplace operator proposed in [Belkin 2008] is used in practice to build a shape descriptor based on the HKS. Independently, this has also been referred to as the *auto-diffusion* function (ADF) [Gebal 2009]: A link between auto-diffusion at small scales and the Gaussian curvature is established, as well as a practical method for finding the skeleton of a shape. Subsequently, the HKS/ADF was used as the basis of a feature-based method for shape retrieval [Bronstein 2011a] allowing to extend the bag-of-features paradigm to 3D shapes and to devise an efficient shape retrieval method.

One of the most attractive features of the heat kernel is that it allows the embedding of a shape into the metric space defined by the spectral decomposition of the heat-kernel matrix. This enables the computation of *spectral distance* [Bronstein 2011b] between any two graph vertices, based on the vertices' spectral coordinates. The simplest distance between two graph vertices is the shortest-path (geodesic) distance, which is very sensitive to graph changes, *e.g.*, insertion and deletion of edges. In the case of visual shapes, the connectivity information is derived from the data, and therefore the geodesic distance is extremely sensitive to variations in the graph's connectivity. Alternatively, the spectral distance is more robust to noise and to local variations in the graph's topology because it averages over many paths [Qiu 2007]. Indeed, the spectral distance decreases as the number of paths connecting two vertices increases and when the average length of these paths decreases. Intuitively, the spectral distance based on the heat kernel captures heat diffusion at time t between two vertices.

Spectral distances were thoroughly studied in the recent past. Diffusion maps, introduced in [Coifman 2006], are based on even positive powers of the transition matrix of a first-order time-reversible Markov chain associated with a graph, *i.e.*, Eq. (2.3). Hence, the power parameter defines a scale-space in this case. The heat kernel belongs to the family of exponential kernels [Kondor 2002] that allow to define diffusion kernels [Kondor 2004]. There are several advantages of the exponential family of kernels over diffusion maps. Indeed, it is not clear how to choose the powers of the transition matrix and particular choices for this power may leave

some of the graph's vertices unreachable. Spectral distances were studied in detail within the framework of shape recognition [Bronstein 2011b]. A 3D shape can be represented based on scale-dependent distance distributions, *e.g.*, histograms, [Mahmoudi 2009, Bronstein 2010d]. The similarity between two shapes is thus converted into the similarity between two point distributions.

Representing a 3D shape using a sparse or dense set of local and global feature descriptors is a natural extension of their 2D counterpart where images are represented using the sparse or dense set of local feature descriptors, *e.g.*, SIFT [Lowe 2004]. There is considerable amount of literature available on 3D shape descriptors, however, it has been largely focused on graphical shapes. The geometry processing community is still searching for a robust generic descriptor for visual shapes.

The existing local feature descriptor methods can largely be categorized as *extrinsic* or *intrinsic* based upon how they characterize the local geometry around a feature point. The extrinsic descriptors typically capture the local Euclidean geometry, *e.g.*, surface normal. Some of the popular extrinsic feature descriptors are [Johnson 1999, Körtgen 2003]. On the other hand, the intrinsic feature descriptors are more preferable for articulated 3D shapes as they capture (pose invariant) intrinsic geometry of the underlying 2D manifold surface. The general idea is to embed the 3D shape in an isometry invariant subspace and compute these descriptors. Thus, many intrinsic feature descriptors are derived by adapting the construction of their extrinsic counterpart, *e.g.*, [Wang 2010]. The heat kernel signature (HKS), introduced in the previous discussion, is an important intrinsic shape descriptor that has been used for 3D shape matching [Sun 2009, Sharma 2011] and shape retrieval tasks [Bronstein 2011a]. Recently, a new intrinsic feature is proposed in [Zaharescu 2012], which captures both the local geometry of the underlying manifold surface and the scale-space differential properties of the real-valued function defined over such surface.

Apart from the local descriptors, the global shape descriptors are more popular for shape retrieval task where a single descriptor can be used for complete shape description. ShapeDNA [Reuter 2006b] is one such intrinsic global shape descriptors for shape retrieval. It describes a 3D shape by the spectra of the Laplacian matrix of underlying graph. A recent extension to this method was proposed in [Wu 2011], where a new global descriptor is computed using the dual Laplacian decomposition for shape matching. In [Wu 2010], a global shape descriptor similar to shapeDNA, is augmented with a local descriptor, which analyze the spectra of the Laplacian matrix of a local patch (bin) around a feature point. A key problem with this method is that it requires a normalized re-sampling of the complete 3D data-set. A detailed survey of various 3D feature descriptors employed for 3D shape matching is presented in [van Kaick 2011].

6.1.2 Contributions

In this chapter, we propose a general framework for describing and analyzing 3D shapes based on heat-kernel embedding. We build on the ideas of representing a polygonal mesh as an

undirected weighted graph and on mapping such a graph onto the metric space associated with the spectral decomposition of the heat-kernel matrix.

Since the heat-kernel matrix is built using the eigenvalue/eigenvector pairs of a graph Laplacian operator, we recall some spectral properties of the graph Laplacians already discussed in detail in Section 2.6.4. The only constraint of this analysis is that the Laplacian is a positive semi-definite symmetric matrix, hence the edge weights must be non-negative. Hence, our framework equally applies to polygonal meshes with Gaussian weights or to triangulated meshes with cotangent weights as discussed in Section 3.3.3. We formally define two heat-kernel embeddings based on the combinatorial and normalized Laplacians. We already know that the eigenvectors of the combinatorial Laplacian are the principal components of a zero-centered distribution formed by the spectral embedding of a 3D shape (see Section 2.6.5). Therefore, 3D shapes lying in $\mathbb{R}^3$ can be represented as distributions of points in $\mathbb{R}^K$ conditioned by a scale parameter (t), where K is the number of principal components of the spectral shape representation. This naturally casts spectral shape analysis within the framework of kernel PCA. However, unlike the general case [Schölkopf 1998a], the heat-kernel and covariance matrices share exactly the same eigenvalue/eigenvector pairs. This provides a principled way to perform dimensionality reduction in spectral space and to choose the K principal vectors. We formally show that at small scales K must be large while at very large scales K corresponds to the multiplicity of the first non-null eigenvalue/eigenvector pair of the Laplacian, namely the Fiedler value/vector. We conclude that a scale-space shape representation must appropriately combine the scale t and the number of principal components K.

One consequence of the use of an exponential kernel, such as the heat kernel, is that the kernel collapses to zero extremely rapidly, when the parameter t goes to infinity. Hence, the distribution associated with the spectral shape representation rapidly collapses to zero as well. This makes the scale space analysis very sensitive to small scale changes. To prevent this, we build on existing approaches [Bérard 1994, Sun 2009] and we propose two normalized spectral representations: (i) a shape embedding mapped on the K-dimensional unit hyper-sphere and (ii) a embedding scaled by the trace of the heat-kernel matrix. We study the behavior of the associated distributions as a function of the scale parameter.

After formalizing the heat kernel framework for 3D shape representation, we discuss the choice of shape representation in the context of visual shape data. This is followed by the definition of a novel, densely computable, multi-scale intrinsic feature descriptor for 3D shapes.

6.2 Heat Diffusion on a 3D Shape

In this section, we will outline the mathematical formulation for heat diffusion on continuous Riemannian manifold surfaces and extend it to undirected weighed graphs, in order to derive the heat-kernel matrix for visual shapes represented by such graphs.

6.2.1 Heat Diffusion on a Closed Riemannian Manifold

Heat diffusion is a fundamental concept in physics. The heat diffusion equation (or more generally the diffusion equation) is a partial differential equation, which describes the distribution of heat (or the variation in temperature) in a given location and over time. Heat diffusion is generally studied in Euclidean spaces but it can be generalized to non-Euclidean ones. In particular, it can be defined on a closed, *i.e.*, compact and without boundary, Riemannian manifold $\mathcal{M}$. In this case, it takes the following analytical form:

$$\left(\frac{\partial}{\partial t} + \Delta_{\mathcal{M}}\right) f(x;t) = 0 \tag{6.1}$$

where $\Delta_{\mathcal{M}}$ denotes the *geometric* Laplace-Beltrami operator, $x \in \mathcal{M}$, $f : \mathcal{M} \to \mathbb{R}$, $f \in L^2(\mathcal{M})$, and $t > 0$. We will refer to $\partial/\partial t + \Delta_{\mathcal{M}}$ as the *heat operator*. More formally, the heat semi group is the family of self ad-joint operators [Bérard 1994]:

$$\left(e^{-t\Delta_{\mathcal{M}}} f\right)(x) = \int_{\mathcal{M}} h_{\mathcal{M}}(x,y;t) f(y) dy, \forall f \in L^2(\mathcal{M}) \tag{6.2}$$

where $h_{\mathcal{M}}$ is a smooth function of $x, y \in \mathcal{M}$ and of $t > 0$, namely it is the *heat kernel* of $\mathcal{M}$. For a closed manifold, the spectrum of the Laplace-Beltrami operator is a sequence of eigenvalues $0 = \lambda_1 < \lambda_2 \leq \lambda_3 \ldots + \infty$. Given an orthonormal basis of real eigenfunctions $\{\phi_i\}_{i=1}^{\infty} \in L^2(\mathcal{M})$ of the Laplacian, one can write:

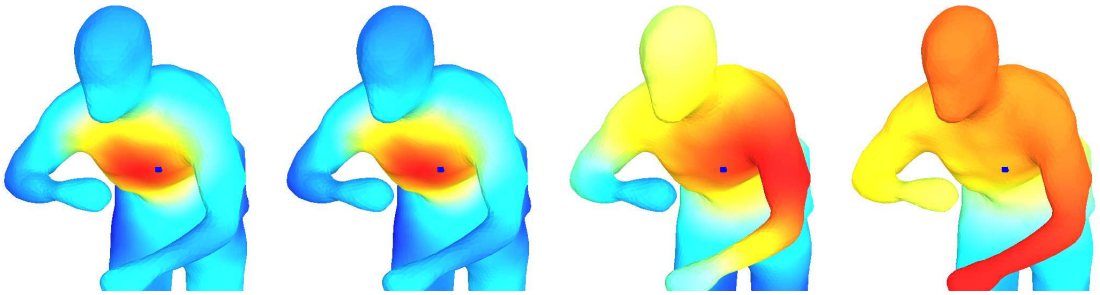
$$h_{\mathcal{M}}(x,y;t) = \sum_{i=1}^{\infty} e^{-\lambda_i t} \phi_i(x) \phi_i(y) \tag{6.3}$$

The operator $e^{-t\Delta_{\mathcal{M}}}$ in (6.2) is the fundamental solution of the heat-diffusion equation on Riemannian manifolds and (6.3) allows to compute the heat kernel using the eigenvalues and eigenvectors of the Laplace-Beltrami operator.

6.2.2 Heat Diffusion on an Undirected Weighted Graph

The definition of heat diffusion on graphs is exactly the parallel of heat diffusion on Riemannian manifolds. We start by recalling some basic graph notations and definitions from Chapter 2 along with properties of combinatorial and normalized graph Laplacian from Section 2.6.4 and appendix A.3.

We can recall from Section 2.2.2 that a connected *undirected weighted graph* is represented as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the vertex set, $\mathcal{E} = \{e_{ij}\}$ is the edge set and $\mathbf{W}$ be the weighted adjacency matrix of this graph, *i.e.*, $w_{ij} > 0$ whenever there is an edge e_{ij} between vertices v_i and v_j, $w_{ij} = 0$ elsewhere, with $w_{ii} = 0$. The degree d_i of a graph vertex is defined by $d_i = \sum_j w_{ij}$. Let $d_{\max} = \max_i(d_i)$. A *regular* graph is such that $d_1 = \ldots = d_i \ldots = d_n$.



(a) $t = 1$ (b) $t = 10$ (c) $t = 100$ (d) $t = 500$

Figure 6.1: Visualization of the heat diffusion on 3D shapes. The color at each vertex v_i encodes the value of the kernel $h(i, j; t)$, where v_j is the blue dot on the torso. The heat diffusion is fairly local for smaller values of time scale parameter t and becomes increasingly global with larger values of t.

The *unnormalized*, or *combinatorial* Laplacian matrix of a graph is defined by (see Section 2.6.1):

$$\mathbf{L} = \mathbf{D} - \mathbf{W} \tag{6.4}$$

Next we consider real-valued functions f over $\mathcal{V}$, $f : \mathcal{V} \mapsto \mathbb{R}$ and we note that $\mathbf{f} = (f_1 \dots f_n)^\top$ is simply a vector indexed by the vertices of $\mathcal{G}$. Using (6.4) one can easily obtain the action of $\mathbf{L}$ on such a vector of functions:

$$(\mathbf{Lf})_i = \sum_{j=1}^{n} w_{ij}(f_i - f_j) \tag{6.5}$$

The unnormalized graph Laplacian can therefore be viewed as a discretization of the Laplace-Beltrami operator (see Section 2.7).Moreover, let:

$$\mathbf{F}(t) = \mathbf{H}(t)\mathbf{f} \tag{6.6}$$

The graph's *heat operator* $\mathbf{H}(t)$, the *heat-kernel matrix* of the graph, is a discretization of (6.2):

$$\mathbf{H}(t) = e^{-t\mathbf{L}} \tag{6.7}$$

where the exponential of a matrix is defined by the following power series:

$$e^{\mathbf{A}} = \sum_{k=0}^{\infty} \frac{\mathbf{A}^k}{k!} \tag{6.8}$$

Notice that the vector $\mathbf{F}(t)$ in (6.6) is a solution to the heat diffusion equation on a graph:

$$\left(\frac{\partial}{\partial t} + \mathbf{L}\right)\mathbf{F}(t) = 0 \tag{6.9}$$

Hence, $\mathbf{f}$ corresponds to some initial heat distribution over the nodes of $\mathcal{G}$ and $\mathbf{F}(t)$ is the heat distribution at time t starting from $\mathbf{F}(0) = \mathbf{f}$. Notice that starting with a point heat distribution at vertex j, $\mathbf{g}_j = (0 \dots g_j = 1 \dots 0)^\top$, the heat distribution at time t is given by the j^{th} column of the heat-kernel matrix, which is denoted by $\mathbf{H}(\cdot, j; t)$:

$$\mathbf{F}(t) = \mathbf{H}(t)\mathbf{g}_j = \mathbf{H}(\cdot, j; t) \tag{6.10}$$

The graph Laplacian $\mathbf{L}$ is a semi-definite positive symmetric matrix since from (6.5) one can easily see that $\mathbf{f}^\top \mathbf{L}\mathbf{f} \geq 0$. Hence, the heat-kernel matrix is semi-definite positive symmetric as well. From (6.10) we obtain a straightforward interpretation of the entries of the heat-kernel matrix, namely each entry of $\mathbf{H}(t)$ corresponds to the amount of heat available at vertex v_i at time t, starting with a point heat distribution at vertex v_j, *i.e.*, $\mathbf{g}_j$:

$$h(i, j; t) = \mathbf{H}(i, j; t) \tag{6.11}$$

The *discrete symmetric function* $h : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$ is the heat kernel of a graph $\mathcal{G}$.

Figure 6.1 visualize heat diffusion at multiple scales on a visual shape represented by a shape graph. It is clearly visible that the heat diffusion is fairly local for smaller values of time scale parameter t and becomes increasingly global with larger values of t. More importantly, heat diffusion follows the topology of the shape graph, *e.g.*, amount of heat reaching left hand is significantly larger than right hand since the point heat source (blue dot) is placed relatively closer to left shoulder.

6.3 Heat-Kernel Matrices

We have already derived various properties of spectral-decomposition of combinatorial and normalized graph Laplacian matrices in Section 2.6.4 and Appendix A.3, respectively. From the spectral decompositions of combinatorial and normalized Laplacian matrices, we further obtain the spectral decomposition of the corresponding heat-kernel matrices:

$$\mathbf{H}(t) = \mathbf{U}e^{-t\mathbf{\Lambda}}\mathbf{U}^\top = \mathbf{U}\mathbf{\Phi}\mathbf{U}^\top \tag{6.12}$$

$$\tilde{\mathbf{H}}(t) = \mathbf{W}e^{-t\mathbf{\Gamma}}\mathbf{W}^\top = \mathbf{W}\mathbf{\Psi}\mathbf{W}^\top \tag{6.13}$$

where both

$$\mathbf{\Phi} = e^{-t\mathbf{\Lambda}} \tag{6.14}$$

$$\mathbf{\Psi} = e^{-t\mathbf{\Gamma}} \tag{6.15}$$

are diagonal matrices with entries:

$$\mathbf{\Phi} = \text{Diag}[\phi_1 \dots \phi_n] = \text{Diag}[e^{-t\lambda_1} \dots e^{-t\lambda_n}]$$

$$\mathbf{\Psi} = \text{Diag}[\psi_1 \dots \psi_n] = \text{Diag}[e^{-t\gamma_1} \dots e^{-t\gamma_n}]$$

Notice that from the properties of the eigenvalues of the Laplacians we obtain:

$$1 = \phi_1 > \phi_2 \geq \ldots \phi_n > 0 \tag{6.16}$$
$$1 = \psi_1 > \psi_2 \geq \ldots \psi_n > 0 \tag{6.17}$$

Let us apply to these matrices a *deflation* such that:

$$\mathbf{H}(t) \longleftarrow \quad \mathbf{H}(t) - \phi_1 \mathbf{u}_1 \mathbf{u}_1^\top \tag{6.18}$$
$$\tilde{\mathbf{H}}(t) \longleftarrow \quad \tilde{\mathbf{H}}(t) - \psi_1 \mathbf{w}_1 \mathbf{w}_1^\top \tag{6.19}$$

Hence $\mathbf{H}(t)\mathbf{u}_1 = \mathbf{0}, \tilde{\mathbf{H}}(t)\mathbf{w}_1 = \mathbf{0}$ and the remaining eigenvalues and eigenvectors remain unchanged. The heat-kernel matrices can then be written as:

$$\mathbf{H}(t) = \sum_{k=2}^{n} e^{-t\lambda_k} \mathbf{u}_k \mathbf{u}_k^\top \tag{6.20}$$
$$\tilde{\mathbf{H}}(t) = \sum_{k=2}^{n} e^{-t\gamma_k} \mathbf{w}_k \mathbf{w}_k^\top \tag{6.21}$$

Both $\mathbf{H}(t)$ and $\tilde{\mathbf{H}}(t)$ are *Gram* matrices: they are symmetric semi-definite positive matrices whose eigenvectors are the eigenvectors of the associated graph Laplacians. From (6.12), (6.13) and (6.28), (6.29) one obtains the following factorizations:

$$\mathbf{H} = \mathbf{U} e^{-t\mathbf{\Lambda}} \mathbf{U}^\top = \mathbf{X}^\top \mathbf{X} \tag{6.22}$$
$$\mathbf{H} = \mathbf{W} e^{-t\mathbf{\Gamma}} \mathbf{W}^\top = \mathbf{Y}^\top \mathbf{Y} \tag{6.23}$$

where $\mathbf{X}$ and $\mathbf{Y}$ are $n-1 \times n$ matrices whose columns are the vertex coordinates in feature-space:

$$\mathbf{X} = \left[\begin{array}{cccc} \mathbf{x}_1 & \ldots \mathbf{x}_i & \ldots & \mathbf{x}_n \end{array} \right] = e^{-\frac{t\mathbf{\Lambda}}{2}} \mathbf{U}^\top \tag{6.24}$$
$$\mathbf{Y} = \left[\begin{array}{cccc} \mathbf{y}_1 & \ldots \mathbf{y}_i & \ldots & \mathbf{y}_n \end{array} \right] = e^{-\frac{t\mathbf{\Gamma}}{2}} \mathbf{W}^\top \tag{6.25}$$

More precisely, the entries of the heat-kernel matrices are:

$$h(i,j;t) = \sum_{k=2}^{n} e^{-t\lambda_k} u_{ik} u_{jk} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \tag{6.26}$$
$$\tilde{h}(i,j;t) = \sum_{k=2}^{n} e^{-t\gamma_k} w_{ik} w_{jk} = \langle \mathbf{y}_i, \mathbf{y}_j \rangle \tag{6.27}$$

with:

$$\begin{aligned} \mathbf{x}_i &= \left(\begin{array}{cccc} x_{i2} & \ldots & x_{ik} & \ldots & x_{in} \end{array} \right)^\top \\ &= \left(\begin{array}{cccc} e^{-t\lambda_2/2} u_{i2} & \ldots & e^{-t\lambda_k/2} u_{ik} & \ldots & e^{-t\lambda_n/2} u_{in} \end{array} \right)^\top \end{aligned} \tag{6.28}$$
$$\mathbf{y}_i = \left(\begin{array}{cccc} e^{-t\gamma_2/2} w_{i2} & \ldots & e^{-t\gamma_k/2} w_{ik} & \ldots & e^{-t\gamma_n/2} w_{in} \end{array} \right)^\top \tag{6.29}$$

To summarize:

- Both $\mathbf{x}_i$ and $\mathbf{y}_i$ are elements of two distinct *feature spaces*, or *embeddings* of the 3D shape into $\mathbb{R}^{n-1}$.
- The heat kernels are *Mercer kernels* [Shawe-Taylor 2004] and they correspond to dot-products in feature space.
- The heat-kernel can be used to define distances and norms in feature space, namely the *heat distances*:

$$\begin{aligned} \|\mathbf{x}_i - \mathbf{x}_j\|^2 &= h(i,i;t) + h(j,j;t) - 2h(i,j;t) && (6.30) \\ \|\mathbf{y}_i - \mathbf{y}_j\|^2 &= \tilde{h}(i,i;t) + \tilde{h}(j,j;t) \\ &- 2\tilde{h}(i,j;t) && (6.31) \end{aligned}$$

  and the *heat norms*, *i.e.*, the auto-diffusion functions:

$$\begin{aligned} \|\mathbf{x}_i\|^2 &= h(i,i;t) && (6.32) \\ \|\mathbf{y}_i\|^2 &= \tilde{h}(i,i;t) && (6.33) \end{aligned}$$

- The points $\mathbf{x}_i$ and respectively $\mathbf{y}_i$ correspond to the coordinates of the graph's vertex v_i in the Euclidean spaces spanned by the non-null eigenvectors of the combinatorial Laplacian and of the normalized Laplacian.

To conclude this section, we provide the trace and determinant of the combinatorial heat-kernel matrix (there are equivalent expressions for the normalized heat-kernel matrix):

$$\begin{aligned} \mathrm{tr}(\mathbf{H}) &= \sum_{k=2}^{n} e^{-t\lambda_k} = \sum_{i=1}^{n} \|\mathbf{x}_i\|^2, && (6.34) \\ \det(\mathbf{H}) &= \prod_{k=2}^{n} e^{-t\lambda_k} = e^{-t\mathrm{tr}(\mathbf{L})}, && (6.35) \end{aligned}$$

and a visualization of the heat-kernel embedding in Figure 6.2.

6.4 Principal Component Analysis of the Heat-kernel Embeddings

The heat-kernel matrices are Gram matrices and this equivalence provides a straightforward tool for characterizing the feature space. In particular there is a duality between the kernel matrix and the feature-space covariance matrix. Let us consider both feature spaces and the corresponding graph embeddings, *i.e.*, $\mathbf{X}$ associated with $\mathbf{H}$ and $\mathbf{Y}$ associated with $\tilde{\mathbf{H}}$, namely equations (6.24) and (6.25).

We denote by $\bar{\mathbf{x}} = 1/n\mathbf{X}\mathbb{1}$ the center of mass of the feature-space points $\mathbf{X}$. From (6.24) and using Proposition 3 we get $\bar{\mathbf{x}} = 0$. We obtain the following result:

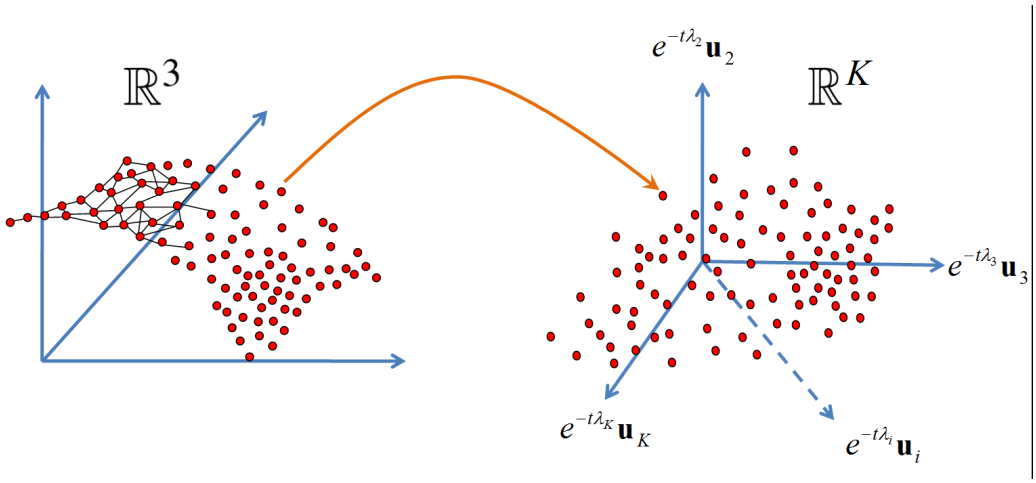


Figure 6.2: Heat-kernel embedding of a discrete surface represented as a sparse undirected weighed graph.

Proposition 4 *The eigenvectors of the combinatorial Laplacian are the directions of maximum variance of the corresponding heat-diffusion embedding. The covariance matrix is*

$$\mathbf{\Sigma}_X = \frac{1}{n} e^{-t\mathbf{\Lambda}} \tag{6.36}$$

Proof: Since the feature-space graph embedding has zero mean, the covariance writes $\mathbf{\Sigma}_X = \mathbf{X}\mathbf{X}^\top$. Using (6.24) and the orthonormal constraint associated with the eigenvectors, one can easily obtain (6.36).

Similarly let $\overline{\mathbf{y}} = 1/n\mathbf{Y}\mathbb{1}$ be the center of mass of the feature-space points $\mathbf{Y}$. Notice that $\overline{\mathbf{y}} = e^{-\frac{t\mathbf{\Gamma}}{2}}\overline{\mathbf{w}}$ where the entries of $\overline{\mathbf{w}} = (\overline{w}_1 \dots \overline{w}_k \dots \overline{w}_n)^\top$ are provided by Eq. (A.4). The covariance matrix of this un-centered data set is $\mathbf{\Sigma}_Y = \frac{1}{n}\mathbf{Y}\mathbf{Y}^\top - \overline{\mathbf{y}}\,\overline{\mathbf{y}}^\top$. Hence, one obtains the following straightforward result:

Proposition 5 *The covariance matrix of the normalized heat-diffusion embedding is given by*

$$\mathbf{\Sigma}_Y = e^{-\frac{t\mathbf{\Gamma}}{2}} \left(\frac{1}{n}\mathbf{I} - \overline{\mathbf{w}}\,\overline{\mathbf{w}}^\top \right) e^{-\frac{t\mathbf{\Gamma}}{2}} \tag{6.37}$$

6.4.1 Choosing the Dimension of the Heat-kernel Embedding

A direct consequence of propositions 4 and 5 is that the combinatorial heat kernel is better suited than the normalized heat kernel for representing the input graph in feature space. Indeed, the embedded graph representation is centered at the origin of the feature space and the

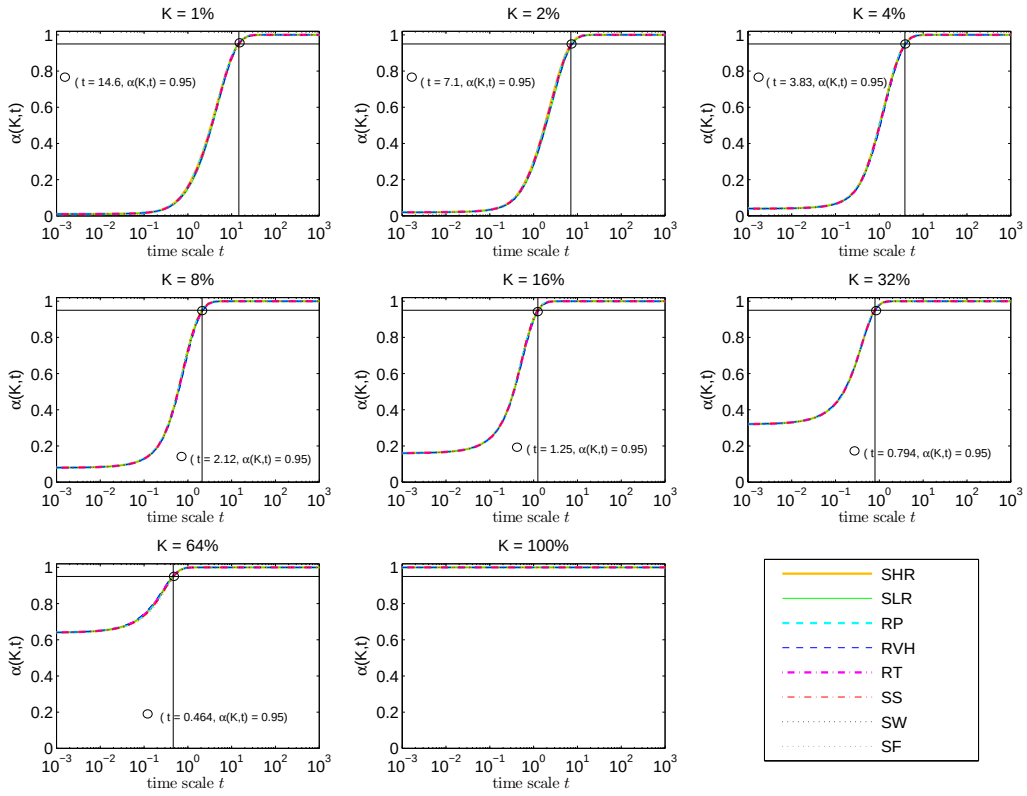


Figure 6.3: $\alpha(K,t)$ curve with varying time scales t for different K values, in which K is always expressed as a percentage of the total number of eigenvalues. The small circle indicates the time scale for which $\alpha(K,t) = 0.95$. (The Gaussian weighting scheme is used to compute the combinatorial Laplacian.)

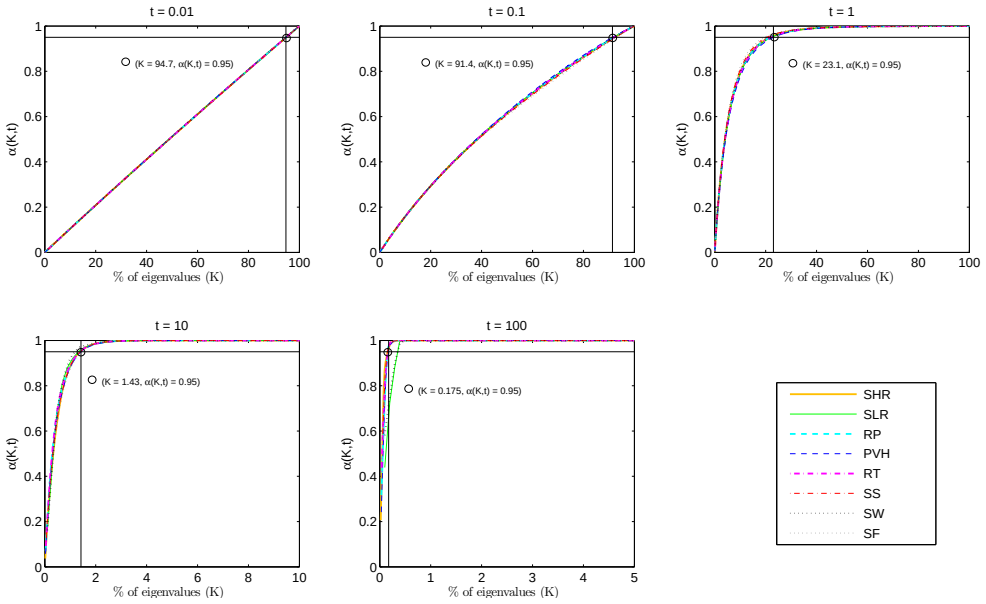


Figure 6.4: $\alpha(K,t)$ curve with varying percentage of eigenvalues for different time scale t. The small circle indicates the percentage of number of eigenvalues for which $\alpha(K,t) = 0.95$. (The Gaussian weighting scheme is used to compute the combinatorial Laplacian.)

eigenvectors of the combinatorial Laplacian correspond to the directions of maximum variance. The *principal* eigenvectors correspond to the eigenvectors associated with the K *largest* eigenvalues of the heat-kernel matrix $\mathbf{H}$, *i.e.*, $e^{-t\lambda_2} \geq e^{-t\lambda_3} \geq \ldots \geq e^{-t\lambda_K}$, or equivalently to the K smallest non null eigenvalues of the combinatorial Laplacian. Notice that in case of regular graphs $\overline{\mathbf{w}} = 0$, *i.e.*, Eq. (A.8), the two covariance matrices in Eq. (6.36) and Eq. (6.37) are strictly equivalent.

The variance along vector $\mathbf{u}_k$ is $e^{-t\lambda_k}/n$. Therefore, the total variance can be computed from the trace of the heat-kernel matrix:

$$\mathrm{tr}(\mathbf{\Sigma}_X) = \frac{1}{n}\mathrm{tr}(\mathbf{H}(t)) \tag{6.38}$$

A standard way of choosing the principal components is to use the *scree diagram*:

$$\alpha(K,t) = \frac{\sum_{k=2}^{K+1} e^{-t\lambda_k}}{\sum_{k=2}^{n} e^{-t\lambda_k}} \tag{6.39}$$

We perform an empirical analysis of the relationship between time-scale parameter and embedding dimension by plotting different scree-curves $\alpha(K,t)$ in Eq. (6.39) for a collection of 3D shapes with varying geometrical characteristics that were shown in the Figure 3.9. The different subplots of Figure 6.3 show the $\alpha(K,t)$-curve with varying time scale parameter t for

different embedding dimensions K, in which K is always expressed as a percentage of the total number of eigenvalues. Similarly, Figure 6.4 shows the scree diagram with varying percentage of the eigenvalues for the same set of shapes.

It is important to note that these overlapping curves, which corresponds to different type of visual and graphical shapes can be understood from the similar behavior of the Laplacian eigenvalue distribution (using the Gaussian weighting) discussed in the previous Section 3.3.3.3 (see Figure 3.13).

Interestingly, these overlapping curves provide two fold conclusion. First, it allow us to choose the minimum time scale depending on the size of Laplacian embedding of one shape. Second, it enable us to use the same time scale while analyzing other shape by selecting the appropriate embedding dimension.

6.5 Normalized Shape Embeddings

One disadvantage of the heat-kernel embeddings presented so far is that, when the scale parameter t is very large, the embedded representations $\mathbf{X}$ and $\mathbf{Y}$ collapse to zero. In this section, we propose three "normalizations". The first one projects the embedding on a unit hyper-sphere. The second one normalizes the embedded coordinates using the trace of the heat-kernel matrix. Finally, the third one integrates the coordinates over time, thus obtaining a representation that does not depend on t anymore.

6.5.1 Unit Hyper-sphere Embedding

In order to avoid the collapsing on $(0\dots0)$ when t goes to infinity, one may re-normalize the embedding such that the embedded vertex coordinates lie on a unit hyper-sphere of dimension K, which yields:

$$\widetilde{x}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|} \tag{6.40}$$

In more detail, the k-th coordinate of $\widetilde{x}_i$ writes as:

$$\tilde{x}_{ik}(t,K) = \frac{e^{-\frac{t\lambda_k}{2}} u_{ik}}{\left(\sum_{l=2}^{K+1} e^{-t\lambda_l} u_{il}^2\right)^{1/2}}, 2 \leq k \leq K+1 \tag{6.41}$$

The heat distance (6.30) becomes equivalent to the geodesic distance on a unit hyper-sphere:

$$d_{\mathcal{S}}(i,j;t,K) = \arccos \widetilde{x}_i^{\top} \widetilde{x}_j \tag{6.42}$$

which is equivalent to:

$$d_{\mathcal{S}}(i,j;t,K) = \arccos \frac{h(i,j;t,K)}{h(i,i;t,K)^{1/2} h(j,j;t,K)^{1/2}} \tag{6.43}$$

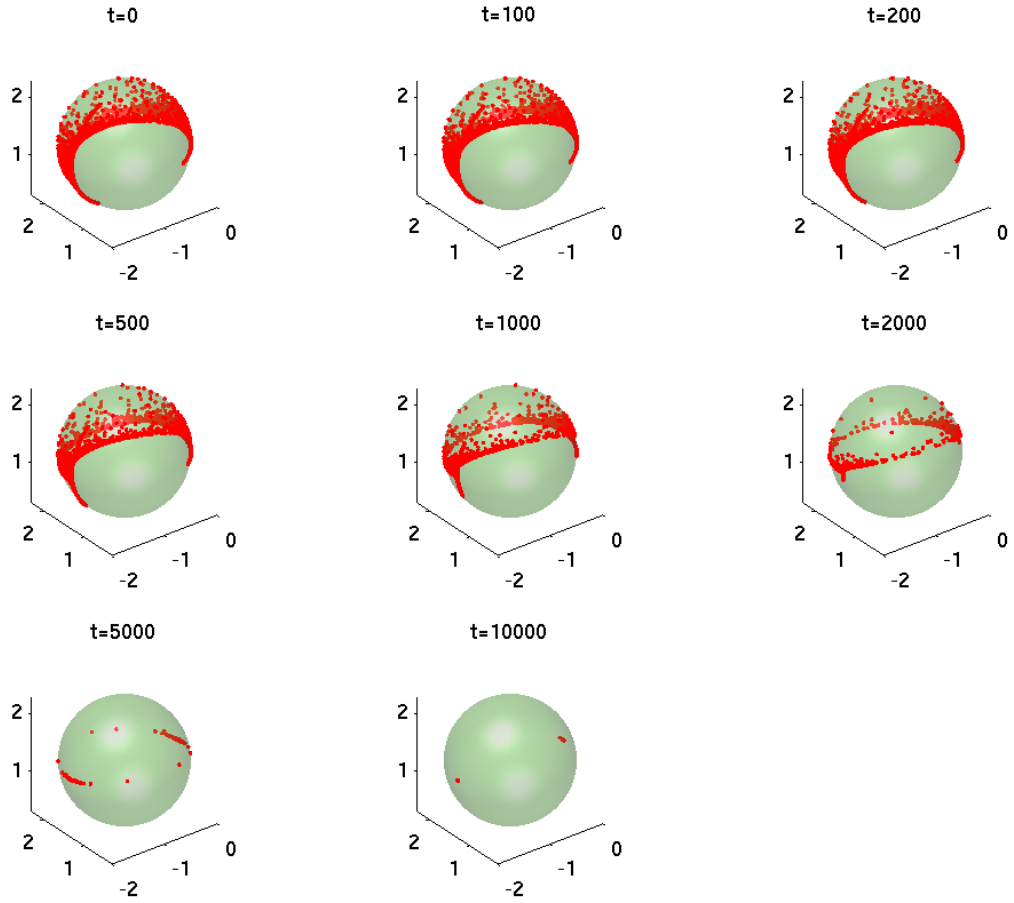


Figure 6.5: A visualization of the unit hyper-sphere normalized heat kernel embedding using the first three non-null Laplacian eigenvectors of a 3D shape. With the increasing values of time scale parameter, the embedding slowly converges to two opposite polar points on the hyper-sphere (in this case at t=10000).The embedding will in turn shrink to a single point on the hyper-sphere for the infinitely very large value of 't'. However, it is difficult to plot the embedding for such value of time scale since the original embedding is of high dimensions as opposed to the plotted three-dimensional visualization.

It is interesting to study the behavior of the embedded points as a function of the time parameter. For that purpose, notice that:

$$\tilde{x}_{ik}(0) = \frac{u_{ik}}{\left(\sum_{l=2}^{K+1} u_{il}^2\right)^{1/2}} \tag{6.44}$$

Moreover, the square of the k-th coordinate of an embedded point can be written as:

$$\tilde{x}_{ik}^2(t) = \frac{e^{-t(\lambda_k-\lambda_2)}u_{ik}^2}{\sum_{l=2}^{m+1} u_{il}^2 + \sum_{l=m+2}^{K+1} e^{-t(\lambda_l-\lambda_2)}u_{il}^2} \tag{6.45}$$

where m denotes the multiplicity of λ_2 (the first non null eigenvalue of the Laplacian matrix) Therefore, there are m mutually orthonormal eigenvectors $\mathbf{u}_2,\ldots,\mathbf{u}_{m+1}$ spanning the eigenspace of dimension m associated with λ_2. Since $\lambda_2 = \ldots = \lambda_{m+1} < \lambda_{m+2} \leq \ldots \leq \lambda_{K+1}$ we obtain:

$$\lim_{t\to\infty} \tilde{x}_{ik}(t) = \begin{cases} \frac{u_{ik}}{\left(\sum_{l=2}^{m+1} u_{il}^2\right)^{1/2}} & \text{if } 2 \leq k \leq m+1 \\ 0 & \text{if } m+2 \leq k \leq K+1 \end{cases} \tag{6.46}$$

Therefore, we obtain the following embedding:

$$\widetilde{x}_i(t\to\infty) = \begin{pmatrix} \frac{u_{i2}}{\left(\sum_{l=2}^{m+1} u_{il}^2\right)^{1/2}} & \cdots & \frac{u_{im+1}}{\left(\sum_{l=2}^{m+1} u_{il}^2\right)^{1/2}} & 0 & \ldots & 0 \end{pmatrix}^\top \tag{6.47}$$

Notice that when $m = 1$, *i.e.*, the smallest non null eigenvalue has multiplicity one, the embedding collapses, at the limit, to the point:

$$\widetilde{x}_i(t\to\infty) = \begin{pmatrix} 1 & 0 & \ldots & 0 \end{pmatrix}^\top \tag{6.48}$$

Figure 6.5 visualize a unit hyper-sphere normalized embedding using the first three non-null Laplacian eigenvectors of a 3D shape. With the increasing values of time scale parameter, the embedding slowly converges to two opposite polar points on hypersphere. The embedding will in turn shrink to a single point on the hyper-sphere for an infinitely very large value of the time scale parameter. However, it is difficult to plot the embedding for such value of the time scale parameter since the original embedding is higher dimensional as opposed to the plotted three-dimensional visualization.

6.5.2 Trace-scaled Embedding

Another possible re-normalization is to re-scale the embedded coordinates using the trace of the heat-kernel matrix, *i.e.*,:

$$\widehat{x}_i = \frac{\mathbf{x}_i}{\mathrm{tr}(\mathbf{H}(t))} \tag{6.49}$$

From (6.32) we obtain that $\mathrm{tr}(\mathbf{H}(t)) = \sum_{i=1}^{n} \|\mathbf{x}_i\|^2$ and hence we have:

$$\sum_{i=1}^{n} \|\widehat{x}_i\|^2 = 1 \tag{6.50}$$

Notice that the trace can also be computed with the sum of the eigenvalues, $\sum_{k=1}^{n} e^{-t\lambda_k}$, or the total variance. If we retain the first K non null eigenvalues, the k-th component of $\widetilde{x}_i$ is given by:

$$\hat{x}_{ik}(t,K) = \frac{e^{\frac{-t\lambda_k}{2}} u_{ik}}{\left(\sum_{l=2}^{K+1} e^{-t\lambda_l}\right)^{1/2}}, 2 \leq k \leq K+1 \tag{6.51}$$

The heat distance (6.30) writes in this case:

$$d_{\mathcal{I}}^2(i,j;t,K) = \sum_{k=2}^{K+1} \hat{\phi}_k (u_{ik} - u_{jk})^2 \tag{6.52}$$

where the *trace-normalized* eigenvalues of the heat-kernel matrix are:

$$\hat{\phi}_k = \frac{e^{-t\lambda_k}}{\sum_{l=2}^{K+1} e^{-t\lambda_l}}$$

The trace-scaled embedding at $t = 0$ corresponds to the Laplacian embedding:

$$\widehat{x}_i(0) = K^{-1/2} \begin{pmatrix} u_{i2} & \dots & u_{iK+1} \end{pmatrix}^\top \tag{6.53}$$

Using a similar development as in Section 6.5.1, we obtain:

$$\lim_{t\to\infty} \hat{x}_{ik}(t) = \begin{cases} \frac{u_{ik}}{m^{1/2}} & \text{if } 2 \leq k \leq m+1 \\ 0 & \text{if } m+2 \leq k \leq K+1 \end{cases} \tag{6.54}$$

which yields:

$$\widehat{x}_i(t\to\infty) = m^{-1/2} \begin{pmatrix} u_{i2} & \dots & u_{im+1} & 0 & \dots & 0 \end{pmatrix}^\top \tag{6.55}$$

Hence, for a large value of t the eigenspace of dimension m, associated with the multiple eigenvalue λ_2, is sufficient to embed the shape. Notice that when $m = 1$ this embedding corresponds to mapping the shape's vertices onto the first non constant eigenvector of the Laplacian matrix, *i.e.*, the Fiedler vector: $\widehat{x}_i(t\to\infty) = \begin{pmatrix} u_{i2} & 0 & \dots & 0 \end{pmatrix}^\top$: Each shape vertex will be represented by a real number along the direction of the Fiedler vector.

6.5.3 Time-invariant Embedding

The discrete heat operator and its associated spectral representations depend on the time parameter t. One can easily obtain a representation that is time-invariant by integration of (6.20):

$$\begin{aligned} \mathbf{L}^\dagger &= \int_0^\infty \mathbf{H}(t) \\ &= \int_0^\infty \sum_{k=2}^{n} e^{-t\lambda_k} \mathbf{u}_k \mathbf{u}_k^\top dt \\ &= \sum_{k=2}^{n} \frac{1}{\lambda_k} \mathbf{u}_k \mathbf{u}_k^\top \\ &= \mathbf{U} \boldsymbol{\Lambda}^\dagger \mathbf{U}^\top \end{aligned} \tag{6.56}$$

with $\mathbf{\Lambda}^{\dagger} = \text{Diag}[\lambda_2^{-1}, \ldots, \lambda_n^{-1}]$. Matrix $\mathbf{L}^{\dagger}$ is called the *discrete Green's function* [Chung 2000] and it corresponds to the Moore-Penrose pseudo-inverse of the combinatorial Laplacian matrix. There is an equivalent definition for the normalized discrete Green's function. This time-invariant representation is also referred to as the *commute-time* embedding (see Section 2.6.5.2). By integration we obtain a number of time-invariant representations such as the commute-time distance (CTD), the commute-time embeddings, the commute-time auto-diffusion function, and the maximum-variance representation. One can easily verify the following formulas when using the first K non constant eigenvectors of the Laplacian matrix:

$$\mathbf{x}_i = \left(\lambda_2^{-1/2} u_{i2} \ \ldots \ \lambda_{K+1}^{-1/2} u_{iK+1}\right)^{\top} \tag{6.57}$$

$$\|\mathbf{x}_i\|^2 = \sum_{k=2}^{K+1} \lambda_k^{-1} u_{ik}^2 \tag{6.58}$$

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = \sum_{k=2}^{K+1} \lambda_k^{-1} (u_{ik} - u_{jk})^2 \tag{6.59}$$

$$\mathbf{\Sigma}_X = \frac{1}{n} \text{Diag}[\lambda_2^{-1}, \ldots, \lambda_{K+1}^{-1}] \tag{6.60}$$

6.6 3D Shape Description using the Heat Diffusion Framework

In this section, we describe a novel 3D shape descriptor using heat diffusion framework. First we discuss the choice of the embedding representation in the context of visual shape data. Next, we outline a key-point detection method for obtaining a set of interest points on 3D shapes. Finally, we present the construction of a new multi-scale intrinsic feature descriptor using the detected key-points and the scale dependent heat diffusion metric.

6.6.1 Choosing Embedding Representation for Visual Shapes

The choice of embedding representation is very important while performing a multi-scale shape analysis. We have already concluded in Section 6.4.1 that the combinatorial heat kernel is better suitable for multi-scale shape representation. The embedding normalization presented in the previous section is also very crucial factor for shape representation. The time-invariant normalization (Section 6.5.3) is by definition suitable for a scale invariant representation. In the case of multi-scale representation, both the unit hyper-sphere normalization (Section 6.5.1) and the trace-scaled normalization (Section 6.5.2) are applicable. We prefer to choose the unit hyper-sphere normalized embedding representation since it also provides an inherent normalization of the spectral distances by projecting all the K-dimensional Laplacian embedding point cloud on a unit hyper-sphere. This is particularly interesting property for visual shape registration where two shapes can have significantly different sampling. Other critical aspect of the shape representation is the choice of embedding dimension as well as the scale of analysis. We have already presented a detailed discussion relating these two shape representation parameters in Section 6.4.1.

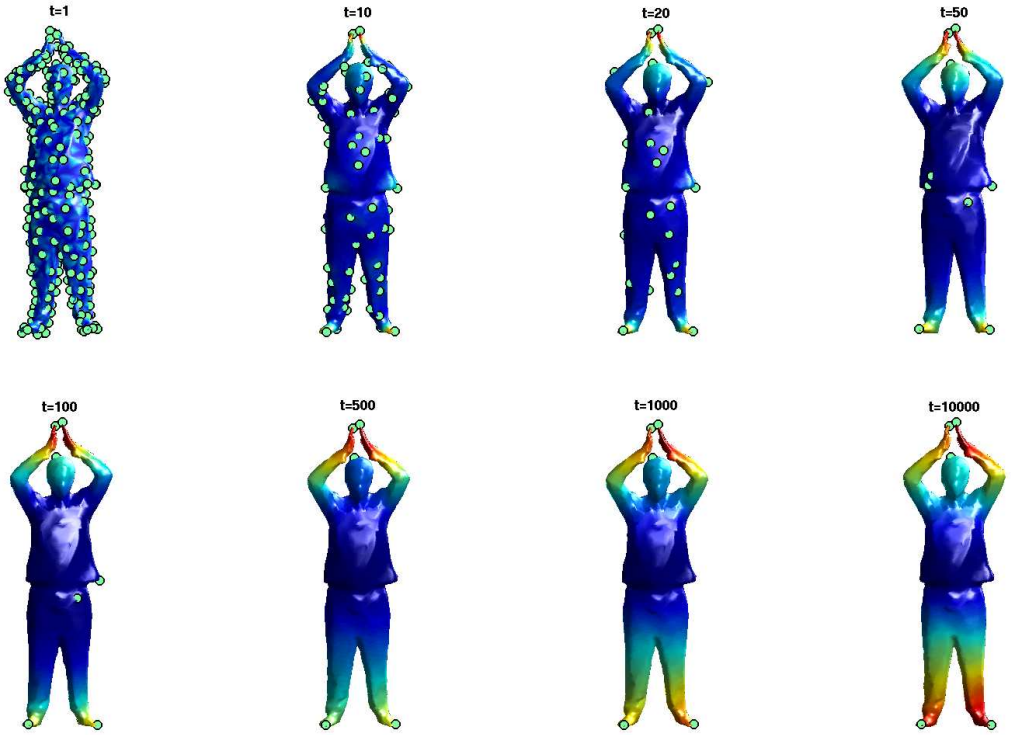
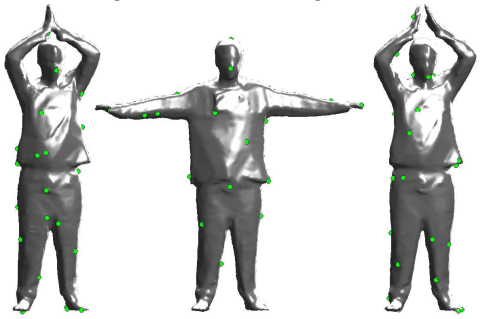
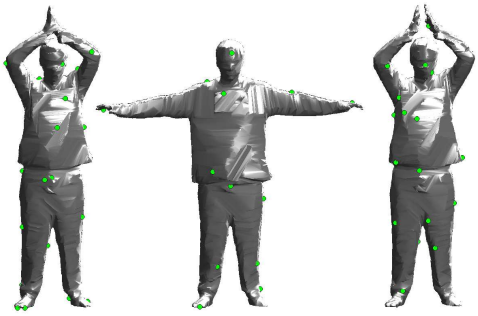


Figure 6.6: Auto diffusion function (ADF) maxima on meshes. Colors plotted on mesh represents the absolute function values and Green dots depicts the function maxima computed locally using a two ring neighboring. At smaller time-scale values, we obtain large number of maxima due to local nature of ADF while at larger time scale values the ADF is global and has small number of stable maxima.

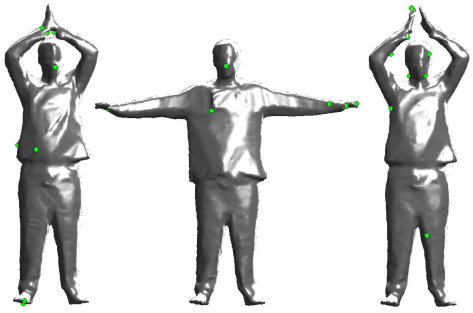
Processed high-resolution triangulated meshes *Low-resolution triangulated meshes*



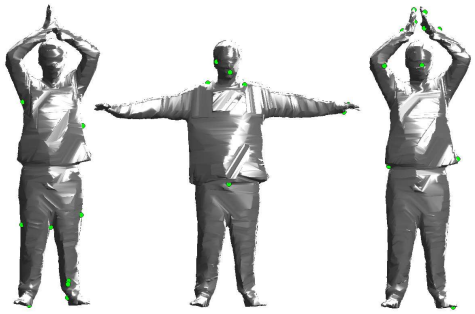
(a) Harris3D dense key-point detection (setup II).



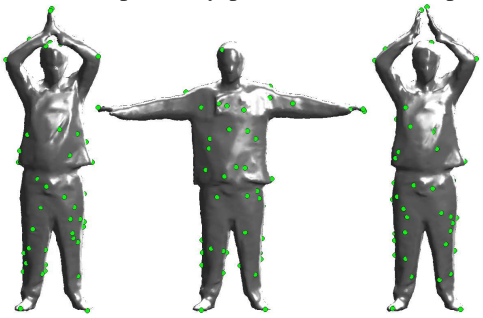
(b) Harris3D dense key-point detection (setup II).



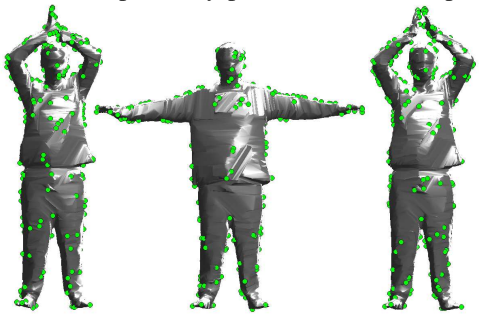
(c) Harris3D sparse key-point detection (setup I).



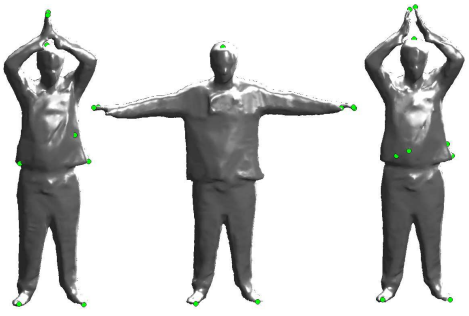
(d) Harris3D sparse key-point detection (setup I).



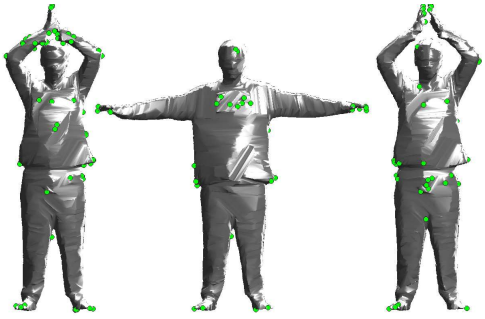
(e) ADF maxima dense key-point detection (setup II).



(f) ADF maxima dense key-point detection (setup II).



(g) ADF maxima sparse key-point detection (setup I).



(h) ADF maxima sparse key-point detection (setup I).

Figure 6.7: Key-point detection on visual shapes using the Harris 3D key-point detector [Sipiran 2011] (a-d) and the maxima of auto-diffusion function, *i.e.*, Eq. (6.32)(e-h). The former misses while the latter manages to repeatedly detect key-points around shape protrusions. Table 6.1 provides the choice of parameters for the two experiments.

6.6.2 Key-point Detection

Feature based representation is very common in the image processing community where first a set of *key-points* or *interest-points* are detected on an image and then the image is described as the vector of features computed on these key-points. A detailed survey on image key-point detection is presented in [Tuytelaars 2008]. Some desired properties for these key-points are *Repeatability, Locality, Distinctiveness, Accuracy* and *Efficiency*. Repeatability is a key aspect for any key-point detection method. Another important aspect is the scale at which key-points are detected. One popular approach is to perform a multi-scale analysis in order to make the detection process repeatable in the presence of various image transforms. The key idea, based on seminal work presented in [Lindeberg 1994], is to select the characteristic scale of a local structure, for which a given function attains an extremum over scales.

On the other hand, the idea of key-points is not well defined in the 3D shape community. A commonly used definition as stated by [Sipiran 2011] is to relate the measure of interest with the level of protrusion of the outstanding local structures. This definition directly points to a multi-scale setup for detection of key-points, as these local structures can exist at multiple scales. Existing multi-scale methods for image key-points detection can easily be extended to 3D shapes by defining smooth differentiable functions on 2D manifold surface. One simple choice of such function is the Euclidean coordinates of 3D shape that can also be combined with surface texture and applied with Gaussian smoothing to obtain a scale-space representation [Zaharescu 2009, Maes 2010]. However, this choice is computationally expensive. A recent work proposed in [Sipiran 2011], which extends Harris operator for 3D shapes, is stated as the best performing feature detector in the existing SHREC benchmark [Boyer 2011]. The proposed method uses an adaptive technique to determine the neighborhood of a vertex, over which the local Harris response is calculated.

However, majority of these methods were primarily design to work on graphical data that usually simulate various shape transforms, but are much simpler than the visual shapes. This could lead to unexpected behavior while operating on visual data with large surface noise and topological discrepancies. One more limitation is that these methods do not guarantee the detection of all major shape protrusions, while seeking a sparse set of key-points. This is particularly more challenging due to the fact that the repeatability of key-points decreases as the number of key-points decreases, as observed in [Boyer 2011].

Another class of approaches uses the geometric diffusion theory for key-point detection. The diffusion operators applied over set of functions defined over the manifold surface reveals the intrinsic geometry of 3D shapes. We adapt the HKS method proposed in [Sun 2009], to detect a set of key-points on a 3D shape as the maxima of the heat norm or the auto-diffusion function shown in Eq. (6.32). Figure 6.6 shows the detected feature points as the maxima of heat norm computed at different time scales. At smaller time scales, we detect large number of key-points due to the local nature of the auto diffusion function, which capture the local surface variations. While at larger time scales, the auto diffusion function is global and hence the detected key-points are sparse and mainly detected on the shape protrusions.

A very recent work in [Tombari 2012] presents an exhaustive discussion on performance evaluation of the existing key-point detectors. They rightfully accept the fact that the HKS detector offers a wider degree of invariance than the other detectors, and is beneficial in applications like non-rigid matching. However, according to them it was less relevant to the scenarios considered in their evaluation methodology and hence only considered as a fixed scale detector.

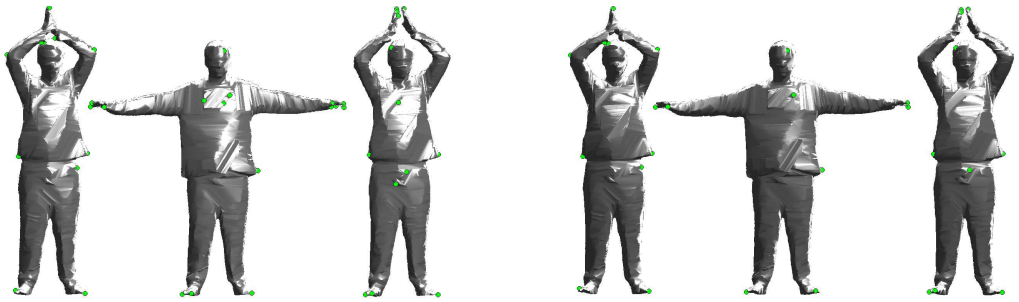
Here it is important to note that, unlike previous similar attempts in the literature, we now have a formalism to choose the appropriate time scale and related embedding dimension. This will provide an inherent normalization to shape sampling while detecting key-point on two visual shapes with significantly different meshing.

Figure 6.7(a-d) shows the results of Harris 3D feature point detection on visual shapes using [Sipiran 2011]. We performed two experiments using two sets of parameters in order to obtain sparse (a,b) and dense (c,d) key-point detection. This method fails to repeat the detection of sparse key-points around the shape protrusions. On the other hand, the repeatability of the sparse key-points detected using the auto-diffusion function in Eq. (6.32) is more reliable, as shown in the Figure 6.7(e-h). We detect sparse and dense key-points using large and small values of time scale parameter, respectively. Table 6.1 presents parameter values used in the experimental setups. It is important to note that the choice of embedding dimension K is consistent with the analysis presented in Section 6.4.1 *e.g*, for sparse key-point detection at $t = 500$ we chose $K = 10$ for visual shapes that has approximate size of $n = 10000$ vertices, which is around 0.1% of the size of complete Laplacian spectrum (see Figure 6.4).

Key-point Detection Method	Setup I -"*Sparse*"	Setup II -"*Dense*"
Harris3D (see [Sipiran 2011]).	type-neighborhood = adaptive, parameter-neighborhood = 0.01, Harris parameter = 0.04, ring-maxima-detection = 2, interest-points-selection = fraction, parameter-selection = 0.001.	type-neighborhood = adaptive, parameter-neighborhood = 0.01, Harris parameter = 0.04, ring-maxima-detection = 2, interest-points-selection = clustering, parameter-selection = 0.001.
ADF Maxima, see Eq. (6.32).	embedding-dimension K=10, time scale parameter $t = 500$, ring-maxima-detection = 2.	embedding-dimension K=50, time scale parameter $t = 50$, ring-maxima-detection = 2.

Table 6.1: Parameter values for key-point detection examples in Figure 6.7.

Interestingly, we also found that by increasing the size of embedding dimension, *i.e.*, K yields the same key-point detection results, and hence verifying the empirical analysis presented in Section 6.4.1. We have another important observation by varying the *ring-maxima-detection* parameter for low-resolution visual shapes. Our results suggests that even for such unprocessed visual shape one can get reliable key-point detection around shape protrusions by increasing the ring size for local maxima detection of ADF function. Figure 6.8 shows one examples with varying ring size for local maxima detection.



(a) *ring-maxima-detection*= 5. (b) *ring-maxima-detection*= 6.

Figure 6.8: Key-point detection on low-resolution triangulated visual shapes using the maxima of auto-diffusion function with varying ring size while performing a local maxima detection. As compared to the key-point detection in Figure 6.7(h), which uses smaller ring size, using the larger ring size for low-resolution triangulated visual shapes yields better key-point detection results.

Hence, we propose to detect a set of sparse but reliable key-points on 3D visual shapes as the maxima of auto diffusion function computed at larger time scales.

6.6.3 Multi-scale Dense Descriptors

As concluded in Section 6.1, the intrinsic feature descriptors are more suitable for 3D articulated shape representation. In this section, we introduce two multi-scale descriptors, which are densely computable for each vertex of the shape graph.

Given a visual shape represented as connected undirected weighted shape graph, we can recall various mathematical notations from Section 6.2.2. A shape graph is represented as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} = \{v_1, \ldots, v_n\}$ is the vertex set. Let $\mathcal{P} = \{p_1, \ldots, p_M\}$ be the set of M sparse key-points detected using the multi-scale analysis presented in the previous section, where $\mathcal{P}$ is essentially a subset of $\mathcal{V}$, *i.e.*, $\mathcal{P} \subseteq \mathcal{V}$.

While defining a multi-scale descriptor, we can use τ number of time scales denoted as $(t_1, \ldots, t_\tau)$. Once these values are decided, one can easily choose the corresponding embedding dimension based upon minimum time scale value using the scree-curves analysis presented in Section 6.4.1.

6.6.3.1 Multi-scale Heat Diffusion Descriptor

Here we outline the construction of heat diffusion descriptor also know as the heat kernel signature (HKS) [Ovsjanikov 2010]. The heat diffusion descriptor for each vertex v_j stores the amount of heat diffused to it in different time steps $t_1, \ldots, t_\tau$ from each of the multiple

point heat sources placed at each of key-point in $\mathcal{P}$. Thus, this can be represented as a 2D array of dimension $\tau \times M$.

Descriptor Construction: Let the amount of heat available at vertex $v_i \in \mathcal{V}$ at time t, starting with a point heat distribution source placed at key-point $p_m \in \mathcal{P}$, where p_m is the j-th vertex in $\mathcal{V}$, is denoted as $h(v_i, p_m; t)$.

We can now represent a τ-dimensional vector for each vertex $v_i \in \mathcal{V}$ that captures the respective multi-scale heat diffusion from a of the key-point $p_m \in \mathcal{P}$ as:

$$\mathbf{h}_{p_m}^{v_i} = [h(v_i, p_m; t_1), \ldots, h(v_i, p_m; t_\tau)]^\top \tag{6.61}$$

In addition to the heat diffusion from these M key-points, we also consider *self diffusion* at multiple times to capture the local geometry. This is encapsulated in a an additional vector:

$$\mathbf{h}_{M+1}^{v_i} = [h(i, i; t_1), \ldots, h(i, i; t_\tau)]\tau \tag{6.62}$$

Finally, considering the heat diffusion from all the key-points and self-diffusion we can define a $\tau \times M + 1$ descriptor for each vertex $v_i \in \mathcal{V}$ of the shape graph $\mathcal{G}$ as:

$$\mathbf{H}^{v_i} = [\mathbf{h}_{p_1}^{v_i} \ldots \mathbf{h}_{p_M}^{v_i} \mathbf{h}_{M+1}^{v_i}]_{\tau \times (M+1)}. \tag{6.63}$$

6.6.3.2 Multi-scale Heat Distance Descriptor

Here we introduce a novel heat distance descriptor, which stores spectral distances between a given vertex to the set of key-points. The descriptor is similar in the spirit to previously proposed intrinsic descriptors in [Ahmed 2008, Ovsjanikov 2010]. The proposed descriptor uses scale dependent heat diffusion metric in Eq. (6.30) introduced in Section 6.3. The use of scale dependent metric enabled a straightforward extension to a multi-scale descriptor, which by storing heat distances computed at multiple-scales from each key-point.

Thus, the heat distance descriptor for each vertex v_j stores the multi-scale heat distances computed at $t_1, \ldots, t_\tau$ time scales from each of the key-point in $\mathcal{P}$. This can also be represented as a 2D array of dimension $\tau \times M$. This is very similar to heat diffusion descriptor outlined in the previous section except that here we store heat distance instead of heat diffusion. However, heat diffusion only consider amount of heat transfer whereas heat distances also captures the total amount of heat at source and destination vertex, thereby capturing more information, see Eq. (6.30).

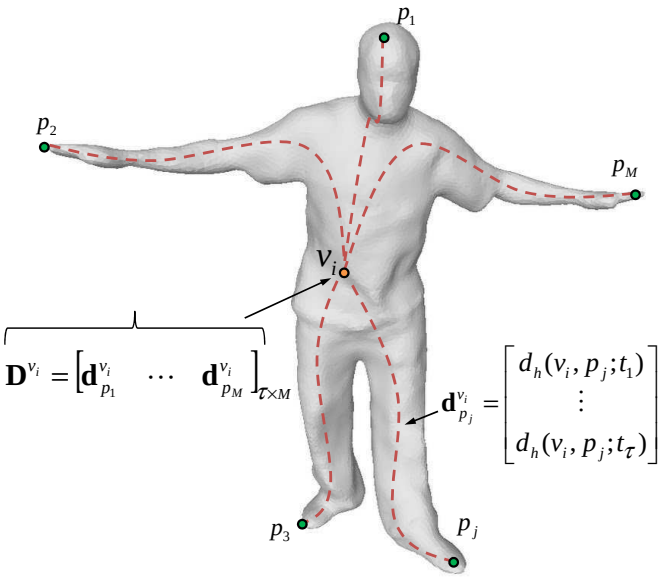


Figure 6.9: Construction of a multi-scale heat distance descriptor. The red dotted line depicts the multi-scale heat distance to a given vertex from a specific key-point. The descriptor stores these distances to a given vertex from all the key-points.

Descriptor Construction: Let's denote scale dependent spectral distance, *i.e.*, heat distance shown in Eq. (6.30) between two vertices v_i and v_j as $d_h(v_i, v_j; t)$. One can also choose an appropriate definition of heat distance from a set of different metrics derived in the Section 6.5. The choice of metric will depend upon the type of embedding normalization considered.

We can compute a τ-dimensional vector for a vertex $v_i \in \mathcal{V}$ that captures the respective multi-scale heat distance from one of the key-point $p_m \in \mathcal{P}$ as:

$$\mathbf{d}^{v_i}_{p_m} = [d_h(v_i, p_m; t_1), \ldots, d_h(v_i, p_m; t_\tau)]^T \tag{6.64}$$

Finally, considering the heat distances from all the key-points, we can define a $\tau \times M$ descriptor matrix for each vertex $v_i \in \mathcal{V}$ of the shape graph $\mathcal{G}$ as:

$$\mathbf{D}^{v_i} = [\mathbf{d}^{v_i}_{p_1} \ldots \mathbf{d}^{v_i}_{p_j} \ldots \mathbf{d}^{v_i}_{p_M}]_{\tau \times M}. \tag{6.65}$$

Figure 6.9 depicts the construction of the proposed descriptor.

6.6.4 Descriptor Matching Score

The matching between two heat distance descriptors defined on different visual shapes will require the pre-computed one-to-one assignment between the set of key-points on two shapes. We will discuss this aspect while performing shape matching in the next section.

By assuming that a one-to-one assignment between the two set of key-points is known, the matching score between a descriptor $\mathbf{D}^{v_i}$ on shape $\mathcal{G}$ and another descriptor $\mathbf{D}^{v'_j}$ on a different shape $\mathcal{G}'$ is computed as:

$$score(\mathbf{D}^{v_i}, \mathbf{D}^{v'_j}) = \sum_{m=1}^{M} ||\mathbf{d}_m^{v_i} - \mathbf{d}_m^{v'_j}||_\infty. \tag{6.66}$$

We choose $L-\infty$ norm while computing the matching score so that for each key-point, the descriptor matching score consider the largest difference in the heat distance values in vector $\mathbf{d}_m^{v_i} - \mathbf{d}_m^{v'_j}$, irrespective of the time scale parameter. This ensure that the matching is not biased to any specific time-scale.

Thus, the vertex descriptors from two different shapes should have a low matching error score if the corresponding vertices have similar multi-scale heat distances from their respective key-points. A similar matching formulation was outlined in [Ovsjanikov 2010] for heat diffusion descriptor.

It is important to note that computing such matching will require a robust alignment of sparse key-points. We will discuss this aspect in detail while performing a sparse visual shape matching in the next chapter.

6.7 Conclusion & Future Work

In this chapter, we have outlined a generalized framework for the representation and analysis of 3D visual shapes based on heat diffusion on undirected weighted graphs. In this context, a detailed mathematical analysis of various spectral constructs has been provided. The presented analysis has lead to a formalism that allowed us to combine the scale parameter of heat diffusion and dimensionality of spectral embedding, providing a basis for performing dimensionality reduction and, more generally, to characterize the statistical properties of the embedded shape representation at multiple scales. In addition to this, we have also proposed a novel multi-scale heat distance descriptor for feature based shape representation.

As part of the future work, we plan to extend this framework to a family of exponential kernels. Although, we did proposed a novel analysis relating the scale of analysis with dimension of the heat-kernel embedding, a more sophisticated method to choose the scale of analysis will be more desirable.

In the next chapter, we will employ the proposed heat descriptors for dense shape registration in the presence of large topological changes.

CHAPTER 7

Topologically-Robust Dense Shape Matching

Contents

7.1 Introduction

We have already introduced the problem of surface mapping in the continuous domain and the graph matching as it's discrete counterpart in Chapter 2. There we presented the spectral formulation for exact graph matching problem and a relaxed isomorphism solution in Section 2.4.4. However, the exact graph matching is typically not suitable for matching the shape graphs representing visual shapes due to the existence of large acquisition noise, non-uniform sampling and topological changes. Instead, the inexact graph matching is the key for visual shapes as it allows one to relax the constraints of strict isomorphism, *e.g.*, a probabilistic many-to-one matching instead of strict binary matching.

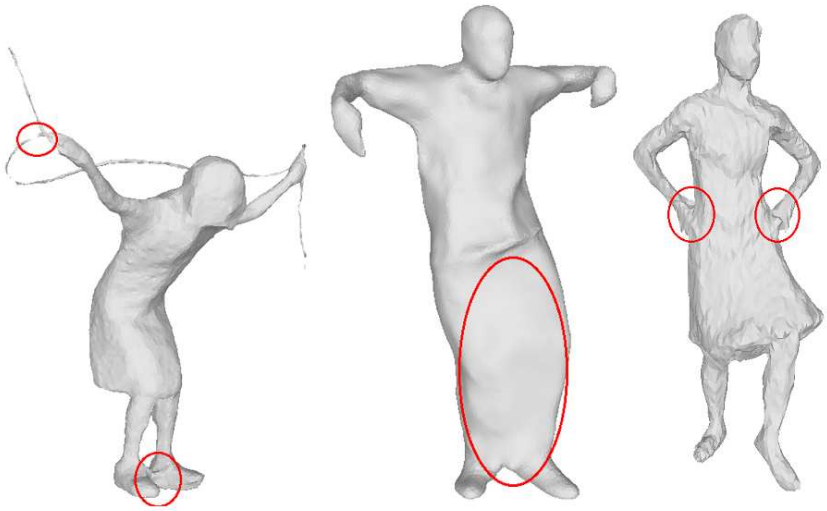


Figure 7.1: Examples of commonly occurring topological changes in visual shapes.

Thus, we propose to employ inexact graph matching methods in order to obtain a dense matching solution for visual shapes. In particular, we focus on commonly occurring topological changes in the visual shapes, which is a major challenge for majority of the existing dense shape matching methods. The problem of topological changes exists in visual shapes obtained from the multi-camera acquisition systems. Such acquisition systems provide independently reconstructed visual shapes for a dynamic 3D object. However, obtaining a 3D animation sequences with spatio-temporal coherence, based on these independently reconstructed shapes, is a challenging task. It inherently involves estimation of *dense* 3D correspondences. This is particularly difficult in the case of articulated shapes due to complex kinematic poses. These poses induce self-occlusions and shadow effects which cause topological changes along the sequence, such as merging and splitting. Figure 7.1 demonstrates some common occurrences of topological changes in independently reconstructed visual shapes that were captured via multi-camera acquisition systems.

We have already seen some examples of topology change in Section 3.4, suggesting that the single scale spectral representation is not suitable for shapes with topological merging and splitting. The multi-scale heat-kernel framework presented in the previous chapter provides a tool to analyze shapes at different scales. Since the topological changes are local phenomenon and does not affect the surface characterization of the regions that are not in the direct proximity of such changes. Hence, we can adapt the multi-scale framework and analyze the visual shapes with topological changes by performing a local analysis at smaller scales, thereby computing the local surface characterization.

In this chapter, we discuss in detail spectral framework for inexact graph matching and employ it for dense matching of visual shapes represented by shape graphs (see Section 3.2).

First we derive a sparse shape matching method for key-point alignment. Next, using the sparse key-point matching, we compute multi-scale heat diffusion descriptors (as outlined in the previous chapter) and employ a novel seed growing algorithm to find a dense binary matching solution for visual shapes. Finally, we propose a method to find a topologically-robust dense probabilistic matching solution for visual shapes with large topological changes.

The chapter organization is as follows. In the beginning, we present a detailed discussion on existing solutions for visual shape matching in the context of inexact graph matching in Section 7.1.1 and list our contributions in Section 7.1.2. This is followed by Section 7.2 where we present two sparse binary shape matching methods using the geometrical and texture features on visual shapes. In Section 7.3, we propose a novel seed-growing algorithm that combines with sparse matching and heat distance descriptors in order to find dense binary shape matching solution. In the next Section 7.4, we propose a method to find topologically robust dense probabilistic matching using the EM based algorithm outlined in Section 7.4.1. In Section 7.5, we present the qualitative and quantitative evaluating of dense matching results that illustrate the robustness of the proposed method. Finally, Section 7.6 concludes the chapter with a brief discussion the on proposed method and future directions of work.

7.1.1 Related Work

A large amount of 3D shape analysis literature exists for shape registration/matching task. In this section, first we summarize the inexact graph matching followed by a compilation of the existing 3D shape matching methods.

The Inexact Graph Matching: We have already mentioned that in computer vision it is popular to cast the problem of shape matching as the graph matching problem where 3D shapes are commonly represented by discrete shape graphs (meshes). It is common to use the inexact graph matching formulations, which relax the constraints of exact graph matching, thereby allowing an approximate matching solution on shape graphs representing challenging visual shapes. The commonly relaxed constraints are many-to-one matching in place of one-to-one matching or a probabilistic soft assignments in place of a strict binary assignments. However, this leads to a bigger solution space and hence the approximate or sub-optimal solutions.

The three common approaches to inexact graph matching are: 1) Transforming one graph to other by penalizing the graph editing operations and choosing the one with minimum editing cost, *e.g.*, [Bunke 1999, Llados 2001, Neuhaus 2006]; 2) Relaxing the strict isomorphism by defining a cost function that penalizing certain matching configurations and finding a graph matching solution with minimum cost, *e.g.*, [Bolles 1982a, Gold 1996, Pelillo 1999, Schellewald 2005, Leordeanu 2005, Cour 2006, Cour 2997]; and 3) Spectral approach to embed the graph in a n-dimensional Euclidean metric space, which reflects the original graph connectivity and then applying a linear or non-linear matching in order to obtain a sub-optimal matching solution, *e.g.*, [Umeyama 1988, Scott 1991, Shapiro 1992, Wang 2006, Luo 2003,

Mateus 2008]. We recommend reading the Section 4.2 in [Mateus 2009] for a detailed survey on inexact graph matching methods.

Sparse and Dense Shape Matching Methods: A popular inexact graph matching technique is to cast the graph matching problem into an integer quadratic problem (IQP) that is equivalent to find a set of mutually compatible nodes in an association graph [Bolles 1982a]. However, this is an instance of intractable *NP hard* problems. Though, there exists sub-optimal solutions that uses various possible relaxations, *e.g.*, spectral relaxation [Leordeanu 2005] and replicator equation method [Pelillo 1999]. These methods are tractable only when the graphs are very small and hence are suitable only for the sparse matching of key-points.

On the other hand, a large number of shape matching methods find vertex-to-vertex dense matching of the corresponding shape graphs. There are two main classes of approaches that achieve dense 3D shape matching. The first class consists of model-based approaches, which by-pass the problem of topological issues by starting with a prior shape model. This model is locally deformed at each time step of the sequence in order to obtain a globally consistent shape representation. This is achieved at the cost of loosing detailed geometric and texture information obtained at each independent reconstruction. Another problem is the accumulation of deformation errors over time. Moreover, in the case of dynamic scenes, the assumption of a prior model is not realistic.

The second class consists of model-free approaches, which do not impose any shape priors. Initially, sparse correspondences are computed between two independently reconstructed shapes using local cues based either on texture or on geometry. These correspondences are then propagated to obtain a dense shape matching. However, it is often the case that these initial sparse correspondences are not uniformly distributed over the shapes, and hence the propagation of these correspondences is a challenging task. It is even more difficult when the two shapes differ significantly. Indeed, one major limitation of this class of methods is that they use a geodesic distance onto the shape manifold, which is not robust to changes in topology.

In this work, we mainly focus on model-free dense matching approaches that primarily use the geometrical cues for 3D shape matching.

There exists a class of solutions based on iterative closest point (ICP) method adapted to 3D shape matching [Sussmuth 2008, Tevs 2009, Cagniart 2010]. These methods perform shape registration in the Euclidean space and hence do not provide a pose invariant matching. The high computation complexity is also an important issue with some existing dense shape matching methods, *e.g.*, an energy minimization solutions based on Markov random fields (MRF) presented in [Starck 2007a], provide decent matching results, but can result in a very high computational costs when the sampling of 3D shapes increases. However, a sparse-to-dense strategy can help to ease this aspect, *e.g.*, a seed-growing method in [Čech 2010].

Many existing methods extends the graph matching to higher order formulations in order to find dense 3D correspondences, *e.g.*, [Zeng 2010a, Duchenne 2011, Smeets 2012]. However,

they mainly show results on graphical shapes.

Another class of intrinsic methods relies heavily on the assumption that different poses in the articulated shapes arises due to non-rigid deformations that are strictly isometric. Hence, the pose invariance is achieved by embedding the shape into an isometry invariant subspace [Starck 2007a, Bronstein 2009, Zeng 2010b, Reuter 2010, Sahillionglu 2010, Ovsjanikov 2010, Sharma 2011, Wu 2011]. However, this is a very strong assumption for visual shapes, as the independently reconstructed shapes are not strictly isometric and also can have topological changes (see Section 3.4 and Section 7.1). Interestingly, an intrinsic dense probabilistic matching method proposed in [Mateus 2008] is a good candidate for matching visual shapes. However, this also fails to handle topological changes and their histograms matching based heuristic to align two embeddings is not robust in case of large variations in the visual shapes.

Some existing methods on shape tracking that inherently performs dense shape matching, employed initial sparse matches to achieve robustness to the topological issues [Vlasic 2008, Huang 2008, Varanasi 2008]. In [Varanasi 2008], a mesh evolution was performed by locally deforming the 3D shape according to the sparse 3D correspondences. These sparse correspondences were obtained by minimizing an error function that evaluates the texture and geometric consistency. However, the texture based error function limits the application of the method and the geometric consistency check was performed using the geodesic metric, which is vulnerable to large shape deformations.

In [Ahmed 2008], initial sparse correspondences were used to compute a set of harmonic functions and each shape vertex was represented by the coordinates of these functions. Then a dense matching was performed based upon computation of the level set of closest initial correspondences on two shapes. However, these harmonic functions are the solution of stationary heat equation and hence are globally affected by the topological issues. An hierarchical assembly of independently reconstructed shapes was performed in [Popa 2010] for computing a globally consistent space-time reconstruction. Recently, [Huang 2011] proposed to find a single temporally consistent representation over the animation sequence by first introducing a global alignment graph structure, which uses shape similarity to identify individual shapes in a temporal registration. This was followed by a graph optimization step, which minimizes the total non-rigid deformation required to register the input sequences into a common structure. A very recent attempt in [Letouzey 2012] claims to recover both the topology and the geometry of a deformable shape over a temporal animation sequence. However, all these methods focus on recovering a single temporally consistent model over the sequence, by mainly exploiting the motion cues, instead of finding a topologically-robust dense binary matching solution.

Another class of intrinsic shape matching methods use sampling based approach and minimize some distance criterion between two shapes, *e.g.*, [Mémoli 2004] compare two point clouds representing graphical shapes (manifold surfaces) using an iterative Farthest Point Sampling algorithm [Moenning 2003], which computes an optimized covering by minimizing an approximate Gromov-Hausdorff-Distance between two shapes. A set of coarsely sampled landmarks (extremal points of a geodesic integral) were used to compute a geodesic diffeomorphism in [Tung 2010, Zhang 2008]. In [Ruggeri 2010] a thresholding of critical

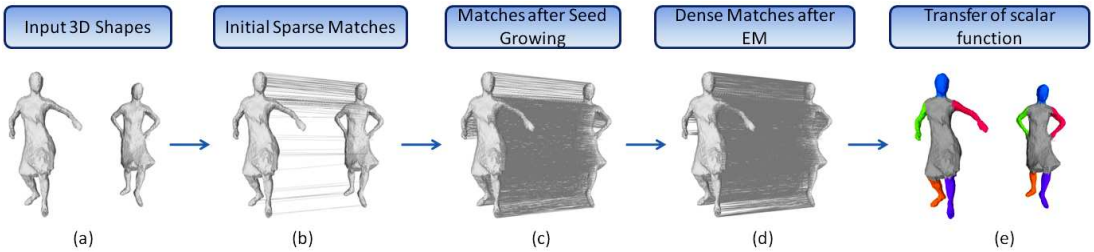


Figure 7.2: Method outline : (a) Input shapes (notice the difference in their topology). (b) Initial sparse matches. (c) Matches obtained with the seed-growing algorithm. (d) Probabilistic dense matches obtained with the EM algorithm. (e) Probabilistic transfer of a scalar function (coloring of shape parts).

points of the Laplace-Beltrami operator were used to compute a set of anchor-points of a shape. These critical-points of Laplace-Beltrami operator are typically located on geometrically and topologically meaningful regions of the shape and are invariant with respect to isometry [Reuter 2009]. A recent shape matching method in [Tevs 2011] propose to first find optimized landmark points and match them for entropy minimization thereby minimizing the sampling costs. However, majority of these methods show results on graphical shapes and are less suitable for visual shapes, *e.g.*, the detection of similar critical points of Laplace-Beltrami operator on two shapes that was used in [Ruggeri 2010] is not reliable for noisy visual shapes.

The closest work to our method in terms of the shape descriptors is [Ovsjanikov 2010]. In their work, a dense shape matching method using a single (or multiple) initial correspondence is used. They propose a detailed theoretical justification for using heat diffusion maps. However, the proposed descriptor, when used in conjunction with a greedy matching method, is vulnerable to the topological issues. In our work, instead, we use a robust seed growing approach for matching, which locally propagate sparse matches and is robust to outliers in the initial correspondences.

7.1.2 Contributions

The main contribution of this chapter is a dense 3D shape matching method that is robust to topological changes in the shape. The method starts by finding *sparse* one-to-one correspondences between set of key-points and produces as output a set of *dense* correspondences. Figure 7.2 sketches the pipeline of the proposed method. Given two input visual shapes with topological changes, Figure 7.2(a), we first obtain sparse matching, Figure 7.2(b). Next, we employ the multi-scale descriptors (introduced in Chapter 6) to perform a dense shape matching. We match these descriptors locally using a novel *seed-growing* method that propagates current correspondences to nearby ones, Figure 7.2(c). The final set of dense correspondences is obtained via a point registration method that uses a variant of the EM algorithm, Figure 7.2(c). A

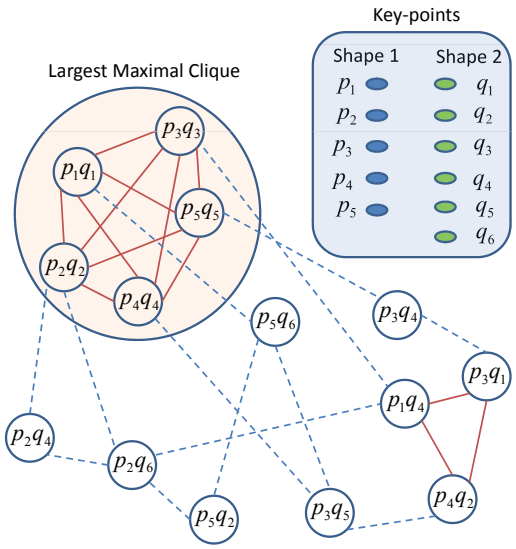


Figure 7.3: Association graph structure: Each node represents a point-to-point binary matching and every edge represents the mutual consistency between two nodes typically measured by positive weights. The edges with low consistency score are depicted with dashed line as opposed to the ones with high consistency score. In the ideal case, only the largest maximal clique should have a complete subgraph structure as shown here.

segmentation transfer introduced in Chapter 5 is employed in order to visualization the dense probabilistic matching results, Figure 7.2(d).

7.2 Sparse Shape Matching

In this section, we present sparse shape matching methods. We outline an association graph method to find one-to-one matching of the geometrical features, followed by a texture based sparse matching method for visual shapes obtained from the same sequence.

7.2.1 Sparse Matching using Geometrical Features

We have already introduced a geometrical feature detection technique for key-point detection in Section 6.6.2. The method uses multi-scale analysis for detection of sparse key-points on visual shapes that typically corresponds to geometrical features on shape protrusions.

Here, we employ the existing graph matching techniques in order to find a one-to-one matching of key-points (geometrical features) on two visual shapes. We have already mentioned in Section 2.4 that the graph matching methods are mostly NP-complete problems and

finding a global optimum is sometime not feasible. Nevertheless, several sub-optimal solutions are available.

One can cast the key-point matching matching problem into an integer quadratic problem (IQP), a *NP-hard* problem, by finding a set of mutually compatible nodes in an association graph [Bolles 1982b], which is equivalent to finding the largest maximal clique of the undirected weighted graph. Other sub-optimal solutions use various possible relaxations of IQP, *e.g.*, spectral relaxation method [Leordeanu 2005, Sharma 2010a], replicator equation method [Pelillo 1999] and re-weighted random-walk method [Cho 2010]. These methods are tractable when the correspondence set is small, namely, of the order of 10^2 and hence they are only suitable when one seeks sparse matching.

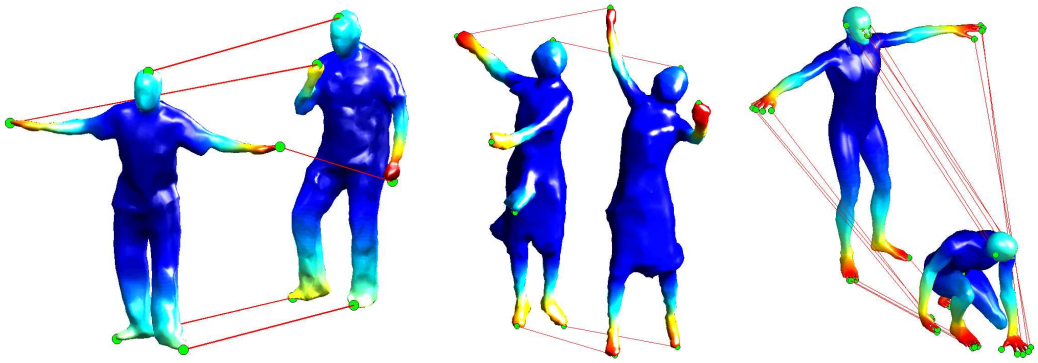
We represent our sparse graph matching problem in the framework of the *association-graph* introduced in [Bolles 1982a]. Each node of the association graph represents a point-to-point matching and every (positive weighted) edge between two nodes represents the mutual (isometric) consistency between respective point-to-point matching. Specifically, two mutually consistent graph nodes are called as strongly connected (*i.e.*, have high edge weights) if the corresponding key-points onto the two shapes have similar spectral distances.

Figure 7.3 illustrates the construction of *association-graph*. The *association-graph* typically has a complete graph structure but for the simplicity of visualization, we have only depicted strong edges and some weak edges. A subset of strongly connected nodes, shown in the big circle (and tagged as the largest maximal clique), represents the largest mutually consistent set of point-to-point matches that are only weakly connected to the other nodes.

Thus, a set of mutually consistent point-to-point matching can be obtained by finding the subset of strongly connected nodes of the *association-graph*. This is an instance of the largest maximum clique problem, which is an *NP-hard* problem. There are many algorithms that find an approximate solution by enforcing different constraints [Pelillo 1999, Leordeanu 2005, Cho 2010]. In [Leordeanu 2005], a spectral relaxation to integer quadratic formulation, followed by one-to-one matching constraints was employed to find a sub-optimal solution. [Cho 2010] obtains a approximate solution by simulating the random walks with re-weighting jumps, thus enforcing the matching constraints on the association graph.

We adapt the energy-minimization framework for the graph isomorphism problem based on an equivalent maximum clique formulation presented in [Pelillo 1999]. One-to-one key-point matching problem is formulated as the quadratic program using the adjacency matrix of association graph and solved using the replicator equations. The method is employed to find one-to-one matching of the key-points between two shapes.

Sparse Matching Results: Figure 7.4 shows the results of key-point matching obtained using this method. It is interesting to note that method provide a decent result for the case when two shapes have different number of key-points, since a new shape protrusion evolved due to dynamic clothing as shown in the Figure 7.4(b).

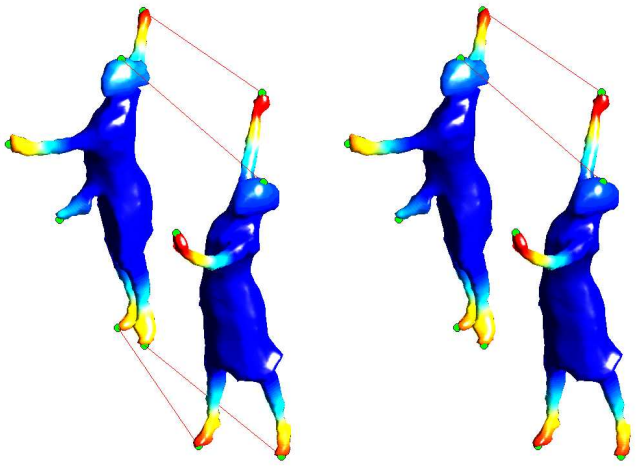


(a) Visual shapes from different acquisition setup.

(b) Visual shapes from same sequence.

(c) Graphical shapes.

Figure 7.4: Sparse key-point matching obtained with association graph method. Interestingly, the method provide a decent result for the case when two shapes have different number of key-points since a new shape protrusion evolved due to dynamic clothing in (b).



(a) (b)

Figure 7.5: Sparse key-point matching obtained with the replicator equation method. Results in (a) and (b) represent different maximal cliques obtained with replicator method for the same association graph.

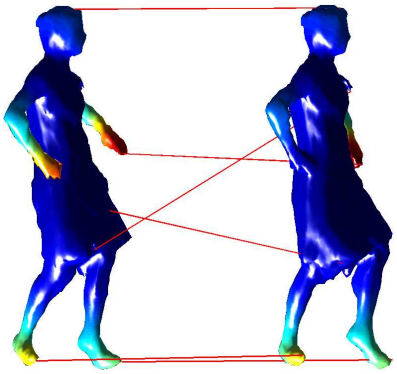


Figure 7.6: A failed key-point matching due to the presence of topological issues (hand on the right shape is merged with the torso).

Limitations of Association Graph Method: One important limitation of the association graph based feature matching methods is the scalability, since the computational cost of these methods grows exponentially with the number of key-points. However, it is common to perform the heuristic elimination of the association graph nodes that corresponds to less probable one-to-one matching in order to address the scalability issue. This can be done by using the extrinsic or intrinsic properties of the local surface around key-points, *e.g.*, texture, curvature, heat diffusion. We propose to compare and threshold the normalized multi-scale auto diffusion values for every possible matching in order to eliminate the weak matching hypothesis and corresponding association graph nodes. Another crucial limitation of the proposed method is that it can find key-point matching solution only up to the symmetry in the given shapes.

While employing the replicator equation method to find the approximate solution to maximum clique of the association graph, one major limitation is that it does not guarantee a global consistency in the matching results. The replicator method can only find one of the many maximal clique since finding the largest maximum clique is not practical. This can lead to different matching solutions for the same data. Figure 7.5 depicts a typical scenario when two different matching solutions are obtained with the replicator method.

Another major limitation of the current method (as well as the majority of existing methods) is that it does not handle the presence of topological discrepancies like merging and splitting in the 3D shapes. This is due to the fact that the spectral distances are sensitive to any large topology change. Figure 7.6 shows a incorrect key-point matching result where two shapes differ in the topology due to merging of hand with torso in one of the shape.

7.2.2 Sparse Matching using Texture based Features

We have already introduced a sparse feature assignment method in the previous section. However, the proposed sparse matching method is susceptible to the presence of topological changes

since the spectral distances used to define the mutual compatibility score of the matching is no more reliable. There are other methods proposed in the past that use local geometry/texture cues to find a set of sparse correspondences [Zaharescu 2009, Ahmed 2008, Thorstensen 2009, Hou 2010].

We adapt a SIFT descriptor [Lowe 2004] based sparse shape matching method proposed in [Ahmed 2008] in order to obtain sparse shape matching in case of visual shapes from the same sequence and with large topological changes.

7.3 Dense Shape Matching

In this section, we present a dense matching method for visual shapes using heat-kernel framework outlined in the previous chapter. Using the sparse matching method presented in the previous section, we can obtain sparse one-to-one correspondences between sets of key-points on two visual shapes. Next, using the same set of key-points, we can define multi-scale descriptors (see Section 6.6.3) for every vertex of the two shape graphs. Hence, given the matched key-point, we can now easily compute the descriptor matching score presented in Section 6.6.4.

Once we have two sets of dense multi-scale heat-kernel descriptors, similar to [Ovsjanikov 2010], we can straightforwardly employ a greedy matching algorithm in order to find dense matching results. This might work really well for graphical shapes where heat diffusion on two shapes is almost the same. However, while dealing with visual shapes, it is not easy to make such an assumption specifically when two shape can also be significantly different in their topology, thus making the heat-kernel descriptors unreliable. Therefore, in case of visual shapes a greedy matching can lead to erroneous arbitrary assignments that can violet the smoothness of the desired correspondence map.

In such scenario, we prefer a sparse-to-dense approach where we start with initial sparse matches and locally propagate them so as to preserve the smoothness of the binary matching. Traditionally, seed-growing algorithms are used to achieve such a sparse to dense matching [Čech 2010].

7.3.1 Correspondence Propagation using Seed Growing

Seed-growing is an interesting class of dense matching method that starts with a set of initial matches and grows them in a local neighborhood. In our case, we already have a set of sparse correspondences, which can be easily used as the seed matches. Hence, we propose to use the dense multi-scale heat-kernel descriptors and a seed-growing algorithm, similar in spirit to the one proposed in [Čech 2010], for propagating sparse key-point correspondences over visual shapes.

Let's define a set of binary variables $\Gamma = \{\gamma_1, \ldots, \gamma_i, \ldots, \gamma_n\}$ for shape $\mathcal{G}$ where γ_i is set to 1 if a vertex v_i from the first shape is assigned to a vertex v'_j from the second shape and 0

otherwise. Similarly, we can define Γ' for $\mathcal{G}'$. Let $\mathcal{S} = \{s_1, \ldots, s_m, \ldots, s_M\}$ be the set of initial seed matches, with $s_m = (v_i, v'_j)$, *e.g.*, $\gamma_i = \gamma'_j = 1$. $\mathcal{S}$ in stored in a priority queue data structure where each seed correspondence s_m is associated with a matching score. The matching score $score(s_m)$ is the heat descriptor matching error computed between the descriptors of v_i and v'_j (see Section 6.6.4). Each time an element is drawn from $\mathcal{S}$, it returns the seed with the minimum matching error score. Initially, the scores of all the seed correspondence are set to zero. This will ensure that all the seed matches are part of the output binary matching. We define a matching threshold β and consider only the correspondences with a matching score less than this threshold. We designate by $Nei(v_i)$ the set of 2-ring neighbors of vertex v_i on the shape graph. The output of our algorithm is the set of accepted binary matches represented as $\Delta = \{\delta_{ij}\}$.

The proposed seed-growing algorithm proceeds as follows. We iteratively draw a seed match with the minimum matching error from $\mathcal{S}$ and accept it as a correct binary match (add to set Δ) if the constituting vertices are not yet assigned to an existing match in Δ. Otherwise, we drop this seed correspondence from S. Once a seed correspondence is accepted, the algorithm searches for all the neighboring vertices of this seed correspondence on each shape that are not yet assigned to any existing binary matches in Δ. It then computes the matching score between every pair of corresponding vertex descriptors and adds the current pair to priority queue structure $\mathcal{S}$ if the matching score is less than β. Pseudo code of the seed-growing method is outlined in Algorithm 6.

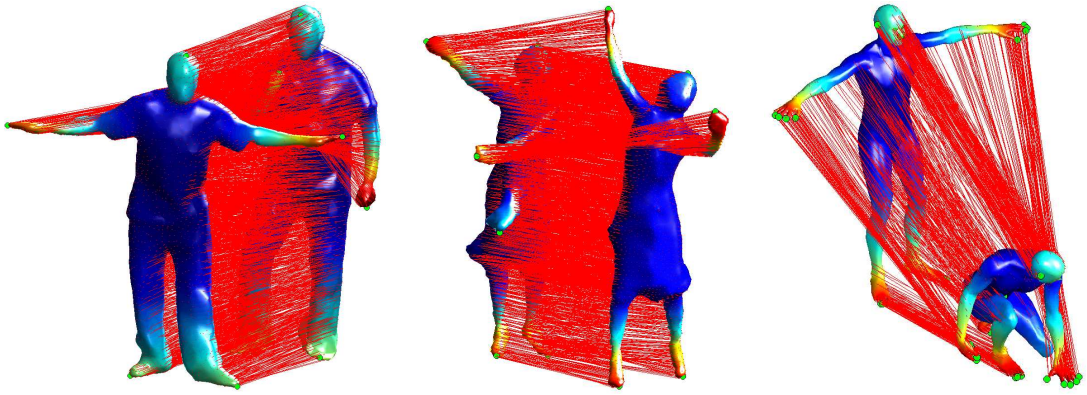
Algorithm 6 *Seed-Growing for Match Propagation*

Input: : Two sets of vertex descriptors $\mathbf{D}$, $\mathbf{D}'$; seed matches $\mathcal{S}$; a set of binary variables Γ, Γ'; the matching score threshold β.

Output: : Dense matches $\Delta = \{\delta_{ij}\}$ where $\delta_{ij} = (v_i, v'_j)$.

while $\mathcal{S}$ *is not empty* **do**
  Draw the seed $s \in \mathcal{S}$ with the minimum matching error, $s = (v_i, v'_j)$.
  if $\gamma_i = \gamma'_j = 0$ **and** $score(s) < \beta$ **then**
    $\Delta = \Delta \cup \{(v_i, v'_j)\}$ and set $\gamma_i = \gamma'_j = 1$.
    for each $v_a \in Nei(v_i)$ **and** $\gamma_a = 0$ **do**
      $v'^{*} = \underset{v'_k \in Nei(v'_j)\ \gamma'_{v'_k}=0}{\operatorname{argmin}} score(v_a, v'_k)$.
      if v'_* exists **and** $score(v_a, v'_*) < \beta$ **then**
        $\mathcal{S} = \mathcal{S} \cup \{(v_a, v'_*)\}$.
      end if
    end for
  end if
end while

The proposed algorithm is partially robust to the initial outlier seeds as those seeds have a low score and will not be propagated due to the priority queue structure and provision of



(a) Visual shapes from different acquisition setup.

(b) Visual shapes from same sequence.

(c) Graphical shapes.

Figure 7.7: Dense 3D shape matching results with the proposed multi-scale heat diffusion descriptor computed using the set of sparse key-points shown in the Figure 7.4.

threshold. However, these outlier matches will still exist as part of output matches as we assign them a zero matching error score. The output can also have some unmatched points if two shapes have different sampling, which is the usual case with the visual shapes or when the matching error score is below some threshold.

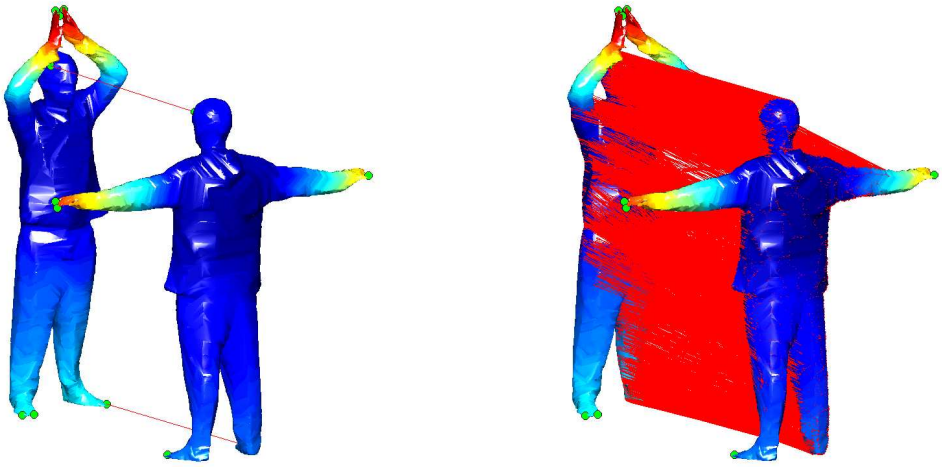
Another major problem is in the case of visual shapes with topological changes where the local heat diffusion properties will not be the same in the areas of topological merging/splitting. This will lead to unmatched set of vertices on two shapes.

Correspondence Propagation Results: We show dense binary matching results for visual shapes using the proposed seed growing algorithm and multi-scale heat distance descriptor. We choose the unit hyper-sphere embedding presented in Section 6.5.1 and the associated diffusion distance metric in Eq. (6.43), while computing the heat distance descriptors. It is important to note that the analysis presented in Section 6.4 allow us to choose the minimum time scale depending on the size of Laplacian embedding of one shape. It also enables us to use the same time scale while analyzing other shapes by selecting the appropriate embedding dimensions. We heuristically chose 25 as the embedding dimension, *i.e.*, K for the first shape and then compute minimum time scale and corresponding K' for the other shape. We heuristically set the value of τ to 5, thereby computing a 5-dimensional vector for each descriptor, capturing the multi-scale heat distances from each key-point. Once the minimum time scale value represented as t_1 is computed by the scree-curve analysis, we choose the remaining scale parameters by logarithmically increasing time scales values.

Figure 7.7, shows the dense matching obtained with multi-scale heat distance descriptors, computed using the corresponding sparse key-points and their binary matching shown in the

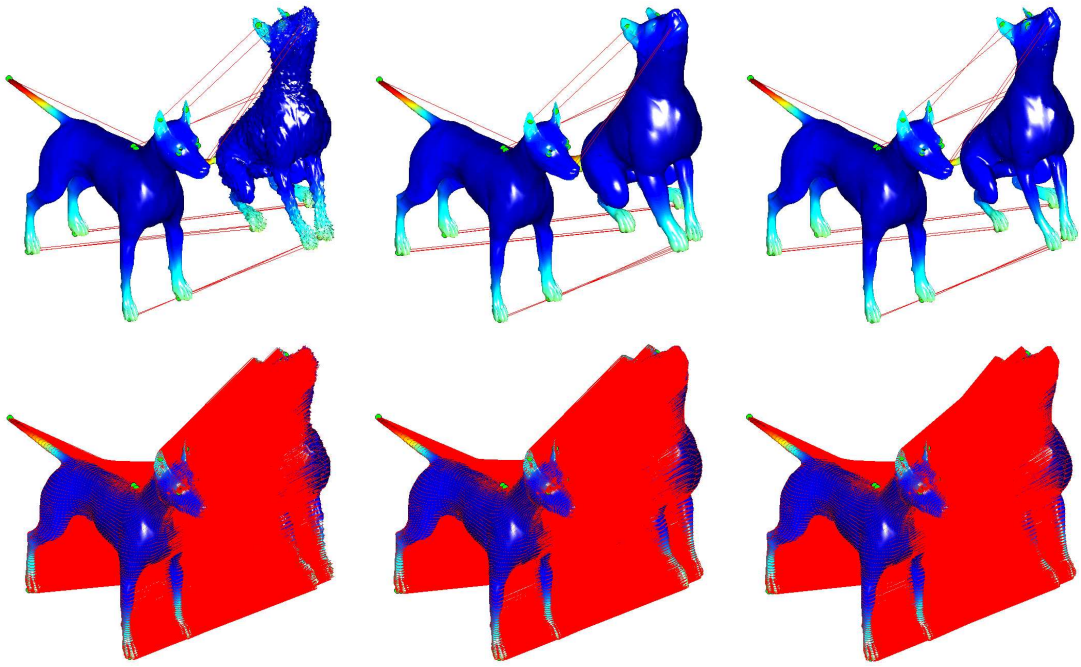
(a) Sparse key-point matching. (b) Dense binary matching.

Figure 7.8: Dense matching of processed high-resolution visual shape with topology change.



(a) Sparse key-point matching. (b) Dense binary matching.

Figure 7.9: Dense matching of unprocessed low-resolution visual shape.



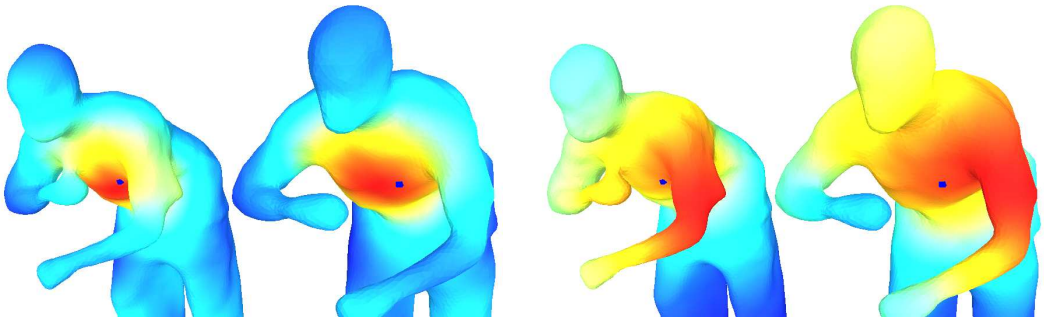
(a)Gaussian noise (b) Scale change (c) Micro holes

Figure 7.10: Sparse key-point matching (top row) and corresponding dense 3D shape matching (bottom row) in the presence of different shape transforms [Bronstein 2010a]. In case of Gaussian noise (a) and scale change (b), we obtained accurate dense binary matching as the two shape graphs have the same topology, *i.e.*, same number of vertices and edges.

Figure 7.4. It is important to note that first two results in the figure are on visual data. In the first case shown in the Figure 7.7(a), two shapes are from two different acquisition setups and thus have completely different meshing. In the second result shown in the Figure 7.7(b), two shapes have different number of protrusions due to dynamic clothing (much clearly visible in Figure 7.4(b)).

In Figure 7.10, we show dense matching results in the presence on different shape transforms available in the SHREC'10 shape benchmark [Bronstein 2010a]. First two results in Figure 7.10(a,b) correspond to an exact vertex-to-vertex binary match, *i.e.*, the proposed method was able to find the original permutation of vertex indices. However, this was the ideal case where two synthetic shapes have an exact same topology. In case of micro holes, the key-point matching is locally failed around the ears, resulting a small matching error in that region. This failure can be explained by the fact that spectral distances at small scales are very sensitive to surface noise like holes and hence the replicator equation method (while finding maximal clique of resulting association graph) finds a locally sub-optimal solution leading to a wrong key-point matching.

A partial matching result was obtained for unprocessed low-resolution shape shown in



(a) (b)

Figure 7.11: Visualization of heat diffusion on 3D shapes. The color at each vertex v_i encodes the value of the kernel $h(i, j; t)$, where v_j is the blue dot on the torso. (a): For small values of t, the heat diffusion map is very similar on both shapes and it is not affected by their topological differences (hand merging with body). (b): For large values of t, the behavior of the diffusion process drastically depends on the shape's topology.

the Figure 7.9. This is mainly because of challenging triangulation of the shape as well as due to the initial matching of only 2 key-points. In Figure 7.8, we show sparse key-point matching and matching for processed high resolution visual shape with topological changes due to merging of hands. Although, the key-point matching using association graph yielded only one match (because of the change in topology), the heat-distance descriptor based dense matching was able match large part of shape, *i.e.*, legs, head and one hand, however, only up to the symmetry in two shapes.

In this section, we have presented dense matching results on visual shapes, obtained with seed-growing algorithm. We were able to find completely unsupervised sparse to dense matching of many visual shapes. However, we can infer from the qualitative results that the proposed method did not performed well on shapes with topological changes. In the next section, we propose to use more local heat diffusion descriptors, coupled with seed-growing and a dense probabilistic matching method to overcome this problem.

7.4 Topologically-Robust Dense Matching

We have already mentioned in the previous section that the proposed seed growing method fails to perform on visual shape with topological changes as the global heat diffusion behavior on two shapes is no more similar. This is mainly because the heat diffusion behavior changes in the presence of topological merging/splitting. Figure 7.11 depicts the heat diffusion phenomenon on visual shapes with topological merging. For small values of the time scale parameter t, the heat diffusion is limited to a local neighborhood (Figure 7.11.a), whereas for large values of t, the heat diffusion is fairly global (Figure 7.11.b). Therefore, at small scales, the

heat diffusion behaves similarly across the two topologically different shapes, while at larger scales, the behavior is affected by the topological discrepancies between the shapes. This motivates the choice of a local descriptor based on heat diffusion at small scales.

Hence, in order to match two topologically different shapes, similar to dense matching attempt in the previous section, we compute dense multi-scale heat diffusion descriptor (see Section 6.6.3.1) for each vertex of the shape graph and employ seed-growing algorithm to propagate initial sparse matches. It is important to note that, first we intentionally use smaller time scales while computing the multi-scale heat diffusion descriptor and second we use the sparse binary matching obtained with texture features (see Section 7.2.2). This should provide robustness to topological changes as now the heat diffusion on shape graphs should be fairly local and hence similar on two visual shapes, except in the region of topological changes. Also, the initial sparse correspondence should be invariant to topological changes due to photometric consistency across the sequence. This however make the application of this method limited to matching visual shapes from the sequence. Nevertheless, these correspondences are very sparse as compare to the sampling of visual shapes and any other sparse feature matching method or even manual input can be used if they can provide such initial matches.

Interestingly, match propagation with seed-growing algorithm is more relevant in this scenario as our heat diffusion descriptor are more local and hence a local propagation algorithm should be preferable than a global algorithm like greedy matching. Nevertheless, even a local propagation algorithm like seed-growing can at best provide only partial matching due to large deformation in heat diffusion behavior in the region of topological changes. Hence, we require a more robust method that can take as a input the partial matching output of seed-growing and provide dense matching results.

7.4.1 Dense Probabilistic Matching with EM

Here we outline a probabilistic method, which takes as input a sparse set of binary correspondences between the two visual shapes and provides as output a dense set of correspondences.

The method is based on a parametric probabilistic model, namely, maximum likelihood with missing data (see 2.8.2 and [Horaud 2011]). Let us consider the Laplacian embeddings of two shapes, *i.e.*, Eq. (6.24) with $t = 0$: $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{n}, \mathbf{X}' = \{\mathbf{x}'_j\}_{j=1}^{n'}$, with $\mathbf{X}, \mathbf{X}' \subset \mathbb{R}^k$, where $k \ll \min\{n, n'\}$ is the common dimension of the two embeddings, $3 \leq k \leq 10$ in our experiments. Without loss of generality, we assume that the points in the first set, $\mathbf{X}$ are cluster centers of a Gaussian mixture model (GMM) with n clusters and an additional uniform component that accounts for outliers and unmatched data. The matching $\mathbf{X} \leftrightarrow \mathbf{X}'$ will consist in fitting the Gaussian mixture to the set $\mathbf{X}'$.

Let this Gaussian mixture undergo a $k \times k$ transformation $\mathbf{Q}$ with $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_k, \det(\mathbf{Q}) = \pm 1$, more precisely $\mathbf{Q} \in O(k)$, the group of orthogonal matrices acting on $\mathbb{R}^k$. Hence, each cluster in the mixture is parametrized by a prior p_i, a cluster mean $\mu_i = \mathbf{Q}\mathbf{x}_i$, and a covariance matrix $\mathbf{\Sigma}_i$. It will be assumed that all the clusters in the mixture have the same priors, $\{p_i =$

$\pi_{in}\}_{i=1}^{n}$, and the same isotropic covariance matrix, $\{\mathbf{\Sigma}_i = \sigma \mathbf{I}_k\}_{i=1}^{n}$. This parametrization leads to the following *observed-data log-likelihood* (with $\pi_{out} = 1 - n\pi_{in}$ and $\mathcal{U}$ is the uniform distribution):

$$P(\mathbf{X}') = \sum_{j=1}^{n'} \log \left(\sum_{i=1}^{n} \left(\pi_{in} \mathcal{N}(\mathbf{x}'_j | \mu_i, \sigma) \right) + \pi_{out}\, \mathcal{U} \right) \tag{7.1}$$

It is well known that the direct maximization of (7.1) is not tractable and it is more practical to maximize the *expected complete-data log-likelihood* using the EM algorithm, where "complete-data" refers to both the observed data (the points $\mathbf{X}'$) and the missing data (the point-to-point assignments). In our case, this expectation writes (see [Horaud 2011] for details):

$$\mathcal{E}(\mathbf{Q}, \sigma) = -\frac{1}{2} \sum_{j=1}^{n'} \sum_{i=1}^{n} \alpha_{ji} (\|\mathbf{x}'_j - \mathbf{Q}\mathbf{x}_i\|^2 + k \log \sigma), \tag{7.2}$$

where α_{ji} denotes the posterior probability of an assignment: $\mathbf{x}'_j \leftrightarrow \mathbf{x}_i$:

$$\alpha_{ji} = \frac{\exp(-\|\mathbf{x}'_j - \mathbf{Q}\mathbf{x}_i\|^2 / 2\sigma)}{\sum_{q=1}^{n} \exp(-\|\mathbf{x}'_j - \mathbf{Q}\mathbf{x}_q\|^2 / 2\sigma) + \emptyset \sigma^{k/2}}, \tag{7.3}$$

where $\emptyset$ is a constant term associated with the uniform distribution $\mathcal{U}$. Notice that one easily obtains the posterior probability of a data point to remain unmatched, $\alpha_{jn+1} = 1 - \sum_{i=1}^{n} \alpha_{ij}$. This leads to the dense matching procedure outlined in Algorithm 7.

Algorithm 7 *Dense matching with EM*

Input: : Two embedded shapes $\mathbf{X}$ and $\mathbf{X}'$;

Output: : Dense correspondences $\Phi : \mathbf{X} \mapsto \mathbf{X}'$ between the two shapes;

1: *Initialization:* Set $\mathbf{Q}^{(0)}$ and $\sigma^{(0)}$;

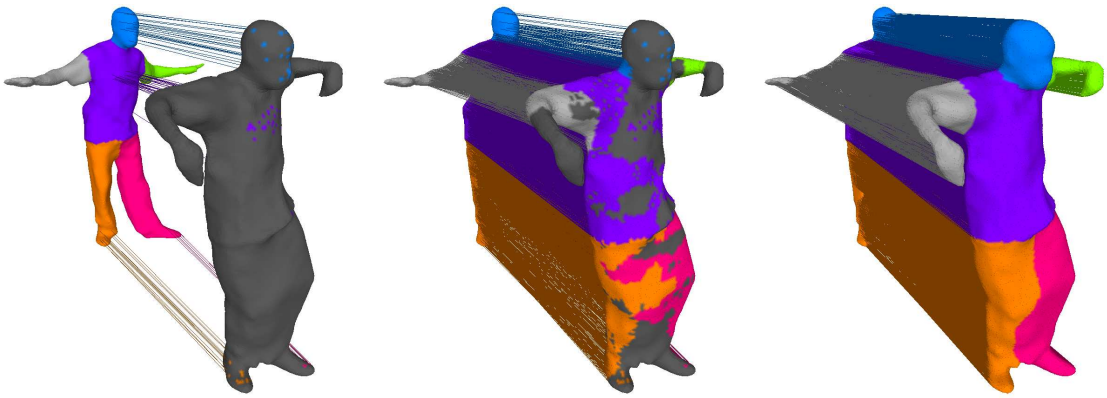
2: *E-step:* Compute the posteriors $\alpha_{ij}^{(q)}$ using (7.3);

3: *M-step:* Estimate the transformation $\mathbf{Q}^{(q)} = \arg\min_{\mathbf{Q}} \sum_{i,j} \alpha_{ij}^{(q)} \|\mathbf{x}'_j - \mathbf{Q}\mathbf{x}_i\|^2$ and the variance $\sigma^{(q)} = \sum_{i,j} \alpha_{ij}^{(q)} \|\mathbf{x}'_j - \mathbf{Q}^{(q)}\mathbf{x}_i\|^2 / k \sum_{i,j} \alpha_{ij}^{(q)}$

4: *Assignment:* Match $\mathbf{x}_i \mapsto \mathbf{x}'_j$ if $\max_i \alpha_{ij}^{(q)} > 0.5$, *i.e.*, $\Phi(i) = j$.

However, the proposed EM algorithm can be easily trapped in a local minimum and the final result crucially depends on correct initialization. Hence, we use the dense binary matching obtained with seed propagation algorithm to initialize our probabilistic dense matching algorithm. Here we assume that the initial sparse matches have enough binary correspondence including the correspondences around the region of topological change in order to find a decent initial estimate of $k \times k$ transformation matrix $\mathbf{Q}$.

The algorithm is robust to those few bad initial sparse matches that were not propagated by the seed-growing algorithm, but still exist in its output. This is because the large number of good correspondences would bias the initialization of the EM algorithm and the probabilistic output matching will completely ignore those bad initial matches.



(a) (b) (c)

Figure 7.12: Dense shape matching: (a) Initial sparse matches, (b) Matches obtained with seed growing, (c) Final matching after EM.

Nevertheless, we agree that the complete absence of sparse initial correspondences around the region of large topological changes can lead to inconsistent matching results.

7.5 Experiments & Results

In this section, we present the qualitative and quantitative evaluation of the dense matching results obtained using the proposed method on visual shapes with significant topological changes. We also presents a comparison with existing dense matching method proposed in [Mateus 2008] an [Ovsjanikov 2010].

While implementing [Ovsjanikov 2010], we computed HKS descriptors using the same set of parameters that were used for our method. As mentioned before, the greedy algorithm for descriptor matching is not suitable for matching the local HKS descriptors that are computed with smaller time scale parameters and particularly, on shapes with topological changes, as it does ensure the neighborhood consistency of the binary matches. Thus, in order to make a fair comparison with [Ovsjanikov 2010], we used our seed growing algorithm (see Section 7.3.1) with HKS descriptors (see Section 6.6.3.1) to find a dense binary matching solution. In practice we use $\tau = 5$ and set $\{t_1, \ldots, t_5\} = \{0, 20, 40, 80, 100\}$ while comping heat diffusion descriptor.

Similarly, we implemented EM algorithm of [Mateus 2008] using the same set of parameters used for our method. Since, both our method and method proposed in [Mateus 2008] compute a probabilistic dense matching solution, we obtain dense matching by thresholding the maximum probability matches and selecting only those matches that have posterior probability value larger than a threshold value of 0.50 (see Algorithm 7). Additionally, we also

impose a binary (one-to-one) matching constraint since the EM algorithm estimates many-to-one probabilistic assignments between shape graph vertices. However, we use the original may-to-one probabilistic matching result for the purpose of segmentation transfer between two shapes (see Algorithm 5).

7.5.1 Qualitative Evaluation

In Figure 7.12, we show *wide-time-frame* matching obtained in the presence of topological merging. For the purpose of visualization, we color code the body parts of one shape and we transfer the corresponding vertex-to-part labels to the other shape, as proposed in 5. This makes use of the fact that EM outputs a posterior probability for each vertex of one shape to be matched with each vertex of the second shape.

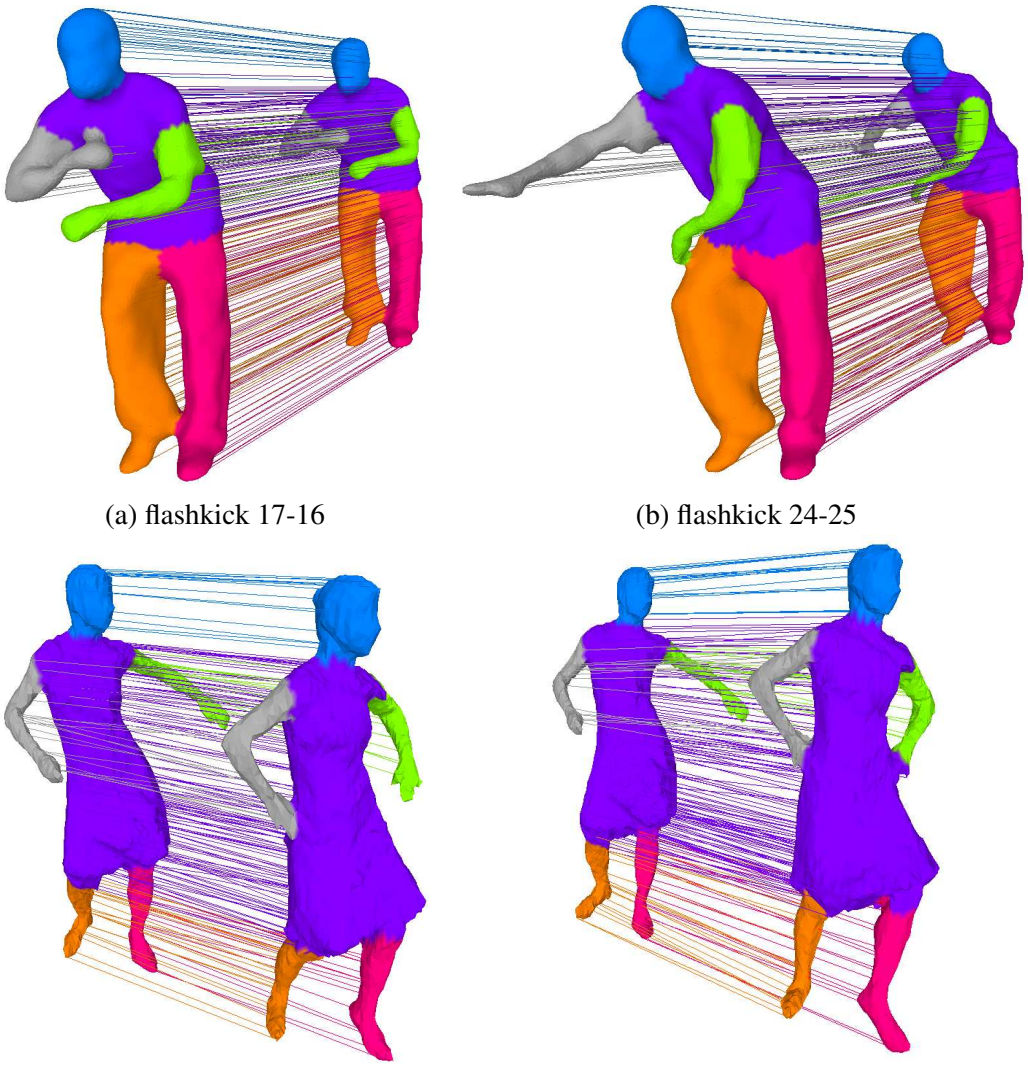
Figure 7.14 shows the results of matching obtained with our method and with two other two methods. The matching based on histograms of Laplacian eigenvectors is not reliable when the two shapes have different topologies and hence the dense matching method presented in [Mateus 2008] fails to provide good results, *e.g.*, Figure 7.14(b). As mentioned earlier, greedy matching [Ovsjanikov 2010] does not consider the neighborhood consistency of matches and leads to wrongly matched patches on shapes, *e.g.*, Figure 7.14(c). We also show our results on another challenging visual data in Figure 7.15. This is an interesting results since the rope is very thin as well as broken at places and we only use one initial match in the rope region. A vertex to vertex color transfer is used for visualization of the dense matching.

However, we do not claim to completely solve such a challenging problem. Sometime the method might yield inconsistent matching results if we do not use enough embedding dimensions in the case when the region of topological change is relatively small and difficult to handle, *i.e.*, Figure 7.16. Nevertheless, our method can still provide a relatively correct matching.

7.5.2 Quantitative Evaluation

In this section, we present our attempt to perform a quantitative evaluation of the proposed dense matching method and compare it with other two methods proposed in [Mateus 2008, Ovsjanikov 2010].

The quantitative evaluation of visual shape matching is indeed a difficult task given the lack of ground truth data. We believe that an evaluation on existing graphical shapes is not a fair measure as these methods are specifically designed to handle large deformations in the data. The qualitative results presented in Section 7.5.1, show dense matching using a probabilistic segmentation transfer method (see Algorithm 5) as well as the down-sampled binary matches obtained by thresholding the maximum probabilistic matches. Here, we conduct two set of experiments to evaluate these two aspects.



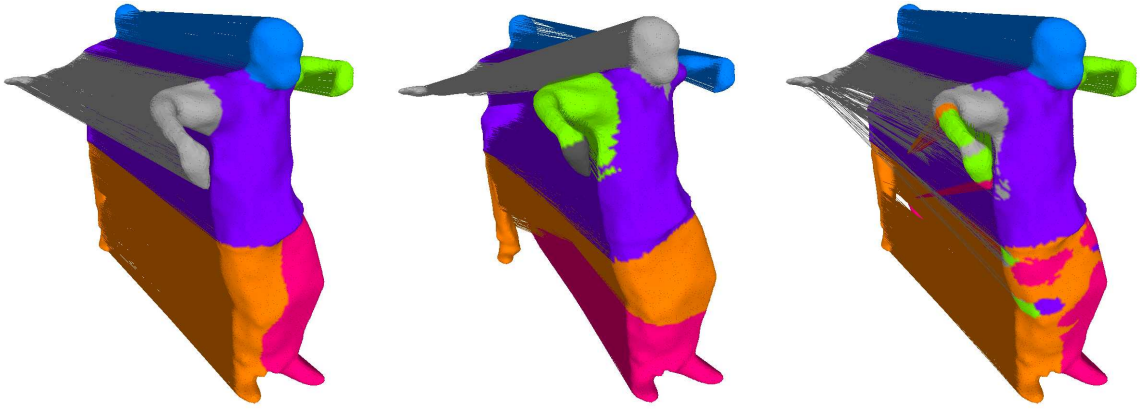
(a) flashkick 17-16

(b) flashkick 24-25

(c) Samba 118-121

(d) Samba 118-123

Figure 7.13: Shape matching results in the presence of topological issues. Only 1% of the total matches are actually shown for ease of visualization.



(a) Proposed method. (b) [Mateus 2008]. (c) [Ovsjanikov 2010] (greedy algorithm).

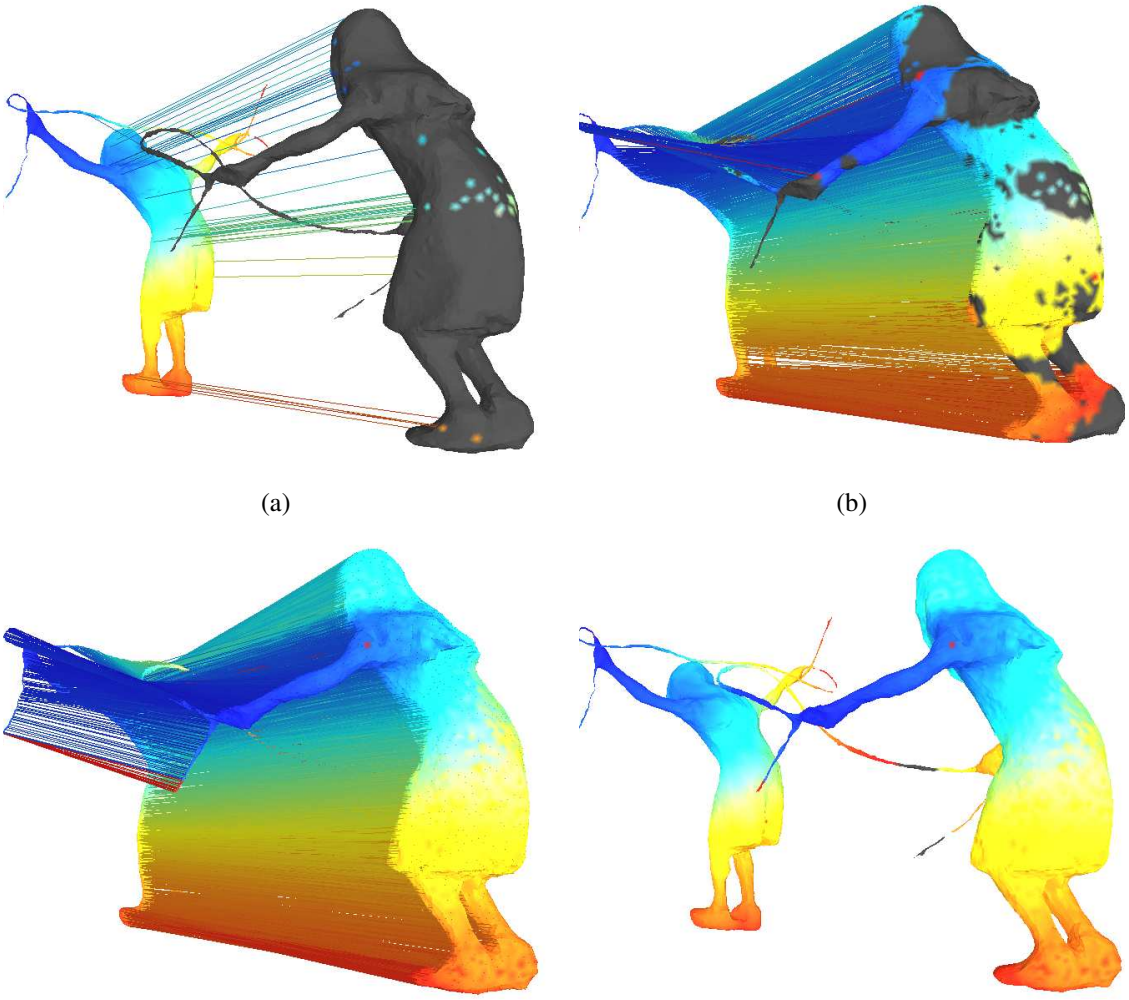
Figure 7.14: Comparison with two existing methods. The colors encode the body-part labels transferred from a segmented shape, *i.e.*, Figure 7.12(a).

In the first experiment, similar to [Mateus 2009], we perform a smoothness evaluation of the binary correspondence map. A smooth binary correspondence map $\Phi : \mathcal{G} \mapsto \mathcal{G}'$ between the two shape graphs $(\mathcal{G}(\mathcal{V},\mathcal{E}), \mathcal{G}'(\mathcal{V}',\mathcal{E}'))$ would ensure that the adjacent vertices $(v_i \sim v_j) \in \mathcal{E}$ should map to neighboring vertices on $\left(v'_{\Phi(i)}, v'_{\Phi(j)}\right) \in \mathcal{E}'$. We need to relax this condition of adjacent neighborhood to a more general neighborhood structure in order to deal with challenging visual shapes with non-uniform sampling and complex topological changes.

Hence, we define r-neighborhood around each vertex v_i by finding the set of vertices that are closest to a given vertex in the local geodesic sense and represent it as $Nei_r(v_i)$. A good approximation of geodesic distances can be computed using the fast marching algorithm proposed in [Sethian 1996]. We use the MATLAB Fast Marching Toolbox[1] to compute the approximate geodesic distances on visual shapes. Hence, we compute in advance, the geodesic distance of each vertex to all the vertices on a shape graph. Unlike [Mateus 2009], we propose to use a relative neighborhood definition in order to handle the sampling variation in visual shapes. Thus, the r-neighborhood of a vertex can be computed by selecting r percentage of total number of vertices $|\mathcal{V}|$ that has the shortest geodesic distance to the given vertex. This definition of neighborhood is locally robust to topological changes in the shape except if the vertex is part of the region where topological change exist.

Once we have a neighborhood for each vertex, we define a binary match $\left(v_i, v'_{\Phi(i)}\right)$ associated with each $v_i \in V$ to be smooth if all the neighboring vertex of v_i are mapped to the

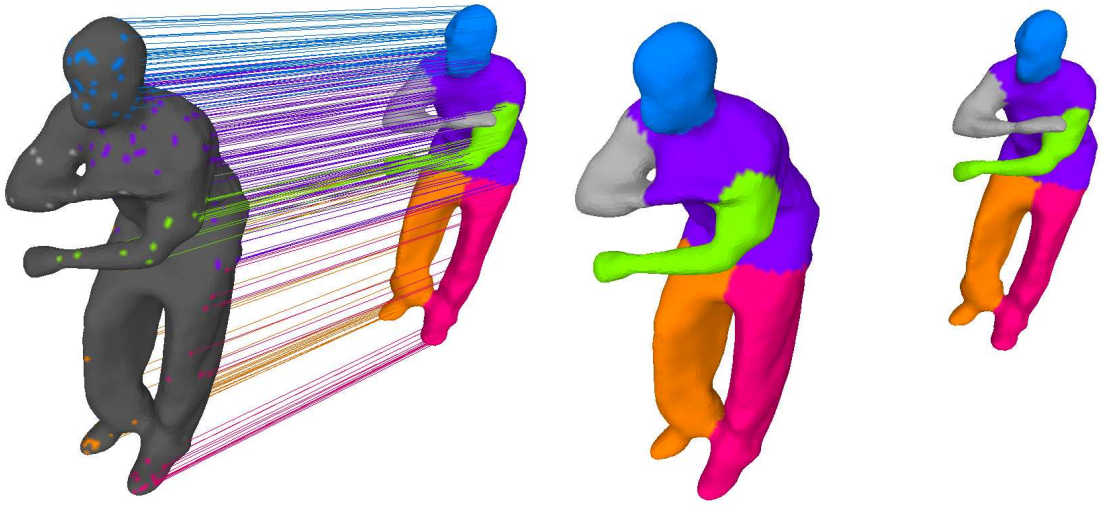
[1] http://www.mathworks.com/matlabcentral/fileexchange/6110



(a) (b)

(c) (d)

Figure 7.15: Dense shape matching: (a) Initial sparse matches, (b) Matches obtained with seed growing, (c) Final matching after EM, (d) Vertex-level color transfer. Only one initial match was provided on the thin rope.



(a) (b)

Figure 7.16: Dense shape matching: (a) Initial sparse matches, (d) Probabilistic color transfer. Clearly, the right hand was only partial matched even though there were some initial matches around it.

neighbors of $v_{\Phi(i)}$. Formally,

$$r\text{-smooth}(i,\Phi(i)) = \begin{cases} 1 & \text{if } \exists \left(v_j, v'_{\Phi(j)}\right) \quad \forall v_j \in Nei_r(v_i), \ \ \forall v'_{\Phi(j)} \in Nei_r(v'_{\Phi(i)}), \\ 0 & \text{otherwise.} \end{cases} \tag{7.4}$$

We can further relax this definition by considering partial smoothness in order to extend the resolution of smoothness measure, *i.e.*, r-smooth$_\theta(i,\Phi(i))$, by considering θ percentage of neighbors instead of all the neighbors. This can be helpful to handle some binary matches obtained by thresholding the matches of similar probability values.

Hence, the smoothness of a correspond map Φ can be computed as:

$$r\text{-smooth}_\theta(\Phi) = \frac{\sum_i r\text{-smooth}_\theta(i,\Phi(i))}{|\mathcal{V}|}. \tag{7.5}$$

However, r-smooth$_\theta(\Phi)$ is a directional quantity, derived only for one shape among the two matched shape and hence not a symmetric metric for measuring the quality of matching. Nevertheless, this can be a decent measure to assess the smoothness of visual shape matching in the absence of ground truth data.

Table 7.1, summarizes the mesh size ($n = |V|$ and $n' = |V'|$), the initial number of anchor correspondences $|\mathcal{A}|$, the number of matches obtained with our seed-growing algorithm and the total number of binary matches obtained from EM algorithm ($|\Phi|$).

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{V}'\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\Delta\rvert$	$\lvert\Phi\rvert$
flashkick 117-130	12041	12656	91	7118	11065
flashkick 024-025	12231	12282	410	10214	12046
flashkick 016-017	12006	12005	379	10191	11599
samba 118-123	4284	4226	167	2641	4211
samba 118-121	4284	4254	200	2432	4210
lucie 340-358	6403	6309	114	3758	6046

Table 7.1: Data-set details: Number of vertices of the input meshes ($\lvert\mathcal{V}\rvert$ and $\lvert\mathcal{V}'\rvert$), the initial number of anchor correspondences ($\lvert\mathcal{A}\rvert$), the number of correspondences found with the seed-growing algorithm ($\lvert\Delta\rvert$) and the total number of binary matches obtained from EM algorithm ($\lvert\Phi\rvert$).

Table 7.2 presents the r-smooth measure, computed for various dense matching results, some of which shown in the figures in previous section. In this work, we consider $\theta = 90\%$ and r-neighborhoods for different values of $r = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.8\}$. The entries of Table 7.2 concludes that the proposed method consistently outperform other two methods as far as the smoothness of dense matching is concerned. However, it is important to note that the smoothness is a desired property for a correspondence map but not enough to classify a map as good or bad. This is because we only consider smoothness of individual matches and it is possible to have a locally smooth but globally incorrect correspondence mapping, yielding large number of smooth matches. Our qualitative assessment shown in the previous section provides the complementary information in this regard.

An interesting observation is that the smoothness measure for some dense matching results obtained using HKS with seed growing is relatively closer to the proposed method (*e.g.*, *flashkick 025-024* and *flashkick 017-016*). This behavior is understandable as in these visual shape pairs, the topological changes are relatively small and hence the output of seed growing algorithm yield large number of binary matches (see Table 7.1). Here, we would like to recall that we use seed growing with HKS instead of greedy matching for a fair comparison with [Ovsjanikov 2010], which otherwise yield much lower performance due to lack of smoothness in greedy matching. The existence of some zero entries in Table 7.2 is due the fact that the corresponding neighborhood size is very small (*e.g.*, 4) and no binary match showed smoothness with respect to such smaller neighborhoods.

Nevertheless, we would like to note that the proposed method also achieves a relatively low smoothness score. We believe the reason behind is that we use smaller embedding size (*e.g.*, $k \leq 10$) while performing the probabilistic matching in Algorithm 7. Hence, only global information about the shape structure is retained while the local information is lost leading to a less accurate binary assignments in a local neighborhood around each vertex.

In the second experiment, we performed quantitative evaluation of the segmentation transfer results obtained by the probabilistic dense matching output of our method and the methods proposed in [Mateus 2008]. We first independently created ground-truth segmentation for

Visual Shapes	r	r-smooth$_\theta(\Phi)$		
		[Ovsjanikov 2010]	[Mateus 2008]	Proposed Method
flashkick 117-130	0.1	8.56	15.93	36.33
	0.2	14.09	20.70	44.30
	0.3	16.47	24.64	53.25
	0.4	17.22	26.28	55.75
	0.5	19.21	28.25	60.14
	0.8	20.43	30.10	65.10
flashkick 025-024	0.1	24.63	16.95	38.60
	0.2	38.96	22.60	52.51
	0.3	50.40	27.02	64.67
	0.4	52.41	28.56	65.16
	0.5	58.99	30.53	71.85
	0.8	64.88	31.66	78.88
flashkick 017-016	0.1	30.47	12.79	39.17
	0.2	48.09	15.65	54.70
	0.3	59.70	18.68	67.56
	0.4	65.41	19.75	70.82
	0.5	68.13	20.49	73.10
	0.8	74.27	22.87	82.83
samba 118-123	0.1	00.00	00.00	00.00
	0.2	07.60	11.61	24.37
	0.3	08.39	15.54	37.00
	0.4	11.65	17.51	47.70
	0.5	10.57	17.79	49.26
	0.8	14.45	20.13	55.82
samba 118-121	0.1	00.00	00.00	00.00
	0.2	07.05	10.53	25.71
	0.3	07.28	13.28	38.62
	0.4	08.43	16.31	45.08
	0.5	09.52	17.58	51.26
	0.8	12.24	20.24	55.76
lucie 340-358	0.1	07.65	10.52	30.64
	0.2	09.01	12.87	36.01
	0.3	10.04	13.71	38.62
	0.4	13.05	17.66	42.44
	0.5	12.82	20.53	49.69
	0.8	16.80	23.81	58.06

Table 7.2: Comparison of the smoothness measure (r-smooth$_\theta(\Phi)$) for dense binary matching results on various visual shape pairs. These dense binary matching were obtained by applying the proposed method and two other existing methods: HKS with seed growing [Ovsjanikov 2010] and EM based probabilistic matching [Mateus 2008] for varying neighborhood sizes, *i.e.*, r.

Visual shapes	Proposed Method		[Mateus 2008]	
	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$	$\bar{m}^{tpr}$	$\bar{m}^{ppv}$
flashkick 117-130	0.9134	0.9400	0.2765	0.2488
flashkick 025-024	0.9548	0.9807	0.2259	0.1647
flashkick 017-016	0.9815	0.9919	0.4791	0.4846
samba 118-123	0.9574	0.9845	0.1308	0.1138
samba 118-121	0.9705	0.9876	0.3829	0.4521

Table 7.3: Comparison of segmentation transfer results to asses the quality of dense probabilistic matching using the quantitative evaluation measure, namely, average recall ($\bar{m}^{tpr}$) and precision ($\bar{m}^{ppv}$) as introduced in Section 5.5.

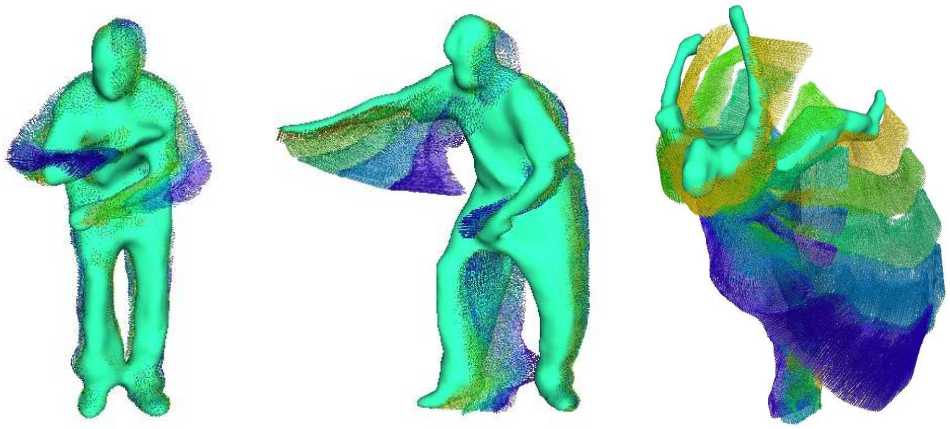
visual shapes and then use the segmentation transfer measurement entities, namely, average recall ($\bar{m}^{tpr}$) and precision ($\bar{m}^{ppv}$) as introduced in Section 5.5 to asses the quality of our results. Table 7.3 summarizes our findings. One can easily infer that the proposed method has higher average recall and precision as compare to other methods. This explains the difference in quality of dense matching in Figure 7.14(a,b). However, we agree that the method proposed in [Mateus 2008] does not use the extra information we have in terms of seed matches as well as their work does not claim to handle visual shapes with topological changes.

7.5.3 Dense Trajectory Results

We have also computed dense binary matching over a sequence of independently reconstructed shapes from [Starck 2007b] using the proposed method, *with no manual intervention*. We use the trajectory plotting software developed in [Petit 2011] to show the frame-wise dense matching of visual shapes. Thus, we do not claim to perform any vertex-level tracking in these results.

Figure 7.17 shows the dense trajectories of the corresponding binary matching of the past five consecutive frames (shapes) for the listed visual shapes. Interestingly, the proposed topologically-robust dense matching method provides continuous dense trajectories even in the presence of large topological change, *e.g.*, Figure 7.17(a-b). Similarly, Figure 7.18 shows the dense trajectories for the nine continuous frames of the flashkick sequence. We can clearly see that the dense matching trajectories are quite stable even though during these frames, it happens twice that the left hand merges with the torso and then again splits.

Data-set The matching results were obtained on publicly available graphical shapes from TOSCA data-set [Bronstein 2008] as well as the visual shapes from INRIA [Franco 2009], University of Surrey [Starck 2007b] (the *flashkick* sequence) and MIT data-set [Vlasic 2008] (the *samba* dance sequence). In case of MIT data-set, we use a simple voxel carving algorithm to compute a visual hull represented as a mesh.



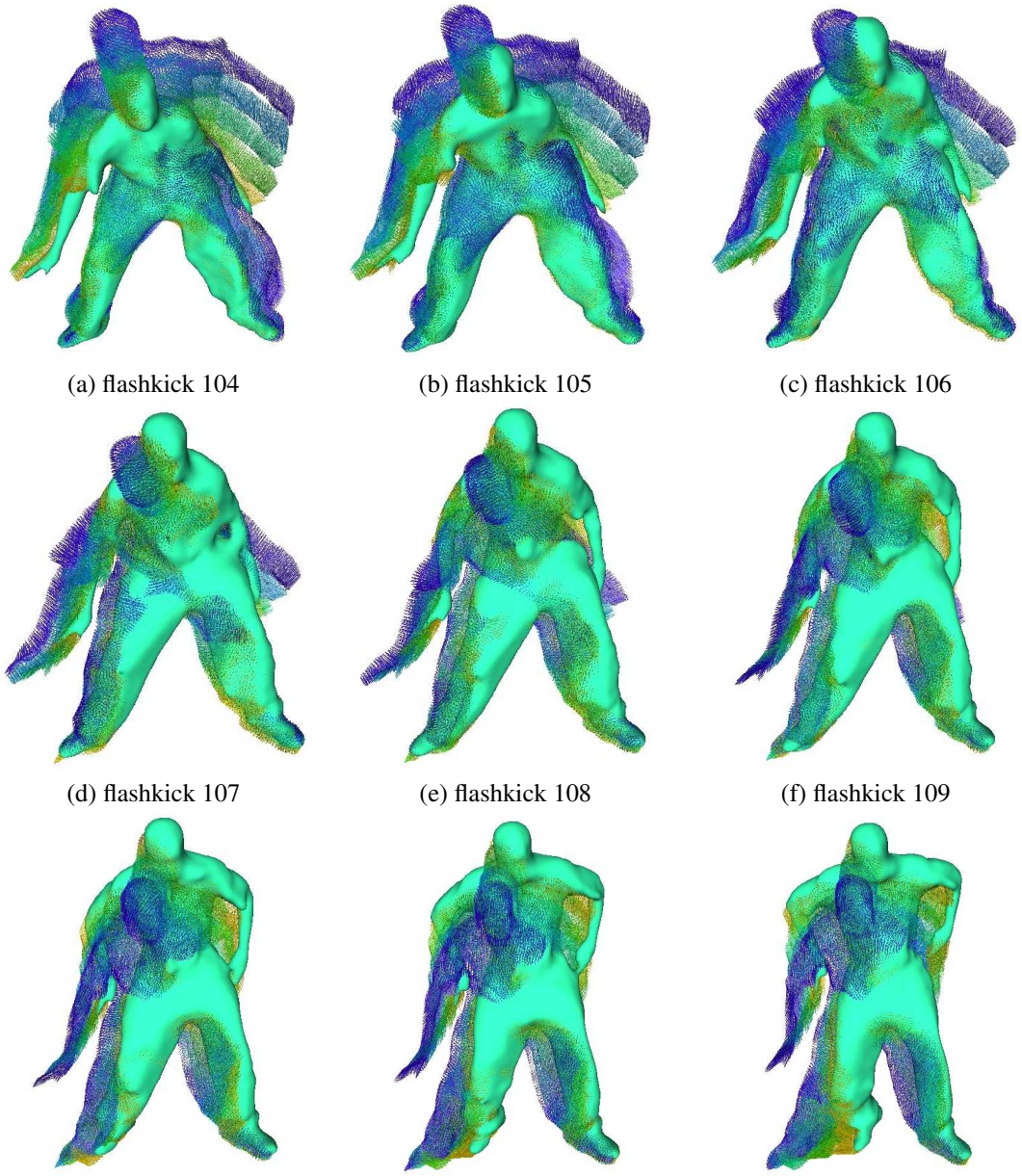
(a) flashkick 019 (b) flashkick 029 (c) flashkick 067

Figure 7.17: Trajectories of the dense binary shape matching over the past five frames in the flashkick sequence. Only the vertex trajectories of the pairwise frames are shown. No explicit vertex-level tracking was performed. The transition of colors from blue to green while plotting the dense trajectories represents the forward traversal in the temporal dimension.

7.6 Conclusion

We proposed a dense shape matching method using a sparse key-point matching association graph method, set of multi-scale heat-kernel descriptors, a seed-growing algorithm that locally propagates only the good matches and a variant of the EM algorithm that eventually registers the two shapes. The key feature of our method is that it considers an intrinsic scale-space representation based on the heat-kernel. This provides a principled framework for defining descriptors at small scales and for robustly propagating correspondences locally, in spite of topological changes. We have shown decent matching results on shapes in the presence of large topological merging. Our method can be used to perform dense shape registration, which stands as the basis of transferring any scalar function defined one shape, to the other one. However, as pointed out earlier, one major limitation of the proposed method is that it requires reliable initial sparse matches, specifically around the area of topological changes. Otherwise, method could yield locally incorrect matching in those regions.

As part of the future work, it will be interesting to employ local heat-kernel descriptors in a dynamic 3D environment where any prior assumption about shape topology is not valid. Additionally, automatic detection and correction of topological issues will be an important direction to explore. It will also be interesting to better understand the EM matching algorithm in the context of multi-scale shape representation as well as the dimension of the spectral embeddings.



(a) flashkick 104 (b) flashkick 105 (c) flashkick 106

(d) flashkick 107 (e) flashkick 108 (f) flashkick 109

(g) flashkick 110 (h) flashkick 111 (i) flashkick 112

Figure 7.18: Trajectories of the dense binary shape matching over the past five frames in the sequence, for nine consecutive frames of the flashkick sequence. Only the vertex trajectories of the pairwise frames are shown. No explicit vertex-level tracking was performed. The transition of colors from blue to green while plotting the dense trajectories represents the forward traversal in the temporal dimension.

CHAPTER 8

Conclusion and Future Work

Contents

The technological advancement in 3D acquisition is leading us to a new era with vast amount of 3D data. However, majority of such data, which we called as the visual shapes, are very challenging as compare to the traditional graphical shapes used in the compute graphics community. Hence, it is difficult to use the traditional shape analysis methods for visual shapes that were primarily designed to deal with graphical shapes.

In this thesis, we have tried to address this issue by proposing new methods that uses spectral graph theory and probabilistic tools to robustly handle visual shapes. In particular, we devised new segmentation and registration techniques for articulated 3D shapes represented as point clouds or meshes by single and multi-scale shape analysis.

8.1 Comprehensive Summary

The derivation of spectral representation for 3D shapes brings together concepts from spectral graph theory, non-linear dimensionality reduction and geometry processing. We adapted Laplacian eigenmap method to isometrically embed a 3D shape represented as sparse and regularly connected shape graph into a subspace spanned by the eigenspace of graph Laplacian matrix. This allowed us to analyze articulated 3D shapes in a pose invariant subspace by assuming the articulated poses as the quasi-isometric deformations. In such space, a shape can be represented as d-dimensional point cloud. Such a representation enabled the application of ideas from machine learning techniques in shape analysis. In particular, we extensively used ideas from spectral clustering for shape segmentation and existing expectation maximization based solutions for point cloud registration.

First, we introduced a novel completely unsupervised visual shape segmentation algorithm based on the PCA interpretation of the Laplacian eigenvectors of the shape graph and on parametric clustering using Gaussian mixtures. We analyzed the geometric properties of these

vectors and we devise a practical method that combines single-vector analysis with multiple-vector analysis. We attempted to characterize the projection of the graph onto each one of its eigenvectors based on PCA properties of the eigenvectors. We devised an unsupervised probabilistic method, based on *one-dimensional* Gaussian mixture modeling with model selection, to reveal the structure of each eigenvector. Based on this structure, we selected a subset of eigenvectors among the set of the smallest non-null eigenvectors and we embedded the shape graph into the isometric space spanned by this selection of eigenvectors. The final clustering was performed via unsupervised classification based on learning a *multi-dimensional* Gaussian mixture model of the embedded graph to obtain segmentation labels.

Next, we proposed a semi-supervised segmentation solution for visual shapes that enabled the use of minimal user interaction to drive the segmentation. We proposed a spectral learning approach to shape segmentation. The method was composed of a new *constrained spectral clustering* algorithm that was used to supervise the segmentation of a shape from a training data set, followed by a *probabilistic label transfer* algorithm that was used to match two shapes and to transfer cluster labels from a training-shape to a test-shape. The novelty resided both in the use of the Laplacian embedding to propagate must-link and cannot-link constraints and in the segmentation algorithm based on a learn, align, transfer, and classify paradigm.

This was followed by the outline of a generalized heat-kernel framework for multi-scale analysis of visual shapes. The representation of visual shapes as undirected weighted graphs allowed us to analyze the heat-kernel within the framework of spectral graph theory, construct heat-kernel matrices well suited for visual shapes, and to represent the latter into the metric space associated with the spectral decomposition of this matrix. We provided a detailed mathematical analysis of various spectral constructs useful for shape representation. We proposed to combine the scale parameter of the heat kernel and dimensionality of the spectral representation, providing a basis for performing dimensionality reduction and, more generally, to characterize the statistical properties of the embedded shape representation at multiple scales. In addition to this, we also introduced a novel multi-scale heat distance descriptor using the proposed heat-kernel framework. The proposed descriptor characterize a point on 3D shape using the scale dependent diffusion metric and a set of sparsely detected key-points.

Finally, we proposed a new shape matching method that is robust to complex topological changes in visual shapes, *e.g.*, merging and splitting of parts. We proposed to combine the multi-scale heat-kernel descriptors (computed at smaller time scales) with sparse matches obtained by a association graph based matching method in a sparse-to-dense propagation approach in order to obtain a densified 3D shape matching for visual shapes. The algorithm starts from a sparse set of *seed* matches and outputs dense matching. At small scales the heat diffusion descriptors behaves locally and hence it is robust to global changes in topology. Therefore, it can be used to build a vertex-to-vertex matching score conditioned by an initial correspondence set. This score was then used to iteratively add new correspondences based on a novel *seed-growing* method that iteratively propagates the seed correspondences to nearby vertices. The matching was further densified via an EM-like method that explores the congruency between the two shape embeddings. The algorithm initialize with output of

seed-growing and yields dense probabilistic matching for visual shapes.

8.2 Contributions

The aim of this thesis was to study the analysis of 3D articulated shapes represented as visual shapes and develop methods that can deal with challenges posed by visual shapes. We mainly investigated single and multi-scale representation and analysis of visual shapes in the context of shape segmentation and registration tasks. The main contributions of this thesis are as follows.

1. A pose invariant spectral representation of articulated visual shapes using the Laplacian eigenvectors of corresponding shape graph (Chapter 3).
2. An unsupervised method for 3D shape segmentation that combines characterization of individual Laplacian eigenvectors with Gaussian mixture model to obtain a pose invariant segmentation of articulated visual shapes (Chapter 4).
3. A semi-supervised segmentation algorithm, based on a learn, align, transfer, and classify paradigm that uses a user guided spectral clustering method to seek a desired segmentation of few visual shapes followed by a probabilistic label transfer to classify a new shape for different segment labels (Chapter 5).
4. A generalized heat-kernel framework for multi-scale representation and analysis of visual shape (Chapter 6).
5. An empirical analysis to find the appropriate minimum scale of analysis for a given embedding representation (Section 6.4.1).
6. A novel heat-kernel descriptor using sparse key points and heat-diffusion metric (Section 6.6.3.2).
7. A dense shape matching method by combining heat-kernel descriptor with novel seed-growing method (Section 7.3.1).
8. A topologically-robust dense shape matching method that start with output of seed-growing method and use EM algorithm to obtain a probabilistic dense matching (Section 7.4).

8.3 Future Work

In this document, we have extensively studied single and multi-scale spectral representation and analysis of 3D articulated shapes and proposed solutions for unsupervised and semi-supervised shape segmentation as well as dense shape registration that are more suitable to visual shapes with large acquisition noise and topological changes.

Nevertheless, there are many aspects that need more understanding efforts and many questions that need clearer answers. One such question can be framed as "what could be the optimal embedding dimension". This needs to be answered in the context of task at hand. Another important open question is the correct scale of analysis for a given task. Although, our empirical analysis showed that these two questions can be related, this needs a better theoretical understanding. We have repeatedly used EM based point registration algorithm for dense matching. However, few aspects like it's sensitivity t0 embedding dimensions as well as scale of analysis needs further investigation. Another important direction to pursue is to deal with more generic 3D data with multiple objects interacting with each other. We believe that heat-kernel based local representation can be key while deal with with such complex data.

APPENDIX A

Appendix

A.1 Permutation and Doubly-stochastic Matrices

A matrix $\mathbf{P}$ is called a *permutation* matrix if exactly one entry in each row and column is equal to 1, and all other entries are 0. Left multiplication of a matrix $\mathbf{A}$ by a permutation matrix $\mathbf{P}$ permutes the *rows* of $\mathbf{A}$, while right multiplication permutes the *columns* of $\mathbf{A}$.

Permutation matrices have the following properties: $\det(\mathbf{P}) = \pm 1$, $\mathbf{P}^\top = \mathbf{P}^{-1}$, the identity is a permutation matrix, and the product of two permutation matrices is a permutation matrix. Hence the set of permutation matrices $\mathbf{P} \in \mathcal{P}_n$ constitute a subgroup of the subgroup of orthogonal matrices, denoted by O_n, and $\mathcal{P}_n$ has finite cardinality $n!$.

A non-negative matrix $\mathbf{A}$ is a matrix such that all its entries are non-negative. A non-negative matrix with the property that all its row sums are $+1$ is said to be a *(row) stochastic matrix*. A *column stochastic matrix* is the transpose of a row stochastic matrix. A stochastic matrix $\mathbf{A}$ with the property that $\mathbf{A}^\top$ is also stochastic is said to be *doubly stochastic*: all row and column sums are $+1$ and $a_{ij} \geq 0$. The set of stochastic matrices is a compact convex set with the simple and important property that $\mathbf{A}$ is stochastic if and only if $\mathbf{A}\mathbb{1} = \mathbb{1}$ where $\mathbb{1}$ is the vector with all components equal to $+1$.

Permutation matrices are doubly stochastic matrices. If we denote by $\mathcal{D}_n$ the set of doubly stochastic matrices, it can be proved that $\mathcal{P}_n = O_n \cap \mathcal{D}_n$ [Zavlanos 2008]. The permutation matrices are the fundamental and prototypical doubly stochastic matrices, for Birkhoff's theorem states that any doubly stochastic matrix is a linear convex combination of finitely many permutation matrices [Horn 1994]:

Theorem 4 *(Birkhoff) A matrix $\mathbf{A}$ is a doubly stochastic matrix if and only if for some $N < \infty$ there are permutation matrices $\mathbf{P}_1, \ldots, \mathbf{P}_N$ and positive scalars $s_1, \ldots, s_N$ such that $s_1 + \ldots + s_N = 1$ and $\mathbf{A} = s_1\mathbf{P}_1 + \ldots + s_N\mathbf{P}_N$.*

A complete proof of this theorem is to be found in [Horn 1994][pages 526–528]. The proof relies on the fact that $\mathcal{D}_n$ is a compact convex set and every point in such a set is a convex combination of the extreme points of the set. First it is proved that every permutation matrix is an extreme point of $\mathcal{D}_n$ and second it is shown that a given matrix is an extreme point of $\mathcal{D}_n$ if an only if it is a permutation matrix.

A.2 The Frobenius Norm

The Frobenius (or Euclidean) norm of a matrix $\mathbf{A}_{n\times n}$ is an *entry-wise* norm that treats the matrix as a vector of size $1 \times nn$. The standard norm properties hold: $\|\mathbf{A}\|_F > 0 \Leftrightarrow \mathbf{A} \neq 0$, $\|\mathbf{A}\|_F = 0 \Leftrightarrow \mathbf{A} = 0$, $\|c\mathbf{A}\|_F = c\|\mathbf{A}\|_F$, and $\|\mathbf{A}+\mathbf{B}\|_F \leq \|\mathbf{A}\|_F + \|\mathbf{B}\|_F$. Additionally, the Frobenius norm is *sub-multiplicative*:

$$\|\mathbf{AB}\|_F \leq \|\mathbf{A}\|_F \|\mathbf{B}\|_F \tag{A.1}$$

as well as *unitarily-invariant*. This means that for any two orthogonal matrices $\mathbf{U}$ and $\mathbf{V}$:

$$\|\mathbf{UAV}\|_F = \|\mathbf{A}\|_F. \tag{A.2}$$

It immediately follows the following equalities:

$$\|\mathbf{UAU}^\top\|_F = \|\mathbf{UA}\|_F = \|\mathbf{AU}\|_F = \|\mathbf{A}\|_F. \tag{A.3}$$

A.3 Spectral Properties of the Normalized Laplacian

The normalized Laplacian Let $\tilde{\mathbf{u}}_k$ and γ_k denote the eigenvectors and eigenvalues of $\tilde{\mathbf{L}}$; The spectral decomposition is $\tilde{\mathbf{L}} = \tilde{\mathbf{U}}\mathbf{\Gamma}\tilde{\mathbf{U}}^\top$ with $\tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top = \mathbf{I}$. The smallest eigenvalue and associated eigenvector are $\gamma_1 = 0$ and $\tilde{\mathbf{u}}_1 = \mathbf{D}^{1/2}\mathbb{1}$.

We obtain the following equivalent relations:

$$\sum_{i=1}^n d_i^{1/2}\tilde{u}_{ik} = 0, \quad 2 \leq k \leq n \tag{A.4}$$

$$d_i^{1/2}|\tilde{u}_{ik}| < 1, \qquad 1 \leq i \leq n, 2 \leq k \leq n. \tag{A.5}$$

Using (2.66) we obtain a useful expression for the combinatorial Laplacian in terms of the spectral decomposition of the normalized Laplacian. Notice, however, that the expression below is NOT a spectral decomposition of the combinatorial Laplacian:

$$\mathbf{L} = (\mathbf{D}^{1/2}\tilde{\mathbf{U}}\mathbf{\Gamma}^{1/2})(\mathbf{D}^{1/2}\tilde{\mathbf{U}}\mathbf{\Gamma}^{1/2})^\top. \tag{A.6}$$

For a connected graph γ_1 has multiplicity 1: $0 = \gamma_1 < \gamma_2 \leq \ldots \leq \gamma_n$. As in the case of the combinatorial Laplacian, there is an upper bound on the eigenvalues (see [Chung 1997] for a proof):

Proposition 6 *For all $k \leq n$, we have $\mu_k \leq 2$.*

We obtain the following spectral decomposition for the normalized Laplacian :

$$\tilde{\mathbf{L}} = \sum_{k=2}^{n} \gamma_k \tilde{\mathbf{u}}_k \tilde{\mathbf{u}}_k^\top. \tag{A.7}$$

The spread of the graph along the k-th normalized Laplacian eigenvector is given by $\forall(k,i), 2 \leq k \leq n, 1 \leq i \leq n$:

$$\bar{\tilde{u}}_k = \frac{1}{n}\sum_{i=1}^{n}\tilde{u}_{ik} \tag{A.8}$$

$$\sigma_{u_k} = \frac{1}{n} - \bar{\tilde{u}}_k^2. \tag{A.9}$$

Therefore, the projection of the graph onto an eigenvector $\tilde{\mathbf{u}}_k$ is not centered. By combining (2.66) and (A.7) we obtain an alternative representation of the combinatorial Laplacian in terms of the the spectrum of the normalized Laplacian, namely:

$$\mathbf{L} = \sum_{k=2}^{n}\gamma_k(\mathbf{D}^{1/2}\tilde{\mathbf{u}}_k)(\mathbf{D}^{1/2}\tilde{\mathbf{u}}_k)^\top. \tag{A.10}$$

Hence, an alternative is to project the graph onto the vectors $\mathbf{t}_k = \mathbf{D}^{1/2}\tilde{\mathbf{u}}_k$. From $\tilde{\mathbf{u}}_{k\geq 2}^\top\tilde{\mathbf{u}}_1 = 0$ we get that $\mathbf{t}_{k\geq 2}^\top\mathbb{1} = 0$. Therefore, the spread of the graph's projection onto $\mathbf{t}_k$ has the following mean and variance, $\forall(k,i), 2 \leq k \leq n, 1 \leq i \leq n$:

$$\bar{t}_k = \textstyle\sum_{i=1}^{n} d_i^{1/2}\tilde{u}_{ik} = \quad 0 \tag{A.11}$$

$$\sigma_{t_k} = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} d_i\tilde{u}_{ik}^2. \tag{A.12}$$

The random-walk Laplacian. This operator is not symmetric, however its spectral properties can be easily derived from those of the normalized Laplacian using (2.68). Notice that this can be used to transform a non-symmetric Laplacian into a symmetric one, as proposed in [Sun 2009] and in [Luo 2009].

A.4 The Least-square Optimization

Many optimization problems that can be related to the *Rayleigh-Ritz* ratio [Horn 1994] are solved using the eigenvalues of Hermitian matrices. Given a $n \times n$ Hermitian matrix $\mathbf{M}$, with eigenvalues in increasing order of magnitude $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. The famous Rayleigh-Ritz theorem states that for a given n-dimensional vector $\mathbf{f}$ following holds:

$$\lambda_1\mathbf{f}^T\mathbf{f} \;\leq\; \mathbf{f}^T\mathbf{M}\mathbf{f} \;\leq\; \lambda_n\mathbf{f}^T\mathbf{f}. \tag{A.13}$$

Using the eigen-decomposition $\mathbf{M} = \mathbf{U}\Lambda\mathbf{U}^T$, we can rewrite $\mathbf{f}^T\mathbf{M}\mathbf{f}$ as:

$$\mathbf{f}^T\mathbf{M}\mathbf{f} = \mathbf{f}^T\mathbf{U}\Lambda\mathbf{U}^T\mathbf{f} = (\mathbf{U}^T\mathbf{f})^T\Lambda(\mathbf{U}^T\mathbf{f}) \tag{A.14}$$

$$\mathbf{f}^T\mathbf{M}\mathbf{f} = \sum_{i=1}^{n}\lambda_i\left((\mathbf{U}^T\mathbf{f})_i\right)^2 \tag{A.15}$$

Interestingly, we can write:

$$\sum_{i=1}^{n} \left((\mathbf{U}^T\mathbf{f})_i\right)^2 = \|\mathbf{U}^T\mathbf{f}\|^2 = \mathbf{f}^T\mathbf{f} \tag{A.16}$$

By dividing Eq. (A.13) with $\mathbf{f}^T\mathbf{f}$ and using Eq. (A.14) and Eq. (A.16), we can write:

$$\lambda_1 \leq \frac{\mathbf{f}^T\mathbf{M}\mathbf{f}}{\mathbf{f}^T\mathbf{f}} \leq \lambda_n. \tag{A.17}$$

From Eq. (A.17), we can deduce:

$$\lambda_{max} = \lambda_n = \max_{\mathbf{f}\neq\mathbf{0}} \frac{\mathbf{f}^T\mathbf{M}\mathbf{f}}{\mathbf{f}^T\mathbf{f}} = \max_{\mathbf{f}^T\mathbf{f}=1} \mathbf{f}^T\mathbf{M}\mathbf{f}, \tag{A.18}$$

$$\lambda_{min} = \lambda_1 = \min_{\mathbf{f}\neq\mathbf{0}} \frac{\mathbf{f}^T\mathbf{M}\mathbf{f}}{\mathbf{f}^T\mathbf{f}} = \min_{\mathbf{f}^T\mathbf{f}=1} \mathbf{f}^T\mathbf{M}\mathbf{f}. \tag{A.19}$$

Thus, the Rayleigh-Ritz theorem gives a variational characterization of the largest and smallest eigenvalues of a Hermitian matrix and gives solution to quadratic optimization problems of the form:

$$\max_{\mathbf{f}\neq\mathbf{0}} \frac{\mathbf{f}^T\mathbf{M}\mathbf{f}}{\mathbf{f}^T\mathbf{f}} \quad \text{and} \quad \min_{\mathbf{f}\neq\mathbf{0}} \frac{\mathbf{f}^T\mathbf{M}\mathbf{f}}{\mathbf{f}^T\mathbf{f}}. \tag{A.20}$$

in terms of these eigenvalues.

The intermediate eigenvalues can be characterized by the orthogonality constraints. Let $\mathbf{f}_1$ be the first eigenvector corresponding to λ_1, then using the constraints $\mathbf{f} \neq \mathbf{0}$ and $\mathbf{f} \perp \mathbf{f}_1$, allows to characterize $\mathbf{f}_2$ as:

$$\min_{\substack{\mathbf{f}\neq\mathbf{0} \\ \mathbf{f}\perp\mathbf{f}_1}} \frac{\mathbf{f}^T\mathbf{M}\mathbf{f}}{\mathbf{f}^T\mathbf{f}} = \min_{\substack{\mathbf{f}^T\mathbf{f}=1 \\ \mathbf{f}\perp\mathbf{f}_1}} \mathbf{f}^T\mathbf{M}\mathbf{f} = \lambda_2. \tag{A.21}$$

This result can be further extended to remaining eigenvectors, by always finding an orthogonal vector to current subspace. Similar analysis is possible for the max case. The Rayleigh-Ritz theorem is also applicable to complex matrices just by replacing the transpose with complex conjugate.

Bibliography

[Ahmed 2008] Naveed Ahmed, Christian Theobalt, Christian Rössl, Sebastian Thrun and Hans-Peter Seidel. *Dense Correspondence Finding for Parameterization-free Animation Reconstruction from Video*. In Computer Vision and Pattern Recognition, pages 1–8, 2008. (Cited on pages 120, 127 and 133.)

[Alpert 1999] C. J. Alpert, A. B. Kahng and S. Yao. *Spectral partitioning with multiple eigenvectors*. Discrete Applied Mathematics, vol. 90, no. 1-3, pages 3–26, 1999. (Cited on page 70.)

[Attene 2006] M. Attene, B. Falcidieno and M. Spagnuolo. *Hierarchical mesh segmentation based on fitting primitives*. Visual Computing, vol. 3, no. 22, pages 181–193, 2006. (Cited on page 70.)

[Belk 2011] Jim Belk. *Pullbacks and Isometries (Revised)*. http://math.bard.edu/belk/math352/, 2011. (Cited on page 22.)

[Belkin 2003] M. Belkin and P. Niyogi. *Laplacian eigenmaps for dimensionality reduction and data representation*. Neural computation, vol. 15, no. 6, pages 1373–1396, 2003. (Cited on pages 11, 35, 36, 39, 40, 45, 46, 57, 62, 70 and 81.)

[Belkin 2004] M. Belkin and P. Niyogi. *Semi-Supervised Learning on Riemannian Manifold*. Machine Learning, vol. 56, pages 1–8, 2004. (Cited on page 83.)

[Belkin 2008] M. Belkin, J. Sun and Y. Wang. *Discrete Laplace operator on meshed surfaces*. In Symposium on Computational Geometry, pages 278–287. ACM, 2008. (Cited on pages 62 and 99.)

[Belkin 2009] M. Belkin, J. Sun and Y. Wang. *Constructing Laplace operator from point cloud in R^d*. In ACM-SIAM Symposium on Discrete Algorithms, pages 1031–1040, 2009. (Cited on page 62.)

[Bérard 1994] P. Bérard, G. Besson and S. Gallot. *Embedding Riemannian Manifolds by their Heat Kernel*. Geometric and Functional Analysis, vol. 4, no. 4, pages 373–398, 1994. (Cited on pages 101 and 102.)

[Berthold 1999] Michael Berthold and David J. Hand, editeurs. Intelligent data analysis: An introduction. Springer-Verlag, 1999. (Cited on page 44.)

[Biasotti 2006] Silvia Biasotti, Simone Marini, Michela Spagnuolo and Bianca Falcidieno. *Sub-part correspondence by structural descriptors of 3D shapes.* Computer-Aided Design, vol. 38, no. 9, pages 1002–1019, 2006. (Cited on page 25.)

[Bilenko 2004] M. Bilenko, S. Basu and R. J. Mooney. *Integrating constraints and metric learning in semi-supervised clustering*. In International Conference on Machine Learning, pages 1–8, 2004. (Cited on pages 83 and 86.)

[Bishop 2006] C.M. Bishop. Pattern recognition and machine learning. Springer, 2006. (Cited on page 34.)

[Biyikoglu 2007] T. Biyikoglu, J. Leydold and P. F. Stadler. Laplacian eigenvectors of graphs. Springer, 2007. (Cited on pages 69, 71 and 73.)

[Boissonnat 1998] J.D. Boissonnat and M. Yvinec. Algorithmic geometry. Cambridge University Press, 1998. (Cited on page 4.)

[Bolles 1982a] R. C. Bolles and R. A. Cain. *Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method*. The International Journal of Robotics Research, vol. 1, pages 57–82, 1982. (Cited on pages 125, 126 and 130.)

[Bolles 1982b] R. C. Bolles and R. A. Cain. *Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method*. International Journal of Robotics Research, vol. 1, no. 3, pages 57–82, 1982. (Cited on page 130.)

[Borg 2005] I. Borg and P.J.F. Groenen. Modern Multidimensional Scaling: Theory and Applications. Springer, 2005. (Cited on page 34.)

[Borman 2004] Sean Borman. *The Expectation Maximization Algorithm: A short tutorial*. http://seanborman.com/publications/EM_algorithm.pdf, 2004. (Cited on page 47.)

[Botsch 2010] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez and Bruno Lévy. Polygon mesh processing. AK Peters, 2010. (Cited on page 53.)

[Boyer 2011] Edmond Boyer, Alexander Bronstein, Michael Bronstein, Benjamin Bustos, Tal Darom, Radu Horaud, Ingrid Hotz, Yosi Keller, Johannes Keustermans, Artiom Kovnatsky, Roee Litman, Jan Reininghaus, Ivan Sipiran, Dirk Smeets, Paul Suetens, Dirk Vandermeulen, Andrei Zaharescu and Valentin Zobel. *SHREC 2011: robust feature detection and description benchmark*. In Eurographics Workshop on 3D Object Retrieval, pages 71–78, 2011. (Cited on page 117.)

[Bronstein 2006] A.M. Bronstein, M.M. Bronstein and R. Kimmel. *Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching*. National Academy of Sciences, vol. 103, no. 5, pages 1168–72, 2006. (Cited on page 34.)

[Bronstein 2008] Alexander Bronstein, Michael Bronstein and Ron Kimmel. Numerical geometry of non-rigid shapes. Springer, 1 édition, 2008. (Cited on page 149.)

[Bronstein 2009] Alexander M. Bronstein, Michael M. Bronstein and Ron Kimmel. *Topology-Invariant Similarity of Nonrigid Shapes*. International Journal of Computer Vision, vol. 81, no. 3, pages 281–301, 2009. (Cited on page 127.)

[Bronstein 2010a] A. M. Bronstein, M. M. Bronstein, U. Castellani, A. Dubrovina, L. J. Guibas, R. P. Horaud, R. Kimmel, D. Knossow, E. von Lavante, Mateus D., M. Ovsjanikov and A. Sharma. *SHREC 2010: robust correspondence benchmark*. In Eurographics Workshop on 3D Object Retrieval, pages 87–91, 2010. (Cited on pages 6, 90 and 137.)

[Bronstein 2010b] A.M. Bronstein, M.M. Bronstein, R. Kimmel, M. Mahmoudi and G. Sapiro. *A Gromov-Hausdorff framework with diffusion geometry for topologically robust non-rigid shape alignment*. International Journal of Computer Vision, vol. 89, no. 2-3, pages 266–286, September 2010. (Cited on pages 83 and 87.)

[Bronstein 2010c] A.M. Bronstein, M.M. Bronstein, M. Mahmoudi, R. Kimmel and G. Sapiro. *A Gromov-Hausdorff framework with diffusion geometry for topologically robust non-rigid shape alignment*. International Journal of Computer Vision, vol. 89, no. 2/3, pages 266–286, 2010. (Cited on page 62.)

[Bronstein 2010d] M. Bronstein and I. Kokkinos. *Scale-Invariant Heat Kernel Signatures for Non-rigid Shape Recognition*. In Computer Vision and Pattern Recognition, pages 1704–1711, 2010. (Cited on page 100.)

[Bronstein 2011a] A. M. Bronstein, M. M. Bronstein, L. J. Guibas and M. Ovsjanikov. *Shape Google: geometric words and expressions for invariant shape retrieval*. ACM Transactions on Graphics, vol. 30, no. 1, pages 1–20, 2011. (Cited on pages 99 and 100.)

[Bronstein 2011b] M.M. Bronstein and A.M. Bronstein. *Shape Recognition with Spectral Distances*. Pattern Analysis and Machine Intelligence, vol. 33, no. 5, pages 1065–1071, may 2011. (Cited on pages 99 and 100.)

[Bruno 2010] Lévy Bruno and Richard Hao Zhang. *Spectral Geometry Processing*. In ACM SIGGRAPH Course Notes, 2010. (Cited on page 66.)

[Bunke 1999] H. Bunke. *Error Correcting Graph Matching: On the Influence of the Underlying Cost Function*. Pattern Analysis and Machine Intelligence, vol. 21, no. 9, pages 917–922, 1999. (Cited on page 125.)

[Burago 2001] Dmitri Burago, Yuri Burago and Sergei Ivanov. A course in metric geometry. AMS Graduate Studies in Mathematics, 2001. (Cited on page 11.)

[Burkard 2009] Rainer Burkard. Assignment problems. SIAM, Society for Industrial and Applied Mathematics, Philadelphia, 2009. (Cited on page 31.)

[Burt 1983] Peter Burt and Ted Adelson. *The Laplacian Pyramid as a Compact Image Code*. IEEE Transaction on Communications, vol. 9, no. 4, pages 532–540, 1983. (Cited on page 99.)

[Cagniart 2010] Cedric Cagniart, Edmond Boyer and Slobodan Ilic. *Probabilistic Deformable Surface Tracking From Multiple Videos*. In European Conference on Computer Vision, pages 326–339, 2010. (Cited on page 126.)

[Čech 2010] Jan Čech, Jiří Matas and Michal Perd'och. *Efficient Sequential Correspondence Selection by Cosegmentation*. Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pages 1568–1581, September 2010. (Cited on pages 126 and 133.)

[Chen 2009] Xiaobai Chen, Aleksey Golovinskiy and Thomas Funkhouser. *A Benchmark for 3D Mesh Segmentation*. ACM Transactions on Graphics, vol. 28, no. 3, pages 73:1–73:12, July 2009. (Cited on page 70.)

[Cho 2010] Minsu Cho, Jungmin Lee and Kyoung Mu Lee. *Reweighted random walks for graph matching*. In European conference on Computer vision, pages 492–505, Berlin, Heidelberg, 2010. Springer-Verlag. (Cited on page 130.)

[Chung 1997] F.R.K. Chung. Spectral graph theory. American Mathematical Society, 1997. (Cited on pages 7, 36, 39, 40, 70, 71, 98 and 158.)

[Chung 2000] F. Chung and S.T. Yau. *Discrete Green's Functions*. Journal of Combinatorial Theory, vol. 91, pages 191–214, July 2000. (Cited on page 114.)

[Coifman 2006] R. R. Coifman and S. Lafon. *Diffusion maps*. Applied and Computational Harmonic Analysis, vol. 21, no. 1, pages 5–30, 2006. (Cited on pages 35, 83, 84, 87 and 99.)

[Cornea 2005] Nicu D. Cornea, M. Fatih Demirci, Deborah Silver, Ali Shokoufandeh, Sven J. Dickinson and Paul B. Kantor. *3D Object Retrieval using Many-to-many Matching of Curve Skeletons*. In Shape Modeling and Applications, pages 368–373, 2005. (Cited on page 25.)

[Cour 2006] Timothee Cour, Praveen Srinivasan and Jinbo Shi. *Balanced Graph Matching*. In Advanced in Neural Information Processing Systems, pages 1–8, 2006. (Cited on pages 26 and 125.)

[Cour 2997] Timothee Cour. *Solving markov random fields with spectral relaxation*. In Artificial Intelligence and Statistics, pages 1–8, 2997. (Cited on page 125.)

[Courant 1962] Richard Courant and David Hilbert. Methods of mathematical physics, volume 2. Wiley-Interscienc, 1962. (Cited on page 64.)

[Crowley 1984] J. Crowley and A.C. Parker. *A Representation for Shape Based on Peaks and Ridges in the Difference of Low Pass Transform*. Pattern Analysis and Machine Intelligence, vol. 6, no. 2, pages 156–170, 1984. (Cited on page 99.)

[Cuzzolin 2008] Fabio Cuzzolin, Diana Mateus, David Knossow, Edmond Boyer and Radu P. Horaud. *Coherent Laplacian 3-D Protrusion Segmentation*. In Computer Vision and Pattern Recognition, pages 1–8, 2008. (Cited on pages 50 and 70.)

[Davies 2001] E. B. Davies, G. M. L. Gladwell, J. Leydold and P. F. Stadler. *Discrete Nodal Domain Theorems*. Linear Algebra Application, vol. 336, pages 51–60, 2001. (Cited on pages 71 and 73.)

[Dempster 1977] A. P. Dempster, N. M. Laird and D. B. Rubin. *Maximum Likelihood from Incomplete Data via the EM algorithm*. Journal of the Royal Statistical Society, vol. 30, no. 1, pages 1–38, 1977. (Cited on page 46.)

[Do carmo 1992] Manfredo Perdigão Do carmo. Riemannian geometry. Birkhauser, 1992. (Cited on page 13.)

[Donoho 2003] David L. Donoho and Carrie Grimes. *Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data*. National Academy of Sciences, vol. 100, no. 10, pages 5591–5596, May 2003. (Cited on page 35.)

[Duchenne 2011] Olivier Duchenne, Francis Bach, In-So Kweon and Jean Ponce. *A Tensor-Based Algorithm for High-Order Graph Matching*. Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pages 2383–2395, 2011. (Cited on page 126.)

[Elad 2003] Asi Elad and Ron Kimmel. *On Bending Invariant Signatures for Surfaces*. Pattern Analysis and Machine Intelligence, vol. 25, no. 10, pages 1285–1295, 2003. (Cited on page 50.)

[Floater 2005] M. S. Floater and K. Hormann. *Surface parametrization: a tutorial and survey*. Advances in Multi-resolution for Geometric Modeling, pages 157–186, 2005. (Cited on page 50.)

[Fraley 2002] C. Fraley and A. E. Raftery. *Model-Based Clustering, Discriminant Analysis, and Density Estimation*. Journal of the American Statistical Association, vol. 97, pages 611–631, 2002. (Cited on page 72.)

[Fraley 2006] C. Fraley and A. E. Raftery. *MCLUST version 3 for R: Normal Mixture Modeling and Model-Based Clustering*. Technical Report 504, Department of Statistics, University of Washington, 2006. (Cited on page 80.)

[Franco 2009] Jean-Sébastien Franco and Edmond Boyer. *Efficient Polyhedral Modeling from Silhouettes*. Pattern Analysis and Machine Intelligence, vol. 31, no. 3, pages 414–427,, March 2009. (Cited on pages 3, 4, 6, 53, 61, 80, 90 and 149.)

[Garey 1990] Michael R. Garey and David S. Johnson. Computers and intractability: A guide to the theory of np-completeness. W. H. Freeman & Co., New York, NY, USA, 1990. (Cited on page 26.)

[Gebal 2009] K. Gebal, J. A. Baerentzen, H. Aanaes and R. Larsen. *Shape Analysis Using the Auto Diffusion Function*. In Symposium on Geometry Processing, pages 1405–1413, 2009. (Cited on page 99.)

[Godsil 2001] C. Godsil and G. Royle. Algebraic graph theory. Springer, 2001. (Cited on page 26.)

[Gold 1996] Steven Gold and Anand Rangarajan. *A Graduated Assignment Algorithm for Graph Matching*. Pattern Analysis and Machine Intelligence, vol. 18, pages 377–388, 1996. (Cited on page 125.)

[Golovinskiy 2008] A. Golovinskiy and T. Funkhouser. *Randomized cuts for 3D mesh analysis*. ACM Transactions on Graphics, vol. 5, no. 27, pages 1–8, 2008. (Cited on page 70.)

[Golub 1989] G.H. Golub and C.F. van Loan. Matrix Computations. The Johns Hopkins University Press, Baltimore, 1989. (Cited on page 39.)

[Grady 2010] Leo Grady and Jonathan R. Polimeni. Discrete Calculus: Applied Analysis on Graphs for Computational Science. Springer, 2010. (Cited on page 36.)

[Gray 2006] Alfred Gray, Elsa Abbena and Simon Salamon. Modern differential geometry of curves and surfaces with mathematica. Chapman & HallCRC, 2006. (Cited on page 13.)

[Grinstead 1998] C. M. Grinstead and L. J. Snell. Introduction to probability. American Mathematical Society, 1998. (Cited on pages 42 and 87.)

[Ham 2004] Jihun Ham, Daniel D. Lee, Sebastian Mika and Bernhard Schölkopf. *A kernel view of the dimensionality reduction of manifolds*. In International Conference on Machine Learning, pages 47–55, New York, NY, USA, 2004. ACM. (Cited on page 41.)

[Hilaga 2001] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura and Tosiyasu L. Kunii. *Topology matching for fully automatic similarity estimation of 3D shapes*. In Computer Graphics and Interactive Techniques, pages 203–212, 2001. (Cited on page 25.)

[Hoffman 1953] A. J. Hoffman and H. W. Wielandt. *The variation of the spectrum of a normal matrix*. Duke Mathematical Journal, vol. 20, no. 1, pages 37–39, 1953. (Cited on page 27.)

[Horaud 2011] Radu P. Horaud, Florence Forbes, Manuel Yguel, Guillaume Dewaele and Jian Zhang. *Rigid and Articulated Point Registration with Expectation Conditional Maximization*. Pattern Analysis and Machine Intelligence, vol. 33, no. 3, pages 587 –602, 2011. (Cited on pages 7, 47, 82, 90, 91, 139 and 140.)

[Horn 1994] R. A. Horn and C. A. Johnson. Matrix analysis. Cambridge University Press, Cambridge, 1994. (Cited on pages 28, 157 and 159.)

[Hou 2010] T. Hou and H. Qin. *Efficient Computation of Scale-Space Features for Deformable Shape Correspondences*. In European Conference on Computer Vision, pages 384–397, 2010. (Cited on page 133.)

[Huang 2008] Q-X Huang, B. Adams, M. Wicke and Leonida J. Guibas. *Non-rigid Registration under Isometric Deformations*. In Symposium on Geometry Processing, pages 1449–1457, 2008. (Cited on page 127.)

[Huang 2011] Peng Huang, C. Budd and A. Hilton. *Global temporal registration of multiple non-rigid surface sequences*. In Computer Vision and Pattern Recognition, pages 3473–3480, 2011. (Cited on page 127.)

[Hummel 1987] R. Hummel. The scale-space formulation of pyramid data structures, pages 107–123. Academic Press, 1987. (Cited on page 99.)

[Jain 2007] Varun Jain, Hao Zhang and Oliver van Kaick. *Non-Rigid Spectral Correspondence of Triangle Meshes*. International Journal of Shape Modeling, vol. 13, no. 1, pages 101–124, 2007. (Cited on pages 50 and 62.)

[Johnson 1999] Andrew Edie Johnson and Martial Hebert. *Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes*. Pattern Analysis and Machine Intelligence, vol. 21, no. 5, pages 433–449, 1999. (Cited on page 100.)

[Kamvar 2003] S. D. Kamvar, D. Klein and C. D. Manning. *Spectral learning*. In International Joint Conference on Artificial Intelligence, pages 1–8, 2003. (Cited on pages 84, 92 and 93.)

[Koenderink 1984] J. Koenderink. *The structure of images*. Biological Cybernetics, vol. 50, 1984. (Cited on page 99.)

[Kondor 2002] R. Kondor and J. Lafferty. *Diffusion Kernels on Graphs and Other Discrete Structures*. In International Conference on Machine Learning, pages 315–322, 2002. (Cited on page 99.)

[Kondor 2004] R. Kondor and J-P. Vert. *Diffusion Kernels*. In B. Scholkopf, K. Tsuda and J.-P. Vert, editeurs, Kernel Methods in Computational Biology. MIT Press, 2004. (Cited on page 99.)

[Körtgen 2003] Marcel Körtgen, G. J. Park, Marcin Novotni and Reinhard Klein. *3D Shape Matching with 3D Shape Contexts*. In European Seminar on Computer Graphics, pages 1–8, 2003. (Cited on page 100.)

[Kulis 2009] B. Kulis, S. Basu, I. Dhillon and R. Mooney. *Semi-supervised graph clustering: a kernel approach*. Machine Learning, vol. 74, no. 1, pages 1–22, January 2009. (Cited on page 84.)

[LAI 2008] Y. K. LAI, S. M. HU, MARTIN R. R. and P. L. ROSIN. *Fast mesh segmentation using random walks*. In Symposium on Solid and Physical Modeling, pages 183–191, 2008. (Cited on page 70.)

[Leordeanu 2005] Marius Leordeanu and Martial Hebert. *A Spectral Technique for Correspondence Problems Using Pairwise Constraints*. In International Conference on Computer Vision, volume 2, pages 1482–1489, 2005. (Cited on pages 125, 126 and 130.)

[Letouzey 2012] Antoine Letouzey and Edmond Boyer. *Progressive Shape Models*. In Computer Vision and Patern Recognition, pages 1–8, 2012. (Cited on page 127.)

[Lévy 2006] Bruno Lévy. *Laplace-Beltrami Eigenfunctions: Towards an Algorithm that Understands Geometry*. In Shape Modeling International, pages 1–8, 2006. (Cited on pages 50, 71, 72 and 98.)

[Li 2009] Z. Li and J. Liu. *Constrained clustering by spectral kernel learning*. In International Conference on Computer Vision, pages 1–8, 2009. (Cited on pages 83, 92 and 93.)

[Lindeberg 1994] T. Lindeberg. *Scale-space Theory: A basic tool for analysing structures at different scales*. Journal of Applied Statistics, vol. 21, no. 2, pages 225–270, 1994. (Cited on pages 99 and 117.)

[Liu 2007] Rong Liu and Hao Zhang. *Mesh Segmentation via Spectral Embedding and Contour Analysis*. Computer Graphics Forum, vol. 26, no. 3, pages 385–394, 2007. (Cited on page 83.)

[Llados 2001] J. Llados, E. Marti and J.J. Villanueva. *Symbol Recognition by Error-Tolerant Subgraph Matching between Region Adjacency Graphs*. Pattern Analysis and Machine Intelligence, vol. 23, pages 1137–1143, 2001. (Cited on page 125.)

[Lloyd 1982] Stuart P. Lloyd. *Least squares quantization in PCM*. Information Theory, vol. 2, no. 28, pages 129–137, 1982. (Cited on page 46.)

[Lowe 2004] D. G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints*. International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, 2004. (Cited on pages 100 and 133.)

[Luo 2003] B. Luo and E.R. Hancock. *A unified framework for alignment and correspondence*. Computer Vision and Image Understanding, vol. 92, no. 1, pages 26–55, 2003. (Cited on page 126.)

[Luo 2009] C. Luo, I. Safa and Y. Wang. *Approximating Gradients for Meshes and Point Clouds via Diffusion Metric*. In Symposium on Geometry Processing, pages 1497–1508, 2009. (Cited on pages 62 and 159.)

[Maes 2010] C. Maes, T. Fabry, J. Keustermans, D. Smeets, P. Suetens and D. Vandermeulen. *Feature detection on 3D face surfaces for pose normalisation and recognition*. In Biometrics: Theory Applications and Systems (BTAS), pages 1 –6, 2010. (Cited on page 117.)

[Mahmoudi 2009] M. Mahmoudi and G. Sapiro. *Three-dimensional point cloud recognition via distributions of geometric distances*. Graphical Models, vol. 71, pages 22–31, 2009. (Cited on pages 62 and 100.)

[Mateus 2008] Diana Mateus, Radu P. Horaud, David Knossow, Fabio Cuzzolin and Edmond Boyer. *Articulated Shape Matching Using Laplacian Eigenfunctions and Unsupervised Point Registration*. In Computer Vision and Pattern Recognition, pages 1–8, June 2008. (Cited on pages 7, 47, 50, 59, 62, 82, 83, 90, 91, 126, 127, 141, 142, 144, 147, 148 and 149.)

[Mateus 2009] Diana C. Mateus. *Spectral Tools for Unsupervised Modeling of Articulated Objects from Multiple-view Videos*. PhD thesis, Institute Polytechnique de Grenoble, Grenoble, France, September 2009. (Cited on pages 51, 66, 126 and 144.)

[Meila 2001] M. Meila and J. Shi. *Learning Segmentation by Random Walks*. In Advances in Neural Information Processing Systems, pages 873–879. MIT Press, 2001. (Cited on page 39.)

[Mémoli 2004] Facundo Mémoli and Guillermo Sapiro. *Comparing point clouds*. In Symposium on Geometry Processing, pages 32–40, 2004. (Cited on page 127.)

[Meyer 2003] M. Meyer, M. Desbrun, P. Schröder and A. H. Barr. *Discrete differential-geometry operators for triangulated 2-manifolds*. In Visualization and Mathematics, pages 35–57. Springer, 2003. (Cited on page 59.)

[Moenning 2003] Carsten Moenning and Neil A. Dodgson. *Fast Marching farthest point sampling for implicit surfaces and point clouds*. Rapport technique, 2003. (Cited on page 127.)

[Mukherjee 2010] S. Mukherjee, Q. Wu and D-X. Zhou. *Learning gradients on manifolds*. Bernoulli, vol. 16, no. 1, pages 181–2007, 2010. (Cited on page 62.)

[Nadler 2006] B. Nadler and M. Galun. *Fundamental Limitations of Spectral Clustering*. In Advances in Neural Information Processig Systems, pages 1–8. MIT Press, 2006. (Cited on page 70.)

[Neuhaus 2006] Michel Neuhaus and Horst Bunke. *Edit distance-based kernel functions for structural pattern classification*. Pattern Recognition, vol. 39, no. 10, pages 1852–1863, 2006. (Cited on page 125.)

[Ng 2001] A. Ng, M. Jordan and Y. Weiss. *On spectral clustering: analysis and an algorithm*. In Advances in Neural Information Processig Systems, pages 849–856. MIT Press, 2001. (Cited on pages 39, 68, 70, 71 and 76.)

[Ovsjanikov 2008] M. Ovsjanikov, J. Sun and L. Guibas. *Global Intrinsic Symmetries of Shapes*. Computer Graphics Forum, vol. 27, pages 1341–1348, 2008. (Cited on page 83.)

[Ovsjanikov 2010] Maks Ovsjanikov, Quentin Merigot, Facundo Memoli and Leonidas Guibas. *One Point Isometric Matching with the Heat Kernel*. Computer Graphics Forum, vol. 29, no. 5, pages 1555–1564, 2010. (Cited on pages 35, 119, 120, 122, 127, 128, 133, 141, 142, 144, 147 and 148.)

[Pelillo 1999] Marcello Pelillo. *Replicator Equations, Maximal Cliques, and Graph Isomorphism*. Neural Computation, vol. 11, pages 1933–1955, November 1999. (Cited on pages 125, 126 and 130.)

[Petit 2011] Benjamin Petit, Antoine Letouzey, Edmond Boyer and Jean-Sébastien Franco. *Surface Flow from Visual Cues*. In Vision, Modeling and Visualization Workshop, pages 1–8, Berlin, Germany, October 2011. (Cited on page 149.)

[Pinkall 1993] U. Pinkall and K. Polthier. *Computing Discrete Minimal Surfaces and Their Conjugates*. Experimental Mathematics, vol. 2, pages 15–36, 1993. (Cited on pages 59 and 72.)

[Popa 2010] T. Popa, I. South-Dickinson, D. Bradley, A. Sheffer and W. Heidrich. *Globally Consistent Space-Time Reconstruction*. In Symposium on Geometry Processing, pages 1–8, 2010. (Cited on page 127.)

[Powers 1988] D. L. Powers. *Graph Partitioning by Eigenvectors*. Linear Algebra and its Applications, vol. 101, pages 121–133, 1988. (Cited on page 71.)

[Qiu 2007] H. Qiu and E. R. Hancock. *Clustering and Embedding Using Commute Times*. Pattern Analysis and Machine Intelligence, vol. 29, no. 11, pages 1873–1890, 2007. (Cited on pages 39, 42, 83, 87 and 99.)

[Reuter 2006a] Martin Reuter. Laplace spectra for shape recognition. Books on Demand GmbH, 2006. (Cited on pages 24, 50 and 53.)

[Reuter 2006b] Martin Reuter, Franz-Erich Wolter and Niklas Peinecke. *Laplace-Beltrami spectra as "Shape-DNA" of surfaces and solids*. Computer-Aided Design, vol. 38, no. 4, pages 342–366, 2006. (Cited on page 100.)

[Reuter 2009] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè and M. Spagnuolo. *Discrete Laplace-Beltrami operators for shape analysis and segmentation*. Shape Modeling International, vol. 1, no. 1, pages 1–12, 2009. (Cited on pages 71, 72 and 128.)

[Reuter 2010] Martin Reuter. *Hierarchical Shape Segmentation and Registration viaÂ Topological Features of Laplace-Beltrami Eigenfunctions*. International Journal of Computer Vision, vol. 89, no. 2, pages 287–308, September 2010. (Cited on pages 83 and 127.)

[Rosenberg 1997] Steven Rosenberg. The laplacian on a riemannian manifold. Cambridge University Press, 1997. (Cited on page 23.)

[Roweis 2000] Sam T. Roweis and Lawrence K. Saul. *Nonlinear dimensionality reduction by locally linear embedding*. Science, vol. 290, pages 2323–2326, 2000. (Cited on page 35.)

[Ruggeri 2010] M. R. Ruggeri, G. Patané, M. Spagnuolo and D. Saupe. *Spectral-Driven Isometry-Invariant Matching of 3D Shapes*. International Journal of Computer Vision, vol. 89, pages 248–265, 2010. (Cited on pages 127 and 128.)

[Rustamov 2007] R. M. Rustamov. *Laplace-Beltrami eigenfunctions for deformation invariant shape representation*. In Symposium on Geometry Processing, pages 1–8, 2007. (Cited on page 50.)

[Sahillionglu 2010] Yusuf Sahillionglu and Yucel Yemez. *3D Shape Correspondence by Isometry-Driven Greedy Optimization*. In Computer Vision and Pattern Recognition, pages 453–458, 2010. (Cited on page 127.)

[Schellewald 2005] Christian Schellewald and Christoph Schnörr. *Probabilistic subgraph matching based on convex relaxation*. In Energy Minimization Methods in Computer Vision and Pattern Recognition, pages 171–186, 2005. (Cited on page 125.)

[Schölkopf 1998a] Bernhard Schölkopf, Alex J. Smola and Klaus-Robert Müller. *Nonlinear Component Analysis as a Kernel Eigenvalue Problem*. Neural Computation, vol. 10, no. 5, pages 1299–1319, 1998. (Cited on page 101.)

[Schölkopf 1998b] Bernhard Schölkopf, Alexander Smola and Klaus-Robert Müller. *Nonlinear component analysis as a kernel eigenvalue problem*. Neural Computing, vol. 10, no. 5, pages 1299–1319, July 1998. (Cited on page 34.)

[Scott 1991] G. Scott and C. Longuet-Higgins. *An algorithm for associating the features of two images*. Biological Sciences, vol. 244, pages 21–26, 1991. (Cited on page 126.)

[Seitz 2006] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein and Richard Szeliski. *A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms*. In Computer Vision and Pattern Recognition, 2006. (Cited on page 6.)

[Sethian 1996] J. A. Sethian. *A Fast Marching Level Set Method for Monotonically Advancing Fronts*. National Academy of Sciences, vol. 93, no. 4, pages 1591–1595, 1996. (Cited on page 144.)

[Shapira 2008] L. Shapira, A. Shamir and D. Cohen-Or. *Consistent mesh partitioning and skeletonisation using the shape diameter function*. Visual Computing, vol. 4, no. 24, pages 249–259, 2008. (Cited on page 70.)

[Shapiro 1992] L.S. Shapiro and J.M. Brady. *Feature-based correspondence: an eigenvector approach*. Image and Vision Computing, vol. 10, pages 283–288, 1992. (Cited on page 126.)

[Sharma 2010a] Avinash Sharma and Radu Horaud. *Shape Matching Based on Diffusion Embedding and on Mutual Isometric Consistency*. In CVPR workshop on Non-Rigid Shape Analysis and Deformable Image Alignment, pages 1–8, 2010. (Cited on page 130.)

[Sharma 2010b] Avinash Sharma, Etienne von Lavante and Radu Horaud. *Learning Shape Segmentation Using Constrained Spectral Clustering and Probabilistic Label Transfer*. In European Conference on Computer Vision, LNCS, pages 743–756, Heraklion, Greece, September 2010. Springer. (Cited on page 62.)

[Sharma 2011] Avinash Sharma, Radu P. Horaud, Jan Cech and Edmond Boyer. *Topologically-Robust 3D Shape Matching Based on Diffusion Geometry and Seed Growing*. In Computer Vision and Pattern Recognition, pages 2481–2488, Colorado Springs, CO, 2011. (Cited on pages 100 and 127.)

[Sharma 2012] Avinash Sharma, Radu P. Horaud and Diana Mateus. *3D Shape Registration Using Spectral Graph Embedding and Probabilistic Matching*. In Leo Grady and Olivier Lezoray, editeurs, Image Processing and Analysing With Graphs: Theory and Practice, pages 441–474. CRC Press, 2012. (Cited on page 59.)

[Shawe-Taylor 2004] J. Shawe-Taylor and N. Cristianini. Kernel methods in pattern analysis. Cambridge University Press, 2004. (Cited on pages 7, 98 and 106.)

[Shi 2000] J. Shi and J. Malik. *Normalized Cuts and Image Segmentation*. Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pages 888–905, 2000. (Cited on pages 39 and 70.)

[Shlafman 2002] Shymon Shlafman, Ayellet Tal and Sagi Katz. *Metamorphosis of polyhedral surfaces using decomposition*. Comput. Graphics Forum, vol. 21, no. 3, pages 219–228, 2002. (Cited on page 70.)

[Siddiqi 2008] Kaleem Siddiqi, Juan Zhang, Diego Macrini, Ali Shokoufandeh, Sylvain Bouix and Sven Dickinson. *Retrieving articulated 3-D models using medial surfaces*. Machine Vision Applications, vol. 19, no. 4, pages 261–275, 2008. (Cited on page 25.)

[Sipiran 2011] Ivan Sipiran and Benjamin Bustos. *Harris 3D: a robust extension of the Harris operator for interest point detection on 3D meshes*. The Visual Computer, vol. 27, pages 963–976, 2011. (Cited on pages 116, 117 and 118.)

[Slabaugh 2001] G. Slabaugh, B. Culbertson, T. Malzbender and R. Schafer. *A Survey of Methods for Volumetric Scene Reconstruction from Photographs*. In International Workshop on Volume Graphics, 2001. (Cited on page 6.)

[Smeets 2012] Dirk Smeets, Jeroen Hermans, Dirk Vandermeulen and Paul Suetens. *Dense Shape Correspondences using Spectral High-Order Graph Matching*. In Computer Vision and Patern Recognition, pages 1–8, 2012. (Cited on page 126.)

[Sorkine 2006] O. Sorkine. *Differential Representations for Mesh Processing*. Computer Graphics Forum, vol. 25, no. 4, pages 789–807, 2006. (Cited on page 50.)

[Spielman 2007] D. A. Spielman and S. Teng. *Spectral partitioning works: Planar graphs and finite element meshes*. Linear Algebra and its Applications, vol. 421, no. 2–3, pages 284–305, March 2007. (Cited on page 81.)

[Starck 2007a] J. Starck and A. Hilton. *Correspondence labelling for wide-timeframe free-form surface matching*. In International Conference on Computer Vision, pages 1–8, 2007. (Cited on pages 126 and 127.)

[Starck 2007b] Jonathan Starck and Adrian Hilton. *Surface Capture for Performance Based Animation*. IEEE Computer Graphics and Applications, vol. 27, no. 3, pages 21–31, 2007. (Cited on pages 6, 61, 80, 90 and 149.)

[Starck 2009] J. Starck, A. Maki, S. Nobuhara, A. Hilton and T. Matsuyama. *The Multiple-Camera 3-D Production Studio*. Circuits and Systems for Video Technology, vol. 19, no. 6, pages 856 –869, 2009. (Cited on page 6.)

[Stoll 2010] Carsten Stoll, Juergen Gall, Edilson de Aguiar, Sebastian Thrun and Christian Theobalt. *Video-based reconstruction of animatable human characters*. In ACM SIGGRAPH Asia, pages 139–139, 2010. (Cited on pages 6 and 90.)

[Sun 2009] J. Sun, . Ovsjanikov and L.J. Guibas. *A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion*. Computer Graphics Forum, vol. 28, no. 5, pages 1383–1392, 2009. (Cited on pages 35, 99, 100, 101, 117 and 159.)

[Sussmuth 2008] J. Sussmuth, M. Winter and G Greiner. *Reconstructing Animated Meshes from Time-Varying Point Clouds*. In Symposium on Geometry Processing, pages 1469–1476, 2008. (Cited on page 126.)

[Szummer 2002] M. Szummer and T. Jaakkola. *Partially labeled classification with Markov random walks*. In Advances in Neural Information Processig Systems, pages 1–8. MIT Press, 2002. (Cited on page 83.)

[Tenenbaum 2000] J. B. Tenenbaum, V. de Silva and J. C. Langford. *A global geometric framework for nonlinear dimensionality reduction*. Science, vol. 290, no. 5500, pages 2319–2323, December 2000. (Cited on page 35.)

[Tevs 2009] Art Tevs, Martin Bokeloh, Michael Wand, Andreas Schilling and Hans-Peter Seidel. *Isometric Registration of Ambiguous Partial Data*. In Computer Vision and Pattern Recognition, pages 1185–1192, 2009. (Cited on page 126.)

[Tevs 2011] Art Tevs, Alexander Berner, Michael Wand, Ivo Ihrke and Hans-Peter Seidel. *Intrinsic Shape Matching by Planned Landmark Sampling*. In Computer Graphics Forum, pages 543–552, 2011. (Cited on page 128.)

[Thorstensen 2009] Nicolas Thorstensen. *Manifold learning and applications to shape and image processing*. PhD thesis, Ècole Nationale des Ponts et Chausseés, Paris, France, November 2009. (Cited on pages 11 and 133.)

[Tombari 2012] Federico Tombari, Samuele Salti and Luigi DiÂ Stefano. *Performance Evaluation of 3D Keypoint Detectors*. International Journal of Computer Vision, pages 1–23, 2012. (Cited on page 118.)

[Tung 2010] Tony Tung and Takashi Matsuyama. *Dynamic Surface Matching by Geodesic Mapping for 3D Animation Transfer*. In Computer Vision and Pattern Recognition, pages 1402–1409, 2010. (Cited on page 127.)

[Tuytelaars 2008] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. Now Publishers Inc., Hanover, MA, USA, 2008. (Cited on page 117.)

[Umeyama 1988] S. Umeyama. *An Eigendecomposition Approach to Weighted Graph Matching Problems*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 10, no. 5, pages 695–703, May 1988. (Cited on pages 30, 31 and 126.)

[Vallet 2008] B. Vallet and Bruno Lévy. *Manifold harmonics*. Computer Graphics Forum, vol. 27, no. 2, pages 251–260, 2008. (Cited on pages 50 and 98.)

[van Kaick 2011] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh and Danial Cohen-Or. *A Survey on Shape Correspondence*. Computer Graphics Forum, vol. 30, no. 6, pages 1681–1707, 2011. (Cited on page 100.)

[Varanasi 2008] Kiran Varanasi, Andrie Zaharescu, Edmond Boyer and Radu Horaud. *Temporal Surface Tracking using Mesh Evolution*. In European Conference on Computer Vision, pages 30–43, 2008. (Cited on pages 5 and 127.)

[Vlasic 2008] Daniel Vlasic, Ilya Baran, Wojciech Matusik and Jovan Popovic. *Articulated Mesh Animation from Multi-view Silhouettes*. ACM Transaction on Graphics, vol. 27, no. 3, pages 97:1–97:9, 2008. (Cited on pages 6, 90, 127 and 149.)

[von Luxburg 2007] U. von Luxburg. *A tutorial on spectral clustering*. Statistics and Computing, vol. 17, no. 4, pages 395–416, 2007. (Cited on pages 11, 36, 40, 45, 46, 62, 68, 71, 82, 83, 92 and 93.)

[Wagstaff 2001] K. Wagstaff, C. Cardie, S. Rogers and S. Schrödl. *Constrained K-means Clustering with Background Knowledge*. In International Conference on Machine Learning, pages 1–8, 2001. (Cited on pages 83 and 86.)

[Wang 2006] Hong Fang Wang and Edwin R. Hancock. *Correspondence matching using kernel principal components analysis and label consistency constraints*. Pattern Recognition, vol. 39, no. 6, pages 1012–1025, 2006. (Cited on page 126.)

[Wang 2010] Xu-Lei Wang, Yi Liu and Hongbin Zha. *Intrinsic Spin Images: A Subspace Decomposition Approach to Understanding 3D Deformable Shapes*. In Symposium on 3D Data Processing, Visualization and Transmission, pages 1–8, 2010. (Cited on page 100.)

[Wardetzky 2007] M. Wardetzky, S. Mathur, F. Kälberer and E. Grinspun. *Discrete Laplace operators: No free lunch*. In Symposium on Geometry Processing, pages 33–37, 2007. (Cited on pages 59 and 67.)

[Wardetzky 2008] M. Wardetzky. *Convergence of the Cotangent Formula: An Overview*. In A. I. Bobenko, John M. Sullivan, Peter Schröder and Günter Ziegler, editeurs, Discrete Differential Geometry, pages 89–112. Birkhäuser, 2008. (Cited on page 62.)

[Wilkinson 1965] J. H. Wilkinson. The algebraic eigenvalue problem. Clarendon Press, Oxford, 1965. (Cited on page 27.)

[Wilkinson 1970] J. H. Wilkinson. *Elementary Proof of the Wielandt-Hoffman Theorem and of its Generalization*. Rapport technique CS150, Stanford University, January 1970. (Cited on page 28.)

[Wu 2010] Huai-Yu Wu, Hongbin Zha, Tao Luo, Xu-Lei Wang and Songde Ma. *Global and local isometry-invariant descriptor for 3D shape comparison and partial matching*. In Computer Vision and Pattern Recognition, pages 438–445, 2010. (Cited on page 100.)

[Wu 2011] Huai-Yu Wu and Hongbin Zha. *Robust consistent correspondence between 3D non-rigid shapes based on "Dual Shape-DNA"*. In International Conference on Computer Vision, pages 587–594, 2011. (Cited on pages 100 and 127.)

[Xing 2002] E. P. Xing, A. Y. Ng, M. I. Jordan and S. J. Russell. *Distance Metric Learning with Application to Clustering with Side-Information*. In Advances in Neural Information Processig Systems, pages 1–8. MIT Press, 2002. (Cited on pages 83 and 86.)

[Yu 2003] S. X. Yu and J. Shi. *Multiclass Spectral Clustering*. In International Conference on Computer Vision, pages 1–8, 2003. (Cited on page 70.)

[Yu 2004] S. X. Yu and J. Shi. *Segmentation Given Partial Grouping Constraints*. Pattern Analysis and Machine Intelligence, vol. 26, no. 2, pages 173–183, 2004. (Cited on page 84.)

[Zaharescu 2009] Andrei Zaharescu, Edmond Boyer, Kiran Varanasi and Radu P. Horaud. *Surface Feature Detection and Description with Applications to Mesh Matching*. In Computer Vision and Pattern Recognition, pages 373–380, Miami Beach, Florida, June 2009. (Cited on pages 62, 117 and 133.)

[Zaharescu 2011] Andrei Zaharescu, Edmond Boyer and Radu P. Horaud. *Topology-Adaptive Mesh Deformation for Surface Evolution, Morphing, and Multi-View Reconstruction*.

Pattern Analysis and Machine Intelligence, vol. 33, no. 4, pages 823 – 837, April 2011. (Cited on pages 3, 4, 6, 52 and 53.)

[Zaharescu 2012] Andrei Zaharescu, Edmond Boyer and Radu P. Horaud. *Keypoints and Local Descriptors of Scalar Functions on 2D Manifolds*. International Journal of Computer Vision, vol. 100, no. 1, pages 78–98, 2012. (Cited on page 100.)

[Zavlanos 2008] M. M. Zavlanos and G. J. Pappas. *A Dynamical Systems Approach to Weighted Graph Matching*. Automatica, vol. 44, pages 2817–2824, 2008. (Cited on page 157.)

[Zelnik-manor 2004] L. Zelnik-manor and P. Perona. *Self-tuning spectral clustering*. In Advances in Neural Information Processig Systems, pages 1–8. MIT Press, 2004. (Cited on page 70.)

[Zeng 2010a] Y. Zeng, C. Wang, Y. Wang, X. Gu, D. Samras and N. Paragios. *Dense non-rigid surface registration using high order graph matching*. In Computer Vision and Patern Recognition, pages 382–389, 2010. (Cited on page 126.)

[Zeng 2010b] Yun Zeng, Chaohui Wang, Yang Wang, Dimitris Samaras, Xianfeng Gu and Nikos Paragios. *Dense Non-rigid Registration Using High Order Graph Matching*. In Computer Vision and Pattern Recognition, 2010. (Cited on page 127.)

[Zhang 2005] Zhenyue Zhang and Hongyuan Zha. *Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment*. SIAM Journal on Scientific Computing, vol. 26, no. 1, pages 313–338, January 2005. (Cited on page 36.)

[Zhang 2008] H. Zhang, A. Sheffer, D. Cohen-Or, Q. Zhou, O. van Kaick and A. Tagliasacchi. *Deformation-Driven Shape Correspondence*. Computer Graphics Forum, vol. 25, no. 5, pages 1431–1439, 2008. (Cited on page 127.)