Università Ca'Foscari Venezia

Scuola Dottorale di Ateneo
Graduate School

Dottorato di ricerca
in Informatica
Ciclo 29
Anno di discussione 2017

Generative Models for Non-Vectorial Data

SETTORE SCIENTIFICO DISCIPLINARE DI AFFERENZA: INF/01
Tesi di Dottorato di Andrea Gasparetto, matricola 812882

Coordinatore del Dottorato	Tutore del Dottorando
Prof. Riccardo Focardi	Prof. Andrea Torsello

Università Ca' Foscari Venezia

Dottorato di Ricerca in Informatica, 29° Ciclo

(A.A. 2012/2013 – 2015/2016)

Generative Models for Non-Vectorial Data

Settore scientifico disciplinare di afferenza: INF/01

Tesi di dottorato di Andrea Gasparetto
Matr. 812882

Tutore del dottorando

Andrea Torsello

Coordinatore del dottorato

Riccardo Focardi

September, 2016

Author's Web Page: http://www.dsi.unive.it/~gasparetto

Author's e-mail: andrea.gasparetto@unive.it

Author's address:

Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348465
fax. +39 041 2348419
web: http://www.dais.unive.it

To my family, Andrea Torsello and the whole computer vision group

- Andrea Gasparetto

Abstract

In this thesis we present several different approaches for constructing generative models to be used in both graphs and shapes. The central problem that we address in this work is how data that are defined in a non-vectorial space can be processed in order to tackle problem like data clustering and classification.

While there exists a huge framework of techniques that deal with classification and regression with real-valued data which can be embedded into the Euclidean vector space, dealing with data that lay in different spaces is another matter. As an example, relational graphs represent a convenient way to represent real world data. However, working with graphs presents two problems: firstly, the order over the vertices of a graph is arbitrary. Secondly, the number of vertices may vary among different graphs. This leads to several practical problems when it comes to define statistical quantities like mean and variance of the data. Working with shapes brings similar issues. Methods of shape analysis field are often defined on manifolds which represent different spaces. In order to define a common intrinsic-space, a map between the surfaces must be defined. However, finding such map in a non-rigid setting presents several issues, in particular from a dimension of the problem point of view.

One solution is to define an embedding space endowed with an intrinsic measure in which a statistical framework is used to learn the parameters of a generative model. Such model can then be used, for example, in classification tasks. In the first part of the thesis we introduce two generative models for shape classification. Both methods are based on spectral descriptors. However, in the first one we lift the one-to-one node-correspondence assumption. In the second approach, in which we study the variations of a metric defined on a special manifold, we deal the problem of missing correspondences casting it as a bipartite matching problem. Finally, we present a technique based on functional correspondences that computes an injective map between two surfaces.

In the second part we deal with the problem of embedding data represented as relational graphs. In particular, the first methods we present is based on the spectral components of a graph which are used to define a generative model. In another work, we lift the problem of the missing common intrinsic space by generalizing a structural kernel based on the Jensen-Shannon divergence. Finally, we investigate the use of multi-graph matching techniques in the context of graph kernels and how to transform any given matching algorithm to a transitive multi-graph matching approach.

Sommario

In questa tesi presentiamo diversi approcci per la costruzione di modelli generativi utilizzati sia su grafi sia su modelli. Il problema principale su cui questo lavoro verte riguarda il problema di processare dati che vengono definiti in spazi non vettoriali in modo da poterli utilizzare in operazioni di clustering e classificazione.

In letteratura sono state proposte un gran numero di tecniche utilizzate per la classificazione e la regressione di dati reali che possono essere immersi in uno spazio vettoriale Euclideo. Processare dati che giacciono in spazi differenti è una questione diversa. Ad esempio, i grafi rappresentano un mezzo conveniente per rappresentare dati provenienti dal mondo reale. Purtroppo lavorare con i grafi presenta diverse difficoltá. Innanzitutto, l'ordine dei vertici di un grafo è arbitrario. Inoltre, il numero di vertici è variabile tra grafi differenti. Questo porta a diversi problemi di natura pratica quando si vogliono definire quantitá statistiche come la media e la varianza dei dati. Lavorare con le forme porta a problematiche simili. I metodi proposti per l'analisi delle forme sono spesso definiti su varietá che rappresentano spazi diversi. Per definire uno spazio intrinseco comune è necessario trovare una mappa tra le superfici. Tuttavia, trovare tale mappa può risultare particolarmente ostico (in particolare se trasformazioni non rigide sono coinvolte), soprattutto dal punto di vista della dimensione del problema.

Una possibile soluzione consiste nella definizione di uno spazio di embedding su cui é stata definita una misura intrinseca. Su tale misura é possibile definire un framework statistico in modo da imparare i parametri che contraddistinguono un modello generativo. Tale modello può quindi essere utilizzato, tra le altre cose, per la classificazione dei dati. Nella prima parte di questa tesi introduciamo due modelli generativi per la classificazione di forme. Entrambi i metodi presentati utilizzano descrittori spettrali. Tuttavia, nel primo approccio non assumiamo che la corrispondenza tra i vertici sia nota. Nel secondo invece, studiamo la variazione di una metrica definita in una varietá speciale e gestiamo il problema della mancanza delle corrispondenze proiettandolo in un problema di matching bipartito. Infine, presentiamo una tecnica basata sulle corrispondenze funzionali che calcola la mappa iniettiva tra due superfici.

Nella seconda parte della tesi affrontiamo il problema dell'embedding dei dati rappresentati da grafi relazionali. In particolare, il primo metodo proposto é basato sulle componenti spettrali di un grafo che vengono utilizzate per definire un modello generativo. In un altro lavoro, eliminiamo l'assunzione di spazio intrinseco comune generalizzando un kernel basato sulla divergenza di Jensen-Shannon. Infine, investighiamo l'utilizzo di

tecniche di matching multi-grafo nel contesto di kernel sui grafi e come é possibile trasformare un qualsiasi algoritmo per il matching in un algoritmo per il matching multi-grafo transitivo.

Contents

List of Figures

List of Tables

Published Papers

[1] Andrea Torsello, Andrea Gasparetto, Luca Rossi, Lu Bai, Edwin R. Hancock Transitive State Alignment for the Quantum Jensen-Shannon Kernel-Structural, *Syntactic, and Statistical Pattern Recognition, (S+SSPR)*, 2014

[2] Michele Schiavinato, Andrea Gasparetto, Andrea Torsello Transitive Assignment Kernels for Structural Classification *Third International Workshop on Similarity-Based Pattern Recognition, (SIMBAD)*, 2015

[3] Andrea Gasparetto, Andrea Torsello A Statistical Model of Riemannian Metric Variation for Deformable Shape Analysis *The IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 2015

[4] Andrea Gasparetto, Giorgia Minello, Andrea Torsello Non-parametric Spectral Model for Shape Retrieval *International Conference on 3D Vision, (3DV)*, 2015

[5] Andrea Albarelli, Luca Cosmo, Filippo Bergamasco, Andrea Gasparetto Objective and Subjective Metrics for 3D Display Perception Evaluation *International Conference on Pattern Recognition Applications and Methods (ICPRAM)*, 2015

[6] Andrea Gasparetto, Giorgia Minello, Andrea Torsello A Non-parametric Spectral Model for Graph Classification *Proceedings of the International Conference on Pattern Recognition Applications and Methods, (ICPRAM)*, 2015

[7] Andrea Gasparetto, Luca Cosmo, Andrea Torsello, Richard Wilson Non-Rigid Dense Bijective Maps *International Conference on Pattern Recognition, (ICPR)*, 2016

1
Introduction

In 1959, Arthur Samuel defined machine learning as a "*Field of study that gives computers the ability to learn without being explicitly programmed*" [158]. In fact, the machine learning field can be seen as the answer given by artificial intelligence researchers on how a machine could learn from data.

The modern approach to the machine learning is closely related to statistics, and provide useful tools in the definition of models that can be used to "understand" data. In these years, the researchers produced a large amount of different approaches that can be classified as machine learning approaches. Decision tree learning is an example of such approaches in which a predictive model is build using a decision tree which maps observations about an object to conclusion about the object's target value. Artificial neural networks are another interesting example of learning algorithm that is inspired by the structure and functionalities of biological neural networks (*i.e.* brain simulation). Several other tools can be counted in the machine learning toolbox, from clustering techniques, Bayesian networks, dictionary learning and genetic algorithms. Most of them will be introduced in the literature review chapter, while the others are just tangential to the objective of this thesis and thus will not be covered.

The majority of the research in the machine learning field concerns with classification and regression for real-valued data which can easily be embedded into the Euclidean vector space. Unfortunately, the real world does not work always so straightforwardly. Indeed, in real world problems data are often highly structured, characterized by combination of features (*e.g.*, natural language and speech processing), a sequence of symbols (*e.g.* bioinformatics), a mixture of different modalities, may have missing variables, *etc.*. The data represented in such structured way are often referred to as non-vectorial data, and the items which belong to this category can be one dimensional structures (*e.g.* sequences), two dimensional (*e.g.* images), three dimensional (*e.g.*molecular descriptions), trees (*e.g.* xml documents), or other hybrid and not-so-easily classified data structures.

More generally, objects which are not given in a vectorial form, and thus objects that do not have an intrinsic-space, fall in the non-vectorial data class. The first issue which raises from the adoption of a non-vectorial representation for the data is the need of a different way to process them and different learning techniques must be employed. To address this problem, various methods have been proposed. A common approach is to embed the structures in Hilbert spaces (*e.g.* via Kernels), or try to extract and select

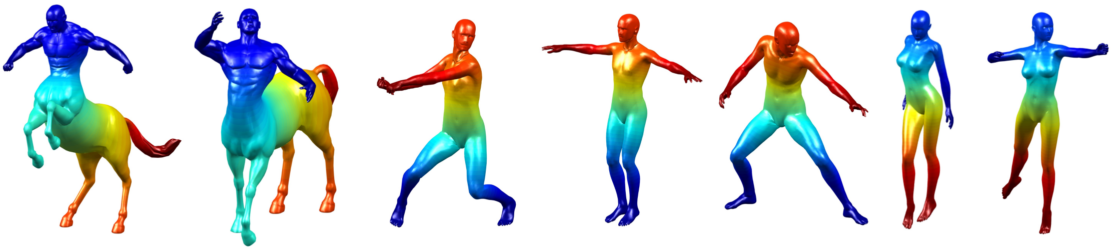


Figure 1.1: Example of surfaces used in the shape retrieval or shape registration tasks. Here, each vertex is coloured with the corresponding component of the first non-trivial eigenvector of the mesh Laplacian computed on the surface itself.

features (and their embedding spaces). Furthermore, proximity based approaches, parameter constraints in Graphical Models, Inductive Logic Programming, Decision Trees, or hand-crafted models have been applied to tackle the non-vectorial data problem by the community of researchers.

The main goal of this thesis is to devise novel learning techniques which are able to deal with data that are represented in a non-vectorial fashion. In particular, we deal with this problem from two points of view through the definition of generative models to be used in classification tasks and explore several approaches to embed non-vectorial data in some common intrinsic space.

In the first part of the thesis we review the literature relevant to the work proposed in the subsequent parts. As mentioned above, even if the main goal is to deal with non-vectorial data in all their forms, here we focus mainly on classification and embedding tasks. In particular, we introduce learning techniques whose input are both shapes and relational graphs (structure-based represented data). Classification methods applied to the surface world are often called *shape retrieval methods*, and regard the ability to retrieve similar 3D objects given a query object. Furthermore, the objects yielded as output should be ordered by *similarity*, which implies that some kind of metric is defined in order to compute numerically how similar two surfaces are. An example of surfaces representing the database to be queried is shown in figure 1.1. Assuming the first shape (centaur on the left) to be the query object, a good shape retrieval algorithm should yield the second shape (the same centaur in a different pose) as the first object of the result set since the similarity computed between the two centaurs should be much higher with respect to the one computed with the other surfaces. Indeed, the first two surfaces represent the same "shape", just in different poses. In the rest of the thesis we will denote with shape the quotient group of surfaces modulo isometries. We refer to surfaces when we want to denote a specific exponent of such group (*i.e.* a shape in a particular pose). More formally, a surface is defined as a smooth compact 2-manifold $\mathscr{S}$ without boundary isometrically embedded in the Euclidean space $\mathbb{R}^3$ with geometry induced by the embedding. Unfortunately, such manifold is usually unknown or too hard from a dimensional point of view to deal with. Hence, a discrete approximation of such surface which takes the name of mesh

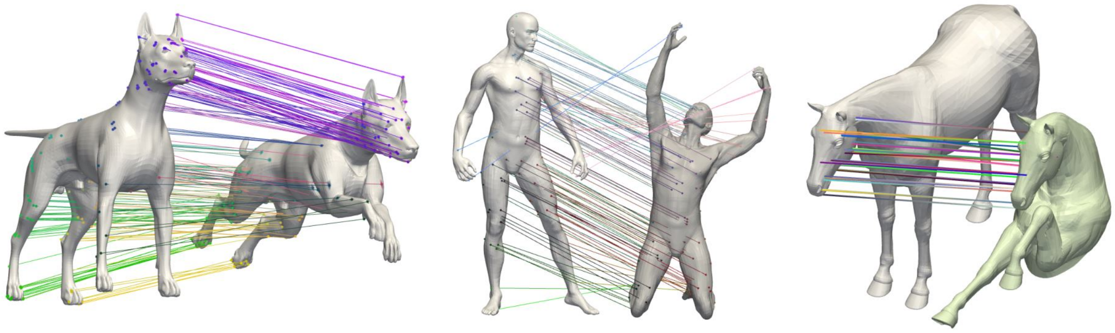


Figure 1.2: Example of a sparse set of correspondences between three pairs of surfaces.

is usually used. It approximates the surface as a triangulated set of vertices.

One of the first problem that must be faced when defining an algorithm that deals with surfaces is the dimension of the problem space. As said above, a mesh is the embed of a surface in $\mathbb{R}^3$. In order to get a good (*i.e.* fine) approximation of the underlying surface, a high number of vertices is needed. But if the number of vertices is too high, the algorithm developed to process them becomes too slow. As often happens in this field of research, the code word here is *compromise*.

Another problem regards the number of classes of transformation that surfaces may undergo. In the simpler scenario, surfaces are all the same up to isometries (*i.e.* rotation, translation, scale changes, *etc.*). In the more challenging setting of non-rigid surfaces, the distances between points are not preserved. Transformations like dilation and stretch are an example of non-isometric transformations. As a result, when devising techniques to deal in this setting through the definition of an embedding method or defining a new similarity measure, the classes of transformation by which the defined metric should be invariant must be chosen accurately.

In this thesis we deal with the shape retrieval problem through the definition of two generative models. Generative models come from probability and statistics and specify a joint probability distribution over observations. In particular, a generative model answer to the question "*how could I generate the data of a certain set of observation?*". Generative models are characterized by a set of parameters which are learned from a training set of data during a learning phase. If employed in the classification task, such parameters are used to define a posterior probability used to assign an observation to the class which yields the highest probability. Generative models are widely used when data are in vectorial form, but due to the problems outlined above, it is difficult to construct vectorial representation out of non-vectorial ones. In particular, the lack of correspondences between the points of two surfaces is the problem to address.

This leads to the second problem this thesis deal with, namely the shape matching problem. The problem can be defined as the finding a meaningful relation (*i.e.* mapping) between points of two or more surfaces. In a discrete setting, this could be interpreted as list of correspondences between vertices of two meshes. Obviously, the problem is not restricted to the discrete case, but is often defined in a continuous setting in which we seek for a continuous smooth map between surfaces. More formally, given two manifolds $\mathcal{N}$ and $\mathcal{M}$, we seek for a map $T : \mathcal{N} \to \mathcal{M}$ such that $y = T(x)$, $\forall x \in \mathcal{N}$ and $\forall y \in \mathcal{M}$. An example of the process is shown in figure 1.2. In particular, the image shows a set of sparse correspondences between three pairs of surfaces. From a practical point of view, the solution to this problem boils down to the formulation of an optimization process which usually try to minimize some intrinsic distortion induced by the mapping. Thus, the distortion metric used in the process is central for the success of the proposed approach, and the same observations about distortion metrics made before still holds. Finally, note that the intrinsic distortion induced by a mapping could be used as a similarity measure of surfaces, hence shape registration algorithms are often employed as part of a shape retrieval pipeline.

In the second part of the thesis we present our result on the shape analysis field. In particular, we present two generative models to deal with the shape retrieval problem. Finally, we present a novel technique based on the functional maps.

The first attempt at solving the shape retrieval problem is showed in Chapter 5. It is defined as a supervised data-driven approach whose main goal is the construction of a generative model based on the spectral decomposition of the Laplace-Beltrami operator of a discrete surface. The statistical framework defined in the chapter models a shape as two independent models for the eigenvector and the eigenvalue components of the spectral representation of the Laplacian, making no assumption about the distribution of the spectral embedding space of the eigenvector part and estimating it through kernel density estimation. Moreover, the eigenvalues are assumed to be log-normally distributed.

In Chapter 6 we show another supervised learning method for the definition of an invariant representation of an arbitrary shape. The main contribution is the definition of an efficient intrinsic metric to approximate the geodesic distance between points over a manifold. Furthermore, we propose an approximative approach to deal with the lack of correspondences between surfaces. In the experimental section we show how the performance of the proposed approach are tied to the quality of the approximate solution to the registration task.

Finally, in Chapter 7 we tackle the surface registration problem directly, defining a novel approach to the computation of dense correspondence maps between shapes in a non-rigid setting. The problem is defined in terms of functional correspondences. One problem with the functional map framework is that the correspondences are only intrinsically captured by the map between the functional spaces and the extracted transformation between functions is not guaranteed to correspond to a bijective point-wise mapping between points in the two shapes. A symptom for this problem is given by the fact that impulse functions on one surface are mapped onto diffuse functions over the other shape, resulting in a fuzzy assignment. Further, even selecting the maximum of the mapped function over the target shape as the corresponding point as suggested in the original work, this results in a non-injective set of correspondences, with several correspondences clustered in relatively few points on the target shape (see Figure 1.3). We deal with

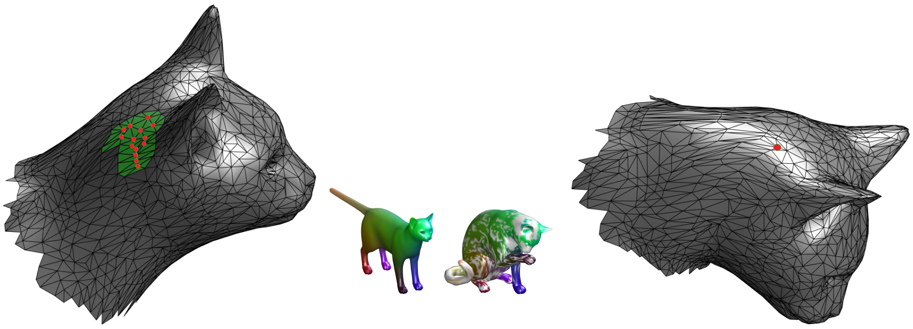


Figure 1.3: Example of non-injectivity of the functional map solution. Here, the maximum of the mapped function over the target shape is selected as the corresponding point, resulting in many-to-one relation between the mesh and the target one.

the non-injectivity of the solution of the functional map framework due to the underdeterminedness of the original problem. Key to the approach is the injectivity constraint plugged directly into the problem to optimize, achieved by casting it as an assignment problem. This leads to an iterative process which yields a high quality bijective map between the shapes. In the experimental section we present both quantitative and qualitative results, showing that the proposed approach is competitive with the current state-of-the-art on quasi-isometric shape matching benchmarks.

In the third part of the thesis we tackle the non-vectorial data problem from the structured-data point of view. In order to do so, we propose a generative model to tackle the graph classification task. Such model is detailed in Chapter 8. In particular, we introduce a novel model of structural representation based on a spectral descriptor of graphs which lifts the one-to-one node-correspondence assumption and is strongly rooted in a statistical learning framework. The generative model is defined as the combination of two separate models, one for the eigenvalues and one for the eigenvectors. We assume these models to be independent, and we tackle the graph classification task through the definition of a posterior probability whose parameters are learned from a training set (supervised learning).

In Chapter 9 we show how to generalize a structural kernel based on the Jensen-Shannon divergence between quantum walks over the graph. In particular, we introduce a novel alignment step which, rather than permuting the nodes of the structures, aligns the quantum states of their walks. We prove that the resulting kernel maintains the localization within the structures, but still guarantees positive definiteness. We test our kernel against a number of alternative graph kernels and we show its effectiveness in a number of structural classification tasks.

In the final chapter of the third part of this thesis we investigate the use of multi-graph matching techniques in the context of graph kernels. By forcing the correspondences between the structures under study to satisfy transitivity, we obtain an alignment kernel that, not only is positive definite, but also makes use of more reliable locational information obtained through the enforcement of global consistency constraints. We proposed a general framework for the projection of (relaxed) correspondences onto the space of transitive correspondences, thus transforming any given matching algorithm to a transitive multi-graph matching approach. The resulting transitive correspondences where used to provide an alignment-based kernel that was able to both maintain locational information and guarantee positive-definiteness. Experimental evaluation shows that the projection onto the transitive space almost invariably increases the classification performance of the alignment kernel, often taking it to a performance level that is at least statistically equivalent to the best performing well-tuned graph kernels present in the literature.

Finally, in Chapter 11 we give our conclusions. In particular, we give a summary of the contributions we have made in dealing with non-vectorial data, which is the common denominator of this thesis. Then, we show a selection of future work that we would like to pursue in the near future.

I

Literature Review

2
Related work

As introduced in previous chapter, the machine learning field can count on a large arsenal of techniques that deal with classification and regression. Unfortunately, the majority of these techniques are defined on real-valued data which can be easily embedded into the Euclidean vector space. Clearly, the world is often far more complex than a vector, and graphs and manifolds are ubiquitous in several research fields, like language processing, bioinformatics or shape analysis. All the problems studied in these fields have a common denominator: they all involve some sort of non-vectorial data.

The goal can be achieved in several ways. For example, a first approach consists of defining techniques directly in the space of the data (*e.g.* graph space). Even if this seems to be the more straightforward solution, there are some issues that restrict the application of such methods. The first one is the dimension of the problem space. For example, search-based deterministic approaches defined to solve the shape matching problem (*i.e.* find an injective map between two shapes) can be applied only to very small shapes. The second issue is related to the lack algebraic properties in those spaces, despite the numerous theoretical results obtained on graphs, for example.

Alternatively, we can define techniques that allow to *embed* non-vectorial data into a vectorial space. One of the drawbacks is the "representativeness" of the embedded data. Indeed, an important aspect to consider when using a technique to embed data in a vectorial space is the amount of information preserved and lost. Usually, different techniques tend to preserve different aspects (portions) of the original data. For example, a technique may focus on preserving the global structural information of the data, while another could focus on local characteristics. Another issue that we are going to deal with when "selecting" the embedding technique to use in the solution of a problem is the similarity metric we are going to employ. Indeed, an embedding technique is worth as much as how discriminative can be the embedded data. The key point of an embedding technique is the definition of the injective map that project the data into another space and some metric that allows to define what similar means in the problem we are dealing with. For example, with a feature extraction technique applied to a graph, we are able to represent a graph as a feature vector. The more similar (for some similarity function) a set of feature vectors (extracted from a set of graphs) are, the less discriminative (with respect to that similarity function) the embedding technique result. So, the problem involves much more than the mere injective map to be used. On the other hand, the definition of an embedding technique into a vectorial space allows to use the whole (and huge) machine learning framework that have been studied and perfected for years by the researcher community.

In this part of the thesis, we introduce some of the most interesting techniques that deal with non-vectorial data. In particular, we present different techniques, some dealing with graphs while others with shapes. The trivial difference between these techniques are the space in which they are defined. For example, shape analysis techniques exploit the properties of topological spaces and the branch of mathematics that studies problems in geometry (*i.e.* differential geometry). This is due to the fact that a shape (*i.e.* surface) is formally defined as a smooth compact 2-manifold without boundary with a neighbourhood that is homeomorphic to the Euclidean space of dimension 2. Furthermore, the surface can be properly embedded within a three-dimensional real space. Hence, we can draw a parallel with the paragraphs above and define both techniques that work directly on the manifold or on the embedding space (with all the issues mentioned above).

Most of these techniques will not work when applied to the graph space, since graphs are mathematical structures used to model pairwise relations between objects. Graphs are usually defined as ordered pair of a set of vertices and a set of links among these vertices (*i.e.* edges), and since the order over the vertices of a graph is arbitrary and the number of vertices in each sample graph may vary, dealing with them is often problematic.

However, there are some frameworks that can be used to deal with both graph spaces and shape spaces. Spectral theory is an example of such frameworks. Spectral theory is a branch of linear algebra that deals with the spectrum of a matrix (*i.e.* the results of an eigendecomposition). The eigen-system of the Laplacian matrix is used, for example, to compute the heat kernel (*i.e.* the fundamental solution to the heat equation). The study of the evolution of the heat kernel over time has been used effectively to characterize the object from which it has been computed. Since the Laplacian matrix is defined in both a graph and a shape version (graph Laplacian and mesh Laplacian respectively), the study of the heat kernel has been used to characterize both shapes and graphs.

In the next two chapters we will propose a review of works related to the goals of this thesis. In particular, in Chapter 3, several shape analysis approaches are presented. Here, we will focus on methods that attempt to solve the shape matching problem (*i.e.* define a correspondence map between two shapes) and methods that deal with the shape retrieval problem. In Chapter 4, we will review some of the most interesting approaches that deal with graphs, like graph features extraction methods, graph similarity metrics and graph matching problem solutions.

3
Shape Analysis

In this chapter we will review literature relevant to the work that is described in the second part of the thesis. In particular, we will explore several approaches that deal with the shape matching problem and the shape retrieval problem. Central in this topic is the definition of shapes. A shape is usually defined as a surface which is invariant to rigid and non-rigid transformations, or, quoting D.G. Kendall [100], a shape is '*what is left when the differences which can be attributed to translations, rotations, and dilatations have been quotiented out*'. A representation of such shape is modelled as a smooth compact 2-manifold without boundary isometrically embedded in the Euclidean space $\mathbb{R}^3$ with geometry induced by the embedding. Unfortunately, the underlying surface is usually unknown (or too complex to work with). Hence, a discrete representation of a particular instance (*i.e.* a particular pose) is used. It is usually called 3D mesh, a triangulated set of points that approximates the surface. In order to get a faithful approximation of the underlying surface, a finer discretization (*i.e.* a discrete representation with a high number of faces) is needed. Unfortunately, this also increase the dimension of the problem space. Indeed, this is one of the first issues that must be faced when working in the shape analysis field. Different approaches deal in different ways to the dimension problem. For efficiency purposes, the meshes are usually simplified in order to work with more manageable data structures. Furthermore, an entire set of meshes representing the same shape in different poses could be needed (in particular if we are dealing with the shape retrieval problem) in order to approximate the shape as defined above. An explanatory example is represented by the human body shape. To understand the shape of a single subject, and in particular to understand the shape of an articulated object, a set of meshes depicting such shape must be used in order to catch the possible variations by which the shape can undergo. Figure 3.1 shows an example of such set. In particular, it shows two shapes (both human being, a female on the left and a male on the right) represented through a set of four meshes of the shape in different poses.

The shape retrieval problem regards the ability to retrieve similar 3D objects given a query object. It has become of great importance in several field, like medical research, automatic information retrieval systems and copyright protection. The main challenge in 3D object retrieval algorithms is to define an invariant representation of a shape capable of capturing geometrical and topological properties of a shape [20, 147, 175]. A large number of methods for 3D shape retrieval have been proposed [10, 98, 127, 140, 167], but

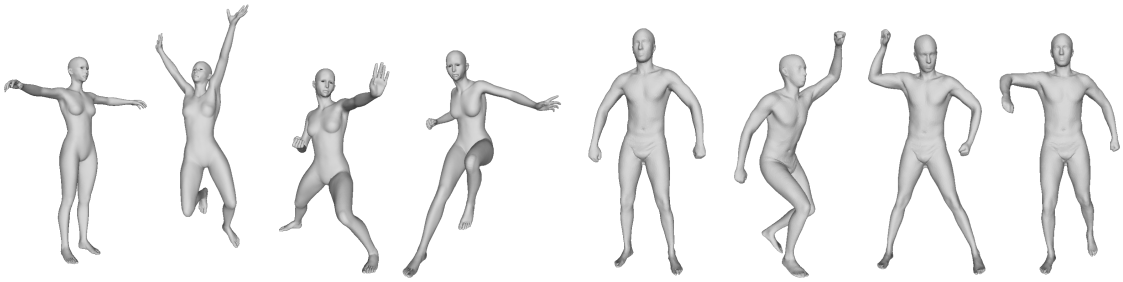


Figure 3.1: Example of two shape (female and male) in different poses, represented by a set of four discrete representation of the shape.

most of them are only suitable for the representation of rigid 3D shapes. Non-rigid 3D shape descriptors are more challenging to define. Indeed, those representations still need to capture the most distinctive properties of a shape, but they should be also insensitive to many other transformations (*e.g.* inelastic transformation, acquisition resolution, noise, *etc.*).

On the other hand, methods that attempt to solve the shape matching problem usually try to retrieve a map between two meshes. It is important to note that the map is not necessarily defined between vertices of the two meshes, but could be, for example, a continuous map between the meshes. Several successful approaches have been proposed in the rigid setting, while in the non-rigid one the problem remains difficult, even if we limit the space of deformations to approximate isometries. This is due to the fact that the rigid matching allows to represent the deformations compactly (*i.e.* a rotation and a translation), while in the non-rigid setting the problem is cast in terms of correspondences (of points or regions), rendering the space of the solution untreatable. Nevertheless, several popular approaches have been proposed in the previous ten years. These approaches can be divided into sparse [32, 102, 142, 151] and dense [103, 141]. Sparse approaches strive to extract a few good point correspondences between the shapes under exam and, in general, trade completeness for precision, while dense approaches strive to extract the full bijection between isometric shapes.

Even if the two problem have significant differences (in both the definition and the approaches employed in their solutions), it is important to note that they are tightly tied since the shape retrieval problem, which regards the clustering of shapes in classes, can make use of shape matching techniques to define similarity metrics. Furthermore, shape matching methods are often employed as a pre-processing step in many shape retrieval methods [69,153]. Indeed, one of the main issues when defining a shape retrieval problem is the unknown correspondence map between two meshes, since it prevents the definition of compact representatives that lie in the same intrinsic space.

3.1 Shape Retrieval

The ability to retrieve similar 3D objects given a query object has become of great importance in several field, like medical research, automatic information retrieval systems and copyright protection. The main challenge in 3D object retrieval algorithms is to define an invariant representation of a shape capable of capturing geometrical and topological properties of a shape [20, 147, 175]. A large number of methods for 3D shape retrieval have been proposed [10,98,127,140,167], but most of them are only suitable for the representation of rigid 3D shapes. Non-rigid 3D shape descriptors are more challenging to define. Indeed, these representations still need to capture the most distinctive properties of a shape, but they should be also insensitive to many other transformations (*e.g.* inelastic transformation, acquisition resolution and noise *etc.*).

Various methods have been proposed to address the non-rigid shape retrieval problem. These can be mainly categorized according to two main directions: topology-based approaches against surface-based methods and the employment of shape descriptors that can be local or global.

3.1.1 Surface-based methods

With surface-based methods we denote the approaches that try to represent a shape as a frequency histogram of deformation invariant local distances or vertex signatures. Surface based methods are quite popular in the research community due to their highly effective representation ability and low memory requirement. In this section we will deal in particular with methods that exploit spectral shape analysis in order to build a spectral descriptor.

Rustamov [155] propose a deformation invariant representation of surfaces, called the *GPS embedding* (***G****lobal* ***P****oint* ***S****ignature*. It uses the eigen-system of the Laplace-Beltrami differential operator. In previous works, the geodesic distance has been used to define a deformation invariant representation, since the geodesic distance is somewhat invariant to natural articulation of shapes. Unfortunately, geodesic distance is sensitive to topology changes. The GPS embedding is a possible answer to this problem and is based on combining the Laplace-Beltrami eigenvalues and eigenfunctions. The Laplace-Beltrami operator is a natural choice for this application since its eigenfunctions are intrinsic in the sense that allows to associate values to points on the surface that are independent on the surface embedding. Given a point p on the surface, the *Global Point Signature* $GPS(p)$ is defined as the infinite-dimensional vector

$$GPS(p) = \left(\frac{1}{\lambda_1}\phi_1(p), \frac{1}{\lambda_2}\phi_2(p), \frac{1}{\lambda_3}\phi_3(p), \dots\right)$$

where $\phi_i(p)$ is the value of the eigenfunction ϕ_i at the point p. The first eigenfunction, which is associated with the eigenvalue 0, is the constant one, hence it is left out in the signature definition. GPS can be seen as a map of the surface into an infinite-dimensional space. The image of such map is what the author call the *GPS embedding* of the surface.

GPS embedding owns several properties. First, GPS embedding is an isometry invariant, which means that two isometric surfaces will have the same image under the GPS mapping. Second, the GPS embedding is absolute in the sense that it is not subject to rotations or translations of the ambient infinite-dimensional space.

From a more practical point of view, the computation of the GPS and its image is made within a discrete setting. In particular, the eigendecomposition is performed on the discrete Laplacian defined in [197]. Once the GPS embedding is computed, it is subdivided into adjacent patches which are used to compute the histogram of pairwise distances between the points sampled from the surface. The result is a set of histograms that capture the distribution of distances between points one of which belongs to one inter-shell patch, the other to another patch. In the experimental section, the author shows the behaviour of the proposed signature in a shape classification task.

Sun *et al.* [174] propose a point signature based on the heat kernel. It takes the name of Heat Kernel Signature (HKS), and is obtained by restricting the heat kernel to the temporal domain. The heat kernel is the fundamental solution to the heat equation. Let M be a compact Riemannian manifold. The heat diffusion process over M is governed by the heat equation

$$\Delta_M u(x,t) = -\partial u(x,t)$$

where Δ_M is the Laplace-Beltrami operator of M. Let $f : M \rightarrow \mathscr{R}$ be the initial heat distribution. Let $H_t(f)$ be the operator that satisfies the heat equation for all t. Furthermore, for the spectral mapping theorem, $H_t(f) = \exp\{-t\Delta_m\}$, meaning that the Laplace-Beltrami operator and the heat operator share the same eigenfunctions and if λ is an eigenvalue of Δ_M, then $e^{-\lambda t}$ is an eigenvalue of H_t corresponding to the same eigenfunction. For any manifold in the form of M, there exists a function $k_t(x,y) : \mathscr{R}^+ \times M \times M \rightarrow \mathscr{R}$ such that

$$H_t f(x) = \int_M k_t(x,y) f(y) dy$$

The minimum function that satisfies the above equation is called the *heat kernel*. The heat kernel represents the evolution of temperature in a region whose boundary is held fixed at a particular temperature. The eigendecomposition of the heat kernel is expressed as

$$k_t(x,y) = \sum_{i=0}^{\infty} \exp(-\lambda_i t)\phi_i(x)\phi_i(y)$$

where λ_i and ϕ_i are the i^{th} eigenvalue and eigenfunction of the Laplace-Beltrami operator Δ_M. The heat kernel has many useful properties. For example, it is symmetric, isometric invariant and very stable. Refer to the original work [174] to get a comprehensive list of the properties and their proofs.

As mentioned above, in order to get a concise feature descriptor, the authors restrict the heat kernel to the temporal domain. As a result, the heat kernel signature is defined as follow

$$k_t(x,x) = \sum_{i=0}^{\infty} \exp(-\lambda_i t)\phi_i^2(x)$$

Unfortunately, like in the work proposed by Rustamov [155] and introduce above, the underlying manifold of the object is not known. Instead, a discrete representation of the shape is used (*i.e.* a mesh). In this setting, the HKS can be computed (approximated) starting from a discrete representation of the Laplace-Beltrami operator. In their work, the authors follow the algorithm proposed by Belkin *et al.* [16], since it offers point-wise convergence guarantees and was experimentally shown to be quite robust. Let L be the mesh Laplacian. The eigensystem is computed through eigendecomposition of L, such that $L = \Phi\Lambda\Phi^T$. In particular, Λ is a diagonal matrix of eigenvalues of L, ordered by magnitude, while Φ is the matrix of corresponding orthonormal eigenvectors. With the spectral components at hand, the heat kernel can be defined as

$$H_t = \Phi \exp(-t\Lambda)\Phi^T$$

The elements $h_t(i,j)$ represents the heat diffusion between vertices i and j after time t. The HKS is then given by the diagonal entries of this matrix, sampled at discrete time intervals.

The HKS presents some limitations. The first one is related to the presence of the time parameter, which is chosen heuristically but with no guarantees. Furthermore, HKS is able to efficiently characterize a shape up to isometries if the eigenvalues of the surface are non-repeating. This is the case of shapes with internal symmetries. Finally, the performance guarantees for HKS only hold for truly isometric transformations. However, deformations for real shapes are often not isometric (*i.e.* topology changes).

In another seminal work, Aubry *et al.* [12] propose the *Wave Kernel Signature* (WKS). Like in the work described above, the goal is to characterize points on non-rigid shapes. The WKS is based on the quantum mechanics theory, since it represents the average probability of measuring a quantum mechanical particle at a specific location. The trick here is to let the energy of the particle to change, which allows the WKS to encodes and separates information from various different Laplace eigenfrequencies.

Like in the definition of the HKS, the basic idea is to characterize a point x by the average probabilities of quantum particles of different energy levels to be measured in point itself. The evolution of a quantum particle on the surface is governed by the wave function $\psi(x,t)$, which is a solution to the of the Schrödinger equation:

$$\frac{\partial\psi}{\partial t}(x,t) = i\Delta\psi(x,t) \tag{3.1}$$

The huge difference between the wave equation and the heat equation lies in the induced dynamics, oscillations in the former and dissipation in the latter. A common limitation between the two approaches is represented by the assumption that the Laplace eigen-system of the shape has no repeated eigenvalues. In this case, the wave function is defined as

$$\psi_E(x,t) = \sum_{i=0}^{\infty} e^{iE_k t} \phi_k(x) f_E(E_k)$$

where f_E is an energy probability distribution with expectation value E. The authors choose to not consider the time parameter since it has no straightforward interpretation in the characteristics of the shape. Hence, they define the WKS in terms of the average probability over time to measure a particle in x, getting to the final (refer to the original work for the details) function definition:

$$WKS(E,x) = \sum_{i=0}^{\infty} \phi_k^2(x) f_E^2(E_k)$$

By fixing a family of these energy distributions $f_i(x)$, the WKS can be obtained as a discrete sequence $\{p_{f_1}(x),\ldots,p_{f_n}(x)\}$.

Another interesting result is the proposed stability analysis of eigenenergies. In particular, they show that the eigenenergies of an articulated shape are log-normally distributed random variables. This results has been used in several contributions of this thesis (*i.e.* [3,4]). In the experimental section, the authors show that the WKS allows for more accurate feature matching even when the data are strongly perturbed.

An extension to the HKS has been proposed by Bronstein and Kokkinos [33]. They propose a scale-invariant version of the heat kernel descriptor. The construction of this new signature is based on a logarithmically sampled scale-space in which shape scaling corresponds to a translation up to a multiplicative factor. The Fourier transform of this HKS changes the time-translation into the complex plane, and the dependency on translation can be eliminated by considering the modulus of the transform.

In particular, given a shape S and given a scale parameter β, the HKS can be define as

$$h_S(x,t) = \sum_{i=0}^{\infty} e^{-\lambda_i \beta^2 t} \phi_i^2(x) \beta^2 = \beta^2 h_S(x, \beta^2 t)$$

The local normalization of the heat kernel signature defined in this work begin by sampling the heat kernel signature logarithmically in time (*i.e.* $t = \alpha^\tau$) for each point of the shape. Then, in order to remove the multiplicative constant from the equation above, the discrete derivative of the logarithm of h_S is computed, turning the multiplicative factor into an additive constant which then vanish in the differentiation. Finally, taking the modulus of the Fourier Transform of the result of the derivative, we obtain the scale-invariant quantity, which takes the name of *SI-HKS*. For the numerical computation of the heat kernel signature, a discrete version of the Laplace-Beltrami operator is used. In particular, the authors rely on the cotangent weight scheme [145]. The experimental section presents results in both shape retrieval and shape matching tasks on the ShapeGoogle Database [31], a popular non-rigid shape retrieval benchmark.

Gong *et al.* [73] propose two kinds of semantic signatures. The key idea is that humans are capable of describing an object using attributes like “symmetric” or “flyable” or using its similarities with other objects. Hence, in this work the authors propose to translate such qualitative descriptions into *attribute signature* and *reference set signature* and use them for the shape retrieval task. The former is defined as a set of attributes (*i.e.* symmetry, flexibility, rectilinearity, circularity, dominant-plane, long, thin, swim, fly, stand with leg and natural, all associate to a binary value). For each attributes a detector (a binary classifier) is trained on a training set (supervised training). The detector will yield a probability measure of the presence of a certain attribute given an object. To train the detector, LIBSVM is employed with an RBF kernel. Since the svm require a vectorial representations of the objects, the authors used the depth buffer descriptor, the wavelet transform and the mutual absolute-angle distance histogram as complementary shape features. Then, the attribute signature is built concatenating detectors output.

$$AS(x) = \{(p(a_i|x)\}_{i=1}^{11}$$

In order to compare two objects x and y, a dissimilarity measure is defined as the weighted sum of a divergence measure:

$$d(x,y) = \sum_{i=1}^{11} w_i f(p(a_i|x), p(a_i|y))$$

where $f(p_x, p_y)$ is the symmetric Kullback-Leibler divergence.

On the other hand, the reference set signature is based on the idea that humans often describe an object by similarities with respect to other known object (or class of objects). Let $C = \{c_j\}_{j=1}^{n}$ be the set of considered classes, and let $s_j(x)$ be the similarity of the object x with respect to class j. Then the reference set signature of the shape x is defined as

$$RSS(x) = (s_1(x), s_2(x), \ldots, s_n(x))$$

The similarity function is compute again with LIBSVM using the same shape features mentioned above. In the experimental section the authors assess the quality of the defined signature over two shape retrieval datasets.

Reuter *et al.* [149], in the seminal work know as “Shape-DNA”, propose a signature (also called fingerprint in the work) of a two or three dimensional manifold (*i.e.* a surface or a solid) by taking the eigenvalues of its Laplace-Beltrami operator. In particular, it is the normalized beginning sequence of the eigenvalues of the Laplace–Beltrami operator. Differently from the approaches seen before, here the authors are not casting the proposed method in a discrete setting by computing the spectrum in the mesh Laplacian. Since the spectrum is symmetry invariant, it is independent of the object’s representation (both parametrization and spatial position). Furthermore, uniform scaling factors can be obtained easily by eigenvalues normalization. Therefore, in order to compare two objects, only the spectrum is needed. The resulting fingerprint consist of a family of non-negative numbers (*i.e.* eigenvalues) that can be compared easily and fast, allowing to employ the method to time critical task like database retrieval. In this work, the input is represented by a parametrized shapes (*e.g.* NURBS curves) on which the Laplace-Beltrami operator is computed. The eigenvalues of such differential operator are computed using the Finite

Element Method. Like the DNA, this signature is not guaranteed to be unique, but it is still capable of performing really well in the task of shape retrieval.

El Khoury *et al.* [101] propose a solution to the shape retrieval problem based on indexed closed curves generated from an invariant mapping function defined on the mesh using the commute time distance. The first step involves the detection of feature points in the shape. These points will then be used in order to define a scalar function based on the commute time distance. The diffusion distance is the metric used to extract feature points. The diffusion distance is computed as the Euclidean distance but in the spectral embedding space. In particular, given the time parameter t and $K(t,x,y)$ the heat kernel between the points x and y, the diffusion distance is defined as

$$d_S(t,x,y) = \|K(t,x,x) - K(t,y,y)\|_{L_2(S)}$$

A point is a feature point if the diffusion distance with respect to its one-ring neighbourhood is the lower.

With a set of feature points at hand, the mapping function can be defined using the commute time distance. Let ψ_i and λ_i be the i-th eigenfunction and eigenvalue computed on the discrete Laplacian of the shape. The commute time distance is define as

$$dc_S(x,y)^2 = \sum_{i=1}^{\infty} \frac{1}{\lambda_i} \left(\psi_i(x) - \psi_i(y)\right)^2$$

Hence, the mapping function F_m is defined for each vertex of the mesh as the maximum commute time distance between the point and all the feature point defined above. The minimum of such function corresponds to the farthest vertex among all feature points, which is then used to generate indexed closed curves. The value of the mapping function at that point is then used as a scalar factor. Here, each curve describes a small region of the shape, and through the similarity measure defined in [94] (a metric that assesses how similar two curves are), a similarity value between the sets of curves of two shapes is produced. Finally, such similarity measure is used to address the shape retrieval task.

Jain and Zhang [88] propose to represent a shape by the eigenvectors of an affinity matrix, forming a spectral embedding invariant to rigid-body transformations, uniform scaling and shape articulation (*i.e.* bending). Given a discrete representation of a shape (*i.e.* a mesh) with n vertices, the authors aim at computing an $n \times n$ affinity matrix A such that each entry a_{ij} represents the affinity between vertex i and vertex j. The eigensystem of the affinity matrix is then computed through eigendecomposition, yielding the ordered vector of eigenvalues and the corresponding normalized eigenvectors. The eigenvectors are then scaled by the square-root of the corresponding eigenvalues (to decouple the eigenvectors value from the number of vertices of the mesh). Finally, the first k scaled eigenvectors are retrieved in order to construct the spectral embedding.

In order to compute a spectral embedding which is invariant to shape articulation, an affinity matrix based on the geodesic distance between intra-points of the mesh is employed. In their work, the authors tackle the problem of disconnected parts within a shape by defining a heuristic approach to the computation of an approximate geodesic distance matrix. Once such matrix is computed, the Gaussian affinity matrix can be built as follow

$$A_{ij} = e^{-\frac{d_{ij}^2}{2\sigma^2}}$$

where d_{ij}^2 is the approximate geodesic distance between vertices i and j while σ is the Gaussian width. In the proposed work, the authors select the maximum distance between all the vertices of the mesh as the Gaussian width. For efficiency purposes, the computation of the affinity matrix and of the eigen-system of A is achieved using Nyström approximation [60].

The first spectral descriptor that is proposed in the work takes the name *Eigenvalue Descriptor*. The eigenvalues of the affinity matrix represent the variation of the shape along the axes represented by the corresponding eigenvectors. The first issue that arises from this definition regards the spaces in which those spectral descriptors lie. Indeed, eigenvalues are affected by both mesh size and resolution. Thus, they cannot be used directly in a comparison between different descriptors. However, since the spectrum is yielded by Nyström approximation (which returns the first k eingevalues), it is sufficient to choose the same k for all the shapes in order to make the affinity matrices comparable. With the descriptor at hand, the authors define the dissimilarity measure as

$$D_{EVD}(X,Y) = \frac{1}{2}\sum_{i=1}^{k} \frac{\left(|\lambda_i^X|^{\frac{1}{2}} + |\lambda_i^Y|^{\frac{1}{2}}\right)^2}{|\lambda_i^X|^{\frac{1}{2}} + |\lambda_i^Y|^{\frac{1}{2}}}$$

The second spectral descriptor that the authors propose takes the name *Correspondence Cost Descriptor*. Here, the distance between two shapes is derived from the correspondence between the vertices of two shapes. In particular, let Φ_X and Φ_Y be the matrix with the first three eigenvectors of the shape X and Y respectively. Let $match_Y(p)$ be a function that yields the vertex of Y which is in correspondence with p. Finally, let $\Phi(i)$ be the i-th row of the spectral embedding. Then, the dissimilarity measure is defined as

$$D_{CCD}(X,Y) = \sum_{x \in X} \|\Phi_X(x) - \Phi_Y(match_Y(x))\|$$

To assess the performance of the proposed descriptors, the authors test both approaches with the McGill database [170] of articulated shapes in the shape retrieval task.

Litman and Bronstein [118] propose a parametric spectral descriptor that is learned taking into account both the statistics on the shape and the class of transformations to which the descriptor is made insensitive. In particular, they propose a data-driven approach to learn the descriptor from examples. To this end, a learning scheme based on the Wiener filter and related to the Mahalanobis metric is presented. The work seeks the solution to the drawbacks from which the heat and wave kernel descriptors suffer. The WKS is known to emphasize local-scale feature. This results in a higher *sensitivity* (*i.e.* enhanced positives identification). On the other hand, HKS emphasizes global features, which leads to a higher *specificity* (*i.e.* enhanced negatives identification). Furthermore, both descriptors suffer of overlapping elements in what the authors call the*frequency response* (*i.e.* the descriptors components).

Another drawback of both the heat and wave kernel descriptors is the fact that the frequency responses forming their elements have significant overlaps. As the result, the descriptor has redundant dimensions. Finally, the only invariance that both the descriptors assure is with respect to truly isometric deformations. This is clearly not the standard setting, since most shapes undergo non-isometric transformations, in particular along the articulations.

The authors suggest to build a descriptor which generalizes the heat and wave kernel emphasizing information-carrying frequencies while attenuating noise-carrying ones through a learning process. The descriptor takes the same form of the HKS and WKS, and can be defined as

$$p(x) = \sum_{i\geq 1} \mathbf{f}(\lambda_i)\phi_i^2(x)$$

parametrized by a vector $\mathbf{f}(\lambda) = (f_1(\lambda),\ldots,f_n(\lambda))^T$ of frequency responses. Differently from the HSK and WKS, here the $\mathbf{f}(\lambda)$ is not restricted to positive values. Let $b_1(\lambda),\ldots,b_m(\lambda)$ be a fixed basis which spans a sufficiently wide interval of frequencies (*e.g.* $[0,\lambda_{max}]$). Then the frequency response vector can be rewritten as $\mathbf{f}(\lambda) = \mathbf{Ab}(\lambda)$, where $\mathbf{A}$ is an $n \times m$ matrix of coefficients representing the response using the basis $\mathbf{b}$. Taking the first k elements of the spectrum, the descriptor definition can be rewritten as

$$p(x) = \mathbf{A}(\mathbf{b}(\lambda_1),\ldots,\mathbf{b}(\lambda_k)) \begin{pmatrix} \phi_1(x) \\ \vdots \\ \phi_k(x) \end{pmatrix} = \mathbf{Ag}(x)$$

where $\mathbf{g}(x)$ is a vector whose elements are defined $g_j(x) = \sum_{i\geq 1} b_j(\lambda_i)\phi_i^2(x)$. $\mathbf{g}(x)$, which takes the name *geometry vector*, captures all the shape-specific geometric information about the point x.

The adopted learning scheme aims at finding the coefficients matrix $\mathbf{A}$ introduced above. Given a set of classified shapes to be used in the learning process, the authors aim at finding a matrix $\mathbf{A}$ minimizing the Mahalanobis distance over the set of positive pairs, while maximizing it over the negative ones. The efficacy of the descriptor is assessed using the the TOSCA and SCAPE datasets in a shape retrieval task, with the spectrum of the shapes computed using the same finite elements scheme introduced above and used to compute the WKS.

Huang *et al.* [84] build the signature directly over local features, selecting discriminative volumetric features over pre-aligned shapes. In particular, the problem of multi-label shape classification (*i.e.* a shape can belong to multiple classes) is addressed. This is a semi-supervised method that takes a homogeneous shape collection with sparse and noisy labels as an input and yields a refined version (label-wise) of such collection.

The approach counts three different steps. Let $S = S_i | 1 \leq i \leq N$ be the shapes collection which belong to the same category. The shapes are assumed to be aligned in the same way (up-right direction along z axis) and normalized. In order to made-up the training set, the labels associated to each shapes are known. In the first step, the shapes are aligned (matched) with each other. The process is divided in two phase. The first is the *global phase*, in which an affine transformation for each shape is computed jointly. The result is a roughly aligned set of shapes. The second phase takes the name of *local phase*, and it consists of a jointly optimization of a free-form deformation for each shape in order to improve the raw alignment.

In the second step a distance metric is learned for each class. The goal here is to define a metric that allows to minimize intra-class differences and maximize inter-class differences. This stage exploits the fact that the shapes belong to the same space (first step result) and the authors define a distance metric as a linear combination of several other primitive distance metrics, each involving a different descriptor.

The final stage of the approach uses the learned distances to build a similarity graph for each class. Then, graph based multi-label classification is performed in order to assign a class (or more) to each shape. The authors propose a select-from-candidate strategy (*i.e.* pre-process a list of good candidate to be tested) in order to handle even large collections of shape. Finally, the authors show in the experimental section that the proposed method is competitive with respect to the state-of-the-art multi-label shape classification techniques.

The aggregation of local descriptors in order to build a global descriptor is a general thread in the literature. For this purpose, the *Bag-of-Features* (BoF) paradigm is quite popular and has been successfully applied to 3D shape description [31, 50, 109, 177]. Li and Hamza [114] used the BoF paradigm combining the exploitation of hierarchical structures of the shape, such as pyramid matching [74] and spatial relationship [31, 50, 109]. They proposed to adopt the eigenfunction associated with the second-smallest eigenvector of the Laplace-Beltrami operator in order to build a global surface coordinate system which is insensitive to shape deformation, showing that the introduction of global spatial context could improve the effectiveness of their descriptor in 3D shape recognition. Spatial pyramid [110, 114, 120], is the term used to identify this approach. Other approaches inspired by text-analysis have been proposed. For instance, in [22, 82] the authors adopt higher-order models defining relations between “geometric words”.

Within the bag of features model, features quantization is generally performed through unsupervised learning. Departing from this approach, Litman *et al.* [117] recently proposed a new supervised BoF framework mapping the discriminative training directly into the dictionary construction step. In particular, the authors define a task-specific dictionary learning approach designed specifically to deal with the shape retrieval task. Interestingly, both positives and negatives examples are used in the learning process of the dictionary. The whole process can be interpreted as supervised metric learning, with the key difference with respect to other approaches of this kind that the metric learning is not pursued linearly (allows to learn more complicated metrics).

Spectral local descriptors like the HKS and its scale-invariant variation (*i.e.* SI-HKS) have been already introduce above. Refer to the beginning of this chapter for their formal definition. Here, we will introduce the *Bag Of Features* approach. Let **X** be a matrix $k \times n$ collecting a set of k dimensional feature vectors, one for each point p in a mesh, such that

$$\mathbf{X} = (x_1, \ldots, x_n) = (x(p_1), \ldots, x(p_n))$$

A Bag of Feature is defined as a global descriptor of a shape built through a quantization process in which the local descriptors are replaced by the closest entries in a geometric dictionary. This first step is followed by the computation of the frequency of appearance of these *geometric words* (*i.e.* representative descriptors). Hence, the geometric dictionary is a collection (column-wise) of geometric words and can be defined as a $k \times v$ matrix $\mathbf{D} = (d_1, \ldots, d_v)$. The geometric dictionary is built commencing from a large dataset of shapes and using clustering techniques on the descriptors computed on them (*e.g. k-means*). So, given a dictionary **D** and the matrix X, the quantization process (also known a *vector quantization*) mentioned above can be defined for each local descriptor x as

$$i^* = \underset{i=1,\ldots,v}{\arg\min} \|\mathbf{x} - \mathbf{d}_i\|_2$$

Hence, the quantization process yields the index of the geometric word of the dictionary which is closest to the descriptor. The index is the used to build an indicator vector in which all components are 0 except the i-th one. As a result, the $v \times n$ matrix $\mathbf{Z}^*$ containing the indicator vector for each point of the shape. With the matrix **X**, **D** and $\mathbf{Z}^*$ at hand, the *pooling* phase can be defined. The goal is to compute the frequency of appearance of different geometric words on the shape while taking into account possible non-uniform sampling (*i.e.* weighting each point by its area element). The result is a v-dimensional vector containing such frequencies and that takes the name of bag-of-features vector.

All the steps from the descriptor matrix to the bag-of-features vector are independent with respect to each other. Since this could lead to worse discriminative performance (*i.e.* loss of invariances), the authors propose a different approach to the construction of the BoF. In their work, the authors propose to replace the quantization step with sparse coding. This results in a $\mathbf{Z}^*$ matrix in which each column has more elements with a value different from 0 which represents the magnitude of similarity with that particular geometric word. Furthermore, instead of the unsupervised dictionary learning, the authors adopt a supervised approach which maximized the end-to-end retrieval performance. In particular, the supervised learning problem is defined as a bi-level minimization problem in which even negative examples are used in the learning process. Also, this supervised learning is embedded into the same optimization problem defined to solve synthesis problem (*i.e.* the problem solved using sparse coding). This reduce the independence between the different stages and which lead to a *task-specific* dictionary that yields better performance.

3.1.2 Topology-based methods

Topology-based approaches capture the global topological structure of the shape. These methods exploit the topological relations between surface features, often coding them as graphs that can be compared through matching algorithms.

Hilaga *et al.* [80] propose a technique called *Topology Matching*, in which similarity between polyhedral models is calculated by comparing a variant of the Reeb graph build on the shape. The Reeb graph tracks topology changes in level sets (*i.e.* a set of points where the function attains a given value) of a scalar function. In particular, in their work the authors define the *Multiresolutional Reeb Graphs*. The key idea is to develop a series of Reeb graphs for an object at various levels of detail. In order to build the Reeb graph for a certain level set, the surface is partitioned into regions which depend on the function used. A new component in the graph is created when a critical point is met (*e.g.* minimum, maximum or saddle point, region centroid, *etc.*). Components that belong to adjacent regions are connected (the type of connection depends on the type of critical point). In order to obtain Reeb graphs of different resolutions, the regions are furtherly partitioned and the process is repeated. In the shape analysis field, the Reeb graph is obtained by partitioning the object surface following regular intervals of the function values, and linking the connected regions to each other. In each interval, a node is associated to a set of connected triangles (the triangles that belong to that particular region). In this work, the authors use the geodesic distance (*i.e.* shortest path over the surface between two points, computed in this work using Dijkstra's algorithm) as the function to be used to partition the shape in intervals since it is invariant to translation and rotation and it is also robust with respect to mesh simplification or subdivision (*i.e.* change in connectivity).

The resulting graph represents the skeletal and topological structure of a 3D mesh at various levels of resolution, and can be used as a search key in a shape dataset. The authors employ a coarse-to-fine (coarse resolution Reeb graphs first) strategy to compute the similarity between 3D objects, trying to establish a correspondence between their parts. The performance of the proposed method is assessed in both matching between shapes and on a small shape retrieval dataset.

Exploiting the invariance of medial descriptor (*i.e.* shock graphs) to transformation like traslation, rotation, scale, articulation and within-class deformation, Macrini *et al.* [126] propose an algorithm to compute a medial shape abstraction that deals with the ligatures instability issue. A medial descriptor describes a shape as a set of connected parts (*e.g.* limbs of a human shape), but often suffers of instability along the parts attachment. They introduce a heuristic-driven ligature analysis whose first step involves the detection of all the ligatures in the skeleton. Then, in order to address the problem that more than one skeletal segment can represent the same shape part, they define a set of rules that analyse ligature properties, detecting locations of over and under segmentation, editing the skeleton accordingly. The result is a new skeleton. In addition, they present a novel graph abstraction of medial structure that partitions the skeleton into its *bone graph*. In the experimental section, the authors assess the efficacy of the resulting bone graph in object recognition and pose estimation tasks.

Li and Hamza [112] propose a skeleton path based approach to analyse and retrieve non-rigid 3D shapes. The main idea is to match skeleton graphs by comparing the geodesic paths between skeleton end-points. In their work, the authors prove the stability of the geodesic path between two endpoints in presence of articulation-point. The approach is defined as a two-phase process.

The first phase aims at computing the skeleton paths of a shape, and begin with the curve-skeleton extraction. In particular, the authors use the algorithm proposed by Cornea *et al.* [47]. The algorithm computes the curve-skeleton by computing repulsive force field over the shape and uses topological characteristics of the resulting vector field like critical points and critical curves to define the curves (sequence of skeleton points that identify a curve) of the skeleton. With the skeleton at hand, the proposed method proceeds with the detection of the endpoints. They are simply defined as the points in the skeleton having only one adjacent point. Articulation-points, on the other hand, are points with three or more adjacent points. Connection points are those points with an adjacency degree of two. The final step of phase one is the construction of the skeleton paths, which are simply the shortest path between all pairs of endpoints detected in the skeleton.

The second phase of the proposed work is the endpoints matching between endpoints of two different skeletons. They define a matching cost based on optimal subsequence bijection algorithm [108] to build a dissimilarity matrix for each pairs of endpoints in the matched skeletons. Finally, the Hungarian algorithm [26] to compute a one-to-one correspondence map between the endpoints. To address the diverse dimensionality problem between skeletons, the authors pad the dissimilarity matrix with constant values based on the dissimilarity computed, obtaining a square matrix.

In the experimental section, the authors assess the performance of the proposed approach by applying it to symmetric components discovery, skeleton matching and shape retrieval.

3.2 Shape Matching

The shape matching problem plays a central role in computer graphics and geometry processing fields. Solutions to this problem have multiple applications, ranging from texture mapping to animation, to name just a few. Several successful approaches have been proposed in the rigid setting, while in the non-rigid one the problem remains difficult, even if we limit the space of deformations to approximate isometries. This is due to the fact that the rigid matching allows to represent the deformations compactly (*i.e.* a rotation and a translation), while in the non-rigid setting the problem is cast in terms of correspondences (of points or regions), rendering the space of the solution untreatable. The shape matching approaches that will be reviewed in this section can be categorized into three different class. In section 3.2.1 methods that deal with the sparse matching problem are presented. These methods usually trade the injectivity of the mapping with a hopefully more reliable sparse correspondence set. On the other hand, in section 3.2.2 we present a collection of methods that deal with the dense matching problem. Figure 3.2

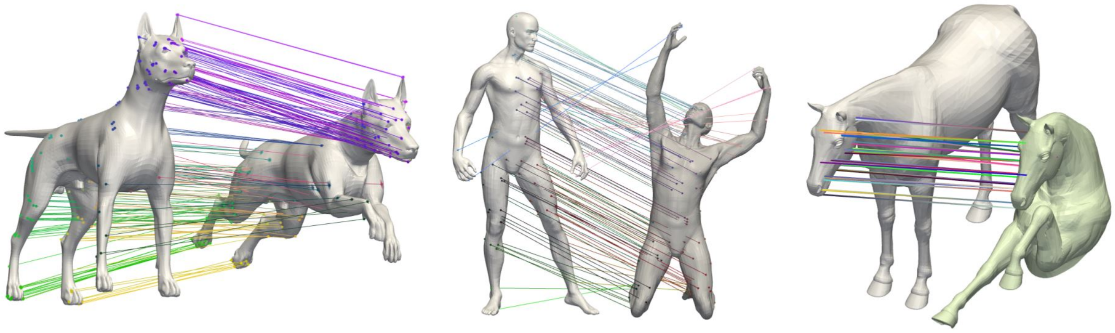


Figure 3.2: Sparse set of correspondences extracted using [151] on three different shapes.

shows some examples of sparse matching results.

3.2.1 Sparse Matching

The first approach that we present in this section is the *Generalized Multidimensional Scaling* [32] (GMDS). This method copes with both a sparse and a dense solution to the matching problem. In particular, it can be defined as a procedure that computes the map that best preserves the inter-geodesic distances while embedding one surface into another.

The input of this method are two surfaces $\mathscr{S}$ and $\mathscr{Q}$ endowed with a metric $d_{\mathscr{S}}$ and $d_{\mathscr{Q}}$ and represented by their discrete approximation $\mathscr{S}_N$ and $\mathscr{Q}_N$. Let Ψ be an isometric embedding from $\mathscr{Q}$ to $\mathscr{S}$, *i.e.* $\Psi : \mathscr{Q} \to \mathscr{S}$. The *partial embedding* distance (PE) is defined as

$$d_{PE}(\mathscr{S}, \mathscr{Q}) = \inf_{\Psi : \mathscr{Q} \to \mathscr{S}} \operatorname{dis} \Psi$$

and measures the metric distortion while using the "most isometric" embedding between $\mathscr{Q}$ and $\mathscr{S}$. Here, dis Ψ is a measure of the distortion of the isometric embedding (*i.e.* if dis $\Psi = 0$, then Ψ is an isometry while $\mathscr{S}$ and $\mathscr{Q}$ are isometric).

Let $\mathbf{D}_{\mathscr{S}_N} = d_{\mathscr{S}_N}(s_i, s_j)$ and $\mathbf{D}_{\mathscr{Q}_N} = d_{\mathscr{Q}_N}(s_i, s_j)$ be respectively the geodesic distance matrices between the vertices of $\mathscr{S}_N$ and $\mathscr{Q}_N$. In particular, in this work the *fast marching method* (FMM) is used to compute such matrices.

The first step of the GMDS algorithm involves the embedding of the sampled surfaces into an m-dimensional Euclidean space by a near-isometric embedding. This is achieved by minimizing the "generalized stress function", a stress function defined as follow.

$$\sigma_p(\mathbf{U}; \mathbf{D}_{\mathscr{Q}_N}, d_{\mathscr{S}}, \mathbf{W}) = \left(\frac{1}{\sum_{j>i} w_{ij}} \sum_{j>i} (w_{ij}(d_{\mathscr{S}}(\mathbf{u}_i, \mathbf{u}_j) - d_{\mathscr{Q}}(q_i, q_j)))^p \right)^{\frac{1}{p}}$$

where $\mathbf{U}$ represents the positions of the points of $\mathscr{S}$ in a local or global parametric coordinates $\mathbf{u}_i$, while $\mathbf{W}$ is a symmetric weight matrix. In the paper the authors show that the minimization of the *generalized stress function* leads to the computation of the *partial embedding distance*. Let $\mathbf{U}_{\mathscr{S}_N} = \arg\min_{\mathbf{U}}; \mathbf{D}_{\mathscr{Q}_N}, d_{\mathscr{S}}, \mathbf{W})$ be the canonical form of $\mathscr{S}$. Both canonical forms are then aligned in $\mathbb{R}^m$ by diagonalizing the matrix of the second-order moments of the canonical form and re-arranging the axes in a decreasing order. Once both shapes are embedded in the new space, a metric based on high dimensional moments is used to compare them from a distortion point of view. Such distance is then used in a rigid matching setting to easily find a match between the meshes.

Lipman and Funkhouser [116] propose an algorithm to find sparse point correspondences between surfaces that are approximately isometric. The method exploits the key idea that isometries are a subset of the Möbius group. Another interesting idea presented by the authors is that the deviation from isometry can be modelled as transportation-like functions between corresponding points.

In particular, given two genus zero surfaces $\mathscr{S}$ and $\mathscr{Q}$, the authors suggest to search for near-isometries in the space of Möbius transformations since it contains the entire set of conformal maps between the surfaces. The low dimensionality of Möbius transformations are exploited by first mapping the two discrete representations of the surfaces (*i.e.* meshes) to the extended plane $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$ with the mid-edge flattening scheme. Then, the meshes are sampled using the farthest point algorithm. This yields two uniform sets of points of both meshes. Finally, the algorithm generate triplets of potential point correspondences randomly and for each triplet the unique Möbius transformation is used to map into a canonical domain. In this space, the deformation error (*i.e.* deviation from isometry) between mapped point is computed. Such error measures are then accumulated in a fuzzy correspondence matrix which is used to find the set of correspondences through a greedy algorithm. As a refinement step, a final set of correspondences is extracted using validating correspondences using the geodesic consistency between matches.

A work based on similar consideration is the one proposed by Kim *et al.* [102]. They propose a method for automatic and robust detection of intrinsic symmetries. The keys idea are that critical point of the *average geodesic distance* function can be exploited to detect symmetry invariant points. Furthermore, intrinsic symmetries are self-isometries of surfaces which are contained in the Möbius group.

The algorithm is divided in three steps. The first one aims at extracting a sample of symmetric points. Let $\mathscr{S}$ be the surface whose intrinsic symmetries must be retrieved. In particular, this step produces two sets of points. A coarse one which is used to produce anti-Möbius transformations (*i.e.* Möbius transformations augmented with a reflection), while the dense one is used to find the final set of correspondences. The key idea in constructing a symmetry-invariant set is to extract critical points of a symmetry invariant function. The *average geodesic distance* is used in this context. In a continuous setting, such function is defined as

$$\Phi_{agd} = \int_{\mathscr{S}} d_g(p,q)\, d\mathrm{vol}_{\mathscr{S}}(q)$$

where $p, q \in \mathscr{S}$. In the discrete setting, given a mesh $S = (V, E)$ where V are the vertices while E the edges, the symmetry invariant sets are computed by approximating the geodesic distances on the surfaces with Dijkstra's algorithm. Then, the average geodesic distance to all vertices is computed as

$$\Phi_{\hat{agd}} = \sum_{v_j \in V} \frac{A_i}{3} d_g(v_i, v_j)$$

with A_i the *area* of the 1-ring neighbourhood of vertex i. The critical points are then selected inside this neighbourhood (both maxima and minima).

The second step seeks for the anti-Möbius transformations and involves the computation of several stages. In the first one, the mesh is mapped to the extended complex plane $\hat{\mathbb{C}}$ introduced above. Then, the coarse symmetry invariant set is used to generate triplets and quadruplets which will be the candidate anti-Möbius transformations. In order to prune the generating set, a heuristic approach is used. In particular, the distance between two matched points is checked and if lower of a threshold, the set is pruned. Even if the average geodesic distance is too similar between two matched points the set is pruned. Finally, each candidate anti-Möbius transformations is associated with a score which measures the quality of the alignment (*i.e.* how well the transformations align the surface with itself).

The final step aims at extracting the correspondences from the dense symmetry-invariant set exploiting the anti-Möbius transformations computed in the previous step. The process commences applying the best anti-Möbius transformations to this set of points and mapping back to the surface. Then, the pairs of points that are geodesically mutually closest are extracted. Those points are then used within the dense set to predict more correspondences. This is done computing a feature vector for each non-assigned point based on the geodesic distance with respect to each assigned points. Then, a deviation measure over these feature vectors is defined and the pairs that yields the smallest deviation are selected as points into correspondence.

The proposed approach is finally tested over several challenging datasets of meshes. Among the limitations, the proposed method cannot deal with small symmetric parts and heavily relies on the smoothness of both the mesh and the symmetry invariant function.

Ovsjanikov *et al.* [142] propose a spectral method to deal with intrinsic symmetries, isometric shape matching and partial and incomplete models matching. The method is based on the heat kernel and on the definition of what is called the *heat kernel map*.

Given a compact Riemannian manifold M without boundary, let $u(x,t) : M \times \mathbb{R}^+ \to \mathbb{R}^+$ be the amount of heat at a point x after timer t for some initial heat distribution f. u satisfied the heat equation and can be computed using the *heat operator* H_t, defined as

$$u(x,t) = (H_t f)(x) = \left(e^{-t\Delta_M} f\right)(x) = \int_M k_t^M(x,y) f(y) dy$$

with Δ_M the usual Laplace-Beltrami operator computed on the manifold M while $k_t^M(x,y)$ is the heat kernel. Intuitively, the heat kernel yields the heat value on the point y from the point x after time t. The heat kernel owns several interesting properties, like its invariance to isometry (*i.e.* intrinsic) and it is multiscale (through the *time* parameter).

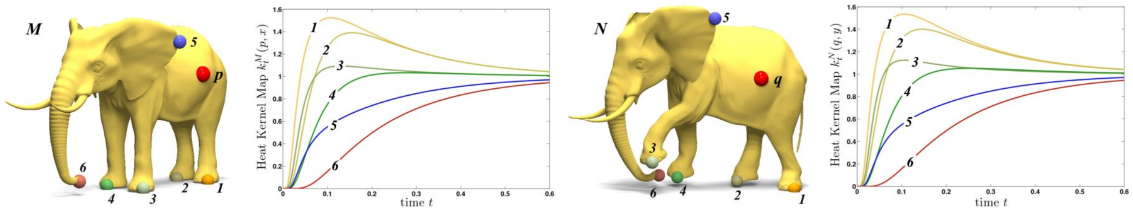


Figure 3.3: Example of feature points extracted from two meshes which represent the same elephant. p and p represent a landmark. In the graph, the Heat Kernel Map $\Phi_p(x)$ computed on the feature points over time. Points with the same color are into correspondence.

Another interesting result reported by the authors is that, given two compact connected Riemannian manifold without boundary M and N, if a map T from M to N then $k_t^M(x,y) = k_t^N(T(x),T(y))$ for all the points in the manifold.

Furthermore, the authors show that the multi-scale property of the heat kernel allows to approximate its value for small values of t by the heat kernel of a small geodesic neighbourhood of a point x. From these considerations, the author introduce the *heat kernel map* (HKM).

Given a fixed source point p, the HKM $\Phi_p^M : M \to F$ (F the space of the functions from $\mathbb{R}^+$ to $\mathbb{R}^+$) of any point x is defined as

$$\Phi_p^M(x) = k_t^M(p,x)$$

In other words, the HKM define a feature vector for each point x of the manifold. Furthermore, any point of the manifold can be used as the source point p. As an extension of the property introduced above, the authors show that under mild genericity conditions, the HKM is *injective*, *i.e.* $\Phi_p^M(x) = \Phi_p^M(y)$ iff $x = y$. Hence, since any isometry $T : M \to N$ must preserve the heat kernel, then for any point $x \in M$, $\Phi_p^M(x) = \Phi_p^N(T(x))$ holds. Thanks to the injectivity of the map, the authors conclude that for every point $x \in M$ there exist a unique corresponding point $T(x) \in N$ such that the heat kernel is preserved.

From a practical point of view, the authors implement what described above as a four stages process. The first one is the *feature detection* and aims at extracting a subset of relevant points. For this purpose, the local maxima of the *heat kernel signature* with a large time parameter t is used. Hence, a point $x \in M$ is extracted as a feature point if $k_t^M(x,x) > k_t^M(y,y)$ for each y belonging to the two ring neighbourhood of x. These feature points, that have been shown to be robust and reliable, are the input of the second stage, namely the *single feature matching*. Here, the compatibility between pairs of feature points of different shapes is tested. In particular, let $p \in M$ be a feature point of M, while $Q \subset N$ is the feature points extracted from N. For each $q_i \in Q$, the quality of the match is computed as

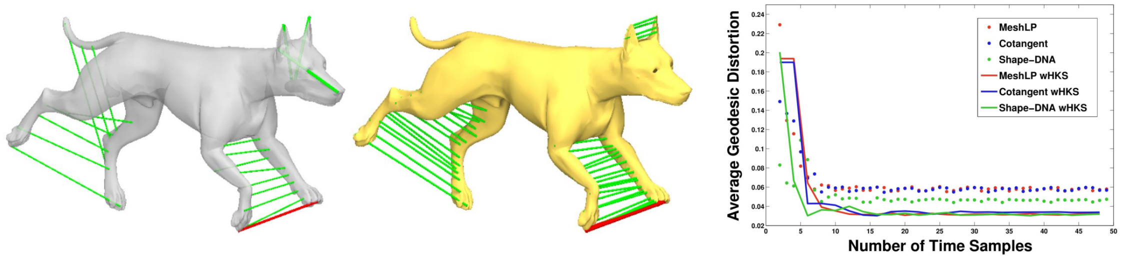


Figure 3.4: Performance of [142] in the symmetry detection scenario. In particular, the right image shows intrinsic detection without the augmentation of the HKM with the HKS. The right one has been augmented. The graph shows the quality of the match using different discretization of the Laplace-Beltrami opearator.

$$E(p, q_i) = \sum_{x \in M} \min_{y \in N} \left\| \Phi_p^M(x) - \Phi_{q_i}^N(y) \right\|$$

and the pairs that yield the highest quality value are selected as initial matches. Which, in turns, are used as the input of the third phase, namely the *correspondence propagation* stage. The authors here use a very simple propagation rule exploiting the injectivity of the HKM. Given a pair of points in correspondence (p, q), the propagation is defined as

$$f(x) = \arg\min_{y \in N} \left\| \Phi_p^M(x) - \Phi_q^N(y) \right\|$$

Thus, each point of N is associated to the point of M whose HKM is closer. The fourth and last stage deals with the non-perfectly-isometric problem. Indeed, most of the shapes are not a perfect isometry with respect to another, even if they represent the same shape. In order to make the proposed method more robust, the authors introduce two heuristics that improve the quality of the results. The first regards the augmentation of the HKM of a certain point by concatenating the HKS computed on the same point. The second assume the knowledge of a *landmark*, which is a known correspondence between two shapes. The *heat kernel signature* compute in both the points of the landmark are then used to augment the HKM again by simply concatenating them. Figure 3.3 shows an example of the proposed approach.

The resulting algorithm is tested against symmetry detection (see 3.4), isometric matching and partial and approximate matching tasks, showing interesting results.

Rodolà *et al.* [151] propose a different solution to the minimum distortion intrinsic correspondence problem defined as a quadratic assignment problem (QAP). The main contribution regards the adoption of L^1 relaxation of the QAP which leverage the game theoretic framework. In particular, the authors firstly show an interpretation of the QAP in terms of the Gromov and Lipschitz distances between metric spaces. The family of distances defined in the paper constitute metrics on the space of equivalence classes of shapes under the isomorphism relation of metric-measure spaces.

The second contribution is the adaptation of the game theoretic framework to solve the relaxed QAP. In particular, the authors model strategies as candidate assignments $(x, y) \in X \times Y$. The choice of these candidates is based on the Euclidean distance of the *scale invariant heat kernel signature* [33] descriptors computed on each point of the shapes. The game-theoretic solver searches for the most coherent group of matches according to a compatibility measure between them (*i.e.* the Gromov-Wasserstein metric). Then, pairs of players (pairs of possible matches) are repeatedly extracted from the population to play a symmetric game, receiving a certain payoff from other players proportional to compatibility measure introduced above. The game is repeated until the strategies that yield larger payoffs are selected as solution to the initial problem. Which means that the most robust matches are extracted from the candidate set.

As a result, the approach yields a sparse set of correspondences between shapes. In the third contribution the authors show how several sparse sets (resulting from several independent iterations of the proposed method) can be aggregated together into a denser correspondence. This is achieved using again a game-theoretic solver. The performance of the proposed method is assessed on the SHREC'10 correspondence dataset [30].

3.2.2 Dense Matching

The approaches presented in this section deal with the dense matching problem. Here the goal is to find an injective map between two shapes, both in the form of a point-to-point correspondence or as a smooth map between surfaces.

In a recent work, Ovsjanikov *et al.* [141] propose modelling the correspondences between the spaces of square-integrable functions $L^2(\mathcal{N})$ and $L^2(\mathcal{M})$ in terms of a linear operator $T : L^2(\mathcal{N}) \to L^2(\mathcal{M})$ mapping functions over the manifold $\mathcal{N}$ onto functions over the manifold $\mathcal{M}$. Clearly, this setting generalizes classical vertex-wise correspondences as this can be achieved by a T that maps delta-functions onto delta-functions.

Given bases $\{\phi_i\}_{i\geq 1}$ and $\{\psi_i\}_{i\geq 1}$ on $L^2(\mathcal{N})$ and $L^2(\mathcal{M})$ respectively, the functional correspondence can be expressed with respect to these bases as follows:

$$\begin{aligned} Tf &= T\sum_{i\geq 1}\langle f,\phi_i\rangle_{\mathcal{N}}\phi_i = \sum_{i\geq 1}\langle f,\phi_i\rangle_{\mathcal{N}} T\phi_i \\ &= \sum_{ij\geq 1}\langle f,\phi_i\rangle_{\mathcal{N}} \underbrace{\langle T\phi_i,\psi_j\rangle_{\mathcal{M}}}_{c_{ij}}\psi_j\,, \end{aligned} \tag{3.2}$$

where the coefficients c_{ij} depend on the choice of the bases. Taking only k elements of each basis, one obtains a rank-k approximation of T as a $k \times k$ matrix $\mathbf{C} = (c_{ij})$.

In order to compute $\mathbf{C}$, the authors assume to be given a set of q corresponding functions $\{f_1,\dots,f_q\} \subseteq L^2(\mathcal{N})$ and $\{g_1,\dots,g_q\} \subseteq L^2(\mathcal{M})$. Denoting by $a_{ij} = \langle f_j,\phi_i\rangle_{\mathcal{N}}$ and $b_{ij} = \langle g_j,\psi_i\rangle_{\mathcal{M}}$ the $k \times q$ matrices of the respective coefficients onto the selected bases, functional correspondence boils down to the linear system

$$\mathbf{CA} = \mathbf{B}\,. \tag{3.3}$$

If $q \geq k$, the system (7.4) is (over-)determined and is solved in the least squares sense to estimate $\mathbf{C}$.

It is convenient to use the eigenfunctions of the Laplace-Beltrami operators of $\mathcal{N}$ and $\mathcal{M}$ as the bases $\{\phi_i, \psi_i\}_{i \geq 1}$, since under these bases truncating the series at the first k coefficients has the effect of "low-pass" filtering over the functional representations. In fact, while the Laplacian eigenbasis suffers from issues like mixing eigenspaces and sign ambiguity, it has been shown in [97] that the space of functions spanned by the first n eigenfunctions of the Laplace-Beltrami operator are stable under near-isometries as long as the n^{th} and the $(n+1)^{th}$ eigenvalues are well separated.

Furthermore, expressed in the Fourier bases, the matrix C has interesting properties making it more efficient to estimate. In fact, Ovsjanikov *et al.* [141] proved that if the correspondence preserves areas, then the matrix C is orthogonal, *i.e.* , $\mathbf{C}^T\mathbf{C} = \mathbf{I} = \mathbf{C}\mathbf{C}^T$. Further, if it is also an isometry, then we have $c_{ij} = \pm\delta_{ij}$.

In the discrete setting, the manifold $\mathcal{N}$ is sampled at n points $x_1, \ldots, x_n$ which are connected by edges E and faces F, forming a manifold triangular mesh (V, E, F). In this setting, a function on the manifold is represented by an n-dimensional vector $\mathbf{f} = (f(x_1), \ldots, f(x_n))^T$. The inner product is discretized as $\langle \mathbf{f}, \mathbf{g} \rangle = \mathbf{f}^T\mathbf{S}\mathbf{g}$, where $\mathbf{S} = \mathrm{diag}(s_1, \ldots, s_N)$ and $s_i = \frac{1}{3}\sum_{jk:ijk \in F} A_{ijk}$ denotes the local area element. Hence, a basis $\mathbf{\Phi} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_k)$ over the vertices of the mesh represents the discretization of an orthogonal functional basis if $\mathbf{\Phi}^T\mathbf{S}\mathbf{\Phi} = \mathbf{I}$.

The discretization of the Laplacian takes the form of an $n \times n$ sparse matrix $\mathbf{L} = -\mathbf{S}^{-1}\mathbf{W}$ using the classical cotangent formula ([52, 125, 145]),

$$w_{ij} = \begin{cases} (\cot\alpha_{ij} + \cot\beta_{ij})/2 & ij \in E; \\ -\sum_{k \neq i} w_{ik} & i = j; \\ 0 & \text{otherwise.} \end{cases}$$

where α_{ij}, β_{ij} denote the angles $\angle ikj, \angle jhi$ of the triangles sharing the edge ij.

Like in most scenarios, the manifolds in which the functional map is built on are unknown and a discrete setting must be defined. The discretized Fourier basis $\mathbf{\Phi} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_k)$ is obtained by finding k orthogonal functions that satisfy the eigenvalue equation for the Laplacian, *i.e.* , $-\mathbf{S}^{-1}\mathbf{W}\mathbf{\Phi} = \mathbf{\Phi}\mathbf{\Lambda}$ with $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$ the diagonal matrix of the corresponding eigenvalues. This results in the standard generalized eigenvalue problem

$$\begin{aligned} & \mathbf{W}\hat{\mathbf{\Phi}} = -\mathbf{S}\hat{\mathbf{\Phi}}\mathbf{\Lambda} \\ s.t. \quad & \mathbf{\Phi}^T\mathbf{S}\mathbf{\Phi} = \mathbf{I} \end{aligned} \tag{3.4}$$

The Fourier coefficients of a (discretized) function $\mathbf{f}$ are then obtained by taking dot products of the function with the basis elements, $a_i = \langle \boldsymbol{\phi}_i, \mathbf{f} \rangle = \boldsymbol{\phi}_i^T\mathbf{S}\mathbf{f}$, resulting in

$$\mathbf{a} = \mathbf{\Phi}^T\mathbf{S}\mathbf{f} \tag{3.5}$$

With the discrete setting in place, the algorithm for the computation of a correspondence map can be introduced. The first step involves the computation of a set of descriptors for each point of the two manifold $\mathscr{M}$ and $\mathscr{N}$. The algorithm is independent on the local descriptor used. Nevertheless, the authors suggest to use the *wave kernel map* (WKM, a declination of the *heat kernel map* introduced in [142] and reviewed above) augmented with the *wave kernel signature*. Encouraging results have been shown even for HKM augmented with the HKS. Interestingly, the descriptor to be used as augmentation is independent on the main descriptor of choice. Hence, it is possible to use the WKM augmented with the HKS.

The descriptor created at the first step is then eventually augmented with descriptor associated to know correspondences between the manifold. Usually, a robust sparse matching method is employed in order to detect a low number of very accurate correspondences between the meshes that can be used to augment furtherly the descriptor defined before. This augmenting procedure, which has been introduced in [142], has been reviewed in section 3.2.1. An alternative way to augment the descriptor is to embed information about the segment it belongs to and a map that matches the segment of the two manifolds. However, the "landmarks augmentation" represents a more robust approach to descriptors augmentation. The augmentations can be seen as constraints that have been added to the problem. The flexibility of the functional representation allows to integrate several other constraints easily. For example, operator commutativity constraints for relevant linear operators (like Laplace-Beltrami or symmetry) are included, leading to a linear system that is solved in the least squares sense in order to get the optimal C.

The authors define additional refinement steps that can be used to furtherly improve the solution, like an alignment process between the columns of the basis (in particular, between the *transferred* base $C\Phi^M$ and the second base Φ^N). To this end, the ICP algorithm is used iteratively in order to find the best C.

Finally, in order to retrieve a point-to-point correspondence, a simple yet effective method involves the definition of a function $f : \mathscr{M} \to \mathbb{R}$ (like the indicator function) and the computation of its image $g = T_F(f)$ using C. Then, $y \in \mathscr{N}$ corresponds to the point at which $g(y)$ obtains the maximum.

Besides the shape matching problem, the authors show that the proposed approach could be used also to improve existing point-to-point maps and finally to transfer segmentation functions between shapes (*i.e.* segments projection).

Kim *et al.* [103] propose an automatic pipeline to address the non-isometric genus-zero shape matching problem. The key idea is to define a weighted combination of already existing maps (*e.g.* Heat Kernel Maps, Möbius Voting Maps, *etc.*). As a result, the map provides low-distortion everywhere even for highly deformed shapes. The process takes the name of *blended maps*.

In particular, the authors aim at computing a map that smoothly blends multiple low-dimensional maps. Given a pair of surface meshes $\mathscr{M}$ and $\mathscr{N}$, the goal is to compute k candidate maps ${m_i}_{i=1}^{k} : \mathscr{M} \to \mathscr{N}$ and a blending weights $b_i(p)$ for every point p such that the final blended map f has low distortion everywhere. Hence, f can be defined as

$$f(p) = \underset{p' \in \mathcal{N}}{\operatorname{argmin}} \sum_{i=1}^{k} b_i(p) d_{\mathcal{N}}(p', m_i(p))^2$$

where $d_{\mathcal{N}}$ is the geodesic distance on the surface $\mathcal{N}$. Intuitively, the process maps every point to the weighted geodesic centroid between the point projected by each map m_i. In this work, the authors focus on algorithms that produce conformal maps to be blended. This allows to simplify the problem of finding blending weights by partitioning it into the two factors computation $b_i(p) = c_i(p) \cdot w_i(p)$, where the $c_i(p)$ measure the *confidence* of the conformal map m_i, while the $w_i(p)$ are the *consistency* weights (and measure to what extent a map should be used in the blending).

The process which leads to the blending map starts with the generation of a set of feature points. This is achieved extracting from the vertex set of a shape the points in which the *Average Geodesic Distance* is at maxima. Then, the conformal maps are generated. To this end, the authors use the same approach used in [102] (Möbius Voting method) which has been review in section 3.2.1. As a result, a set of candidate conformal maps are generated. With the candidate maps at end, the authors show how to compute the confidence weights and the consistency weights. The computation of the former is based on how well the area is preserved while mapping a point into the other surface. The latter is computed minimizing an objective function that ideally yields zero for conformal maps that induce high distortion while yielding non-zero weights for conformal maps that are consistent with one another.

With the conformal maps and the weights at hand, the blended map can be computed as defined above. In order to get a smooth map as a result of the process, the confidence weights must change smoothly over the surface.

Rodolà *et al.* [153] propose a shape matching method that produces dense correspondence tuned to a specific class of shapes and deformations. In particular, the authors devise an application-specific data-driven algorithm. Hence, a set of training meshes which represent the same shape up to some class of non-isometric deformation is used as the input of the learning process proposed in the work. This learn process produces a shape descriptor which is able to capture deformation variations in the given shape. One of the major contribution is the employment of a random forest classifier which allows the WKS (used as descriptor) to detect a wider set of classes of deformation.

In order to use the random forest classifier to compute the map between two surfaces, a canonical transformation is learned from examples. Random forests [49] are sets of decision trees. In this context, each decision tree routes a point x of a shape $\mathcal{S}$ from the root to a leaf node, where a probability distribution defined on a discrete label set L is assigned to the point. The route of each points is determined by a binary decision function, the *split function*, which yields *left* (L) or *right* (R). In other words, every non-leaf node is a split function that will forward the point on the left or right sub-tree of the decision tree. This leads to the posterior probability $P(l|x,t)$ for each tree t of the forest $\mathcal{F}$ of label $l \in L$ given a point $x \in \mathcal{S}$ and is associated to the leaf node reached by a certain point. The process is repeated for each point and for each tree of the forest, leading to global average posterior probability computed as

$$P(l|x,\mathscr{F}) = \frac{1}{|\mathscr{F}|}\sum_{t\in\mathscr{F}} P(l|x,t)$$

The learning phase uses a set of training shapes whose canonical transformation is known in order to build the trees of the forest. The canonical transformation represents a bijection T which map the vertices of a shape into the label set L. Hence, let R_i be a shape whose canonical transformation is T_i $(1 \le i \le m)$, then the training set is the set of edges induced by the mesh set and the canonical transformation set. Hence, we define it as $\mathbb{T} = \{(r, T_i(r) : r \in V_{R_i}\}_{1\le i\le m}$. The trees that will compose the forest are then generated randomly while a recursive procedure iteratively split the terminal nodes according to some policy while each point of the training set is routed in the tree. Central in the whole definition of random forest is the concept of split function. It can be seen as the rule by which a certain tree in the forest is created through the learning process or it determines which route a shape point will take while traversing a tree. To this end, the authors propose to define a family of randomized split function based on the WKS introduced above. From a physical point of view, the WKS evaluates the probability of a quantum particle to be located at a point x of a shape $\mathscr{S}$ under a certain energy distribution. Let $\Delta_{\mathscr{S}}$ be the Laplace-Beltrami operator, whose eigensystem components are the first k eigenvalues λ_k and corresponding eigenfunctions ϕ_k and e an energy level. Then, the WKS can be defined as

$$p(x;e,k) = \sum_{i=1}^{k} f_e^2(\lambda_i)\phi_i^2(x)$$

In order to generate a randomized split function for each node of a tree, an energy level e, an embedding dimension of the spectra k and a threshold τ are sampled, allowing to define a function as

$$\Psi(x;e,k,\tau) = \begin{cases} \text{L if } p(x;e,k) > \tau \\ \text{R otherwise} \end{cases}$$

The training process suffers of a major drawback: it is based on local information (point-wise computation) and it ignores completely the structure of the underlying manifold. To address this problem, the authors augment the prediction formulation with a spatial regularization term based on the functional maps. The effectiveness of the proposed approach is assessed on the TOSCA and Kids datasets, comparing it to other dense matching approaches and showing interesting results.

Chen and Koltun [43] present a non-rigid registration approach of surfaces which casts isometric embedding as a Markov Random Field (MRF) optimization whose solver is based on a linear programming relaxation. Differently from other approaches, this method is able to deal with topological variations of the surfaces.

The authors follow Bronstein *et al.* [32] and define the problem as an embedding of a surface into another, without intermediate embeddings. As a reminder of what has been introduce in section 3.2.1, the GMDS is the embedding objective in terms of the distance distortion induced by the mapping and is defined as

$$E(f) = \sum_{i,j} w_{i,j} |d_{\mathscr{S}}(s_i, s_j) - d_\tau(f(s_i), f(s_j))|^p$$

where $d_{\mathscr{S}}$ and d_τ are geodesic distances while $w_{i,j}$ are weights that can be used to disable some distortion terms. The authors express this objective function in terms of a MRF. Since it is continuous, the first step involves the discretization of the label space. Let $T = \{t_j\}_{j=1}^{m}$ be the mesh which densely represents the underlying surface τ. Let L be a set of m labels. The goal is to define a map $: S \to L$. Using standard MRF notation, the objective define above can be rewritten as labelling * that minimizes an energy function. Let $\theta_{ij}(l_i, l_j) = w_{i,j} |d_{\mathscr{S}}(s_i, s_j) - d_\tau(f(s_i), f(s_j))|^p$, where $l_i = (s_i)$, then the LP relaxation is defined as

$$\begin{aligned}
\min\nolimits_{0 \le x \le 1} \quad & \sum_{i,j} \sum_{a,b} \theta_{ij}(a,b) x_{ij}^{ab} \\
\text{s.t.} \quad & \sum_{b} x_{ij}^{ab} = x_i^a, \quad \forall i, j \in S, \forall a \in L \\
& \sum_{a} x_i^a = 1 \qquad \forall i \in S
\end{aligned}$$

where x_i^a and x_{ij}^{ab} represents distributions over the space of labellings. A linear programming solver can then be used to solve the problem above. Note that the authors rewrite the problem to both solve it in a more efficient way and to exploit the structure of the problem, relying on a tree-reweighted message passing algorithm. Furthermore, a penalty function is added to the objective function in order to deal with topological variations among the shapes. In particular, such function is used to bound penalties imposed on good mappings due to topological violations.

To assess the efficacy of the proposed approach, the authors make extensive test on FAUST [23] dataset, achieving state-of-the-art results on very challenging real-world data.

4

Structure-based representation

In this chapter we will review literature relevant to the work that is described in the third part of the thesis, that is, structure based representation of data and embedding techniques of such data in order to achieve, among the other, classification task. The first section of the chapter considers the work on the extraction of graph features. Representing a graph as a set of features allows a more compact and expressive form. Feature selection and extraction represent a viable solution when dealing with complex structures with intrinsically high dimensionality.

We then discuss graph similarity. The works described in this section try to answer to the question of how two graphs can be compared. The problem is often solved through the definition of a similarity measure between the graphs, which opens to a debate about the meaning of similarity. In one application we could be interested in assess in two graphs are identical, while in another we may be more interested in how many changes a graph must undergo in order to be identical to the reference one.

This leads naturally to the discussion on graph matching algorithms since the may also be used to assess similarity. The techniques reviewed in this section aim at finding a correspondence map between the nodes of two or more graphs. This is clearly a hard task due to the arbitrary order of the nodes. And it gets even more complex when we deal with more than two graphs. In this scenario, properties like transitiveness of the found matches are a very interesting yet very difficult result to achieve.

We then discuss the graph segmentation field, which considers how a graph may be decomposed in a set of subgraphs in order to maximize some similarity measure intra-subgraph while minimizing the same measure between the nodes that belong to different subgraphs.

In the last section of this chapter, we discuss graph means and medians and then detail approaches to building generative models of graphs. The set of techniques introduced in this section allows to define models that can be used as a compact representative of a set of graphs or to study the variation among the graphs of a set.

4.1 Graph Features

The techniques discussed in this section deal with the problem of how the best set of features can be extracted from a graph representation. The study and the use of methods that allows to extract features from graphs is of great importance due to intrinsically high dimension of graphs. This phenomenon is known as the *curse of dimensionality* [186]. In the seminal book [21], it has been shown that for a simple look up classifier (that associates a cell of the feature space with its classification) the number of training examples required to accurately train the system is exponential in the number of features. Therefore, the construction of a good set of features accurately describing the structure of a graph can be a useful tool.

As a first example of a feature extraction process, one can think about the eigendecomposition of a matrix representation of a graph, like the adjacency matrix. Such decomposition leads to the spectral representation of the graph, whose components can be used or combine a feature set of the graph itself. This field take the name of spectral graph theory [45] and has led to many different approaches and applications [191].

More formally, let X be a matrix representation of a graph. Then the spectral decomposition $X = VEV^T$ results in a set of eigenvalues E (one for each node of the graph) and eigenvectors V. Let suppose that the eigenvalues are ordered by magnitude. Then, such eigenvalues can be used as a feature vector for the graph. Furthermore, the following properties hold:

- The eigenvalues are invariant to the order of the vertices of the graph
- The eigenvalues are able to capture some structural information about the graph

The vector of eigenvalues associated to a graph is known as its spectrum.

In order to assess the uniqueness of the spectrum associated to several matrix representations of a graph, Haemers and Spence [76] have enumerated all graphs whose cardinality is 11 or less and retrieving the spectrum of respectively the adjacency, the Laplacian and the signless Laplacian matrices of the graph. In this work the authors coined the term *cospectral mate*. A graph that is structurally different but share the same spectrum of another one is called cospectral mate of the other. The main result of the paper is to give the fraction of all graphs with at most n vertices that have cospectral mates. It is easy to see that the lower this fraction is, the better is the discriminative power of the features extracted from a particular matrix representation of the graph. The results show that the signless Laplacian representation has the lowest fraction of graphs with cospectral mates, followed by the Laplacian and then the adjacency matrix. In fact, most of the approaches based on spectral graph theory make use of the signless Laplacian matrix as matrix graph representation.

Luo *et al.* [123] work with the spectral decomposition of the adjacency matrix of a graph k. For each eigenmode ω, an adjacency matrix is built as

$$S_k^{\omega} = \phi_k^{\omega} ({}_k^{\omega})^T \tag{4.1}$$

where ϕ^i is the i-th eigenvector. The authors investigate whether graphs can be represented in a stable way using vectors of spectral attributes for the leading eigenmodes of the adjacency matrix. The attributes studied include the perimeter length, the volume and the Cheeger constants for the mode adjacency matrices. In order to vectorize the graphs, each component of the vector is taken to represent a different spectral mode of the original graph adjacency matrix. The order of the components of the vector is the magnitude order of the eigenvalues of the adjacency matrix. For each spectral mode, the components of the associated eigenvectors are used to compute spectral attributes. In this way, invariance in vertex ordering is achieved.

Zhu and Wilson [205] have studied the issue of stability and discriminative power of the spectrum obtained from computed on several matrix representations of graphs. In particular, they test the stability of the spectrum computed on the adjacency matrix, combinatorial Laplacian, normalized Laplacian matrix, heat kernel and path length distribution matrix. In their work, they use two methods to evaluate the performance of each representation. The first experiment attempts to find if the relationship between the graph edit distance and the distance between graph spectra is linear. The second test considers the classification accuracy of each representation method when some of the graph structure is perturbed. The evaluation section shows that the Euclidean distance between spectra tracks the edit distance over a wide range of edit costs. An analysis of the stability of this relationship is proposed. Finally, they employ the spectra to match and classify the graphs and demonstrate the effect of the graph matrix formulation on error rates. In both test sessions, the heat kernel and path length distribution representations yield the best results, followed by the various Laplacian representations, with the adjacency matrix whose spectrum is the less stable and discriminative among the others.

Wilson *et al.* [194] tackle the problem of embedding graphs in low dimensional pattern spaces using manifold learning methods. In particular, spectral methods are employed in order to vectorize graphs. The main advantage of the proposed method regards the computation of spectral features which are permutation invariants and which make use of the full spectral matrix. To construct these invariants, symmetric polynomials are used. The arguments of the polynomials are the elements of the spectral matrix. The authors use the values of the symmetric polynomials to construct graph pattern-vectors, while the differences between graphs are tackled by padding the spectral matrix with trailing vectors of zeros, which is equivalent to pad graphs with dummy nodes.

The authors explore how the spectral feature vectors may be used to construct pattern spaces for sets of graphs through a number of different approaches, starting with the application of the principal components analysis to the covariance matrix for the vectors. This has shown to locate a variance preserving embedding for the pattern vectors. The second approach is multidimensional scaling. In this case, the distance between vectors is preserved. The third approach, which demonstrates to be the one yielding the better results, is locally linear projection, which aims to preserve both the variance of the data and the pattern of distances.

Another interesting method of embedding graphs in a vector space comes from Riesen *et al.* [150]. In their work, the authors try to achieve their goal using a dissimilarity measure. Basically, given a set of graphs $\mathscr{S}$, the authors selected a subset of graphs $\mathscr{P} \subset \mathscr{S}$ to use as a prototype set of the original set. In other words, the graphs in $\mathscr{P}$ are elected as representatives of the population of $\mathscr{S}$. Like in many other approaches, the edit distance is used in this work as a similarity metric. In particular, the edit distance between a graph and all the prototypes is computed and then used to build a dissimilarity vector for that particular graph of dimension $|\mathscr{P}|$. One of the main drawback of the proposed method is that it heavily relies on the graphs that belong to the prototype set $\mathscr{P}$. Indeed, a poor selection of graphs could result in a not meaningful feature vector. The authors suggest several possible solutions to this problem, ranging from a random selection of the graphs to use as the prototype set to the employment of a k-means clustering like approach to define the prototype sets.

Differently from the approaches described above, Caelli and Kosinov [40] suggest to embed each vertex in a vector space rather than the graph as a whole. The reason behind this is to allow inexact graph matching. In particular, they compute the spectral decomposition of the adjacency matrix and then project each column of the adjacency matrix into the eigenspace resulting from the decomposition. Since the features constructed for two different graphs result from two different spectral decompositions, they belong to different spaces. To deal with this issue, a re-normalization step is applied. Furthermore, in order to handle graphs of different sizes, the spectral information is truncated to a common dimension (the lower one), resulting in the same dimensionality for both graphs. In their evaluation section, the authors show that even if the individual features are not very useful, the relative distances between the features and the trajectory of the features as a whole provide much information about the graph.

4.2 Graph Similarity Methods

A common problem in structural pattern recognition regards the definition of good similarity measure of two or more graphs. For instance, in a data retrieval problem we would like to be able to query a large database by computing the similarity between a query structure (query graph) and all the structures in the database. This goal is often achieved through the definition of metric on the features extracted from the graphs, like seen in the previous section. In this section, we will review some of the most widespread methods to define similarity measure between graphs. In particular, we will start this section with the introduction of a simpler, more intuitive and widely used measure of graph similarity known as graph edit distance. In the second part of the section, iterative methods for graph similarity will be presented.

Graph Edit Distance

One of the most consolidated technique to compute the similarity between two graphs is the graph edit distance [36, 66, 157]. It is based on the simple concept that a graph can be transformed to another one by a finite sequence of graph edit operations. The

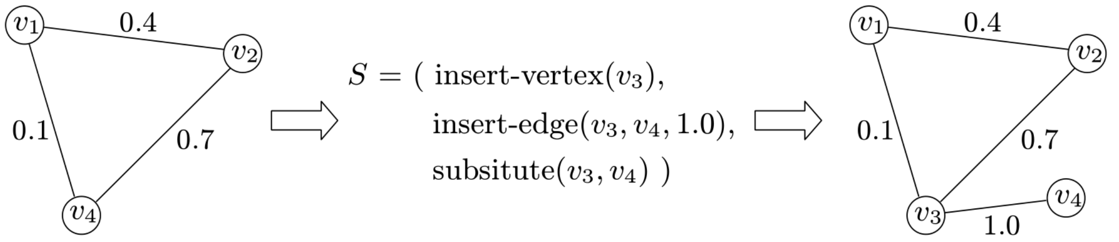


Figure 4.1: A visual representation of a transformation of a graph in another. The collection *S* represent the edit operation sequence.

list of possible edit operations (and the cost of each operation) depends on the algorithm employed, but each algorithm yields the operation sequence with the minimum cost. A visual example of such operation can be seen in figure 4.1.

As said above, a cost function is defined for each operation and the cost for this edit operation sequence is the sum of costs for all operations in the sequence. Even if the sequence of operation needed to transform a graph into a reference one is not unique, the one with the minimum cost is a good starting point. Hence, the similarity of two graphs is inversely proportional to the graph edit distance.

The list of edit operations that can be used to build the edit sequence and how much each edit operation cost are the key issues in the methods that employ the graph edit distance as similarity measure. There are a lot of literature in this regards. These cost functions are usually defined in a manual fashion, adapting the cost of each edit operation to the data the researcher is working with. Of course, this process is highly prone to errors. In this regard, Neuhaus and Bunke [137] proposes a method to automatically learn cost functions from a labelled sample set of graphs. In their work, the authors formulate the graph edit process in a stochastic context and perform a maximum likelihood parameter estimation of the distribution of edit operations. The underlying distortion model is learned using an Expectation Maximization algorithm. The cost functions are then derived from the model.

The idea of using graph edit distance as a similarity measure appears frequently in the literature and methods of improving its usability have been developed. For example, Bunke [34] has established a relationship between the graph edit distance and the maximum common subgraph under a certain cost function. The maximum common subgraph, termed $mcs(G_1, G_2)$, is a subgraph of both G_1 and G_2 such that there is no subgraph possible with more vertices. The cost function in question is as follows: vertex insertions and deletions have cost 1, identical vertex and edge substitutions have cost 0 but different substitutions have cost inf and finally edge insertions or deletions have no cost. While this cost function appears simple, the authors only consider complete graphs. This simplifies the proofs and results in no loss of generality. A standard graph is mapped onto a complete graph by setting the weights of edges that do not appear in the standard graph to zero in the complete graph. In this way the vertex insertions and deletions actually encode the edge insertions and deletions. The graph edit distance $d(G_1,G_2)$ and the maximum common subgraph are related as follows:

$$d(G_1,G_2) = |G_1| + |G_2| - 2|mcs(G_1,G_2)|$$

Bunke [35] has conducted an extensive analysis of cost functions and related the concept of graph edit distance to graph isomorphism and subgraph isomorphism under certain cost functions. In the tree domain, Torsello and Hancock [183] have shown how the tree edit distance may be estimated from the minimum description length of a tree set that they use in their generative model of tree structure.

Iterative Methods

The other approach that will be introduced in this section of the chapter is the iterative one. The basic idea behind the iterative approach is quite simple: two nodes are similar if their neighbourhoods are also similar. This leads naturally to an iterative approach for computing similarity scores for the elements of these graphs, in which scores for similarity between elements propagate along to neighbouring elements at each time step. The process ends when convergence is achieved.

There are several examples of successful algorithms that belong to this category. The flooding algorithm by Melnik *et al.* [128] is one of them. In their work, they propose a structural algorithm that can be used for matching of diverse data structures. Such data structures are called *models*. The algorithm is based on the idea of converting the models to be matched into directed labeled graphs. Then, these graphs are used in an iterative computation whose results tell us what nodes in one graph are similar to nodes in the second graph. The similarity computation follows directly from the iterative methods basic idea: two distinct models are similar when their adjacent elements are similar. In other words, a part of the similarity of two elements propagates to their respective neighbours, with a mapping between two graphs as the final result. In their work,

Jeh *et al.* [89] propose a different approach, in which the self-similarity of a graph is measured. In other words, the algorithm assesses the similarities between all pairs of nodes in one graph. The algorithm computes iteratively all pairs similarity scores, by propagating similarity scores in the A^2 matrix, where A is the adjacency matrix of the graph. The process ends when convergence is achieved.

Zager and Verghese [203] propose another graph similarity measures, applied to the task of graph matching. Like in the approaches seen before, the graph similarity measures they define uses the structural similarity of local neighbourhoods to derive pairwise similarity scores for the nodes of two different graphs. Interestingly, in their work they introduce the idea of coupling the similarity score of nodes and edges in order to compute the similarity between two graph.

Finally, Bayati *et al.*in [15] propose an algorithm to find approximate isomorphisms, or alignments, between large graphs. The problem is formulated in a classic fashion as graph isomorphism, maximum common subgraph, and the quadratic assignment problem.

Since all these problems are intractable, the authors focus on practical heuristics, solving it using a Belief Propagation approach.

4.3 Graph Matching Methods

The graph matching problem consists of mapping two or more graphs to each other such that a given cost function is optimized. Hence, a solution of the matching problem should provide a map that makes the graphs as similar as possible, where the similarity is assessed by the cost function. The efficacy of the similarity measure is central in the definition of a good solution to the matching problem.

The graph matching problem is tightly tide to the isomorphism between graphs. Indeed, if a bijective matching between two graphs exists, then an isomorphism exists. This is known as the graph isomorphism problem, a well-studied graph problem with no deterministic solution in polynomial time. We can consider the graph matching problem as a more general instance of the graph isomorphism problem since the former seeks an alignment even between different graphs (different dimension, in which case we look for a partial matching, or a matching between subgraphs).

Due to the complexity of the graph matching problem, researchers often rely on heuristics approaches for practical applicability. In this section, after a brief more formal introduction of the problem, we will introduce some of the most successful methods that deal with the graph alignment problem.

Problem Definition

In this section, we will briefly introduce a general definition to the graph matching problem.

Let $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ be two undirected and unweighted graphs. The graph matching problem consists of finding a bijective map $f : V_1 \to V_2$ such that the score of the alignment, denoted by $S(G_1, G_2, f)$, is maximized. The function S represents our similarity function. Hence, it evaluates how similar the two graphs are with each other given the map that transform G_1 into G_2. In other words, we can define the graph matching problem as

$$\max_{f} \quad S(G_1, G_2, f)$$

It is important to note that f is required to be total and injective. This will ensure that all the nodes of the domain of the function (G_1) are mapped to different nodes of G_2. Since we are not requiring f to be a bijection, we can use the solution to the graph matching problem even when the graphs have different cardinality ($|V_1| \leq |V_2|$).

Main approaches

The methods that attempt to solve the graph matching problem can be divided in two macro groups. The first group contains the search based methods, while in the second we find methods based on non-linear optimization. The former tries to build a state space representing all possible mappings between two graphs. Then, this space is searched in order to find the mapping that minimizes some criterion.

For example, Wang *et al.* [190] propose a genetic-based search approach to solve the problem of error-correcting graph isomorphism. Genetic algorithms [72] are adaptive search methods which are devised based on the natural genetic processes. They are general-purpose optimization methods which have been successfully applied to difficult search, optimization, and machine learning tasks. The author, to address the performance issues of a direct implementation of the genetic algorithm on this problem, defines two local search strategies in order to form a hybrid search mechanism. Furthermore, in their work they also address the premature convergence problem, which is a well-known issue of genetic algorithms.

Unfortunately, the state space grows exponentially in the number of vertices of the graphs. As a results, even with very advanced technique like genetic algorithms, those approaches cannot provide a solution in a reasonable time. Non-linear optimization methods are the answer to this problem.

Umeyama [187] propose a solution to the weighted graph matching problem in both the undirected and directed cases. In its work, the author uses the eigendecomposition of the adjacency matrix of a graph and shows that it is almost always able to retrieve the optimum matching when a pair of graphs are nearly isomorphic. Let two graphs G_1 and G_2 be isomorphic. Then there exists a permutation matrix P that solve

$$PA_1P^T = A_2$$

which represents the map between G_1 and G_2. But, since this equation is difficult to solve, the author devises a method to estimate a permutation matrix P' which represents a very good estimation of the optimum permutation. In particular, he shows that the permutation matrix P' that maximizes

$$\mathrm{Tr}\left(P'^T \bar{\phi_1} \bar{\phi_2}\right)$$

will be very close to the optimum permutation matrix (still assuming G_1 and G_2 nearly isomorphic). Here, $\bar{\phi_1}$ and $\bar{\phi_2}$ are the absolute eigenvectors of G_1 and G_2). The Hungarian algorithm [26] can maximizes the trace and thus find the solution to the problem. The two biggest drawback of Umeyama's work are that it does not work if the graphs are not sufficiently close to each other (in which case the proposed method could fail to give the optimum matching) and it requires a pair of graphs of the same size.

Gold and Rangarajan [71] propose a different solution to the graph matching problem. The main contributions of their work is two-fold. Firstly, they use iterative projective scaling in order to enforce two-way assignment constraints (*i.e.* the vertices of both graphs are equally constrained). Secondly, they use the graduated non-convexity technique as optimization technique. This global optimization technique usually try to break a difficult optimization problem into a sequence of optimization problems, such that the first problem is convex and the next ones are always "less convex", until the last function solved is the non-convex initial one. This technique allows to avoid getting stuck in local minima. In this particular instance, graduated non-convexity is used to turn a matrix describing the discrete mappings into one containing continuous values. Thanks to a control parameter, the proposed method can move the solution from continuous (initial execution) to almost discrete (in the last iterations). Unfortunately, event at the highest value of the control parameter, the resulting solution is just "almost" discrete. A post-process must be applied in order to get the final map. In the original work, they adopt a naive solution in which the maximum element in each column is set to 1, while all other values are set to 0. This could lead to a sub-optimal solution since this *discretization* process is done sequentially on each columns, and not simultaneously. In order to find a global solution (*i.e.* all the mappings are considered simultaneously), algorithms like Auction algorithm [17] or the Hungarian algorithm [107] could be used.

A similar proposal comes from Van Wyk and Van Wyk [188]. In their work, they describe an algorithm similar to the one proposed by Gold and Rangarajan. Differently, they do not employ the iterative projective scaling technique to enforce two-way assignment constraints. Instead, they use a POCS (Projections onto Convex Sets) method. The two convex sets used in this case are the set of matrices that satisfy assignment constraints on the columns and the set of matrices that satisfy assignment constraints on the rows. Each of these corresponds to the assignment constraints in one direction and used together they can enforce two way constraints. The algorithm proceeds as that of Gold and Rangarajan's with the row and column normalizations being performed using the POCS approach.

The last method we will review try to solve a slightly different problem. All the approaches that we have seen before try to find a solution to the one-to-one vertex correspondence problem. However, sometimes it can be more convenient to produce a many-to-many solution. This allows to define more flexible frameworks and to find better matching too. The graph features described in section 4.1 by Caelli and Kosinov [40] is an example of such approach. In their work, they suggest to embed each vertex in a vector space rather than the whole graph. We can do the same with the vertices of two graphs, projecting them in the same subspace. This results in two sets of points representing the vertices of the graphs. Then, a clustering technique can be applied to these groups of points in order to map the vertices of two graphs. Clearly, the clustering technique allows to retrieve a many to many correspondence map between the vertices (*i.e.* with a density based clustering technique, in crowded region). In fact, the main advantage of the approach is its ability to perform matching between structures of different sizes where it is suited, while performing a one to one matching in others.

4.4 Segmenting Graphs

Clustering, also known as *clustering analysis*, is a set of techniques which aim at selecting and grouping homogeneous elements in sets of data. Usually, the criteria by which an element is assigned to a certain group is a metric between the elements (*i.e.* a distance measure in some multidimensional space). It is easy to see how this is related to the concept of similarity. The final objective is to group together "near" elements (*i.e.* elements whose distance is low) minimizing the intra-group distances. Like for the methods introduced in section 4.2, the performance of a method heavily relies on the quality of the metric used. Furthermore, different metrics allows to find different relations between the data we are working on.

Graph segmentation is, from a certain point of view, very similar to a clustering technique. Indeed, graph segmentation methods aim at splitting a graph into k subgraphs in order to maximize some measure of similarity between the vertices within the same subgraph while minimizing the similarity between vertices that belong to different subgraphs. One of the most used technique to accomplish this goal is the *graph cut*. As the name suggests, it aims at selecting a number of subsets of vertices of a graph by "*cutting*" some edges of the graph, resulting in the creation of several virtual connected components.

Hence, let $G = (V, E)$ be a graph. Let G_1 and G_2 be the two subgraphs that we obtain when we split G. Let V_1 and V_2 be the sets of vertex of the subgraphs G_1 and G_2 respectively. Then, the value of the cut is the sum of all the edges that begin in one subgraph and end in the other.

$$cut(V_1, V_2) = \sum_{u \in V_1, v \in V_2} w_\epsilon(u, v) \tag{4.2}$$

where $w_\epsilon(u, v)$ is a weight function on the edge (u, v). Clearly, if the edge does not exist in G, *i.e.* $(u, v) \notin E$, the weight function would yield 0. Otherwise, if the edge exists, the function would return some value, depending on how the weight function has been defined. Various criteria have been proposed to split the graph in segments. One of the most famous is the Minimum Cut by Wu and Leahy [195]. In particular, the method they propose is applied to image segmentation. In their work, they take the image pixels to be vertices in an undirected graph with weighted edges between neighbouring pixels and solve the min-cut/max-flow problem between every pair of pixels in the image, using the K smallest cuts to partition the original image into K regions. The problem is solved by recursively finding the minimum cuts that bisect the existing segments. The main drawback of this approach is that it tends to cut very small subgraphs without taking into account the global structure. This is due to the fact that 4.2 increases with the number of edges involved in the cut.

This problem has been addressed by Shi and Malik in their Normalized Cut method [166]. In their work, they propose a new measure of disassociation between two groups. Instead of looking at the value of total edge weight connecting the two partitions, their measure computes the cut cost as a fraction of the total edge connections to all the nodes in the graph. This *disassociation* measure is called *normalize nut* and is defined as:

$$Ncut(V_1, V_2) = \frac{cut(V_1, V_2)}{assoc(V_1, V)} + \frac{cut(V_1, V_2)}{assoc(V_2, V)}$$

where $assoc(V_1, V) = \sum_{u \in V_1, v \in V} w(u, v)$ is the total connection from nodes in A to all nodes in the graph. This alternative definition of the cut score avoids the issue introduced before because the cut that partitions out small isolated points will no longer have small cut value, since the cut value will almost certainly be a large percentage of the total connection from that small set to all other nodes.

The algorithm begins from an image (or a sequence of images) that is used to build a graph where the weight on the edge connecting two nodes is set to be a measure of the similarity between the two nodes. With this graph at hand, the discrete Laplacian is built as $\mathbb{D} - \mathbb{W}$, with $\mathbb{D}$ the diagonal matrix whose diagonal elements are the degree of a node and $\mathbb{W}$ the adjacency matrix of a weighted graph. An eigendecomposition is performed on the Laplacian and the eigenvector associated to the second smallest eigenvalue is used to bipartite the graph. Then the process is repeated recursively with the two partition created in the previous step, using some measure to determine if the process can be stopped (if the graph has been segmented enough or if some fixed number of partitions is reached) or if it has to continue. In their test section, the authors show that the propose approach is able to give good performance when used in still images and in motion sequences.

4.5 Median and Mean Graphs

One of the main goal pursued in this thesis, and in the machine learning field in general, regards the definition of algorithms by which a system can automatically construct models. And part of the job, regards the construction of such models from a bunch of data. The advantages of building models and prototypes are various, but mainly we aiming for compact summaries of the data that can be managed in a faster fashion with respect to the whole database of data. That is one of the reason that median graph has received much attention in the literature.

Before getting to the heart of the topic, note that in graph theory the term median graph is also used to define a specific class of graph [14]. According to [134], a median graph is a connected graph, such that for any three vertices u, v and w there is exactly one vertex x that lies simultaneously on a shortest (u, v)-path, a shortest (v, w)-path and a shortest (u, w)-path. “Famous” median graphs are trees and hypercubes.

Instead, in this section with median graph we refer to the solution to the *generalized median problem*, in which we try to find a good representative (a prototype) of a set of graphs. Indeed, one use is as a model for representing a set of graphs in a compact notation. It can also serve as the starting point for generating new graphs or expressing the variation in a graph set. Given the mean graph, each graph in the set can be produced from the mean by applying some transformation. If the distribution of these transformations can be found, then it can be sampled from and used to generate new graphs.

The problem was originally stated by Jiang *et al.* [92]. In their work, the authors propose two different definitions for median graphs, one called *generalized median graphs* and the other *set median*. The difference lies in the space of graphs where the respective median is searched for, while in both cases the problem aims at minimizing the sum of distances to all input graphs. Indeed, central in the concept of median graph is a distance function $d(g_1, g_2)$ that measure *dissimilarity* between two graphs g_1 and g_2.

In particular, the *generalized median graph* is defined on the space of all graphs. Let $\mathscr{G}$ be the set of all graphs. Let $G = \{g_1, g_2, \ldots, g_n\}$ be the set of graphs whose median graph we are trying to retrieve. Then, the generalized median graph $\bar{g}$ is defined as

$$\bar{g} = \arg\min_{g \in \mathscr{G}} \sum_{i=1}^{n} d(g, g_i)$$

The second definition, which takes the name of *set median*, the space where the media graph is searched for is the set of graphs in question. Hence, the *set median* $\hat{g}$ is defined as

$$\hat{g} = \arg\min_{g \in G} \sum_{i=1}^{|G|} d(g, g_i)$$

It is easy to see that, since the generalized median graph is drawn from a larger set of potential medians, it is a better representation than the set median. Unfortunately, the complexity of finding the generalized median graph is much greater than finding the set median. For this reason, the authors investigated in a search strategy that provides good approximate solutions in a reasonable time. For this reason, they developed a genetic search algorithm to tackle the optimization task. In their results, they show the advantage of generalized median graphs in comparison to set median graphs.

Bunke and Günter [38] present a solution to a similar but simpler problem. In their work, they aim at finding the mean graph between a pair of graphs, which takes the name of *weighted mean*. They take the standard weighted mean defined on a n-dimensional vector space as $\mathbf{z} = \gamma\mathbf{x} + (1-\gamma)\mathbf{y}$ and define the concept in the domain of graphs. Given two graphs G_1 and G_2 and a scalar γ, the weighted mean of G_1 and G_2 is defined as the graph $\bar{G}$ for which

$$d(G_1, \bar{G}) = (1-\gamma)d(G_1, G_2) \quad and \quad d(\bar{G}, G_2) = \gamma d(\bar{G}, G_2)$$

holds, with $0 \le \gamma \le 1$. Hence, the authors firstly find the set of edit operations that transform one graph into the other. As we seen in section 4.2, when we deal with a graph edit distance a cost function must be assigned to each edit operation. This bring us to the definition of the total cost c as the sum of all the costs of the edit operations in the edit sequence.

We are now able to produce the mean graph by applying the the subsequence of edit operations with cost $\frac{c}{\gamma}$. Note that this weighted median may not be unique. The authors apply their approach to line drawing analysis and produce sequences of weighted means between the line drawings of two different letters.

Bunke *et al.* [39] present two solutions to the generalized median graph problem. The first approach is based on combinatorial search in which the simultaneous transformation of an initially unknown graph into the set of graphs questioned is examined. This simultaneous transformation is described by a structure termed *multi match*. Several *multi match* structure are defined. The search for the generalized media graph is done with an algorithm based on $A*$ whose space of search is represented by the multi matches found. The second approach employs a genetic algorithm in order to solve the initial problem and comes in two variants. The first, called *GA Random*, initialize the population completely randomly. The second one takes the name of *GA Init*. In this case, the initial population are initialized with transformations that represent the given input graphs. From the performances point of view, the two algorithms are very different. The combinatorial search algorithm has exponential worst case execution time and very high memory requirements, meaning that the approach can be applied only to a set of graphs with a low number of nodes. On the contrary, the genetic algorithm performs much better in both its incarnations.

The common issue in all the works presented in this section is the dimension of the search space when it comes to search for the generalized median graph. An attempt to solve this problem is proposed by Ferrer *et al.* [55], in which they propose a method to place an upper bound on the sum of distances (SOD) from the generalized median graph to all graphs in the set. This result is achieved using the concept of maximum common subgraph of a graph set and establishing a relation with the generalized median graph itself. A maximum common subgraph of a graph set aims at finding the largest subgraph of a reference graph of the set which is isomorphic to a subgraph of each other graph of the set. This is a problem known to be *NP-hard*. Let G_e denote the empty graph while G_u be the union graph set. For any pairwise partition p of the set G, with $SOD(p)$ the sum of distances between the elements of the partition, then the bound on such distances from the median graph $\bar{g}$ to all the graphs in G can be defined as

$$max(SOD(p)) \leq SOD(\bar{g}) \leq min(SOD(G_e), SOD(G_u))$$

In their work, the authors prove that the sum of distances to the generalized median $\bar{g}$ has as an upper bound the sum of distances with respect to the maximum common subgraph $mcs(G)$. Hence, the following holds.

$$max(SOD(p)) \leq SOD(\bar{g}) \leq SOD(mcs(G)) \leq min(SOD(G_e), SOD(G_u))$$

In another work, Ferrer*et al.* [56] use again the concepts of maximum common subgraph (*mcs*) and minimum common supergraph (*MCS*) to present a reduction to the bounds defined in [92] in order to improve the efficiency of the algorithms used to tackle the problem of finding the median graph. In their work, they show that the number of nodes of the median graph must be in between the number of nodes of the maximum common subgraph and the number of nodes of the minimum common supergraph of a set of graphs:

$$0 \leq |mcs(G)| \leq |\bar{g}| \leq |MCS(G)| \leq \sum_{i=1}^{|G|} |G_i|$$

In their paper, the authors prove this result both from a theoretical point of view and give a detailed example to validate the theoretical results. Furthermore, the experimental results show a lower search time and hence an improved performance for both the combinatorial search and genetic algorithm presented above in this section.

Taking advantage of the results in both [55] and [56], Ferrer *et al.* [57] propose a new genetic algorithm for finding the approximate generalized median graph. The new bounds are used to reject population representing median graphs that fall outside the space where the median graph is supposed to be. They tested their approach on web pages (represented as graphs), and the results they show are more than positive.

Ferrer *et al.* [58,59] propose other two methods to tackle the generalize median graph problem. The first searches for generalized median graph in a reduced space. The second directly computes an approximation of the generalized median graph.

They both make use of an interesting graph embedding method proposed by Riesen *et al.* [150]. The key idea in Riesen's work is to use the distances of an input graph to a number of training graphs as a vectorial description of the graph. In Ferrer *et al.*search based method [58], this embedding technique is used to produce a vector describing each graph in a set G. All the graphs in the set are used as prototypes, meaning that the dissimilarity matrix produced by the algorithm is of dimension $|G| \times |G|$, while a vector describing a graph contains the dissimilarity with respect to each other graph of the set G. The dissimilarity function is a graph edit distance function with costs deletions and insertions of vertices of 1. The deletions and insertions of edges cost 0. Finally, edge and vertex substitutions have cost of 0 or inf (identical substitution or not). The result is that the Manhattan distance can be employed to compute the mean feature vectors. The mean feature vector represents the median graph feature vector. Unfortunately, this feature vector cannot be translated back to a graph directly. However, it can be used to limit the search space of the median graph. Under the cost function introduced above, the search space is composed of the vertex induced subgraphs of the minimum common supergraph of G. Furthermore, in their work the authors describe the search space of the generalized median graph as a rhombus with the minimum common supergraph at one end and the empty graph at the other. The graphs from the set G must all fall within this rhombus.

This rhombus search space is restricted by the graph in the set G with the smallest dissimilarity with respect to the median (smallest edit distance). Thus, the graph that has the minimum *sum of distances* to the graphs in G which is selected from the reduced search space must be the generalized median. The authors describe good results, in terms of time complexity, when the approach is applied to line drawings of letters.

The second method proposed by Ferrer *et al.*in [59] aims at computing an approximation of the generalized median graph. The method shares the same steps seen in the [58] until the search stage. In this work they employ the weighted mean of a pair of graphs

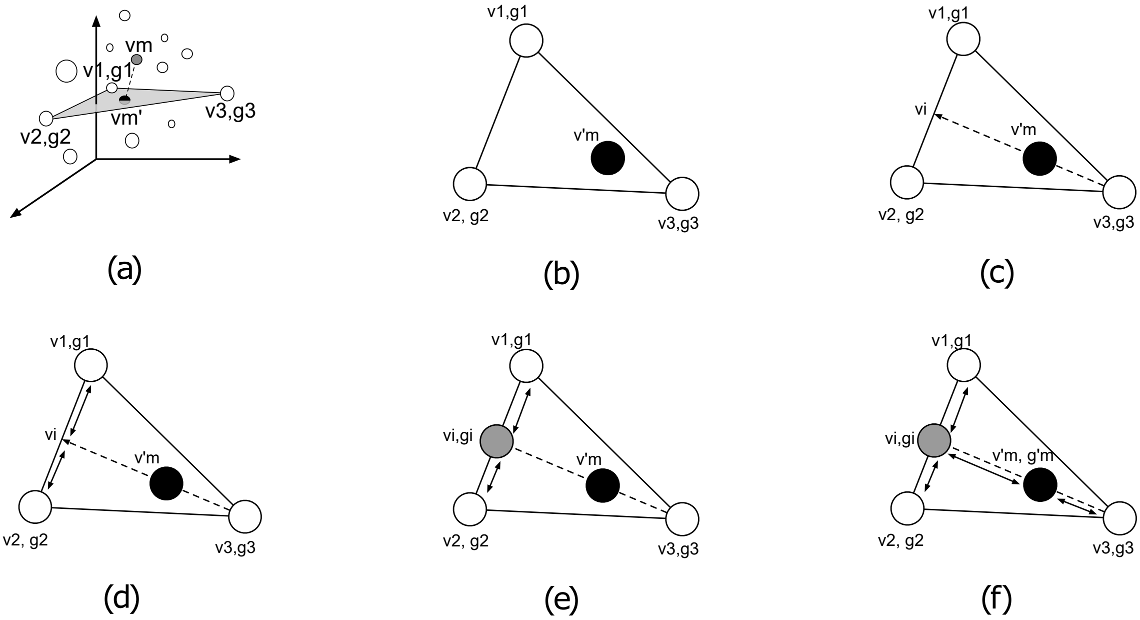


Figure 4.2: Triangulation steps from the approximate generalized median algorithm [59].

introduced at the beginning of this section (see [38]) in order to compute the approximate generalized median. In particular, the triangulation procedure depicted in figure 4.2 is performed. Figure 4.2(a) show the n-dimensional ($n = 3$ in the figure) points representing every graphs in the set G. The first step involves the selection of the three closest graphs to the mean point v_m. Let $g_1, g_2, g_3 \in G$ be these graphs, while v_1, v_2 and v_3 are the respective points. The median vector v'_m. Two out of three points are selected (*e.g.* v_1 and v_2), and the remaining points (v_3) is projected onto the line joining v_1 and v_2, obtaining the point v_i as Figure 4.2(c) shows. The next step involves the computation of the percentage of the distance of v_i between v_1 and v_2 (see Figure 4.2(d)). This allows to retrieve the graph g_i corresponding to the point v_i (Figure 4.2(e)). With g_i at hand, the percentage of distance between v_i and v_3 of v'_m can be computed, allowing to retrieve the graph g'_m (Figure 4.2(f)), which represents the approximate median graph the algorithm was looking for.

In the evaluation section, the authors compute the *sum of distances* of both the approximate median graph and the *set median* with respect to all the graphs in the set. In their tests, the approximate median graph results to outperform the set median. Unfortunately, a comparison with the exact generalized median graph is not possible due to size of the data sets, which make its computation infeasible from a computational time point of view.

Another very interesting results by Ferrer *et al.*is their extension of the definition of generalized median and set median in the spectral domain [54]. The space in which the median graph (in this case called respectively *generalized spectral median* and *set spectral median*) is searched is the same as in the original definition.

To conclude, we have seen several different approaches that try to tackle the generalized median graph problem from different perspectives. Even if initially the solutions to this problem were computationally too heavy to be applied to something different from a simple example, the recent approaches proposed in [57–59] showed that the usage for real world data sets is becoming feasible.

4.6 Generative Model

The last section of this chapter will deal with generative models for graphs. Generative models allow to define a probabilistic model that learn how data have been generated. This is usually done through the learning of some parameters which can then be used in tasks like categorization of new data or generation of new data. A lot of effort has been put in the research community on generative models in vectorial domain. Unfortunately, the contribution in the graphs domain is minor. In this section we will review some of the most interesting approaches.

Luo *et al.* [122] propose a method to construct a linear deformable model for graph structure through principal component analysis of the vectorized adjacency matrix of each sample graph. This allows to build a vectorial description of the whole set of graphs. The first issue that has been addressed before the construction of the vectorial descriptor is the lack of correspondences between the nodes of the graphs belonging to the sample set. Thus, an alignment step is defined. For this task, the algorithm for inexact graph matching proposed by Luo and Hancock [121] is used. Furthermore, to tackle the problem of graphs of different size, all the graphs in the sample have been padded adding dummy vertices (vertex with no connections). Once this pre-processing stage has been completed, the aligned and padded adjacency matrix A_k of each graph are vectorized.

$$a_k = vec(A_k)$$

These vectors are used to build the linear deformable model in order to express structural variations among the graphs in the set. Practically speaking, the mean μ and the covariance matrix Σ of the vectors representing the graphs are computed. Then, the eigendecomposition of the covariance matrix is performed, resulting in the ordered eigenvalues Λ and the corresponding eigenvectors Φ. The former matrix is used to retrieve the variance in the data through principal component analysis, while the eigenvalues determines the degree of variance associated with each component. The vectorial descriptor of the sample data is thus defined as the mean summed with a vector which encodes how the graph varies structurally. The vector is projected into the principle components of the distribution:

$$\hat{a}_k = \mu + \Phi b$$

where b is a parameter vector describing the degree of variation experienced by a graph k in the direction of the principle component. The authors test the model defined above with a dataset of face-representing graphs.

In another work, Xiao and Hancock [196] propose a similar method to build a generative model for graph structure. In their work, the authors commence from a sample of graphs where the correspondences between nodes are unknown. The proposed method deals with structural differences between graphs (*i.e.* different number of nodes in the graphs that belong to the sample set). The key idea is to embed the nodes of the graphs into a vector space by performing *kernel PCA*. Differently from the approach introduced before, here the PCA is performed on the heat kernel computed on each graphs. After the computation of the heat-kernel H_k representation of each graph, the Young-Householder decomposition [201] $H_k = Y_k^T Y_k$ is performed. Here, Y_k is the coordinate matrix and each column u of Y represents the vectorial coordinate of the node u of the graph k. Since the heat kernel is defined as $H_t = \Phi \exp[-\Lambda t]\, \Phi^T$ (with t the time parameter, Λ and Φ the eigenvalues and the eigenvector of the normalized Laplacian associate to the graph k), then we can define the coordinate matrix as

$$Y_k = exp\left[-\frac{1}{2}\Lambda_k t\right]\Phi_k^T$$

When different graphs undergo structural variations, the coordinate matrices are truncated. Furthermore, to deal with the non-correspondence problem, the algorithm proposed by Scott and Longuet-Higgins [163] is used to align the coordinate matrices, electing the largest graph as the reference one. Finally, the model is built performing the principal component analysis on the resulting coordinate matrices.

In the experimental section, the authors assess the performance of the proposed model for shape-analysis. In particular, experiments on the COIL dataset [135] are performed. The results show that the model can be adopted to both construct pattern spaces for sets of graphs and to perform graph clustering. Also, a scalability analysis of the model (*i.e.* compactness of the model) is performed, showing adequate results with very compact models.

Torsello [178] addresses the problem of learning archetypal structural models from examples. In particular, a generative model for graphs is devised. In this work, the author assumes nodes and edges to be governed by a set of independent Bernoulli trials, whose parameters must be estimated. Furthermore, the correspondences between the nodes of the samples are unknown. The parameters are estimated maximizing the likelihood of the observed graphs, marginalizing it over all possible node correspondences. To limit the exponential explosion of the set of correspondences, an Importance Sampling approach is employed. In the experimental section, the author uses the generative model in two different structural abstraction of shapes. The first one is represented by the Delaunay graph build over a set of image features. The second test bed is made up of shock graphs constructed from objects.

An extension of this work is presented Torsello and Dowe [180]. In this work, the authors define the generative models in the same way, assuming that each vertex and edges of the graphs are distributed following a set of independent Bernoulli trials. The assumption of no node-to-node correspondence is still valid. In this work, this problem is tackled by an EM-like approach in which the estimation of the node correspondences is alternated with the model parameters estimation. In particular, the first phase of the process is cast as an instance of graph matching. The new correspondence set is then used to estimate the model parameters while the second phase. The model parameters are estimated employing a Minimum Message Length criterion which states that the best model describing the data is the one resulting in the shortest length description of both the model and the encodings of the data given the model. The model is then tested on a shape recognition task, where the effectiveness of the learning approach is assessed.

Extending this work, Torsello and Hancock [183] propose an unsupervised learning of tree structures in an information theoretic setting. The problem is posed as that of learning a union structure from the set of examples with unknown correspondences. The structure is built through a set of edit operations. Furthermore, each node is endowed with a random variable which represents its probability. In order to estimate the correspondences, the union structure and the node probabilities, an iterative optimization process is employed. The optimization process is based on the assumption (proved in the paper) that a relation between the edit distance and the description length exists. The optimization process counts three steps. In the first one, correspondences are located in order to minimize the edit distance between graphs. The second step involves the construction of the union structure by editing an initial one. In this step, the description length is the function to minimize. The last step sees a maximum likelihood estimation of the nodes probabilities. The model is tested in classification and clustering tasks of a set of shapes based on their shock graphs.

II

Non-Rigid Shape Analysis

5

Non-Parametric Spectral Model for Shape Retrieval

Non-rigid 3D shape retrieval is an active and important research topic in content based object retrieval. This problem is often cast in terms of the shapes intrinsic geometry due to its invariance to a wide range of non-rigid deformations. In this chapter, we devise a novel generative model for shape retrieval based on the spectral representation of the Laplacian of a mesh. Contrary to common use, our approach avoids the ubiquitous correspondence problem by transforming the eigenvectors of the Laplacian to a density in the spectral-embedding space which is estimated non-parametrically.

In this chapter we show that this model can efficiently be learned from a set of 3D meshes. The experimental results on the SHREC'14 benchmark show the effectiveness of the approach compared to the state-of-the-art.

5.1 Introduction

Non-rigid 3D shape retrieval is an active and important research topic in content based object retrieval. 3D models are widely-used in many application areas, such as computer aided design, medical modelling, bioinformatics, and a large number of 3D models have become available on the web. Hence, the ability to recognize a deformable object's shape has become central in modern shape retrieval methods. A shape retrieval algorithm can be defined as a query executor where, given a shape, the result set is the collection of shapes which belong to the same class. Moreover, the shapes returned should be ordered by decreasing similarity to the query shape. Several different approaches have been proposed to address the non-rigid shape retrieval problem. Depending on the paradigm used to represent the objects, there are view based methods and model based methods. The former try to exploit the observation that similar 3D shapes look similar from the same perspective. Hence, several 2D projections have been used to represent the shape, *e.g.* silhouettes [42]. On the other hand, model based techniques make use of the 3D shape directly. Both structural and topological techniques and geometric techniques belong to this class of approaches. Structural and topological techniques take into account structural properties like connected components or the holes in the shape. An example of such approach is the Surface Penetration Map by Yu *et al.* [202], where topological information is extracted from a model by morphing it into a sphere. Finally, geometric techniques make use of the quantitative properties of the shapes (*e.g.* volume, aspect ratio, surface area, curvature) that could be used to characterize the shape either globally or locally. Global methods aim to capture the characteristics of the whole shape and tend to be more computationally efficient. As an example, Zhang and Chen [204] propose several methods to compute efficiently global features and use them for 3D shape retrieval purpose. Features are also employed indirectly in the methods which exploit the distributions of those measurements in place of the value of the feature itself [140]. With respect to global methods, local approaches can be used for partial matching, but they are usually less computationally efficient. These methods consider local properties around the neighbourhoods of points on the surface, such as curvature, volume and area [93].

5.1.1 Related works

One of the most popular approach within the geometric techniques involves the definition of an invariant representation of a shape capable of capturing its geometrical and topological properties, but at the same time being insensitive to transformations like bending and stretching, and robust to acquisition resolution or noise. On top of that, storage and computational costs of the representation should be taken into account. For these reasons, a widely used approach to define shape descriptors is through spectral shape analysis. Research efforts have recently resulted in several spectral descriptors [12, 149, 174] usually based on the spectral decomposition of the Laplace-Beltrami operator, due to its invariance to isometries. In particular, the mesh-Laplacian, the discrete counterpart of the Laplace-Beltrami operator applied to surfaces, has been extensively used to provide spectral representations of structures [118]. Reuter *et al.* [148] suggest to use the sequence of eigenvalues (spectrum) of the Laplace-Beltrami operator of a surface as fingerprints, while Jain and Zhang [88] propose to use the eigenvalues of the geodesic distance matrix of a 3D object in order to build the associated shape descriptor. Huang *et al.* [84], on the other hand, build the signature directly over local features, selecting discriminative volumetric features over pre-aligned shapes.

The aggregation of local descriptors to build a global descriptor is a general thread in the literature. For this purpose, the *Bag-of-Features* (BoF) paradigm is quite popular and has been successfully applied to 3D shape description [31, 114, 120, 177]. Li and Hamza [114] used the BoF paradigm combining the exploitation of hierarchical structures of the shape, such as pyramid matching [74] and spatial relationship [31]. They proposed to adopt the eigenfunction associated with the second-smallest eigenvector of the Laplace-Beltrami operator in order to build a global surface coordinate system which is insensitive to shape deformation, showing that the introduction of global spatial context could improve the effectiveness of their descriptor in 3D shape recognition. Spatial pyramid [114, 120], is the term used to identify this approach. Other approaches inspired by text-analysis have been proposed. For instance, in [22, 82] the authors adopt higher-order models defining relations between ‘geometric words’. Within the bag of features model, features quantization is generally performed through unsupervised learning. Departing from this approach, Litman *et al.* [117] recently proposed a new supervised BoF framework mapping the discriminative training directly into the dictionary construction step. Finally, methods which aim at finding the correspondences between two or more shapes have been used for shape retrieval, using the correspondences found as a similarity measure between shapes and classifying the query shape accordingly [18, 41].

5.1.2 Main contribution

In this chapter, we propose a novel and efficient supervised approach for the construction of a generative model based on the spectral decomposition of the Laplace-Beltrami operator. To this end, we define a statistical framework that models a shape as two independent generative models for the eigenvector and the eigenvalue components of the spectral representation of the Laplacian. In particular, we assume the spectral embedding space of the eigenvector part to be a set of independent observations which follows an unknown distribution. We estimate the underlying distribution in a non-parametric way, through kernel density estimation. The two most important benefits brought by this point-wise approach regard the robustness (and somehow, the scalability) of the algorithm and its acquired invariance to vertex correspondences. In fact, the adoption of kernel density estimation allows to avoid sampling density issues, that is, the underlying probability density remains the same even if the meshes are sampled with different resolutions. Finally, two alignment steps are defined and applied to the eigenvector model in order to take care of the residual rotations between eigenvector matrices. On the other hand, in order to define a descriptor that is robust to small non-isometric perturbations, we assume the eigenvalues to be log-normally distributed.

5.2 Background

In this section we briefly introduce the notations and some theoretical aspects used in this work. A surface is a smooth compact 2-manifold $\mathscr{S}$ without boundary isometrically embedded in the Euclidean space $\mathbb{R}^3$ with geometry induced by the embedding. With shape, we denote the quotient group of surfaces modulo isometries. Finally, we define a mesh as a discrete representation of a surface embedded in $\mathbb{R}^3$. In order to address the shape retrieval task, we define an invariant representation of a shape obtained as the result of a data-driven process. To this end, let us define the discrete Laplacian as the discretization of the continuous Laplace-Beltrami operator on the mesh. The Laplacian operator is a differential operator which stores intrinsic geometry information (vertex-wise) about a mesh, allowing to preserve the relationship among vertices when isometric transformations are applied. Several approaches have been proposed to compute the Laplacian matrix from a mesh. In this work we adopt the algorithm proposed by Belkin *et al.* [16] which offers point-wise convergence guarantees and was experimentally shown to be quite robust. In particular, it approximates the Laplace operator of a surface from a mesh with point-wise convergence and they show that it converges (for fine meshes) to the Laplace-Beltrami operator. The Laplacian matrix is computed as $\mathscr{L} = A^{-1} * W$, with A the area elements on the diagonal and W a symmetric weight matrix. $\mathscr{L}$ is not symmetric but satisfy $\Phi^T A\, \Phi = I$ (I the identity matrix). By setting $\Psi = A^{\frac{1}{2}}\Phi$ and $\mathscr{L}_N = A^{1/2}\mathscr{L}A^{-\frac{1}{2}} = A^{-\frac{1}{2}} * WA^{-1/2}$, the normalised Laplacian matrix L_N has the same eigenvalues as $\mathscr{L}$ and its eigenvectors Ψ are a discrete approximation of the eigenfunctions by taking integrals over the area elements rather than point samples over the vertices as in Φ. Indeed $\mathscr{L}_N$ is symmetric, positive definite and its eigenvectors are robust with respect to changes in sampling densities. The spectral representation of a mesh can be computed from the Laplacian through singular value decomposition. Given a discrete Laplacian $\mathscr{L}$, its eigendecomposition is $\mathscr{L} = \Phi\Lambda\Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the matrix whose diagonal contains the ordered eigenvalues, while $\Phi = (\phi_1|\phi_2|...|\phi_{|V|})$ is the matrix whose columns are the ordered eigenvectors. This decomposition is unique up to a permutation of the vertices of the mesh, a change of sign of the eigenvectors, or a change of basis over the eigenspaces associated with a single eigenvalue, *i.e.*, the following properties hold:

$$\begin{aligned} \mathscr{L} &\simeq P\mathscr{L}P^T = P\Phi\Lambda(P\Phi)^T && (5.1) \\ \mathscr{L} &= \Phi\Lambda\Phi^T = \Phi S\Lambda S\Phi^T && (5.2) \end{aligned}$$

where $\simeq$ indicates isomorphism of the underlying surfaces, P is a permutation matrix and S is a diagonal matrix with diagonal entries equal to ± 1.

5.3 Model definition

The main contribution of the proposed work consists in the definition of a generative model based on the spectrum of the Laplacian matrices associated to a set of meshes representing the same shape. The defined model is able to capture the structural variations between the meshes that belong to the same class, and it takes into account all the invariances of the spectral representation. The use of spectral representation to perform a vertex embedding is an approach underlying several approaches. Many use the so called *heat kernel embedding* [31, 118, 174] using a low rank approximation of the Heat kernel to embed vertices in an Euclidean space in a manner that is invariant to surface isometries

$$\begin{aligned} XX^T &\approx \exp(-\mathscr{L}t) = \Phi\exp(-\Lambda t)S\Phi^T && (5.3) \\ X &\approx \Phi\exp(-\Lambda\frac{t}{2}) && (5.4) \end{aligned}$$

However, this representation is unique only up to isometries in the Euclidean embedding space. Indeed, applying an isometric transformation $\mathscr{O}$ to X we obtain $X\mathscr{O}\mathscr{O}^T X^T = XX^T$. Hence, this representation is insensitive to eigenvectors orthogonal mixing, in particular for the eigenvectors associated to very similar eigenvalues. Similar analysis holds with the *commute time* embedding [78]. In all these approaches the embedding model mixes eigenvectors and eigenvalues, using the eigenvalues to modulate the variability of the eigenvector, with an exponential decay in the case of the heat kernel embedding, and with an inverse-law in the case of the commute time. In the definition of our model, we follow White and Wilson [192] and we build two separate and independent models, one for the eigenvectors and one for the eigenvalues of the Laplacian matrix. The lack of eigenvalue modulation in the spectral embedding is compensated by a data driven modelling of the variation in each spectral dimension through non-parametric density estimation. Hence, let M be a mesh, and let $\mathscr{L}_M = \Phi_M \Lambda_M \Phi_M^T$ be the eigendecomposition of the associated Laplacian matrix. The eigenvector component Φ^M is modelled as an unknown distribution $\mathscr{F}$ on the d-dimensional spectral embedding space $\Omega_d \subseteq \mathscr{R}^d$. The d-dimensional embedding of the eigenvector matrix is obtained from a simple sub-sampling operation over the matrix Φ^M, taking the first d columns (first d non-constant eigenvectors) of the matrix or, in other words, taking the first d non-constant eigenvectors which are associated with the d smallest non-trivial eigenvalues. The resulting $n \times d$ matrix is assumed to be a collection of n d-dimensional points that belong to the embedding space Ω_d. Eigenvectors are often assumed to be of unit Euclidean norm. Unfortunately, this results in a compression of the value of the components of the spectral embedded points as the number of the vertices of the mesh (or alternatively, the n points of our representation) increases. To overcome this issue, we scale the d eigenvectors which belong to the embedding by multiplying for the number of vertices of the mesh n. So, let $\hat{\Phi}^M = \left(\bar{\Phi}^M\right) \cdot n$, with $\bar{\Phi}^M \in \mathscr{R}^{n \times d}$, be the resulting embedded eigenvector matrix. Note that we are not assuming nor requiring the meshes to have the same number of vertices. Indeed, the number of vertices of the meshes belonging to the datasets used in the experimental section are different.

After this pre-processing stage, we are able to cast the learning phase of the eigenvector component into a non-parametric density estimation of the distribution of the spectral embedding points $\hat{\phi}_1^M, ..., \hat{\phi}_n^M$. With those assumptions in mind, let the eigenvector model parameter Θ^Φ consists of a collection of N d-dimensional vectors $\theta_1^\Phi, ...\theta_N^\Phi$. We assume that all these points are samples from the unknown density function $\mathscr{F}$. In the learning phase, the parameter Θ^Φ is obtained aligning and stacking (and eventually sub-sampling) all those spectral embedding points together from the sample meshes which belong to the same class, or in other words, all the meshes that belong to the training set and represent the same shape. The density model allows us to avoid the vertex-alignment problem allowing many-to-many relations among vertices from different meshes and the separation of eigenvector and eigenvalue constrains the set of isometries of the embedding space to the discrete set of sign changes. However, sampling noise and approximate isometries of the surfaces result in mixing of the eigenspaces, especially between those with similar eigenvalues. We address this by adopting a kernel Procrustes alignment approach, locally optimizing over the orthogonal group $\mathbb{O}(d)$. Thus, the posterior probability $P\left(\Phi^M | \Theta^\Phi\right)$ can be computed solving the problem:

$$\max_{\mathscr{O}\in\mathbb{O}(d)} \max_{S\in\{\pm1\}^d} (Nh^d)^{-n} \prod_{i=1}^{n} \sum_{j=1}^{N} e^{-\frac{\|\mathscr{O}S\phi_i^M - \theta_j^\Phi\|^2}{2h^2}} \tag{5.5}$$

expressing the optimization problem in terms of the product of Parzen-Rosenblatt kernel density estimators. In particular, ϕ_i^M is one of the sample points obtained taking the first d components of the i-th row of the eigenvector matrix associated to the mesh M, while θ_j^Φ is the j-th component (still a d-dimensional row vector) of the eigenvector model Θ^Φ. Here the eigenvector model is assumed to be a collection (an array) of samples taken from the training set during the learning phase. Furthermore, in order to minimize the variance between all the eigenvector matrices of the meshes of the training set (and simultaneously increasing the variance with the eigenvector matrices which does not represent the same shape), we introduce two alignment steps. In particular, the matrix $\mathscr{O}$ that appears in the optimization problem 5.5 represents an orthogonal transformation used to align as much as possible all the points of our representation, while S (to not be confused with S and $\mathscr{S}$ introduced in section 5.4) is a matrix containing just ±1 used to solve the sign ambiguity problem (see 5.4.1).

In the computation of the kernel density estimator, a rule of thumb must be chosen in order to estimate the bandwidth h of the kernel. In this work, we employ the Silverman's rule of thumb for the multivariate case [171] and we estimate the bandwidth as

$$h = \left(N\frac{d+2}{4}\right)^{-\frac{1}{d+4}} \sigma \tag{5.6}$$

where d is the embedding dimension, N is the number of meshes employed in the construction of the model Θ^Φ (*i.e.* the dimension of the training set) and σ is the standard deviation computed as the squared root of the mean variance (the trace of the covariance matrix Σ divided by the n nodes of the model) of the eigenvector model

$$\sigma = \sqrt{\frac{1}{n}Tr(\Sigma)} \tag{5.7}$$

Note that we are not assuming that the eigenfunctions are independent. Indeed, we build our eigenvector model as a point cloud. Given an eigenvector matrix (as columns), each point is given by a row of this eigenvector matrix. We assume those points to be independent observations of an unknown underlying distribution which we estimate non parametrically. And that's why we state that our model is independent of the sampling, since even if we prune out part of those points, the underlying distribution remains the same. Robustness analysis presented in section 8.3 confirms the assumption. A more practical analysis about the construction of such model is demanded to section 5.4.

As regards to the eigenvalue model Θ^Λ, we follow Aubry *et al.* [12] for the choice of the distribution to be used. In their work, they showed empirically that the eigenenergies of an articulated shape are log-normally distributed random variables due to stability considerations derived from matrix perturbation theory. As a result, we model the set of eigenvalues as a series of independent log-normal distributions, one for each of the

d eigenvalues involved in the construction of the model. More formally, we define the posterior probability (relative to the eigenvalue component) of a mesh M to belong to the class modelled by Θ^Λ as

$$P(\Lambda^M|\Theta^\Lambda) = (2\pi)^{\frac{d}{2}} \prod_{i=1}^{d} \frac{1}{\lambda_i \sigma_i} e^{\frac{-(\ln \lambda_i - \mu_i)^2}{2\sigma_i^2}} \tag{5.8}$$

In particular, μ_i and σ_i represent the model parameters to be learned from the training set and d is the number of eigenvalues used in the model (*i.e.* the embedding dimension). Note that the log-normal distribution concerns corresponding eigenvalues among the different meshes of the training set. Finally, note that we are not assuming the eigenvalues to be independent, but rather that they follow the same underlying distribution. It is the deviation from that distribution that is considered as independent noise and modelled as log-normals following [16].

To address the classification task, those models are employed inside a statistical framework in the prediction phase of a new mesh. Let $\Theta = \left\{\Theta^\Lambda, \Theta^\Phi\right\}$ be such model. Still assuming the independence of the two models, we define the posterior probability of a given mesh M, whose spectral components of the associated Laplacian matrix are the eigenvalue matrix Λ^M and the eigenvector matrix Φ^M, as

$$P(M|\Theta) = P(\Lambda^M|\Theta^\Lambda)P(\Phi^M|\Theta^\Phi) \tag{5.9}$$

5.4 Learning phase

In order to address the retrieval task, we define two separate phases. The first one is the learning phase. The main goal of the learning process is to estimate the model parameters for both the eigenvector model and the eigenvalue model. Before estimating the parameters, the training dataset must undergo a pre-processing phase. Given a set of meshes $S = \{M_1, M_2, \dots, M_N\}$ representing the same shape $\mathscr{S}$ in different poses (and which represents our training set), the first step involves the computation of the Laplacian matrix associated to each mesh M_i, obtaining the set $\{\mathscr{L}_1, \mathscr{L}_2, \dots, \mathscr{L}_N\}$. Applying a singular valued decomposition to each Laplacian matrix, we obtain the eigendecomposition $\mathscr{L}_i = \Phi_i \Lambda_i, \Phi_i^T$ where the diagonal elements of Λ_i are in ascending order and we remove the trivial eigenvalue and the corresponding eigenvector (which results to be constant). Finally, we are not interested in the whole Laplacian eigendecomposition, but just an embedding of it. In other words, given d as the embedding dimension, we keep the first d eigenvalues and the first d eigenvectors of the eigendecomposition of the Laplacians. As a result of this pre-processing phase, we obtain the set $\{(\Lambda_i, \Phi_i)\}_{i=1,\dots,N}$, where $\Lambda_i \in \mathscr{R}^{d \times d}$ and $\Phi_i \in \mathscr{R}^{n_i \times d}$ (n_i the number of vertices of the mesh i).

5.4.1 Eigenvector model

As said before, the result of the pre-processing phase is the set of embedded eigenvector and eigenvalue matrices of the Laplacians of the meshes which represent the same shape $\mathscr{S}$. The next step in the construction of the eigenvector model involves a simple stacking operation of all the matrices that belong to the set. Hence, the eigenvector model of the shape $\mathscr{S}$, denoted as $\Phi^{\mathscr{S}}$, is defined as

$$\Phi^{\mathscr{S}} = \begin{bmatrix} \phi_1^1 & \phi_2^1 & \dots & \phi_d^1 \\ \phi_1^2 & \phi_2^2 & \dots & \phi_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1^N & \phi_2^N & \dots & \phi_d^N \end{bmatrix}$$

In particular, ϕ_j^i denotes the j-th non-trivial eigenvector (which is still a column vector) of the i-th mesh of the training set. In other word, we perform a simple vertical concatenation of the embedded eigenvector matrices of the meshes in the training set. Therefore, the final dimension of the model $\Phi^{\mathscr{S}}$ can be expressed as $\left(\sum_{i=1}^{N} ||M_i||\right) \times d$, with $||M_i||$ the number of vertices of the i-th mesh.

As the quantity of available training data grows, this matrix becomes very big. The main advantage of the point-wise approach adopted by the proposed method consists in the ability to overcome the issue deriving from the dimension of the model. In particular, treating each row of the eigenvector model as an independent observation in order to estimate the posterior probability through equation 5.5, allows to randomly sub-sample the model $\Phi^{\mathscr{S}}$ still preserving the underlying distribution of the points.

Eigenvector sign ambiguity

It is well known that the eigenvectors computed by eigendecomposition are unique only up to a linear transformation. In particular, since eigenvectors are often assumed to be of unit Euclidean norm, they are recovered up to a sign factor. Since the proposed method characterize each vertex of a mesh with a feature vector, a sign disambiguation is mandatory. There are several approaches that can be used to solve the sign ambiguity issue. A reliable approach involves the computation of the correlation between two functions (the probability density function in our instance). If the correlation grows after a sign flip, then the eigenvector should be flipped. However, this technique results computationally heavy as the size of the model grows. For this reason, we have to rely on an heuristic-based method in order to solve the sign ambiguity issue, even if it does not guarantee the identification of all the sign flips that should be performed. Given the eigenvector matrices Φ^A and Φ^B, computed through the eigendecomposition of the Laplacian of two meshes M_A and M_B (which represent the same shape in different poses), let ϕ_i^A and ϕ_i^B be the i-th eigenvector. We assume the eigenvectors to be random variables whose probability density functions are unknown. In addition, we assume all i-th eigenvectors associated to each mesh representing the same shape to share a similar probability density function (up to a linear transformation). Such linear transformation does not influence the shape of

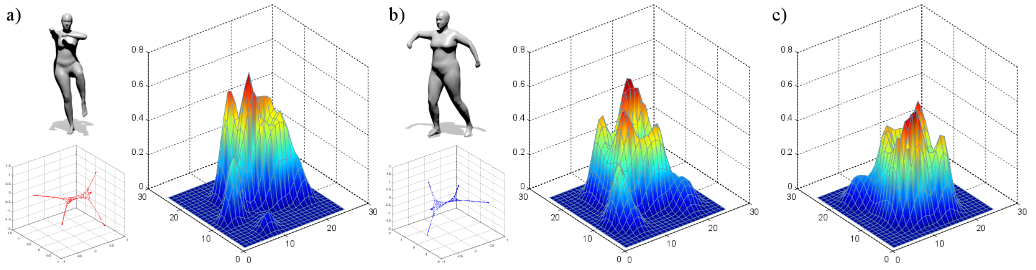


Figure 5.1: Orthogonal transformation of the eigenvector matrix example. *a)* shows the mesh which is used as the reference mesh. The graph below shows the plot of the first three non trivial eigenvectors of the reference mesh, while the graph on the right of the mesh shows the kernel density estimation computed on the eigenvector matrix. Hence, here we want to align the eigenvectors of the mesh *b)* according to the eigenvectors of *a)*. The result of the optimization process can be seen in *c)*.

the *pdf*, but it influences the sign of the peak of such function. In other words, if the sign of the peaks disagree, a sign flip occurs. More formally, after the selection of a mesh as reference mesh (for example A), we define our heuristic approach to the sign ambiguity solution as

$$\phi_i^B = \begin{cases} \phi_i^B(-1) & \text{if } x_i^{A*} < 0 \text{ and } x_i^{B*} \geq 0\ , \\ \phi_i^B(-1) & \text{if } x_i^{A*} \geq 0 \text{ and } x_i^{B*} < 0\ , \\ \phi_i^B & \text{otherwise.} \end{cases} \tag{5.10}$$

where x_i^{A*} and x_i^{B*} are the peaks of the pdf computed through kernel density estimation. The densities are evaluated at 100 points covering the range of values in the eigenvectors. Hence, before the construction of the eigenvector model $\Phi^{\mathscr{S}}$, we randomly select a mesh M_j as the reference mesh. Then we flip the eigenvector matrices associated to the other meshes according to 8.9. As mentioned in section 5.3, the main drawback of the proposed solution to the sign-ambiguity problem is its non-robustness to multiplicity of eigenvalues. Furthermore, similar eigenvalues could result in mixing of the eigenspaces. The problem is addressed through an additional alignment step, which is explained in details in section 5.4.1.

Eigenvector orthonormal transformation

The alignment step introduced in section 5.4.1 acts as an orthonormal transformation and thus it does not violate the orthogonality of the basis. Nevertheless, an isometric difference between the eigenvector matrix of the reference mesh and the eigenvector matrix of an another mesh is expected. In order to overcome the misalignment and consequently to minimize distances between feature vectors of the two matrices, a further alignment step is employed. From this viewpoint, the formulation of such optimization problem is similar to the Orthogonal Procrustes Problem (OPP), that is solving for the optimal orthogonal matrix that maps the first set of points into the second one. However, our algorithm differs from the standard OPP since we seek for the orthonormal transformation which maximizes a certain probability. In particular, we define the probability density in terms of Parzen-Rosenblatt kernel density estimator (Gaussian kernel)

$$P_h(x) = \frac{1}{n \cdot h} \sum_{j=1}^{n} e^{-\frac{1}{2}\frac{\|x-y_j\|^2}{h^2}} \tag{5.11}$$

By integrating the constraints of the optimization problem into our definition of probability density, we get

$$\underset{\mathcal{O}\in\mathbb{O}(d)}{\arg\max} \prod_{i}^{m} \sum_{j}^{n} e^{-\frac{1}{2}\frac{\|\mathcal{O}x_i-y_j\|^2}{h^2}} \tag{5.12}$$

where m is the number of vertices of the mesh we want to align to the reference mesh y and the parameter h is the bandwidth computed according to equation 8.7.

In order to solve 5.12, we firstly have to calculate the gradient with respect to an incremental variation and later compute iteratively the orthonormal transformation. The log-likelihood of the problem, after the introduction of the additive rotation $\mathcal{T}$, is given by

$$\ell_{\mathcal{L}}(y|x) = \sum_{i}^{m} \log\left(\sum_{j}^{n} e^{-\frac{1}{2}\frac{\|\mathcal{T}\mathcal{O}x_i-y_j\|^2}{h^2}}\right) \tag{5.13}$$

Let $\alpha_{i,j}$ be defined as $\alpha_{i,j} = e^{-\frac{1}{2}\frac{\|\mathcal{O}x_i-y_j\|^2}{h^2}}$. Then, by deriving the function in 8.13, with respect to the additional rotation $\mathcal{T}$, we get

$$\frac{\partial \ell_{\mathcal{L}}}{\partial \mathcal{T}} = \sum_{i} \frac{\sum_j \alpha_{ij}\left(-\frac{1}{2}\frac{\frac{\partial}{\partial\mathcal{T}}\|\mathcal{T}\mathcal{O}x_i-y_j\|^2}{h^2}\right)}{\sum_j \alpha_{ij}} \tag{5.14}$$

In particular, the partial derivative obtained by deriving with respect to the identity $\mathcal{I}$ ($\mathcal{T} = \mathcal{I}$), results

$$\frac{\partial}{\partial\mathcal{T}}\|\mathcal{T}\mathcal{O}x_i - y_j\|^2 = -2y_j(\mathcal{O}x_i)^T \tag{5.15}$$

$$= -2y_j x_i^T \mathcal{O}^T \tag{5.16}$$

We can rewrite 5.14 as

$$\frac{\partial \ell_{\mathcal{L}}}{\partial \mathcal{T}} = \underbrace{\left(\sum_{i} \frac{\sum_j \alpha_{ij} h^{-2} y_j x_i^T}{\sum_j \alpha_{ij}}\right)}_{\text{A}} \mathcal{O}^T = A\mathcal{O}^T \tag{5.17}$$

In order to find the maximum, we want to project the gradient to the null space. $\mathscr{T}$ is an orthogonal rotation matrix, hence it belongs to the Lie group $\mathbb{O}(d)$ (with d the embedding dimension). The tangent space (which we compute through the gradient of the log-likelihood 5.17) at the identity element of a Lie group is its Lie algebra, which represents the skew-symmetric matrices space. Since the skew-symmetric component of a matrix B is given by $\frac{B-B^T}{2}$, in order to nullify such component, we want to make B symmetric. More precisely, in order to find the maximum we want to make $A\mathscr{O}^T$ symmetric (which means $A\mathscr{O}^T = (A\mathscr{O}^T)^T$. Indeed

$$\frac{A\mathscr{O}^T - (A\mathscr{O}^T)^T}{2} = 0 \tag{5.18}$$

Treating the problem as an Orthogonal Procrustes Problem, the rotation matrix $\mathscr{O}$ which symmetrizes $A\mathscr{O}^T$ is computed through singular value decomposition of the matrix A. The decomposition yields $svd(A) = ULV^T$, while we can compute the rotation matrix as $\mathscr{O} = UV^T$. It is easy to see that $\mathscr{O}$ symmetrizes the gradient defined above, indeed

$$A\mathscr{O}^T = (ULV^T)(VU^T) = ULU^T \tag{5.19}$$

To summarize, the computation of the rotation matrix $\mathscr{O}$ is achieved using the following algorithm.

1. Initialize $\mathscr{O} = I$
2. Compute α_{ij} (8.14) for each $i = 1,\dots,n$ (n the number of vertex of a mesh) and $j = 1,\dots,N$ (N the number of points of the model)
3. Compute the matrix A (according to 5.17)
4. $svd(A) = ULV^T$
5. $\mathscr{O} = UV^T$
6. If the convergence is achieved, *i.e.* $A \approx A^T$, or the maximum number of iterations allowed is reached, end the algorithm, otherwise repeat from 2

Once both the alignment steps are computed on the spectral decomposition of each mesh belonging to the training set, we are able to build the matrix $\Phi^{\mathscr{S}}$ through the stacking operation mentioned at the beginning of this section. We decide to not combine the two alignment steps (the sign disambiguation and the orthonormal transformation) since the former represents just a coarse alignment, while the latter represents a fine alignment step which is effective only for smaller and local alignments.

5.4.2 Eigenvalue model

The second independent model involved in the proposed method regards the eigenvalues computed on the Laplacian of each mesh of the training set S (see section 5.4). The eigendecomposition produces the set of pairs $\{(\Lambda_i, \Phi_i)\}_{i=1,\dots,N}$. Especially, Λ_i contains the first d non-trivial eigenvalues of the i-th mesh of the training set as its diagonal elements. Let $\Lambda^{\mathscr{S}}$ be a $N \times d$ matrix whose rows are the eigenvalues extracted from the Λ_is.

$$\Lambda^{\mathscr{S}} = \begin{bmatrix} diag(\lambda_1^{\mathscr{S}}) \\ diag(\lambda_2^{\mathscr{S}}) \\ \vdots \\ diag(\lambda_N^{\mathscr{S}}) \end{bmatrix} = \begin{bmatrix} \lambda_{1_{1,1}}^{\mathscr{S}} & \cdots & \lambda_{1_{d,d}}^{\mathscr{S}} \\ \lambda_{2_{1,1}}^{\mathscr{S}} & \cdots & \lambda_{2_{d,d}}^{\mathscr{S}} \\ \vdots & \ddots & \vdots \\ \lambda_{N_{1,1}}^{\mathscr{S}} & \cdots & \lambda_{N_{d,d}}^{\mathscr{S}} \end{bmatrix}$$

We assume that all the j-th eigenvalues of $\Lambda^{\mathscr{S}}$, with $j = 1, \dots, d$, are distributed as a log-normal distribution (see equation 8.5). In order to learn the model parameters μ (the mean) and σ^2 (the variance), we do a maximum likelihood estimation using

$$\hat{\mu} = \frac{\sum_i \ln x_i}{N}, \quad \hat{\sigma}^2 = \frac{\sum_i (\ln x_i - \hat{\mu})^2}{N} \tag{5.20}$$

As a result of this learning phase, we compute a pair of parameter (μ, σ^2) for each of the d eigenvalues of the model. Hence, the eigenvalue model for a particular shape $\mathscr{S}$ is represented by the set $\Theta^{\Lambda} = \left\{(\mu_i, \sigma_i^2)\right\}_{i=1,\dots,d}$.

5.5 Model inference

The learning phase of the proposed method produces two separate models, both referring to a shape $\mathscr{S}$. The eigenvector model Θ^{Φ} contains the embedded eigenvector matrices stacked together after the two alignment steps introduced in section 5.4.1 and 5.4.1, while the eigenvalue model Θ^{Λ} contains the parameters set learned on the eigenvalues of the training set. Once both models are computed, we can combine them to compute the posterior probability of a new mesh M^* (whose spectral decomposition is (Φ^*, Λ^*)) with respect to the model representing a mesh $\mathscr{S}$. Assuming the independence of the two models, we can define the conditional probability as

$$P(\mathscr{S} \mid M^*) = P(\Phi^* \mid \Theta^{\Phi}) P(\Lambda^* \mid \Theta^{\Lambda}) \tag{5.21}$$

Since both $P(\Phi^* \mid \Theta^{\Phi})$ and $P(\Lambda^* \mid \Theta^{\Lambda})$ come from a log-derivation, equation 5.21 can be rewritten as

$$\log \quad P(\mathscr{S} \mid M^*) = \ell_{\mathscr{S}}(\Phi^* \mid \Theta^{\Phi}) + \ell_{\mathscr{S}}(\Lambda^* \mid \Theta^{\Lambda}) \tag{5.22}$$

where the eigenvector model log-likelihood is defined as

$$\ell_{\mathscr{L}}(\Phi^*|\Theta^\Phi) = \prod_{i=1}^{n} P(x_i) = \sum_{i=1}^{n} \log P(\bar{x}_i|\Theta^\Phi) \tag{5.23}$$

Here, n is the number of vertices of the mesh M^*, while $\bar{x}_i$ is a d-dimensional row vector representing the i-th feature vector of the mesh. Note that the eigenvector matrix associated to M^* undergoes the same alignment steps defined in section 5.4.1 and 5.4.1. On the other hand, the eigenvalue model log-likelihood is defined as

$$\ell_{\mathscr{L}}(\Lambda^*|\mu_i^\Theta, \sigma_i^\Theta) = \prod_{i=1}^{d} P(\lambda_i) = \sum_{i=1}^{d} \log P(\lambda_i) \tag{5.24}$$

where μ_i^Θ and σ_i^Θ are the mean and the variance parameters learned according to 8.22. The posterior probability is computed against each model of each shape contained in the dataset. As a decision rule, we classify a certain mesh as representing the shape whose model yields the highest probability. Even though we employ the proposed method as a classifier, the output of the approach is still a probability value which can be employed as a similarity score.

5.6 Experimental results

In this section we show the performance achieved by the proposed method while addressing a classification task. The evaluation is based on the comparison of the results obtained by our method with respect to the current state-of-the-art on a very popular shape retrieval benchmark, *i.e.* SHREC'14 benchmark. The implementation of our method has been done in MATLAB, and the source code is available on the author web page. The efficacy of the method is evaluated in terms of mean classification accuracy. Besides, we test the robustness and its sensitiveness to sub-sampling, showing the retrieval performance with different amount of data employed in the learning process and with different embedding dimensions. Finally, a short performance analysis (in terms of execution time) is proposed at the end of the section.

SHREC'14 Humans [144] is a dataset containing a large number of meshes which represent 55 different shapes. It consists of two different sub-sets. The first one (synthetic) contains 15 different human models each in 20 different poses, while the second one contains scans of 40 human subjects, each in 10 different poses. All shapes were downsampled to have about 6×10^3 triangles for efficiency purpose. Both datasets are extremely challenging, as they contain geometrically similar human shapes. We employed a 10-fold test over the datasets resulting respectively in 2 positives and 28 negatives per query for the synthetic dataset and 1 positive and 39 negatives for the scanned dataset. The process was repeated for each possible subset of the datasets and the whole test was averaged over 100 iterations. We compare our approach with the most accurate methods that participate

Method	Synthetic	Scanned
ISPM [113]	90.2	25.8
DBN [144]	84.2	30.4
R-BiHDM [200]	64.2	64.0
HAPT [70]	81.7	63.7
ShapeGoogle (VQ) [31]	81.3	51.4
Unsupervised DL [117]	84.2	52.3
Supervised DL [117]	95.4	79.1
NPSR	95.0	79.0

Table 5.1: Comparison of different retrieval methods in terms of average retrieval precision on the SHREC'14 Humans datasets.

to the SHREC'14 benchmark: Histograms of Area Projection Transform (HAPT) [70], Deep Belief Network (DBN) [144], Intrinsic Spatial Pyramid Matching (ISPM) [113], Reduced Bi-harmonic Distance Matrix (R-BiHDM) [200] and to a group of approaches based on the Bag-of-Features paradigm ([117], [31]). Table 6.3 shows the results yielded by those methods. The proposed method performs well and is competitive with the current state-of-the-art. We are able to achieve high retrieval accuracies in both the synthetic and scanned datasets, while the differences with respect to the current state-of-the-art are not statistically significant.

Finally, we present a robustness analysis of the proposed method. The random sub-sampling has been performed directly on the Laplacian matrices of the meshes which belong to the same training set. Indeed, performing the sub-sampling after the model construction through the elimination of some feature vectors (which means, taking out some rows from the model Θ^{Φ}) could lead to biased results since both the alignment steps employed in our learning phase would have benefited from the contribution of data that would no longer be available in the next steps. Figure 5.2 shows the average precision while both the embedding dimension and the sampling percentage vary. As you can see, for the lowest level of sub-sampling (respectively, 75% and 100% of data used in the learning process) we were not able to produce the results with respect to all the embedding dimensions due to the dimension of the model. Even so, it is easy to see that the performance achieved are consistent at every sub-sampling level. In particular, the results show that the defined model is robust with respect to the sub-sampling of points. This allows to greatly reduce the dimension of the model itself and consequently increasing the performance of the whole pipeline.

5.7 Conclusion

In this work, we proposed a supervised data-driven approach for the construction of a generative model based on the spectral decomposition of the Laplace-Beltrami operator. We defined a statistical framework that models a shape as two independent generative models for the eigenvector and the eigenvalue components of the spectral representation of the Laplacian, making no assumption about the distribution of the spectral embedding space of the eigenvector part and estimating it through kernel density estimation. Moreover, we modelled the eigenvalues as a log-normal distribution. We tested our method on one of the most recent and popular dataset for shape retrieval. Experimental results show that the proposed approach is competitive with the current state-of-the-art and it is able to achieve high retrieval accuracy. Finally, robustness analysis showed that the proposed method is resilient to sampling, allowing to remove data from the models in order to improve the computation efficiency.

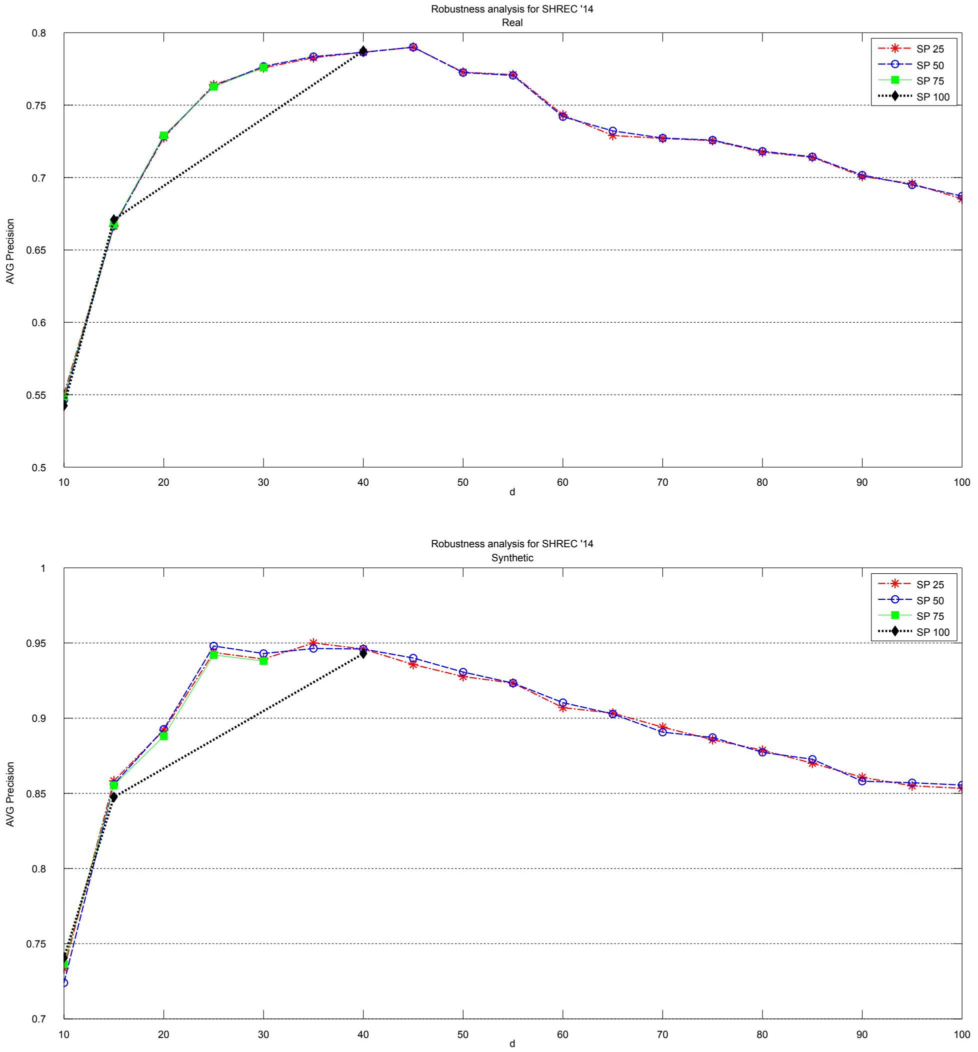


Figure 5.2: Robustness analysis of the proposed method in both Real (top) and Synthetic (bottom) datasets. The marks represent the average precision obtained with a certain embedding dimension. Different lines represent different sub-sampling percentage, *e.g.* the red line represents the average accuracy with respect to different embedding dimensions after removing 75% of the data on the model (*i.e.* keeping only 25% of the data).

6

Statistical Model of Riemannian Metric Variation for Deformable Shape Analysis

The analysis of deformable 3D shape is often cast in terms of the shape's intrinsic geometry due to its invariance to a wide range of non-rigid deformations. However, object's plasticity in non-rigid transformation often result in transformations that are not completely isometric in the surface's geometry and whose mode of deviation from isometry is an identifiable characteristic of the shape and its deformation modes. In this chapter, we propose a novel generative model of the variations of the intrinsic metric of deformable shapes, based on the spectral decomposition of the Laplace-Beltrami operator. To this end, we assume two independent models for the eigenvectors and the eigenvalues of the graph-Laplacian of a 3D mesh which are learned in a supervised way from a set of shapes belonging to the same class. We show how this model can be efficiently learned given a set of 3D meshes, and evaluate the performance of the resulting generative model in shape classification and retrieval tasks. Comparison with state-of-the-art solutions for these problems confirm the validity of the approach.

In Chapter 5 we proposed another approach to tackle the shape retrieval problem. Both methods devise generative models based on the spectrum of shapes. The central difference between the approaches is the assumption on which they are based. In the method proposed in this chapter, a common space is assumed to be known (or somehow computed) between the meshes. In this space, a novel intrinsic metric is defined and used to compute the invariant representation of a shape. On the other hand, in the method introduced in the previous chapter, this space is not assumed nor necessary in the definition of the generative model.

6.1 Introduction

The ability to retrieve similar 3D objects given a query object has become of great importance in several field, like medical research, automatic information retrieval systems and copyright protection. The main challenge in 3D object retrieval algorithms is to define an invariant representation of a shape capable of capturing geometrical and topological properties of a shape [20, 147, 175]. A large number of methods for 3D shape retrieval have been proposed [10, 98, 127, 140, 167], but most of them are only suitable for the representation of rigid 3D shapes. Non-rigid 3D shape descriptors are more challenging to define. Indeed, those representations still need to capture the most distinctive properties of a shape, but they should be also insensitive to many other transformations (*e.g.* inelastic transformation, acquisition resolution, noise, *etc.*).

Various methods have been proposed to address the non-rigid shape retrieval problem. These can be mainly categorized according to two main directions: topology-based approaches against surface-based methods and the employment of shape descriptors that can be local or global. The first solution usually capture the global topological structure of the shape [112, 126, 130], while the surface-based methods usually exploit spectral shape analysis in order to build a spectral descriptor [12, 155, 174] to be used for the task. Finally, one can employ global descriptors in order to characterize the whole shape [33, 174] or define a set of of local descriptors [64, 73] that characterize only parts of the object (*e.g.* segments or points).

6.1.1 Related works

The methods which define a shape descriptor (both local and global) to tackle the non-rigid 3D shape recognition are the ones we are most interested in. A popular approach to define shape descriptors is through spectral shape analysis. Research efforts have recently resulted in several spectral descriptors [12, 101, 149, 174] usually based on the spectral decomposition of the Laplace-Beltrami operator. In particular, the graph-Laplacian, the discrete counterpart of the Laplace-Beltrami operator, has been extensively used to provide spectral representations of structures [118]. Reuter *et al.* [148] suggest to use the Laplace-Beltrami spectra as fingerprints of surfaces, while Jain and Zhang [88] propose to use the eigenvalues of the geodesic distance matrix of a 3D object in order to build the associated shape descriptor. Huang *et al.* [84], on the other hand, build the signature directly over local features, selecting discriminative volumetric features over pre-aligned shapes.

The aggregation of local descriptors in order to build a global descriptor is a general thread in the literature. For this purpose, the *Bag-of-Features* (BoF) paradigm is quite popular and has been successfully applied to 3D shape description [31, 50, 109, 177]. Li and Hamza [114] used the BoF paradigm combining the exploitation of hierarchical structures of the shape, such as pyramid matching [74] and spatial relationship [31, 50, 109]. They proposed to adopt the eigenfunction associated with the second-smallest eigenvector of the Laplace-Beltrami operator in order to build a global surface coordinate system which is insensitive to shape deformation, showing that the introduction of global spatial context could improve the effectiveness of their descriptor in 3D shape recognition. Spatial pyramid [110, 114, 120], is the term used to identify this approach. Other approaches inspired by text-analysis have been proposed. For instance, in [22, 82] the authors adopt higher-order models defining relations between ‘geometric words’.

Within the bag of features model, features quantization is generally performed through unsupervised learning. Departing from this approach, Litman *et al.* [117] recently proposed a new supervised BoF framework mapping the discriminative training directly into the dictionary construction step.

6.1.2 Main contribution

In this chapter, we propose a new supervised technique to learn a statistical model build on the Riemannian metric variations on deformable shapes based on the spectral decomposition of the Laplace-Beltrami operator. To this end, we define a statistical framework that models a shape as two independent models for the eigenvectors and for the eigenvalues. The eigenvector and eigenvalue matrices of a set of discrete representations (*i.e.* meshes representing the shape in different poses) are assumed to be points of two separate smooth Riemannian manifolds. The invariant representation of a certain shape is computed as the centroid of such manifolds. The shape centroid acts as our invariant representation of the shape. One of the main contributions is indeed the definition of an efficient algorithm that compute the spectral decomposition of this centroid. In particular, we define it as a supervised data-driven optimization process. For this purpose, we define a computationally efficient intrinsic distance metric over the eigenvectors manifold. Finally, we define a statistical model over the variations of such metric, assuming that the squared geodesic distances follow a Γ-distribution. Moreover, in order to define a descriptor that is robust to small non-isometric perturbations, we assume that the eigenvalues are log-normally distributed for the same stability reasons presented by Aubry *et al.* [12].

6.2 Background

We model a shape as a surface which is invariant to rigid and non-rigid transformations, or, quoting D.G. Kendall [100], a shape is 'what is left when the differences which can be attributed to translations, rotations, and dilatations have been quotiented out'. A representation of such shape is modelled as a smooth compact 2-manifold $\mathscr{S}$ without boundary isometrically embedded in some Euclidean space $\mathbb{R}^3$ with geometry induced by the embedding. A 3D mesh is a discretization of a shape embedded in $\mathbb{R}^3$, and a discrete Laplacian is a discretization of the continuous Laplace-Beltrami operator on the mesh. There are various ways by which such discretization can be obtained from the mesh. Here we adopt the algorithm proposed by Belkin *et al.* [16] which offers point-wise convergence guarantees and was experimentally shown to be quite robust.

The spectral representation of the mesh can be obtained from the Laplacian through singular value decomposition. Given a Laplacian $\mathscr{L}$, its decomposition is $\mathscr{L} = \Phi\Lambda\Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the matrix whose diagonal contains the ordered eigenvalues, while $\Phi = (\phi_1|\phi_2|...|\phi_{|V|})$ is the matrix whose columns are the ordered eigenvectors. This decomposition is unique up to a permutation of the vertices of the mesh, a change of sign of the eigenvectors, or a change of basis over the eigenspaces associated with a single eigenvalue, *i.e.* , the following properties hold:

$$\mathscr{L} \simeq PLP^T = P\Phi\Lambda(P\Phi)^T \tag{6.1}$$

$$\mathscr{L} = \Phi\Lambda\Phi^T = \Phi S\Lambda S\Phi^T \tag{6.2}$$

where ≃ indicates isomorphism of the underlying surfaces, P is a permutation matrix and S is a diagonal matrix with diagonal entries equal to ±1.

6.3 Statistical model definition

One of the main contribution of this part of the thesis is the definition of an efficient algorithm to compute the spectral decomposition of the manifold centroid from a set of points lying on it. In order to address the classification task, we study the variations of the metrics in the manifold, casting them into a statistical framework. So let $M^{\mathscr{S}} = \{M_1^{\mathscr{S}}, M_2^{\mathscr{S}}, \dots M_N^{\mathscr{S}}\}$ be a set of meshes which represent a discretization of the same underlying shape $\mathscr{S}$ in different poses. In the proposed method, we assume that each component of the spectral decomposition of the Laplacian of each mesh $M_i^{\mathscr{S}}$, $i = 1, \dots, N$, lies on a Riemannian manifold. Let $\Delta_{M_i^S}$ be the Laplace operator introduced in section 7.2 applied to the mesh $M_i^{\mathscr{S}}$. The spectral decomposition of $\Delta_{M_i^S}$ yields the eigenvectors $\Phi_{M_i^{\mathscr{S}}}$ and the eigenvalues $\Lambda_{M_i^{\mathscr{S}}}$ (Φ_i and Λ_i in short) such that

$$\Delta_{M_i^{\mathscr{S}}} = \Phi_{M_i^{\mathscr{S}}} \Lambda_{M_i^{\mathscr{S}}} \Phi_{M_i^{\mathscr{S}}}^T \tag{6.3}$$

These components are used as a basis for our invariant representation of the shape $\mathscr{S}$. Let $M_0^{\mathscr{S}} = \{\Phi_0^{\mathscr{S}}, \Lambda_0^{\mathscr{S}}\}$ be such invariant representation. We treat the spectral components independently, building a model for the eigenvectors and a model for the eigenvalues. Since the Laplacian matrix is a positive-semidefinite matrix, the eigendecomposition of such matrix yields an orthogonal basis of eigenvectors, $\phi \in \mathscr{O}(n)$.

The set of orthogonal matrix is composed by two disjoint subsets, one with matrices with determinant +1 (*i.e.* the special orthogonal group $\mathscr{S}\mathscr{O}(n)$) and one with matrices with determinant −1. In the computation of the geodesic distance, the two orthogonal matrices must belong to the same subset. However, the arbitrarity of the sign of the eigenvectors guarantee that we can always find representatives in the same connected component. In particular, choosing the sign-flip S that minimizes the Frobenius distance between the eigenvector matrices, we guarantee that we pick representatives that belong to the same connected component and which are closest to one another in terms of geodesic distance. This requires a pre-processing of the eigendecomposition of the Laplacians of the whole class in such a way that the direction of the eigenvectors are concordant. To this end, we take a reference mesh and flip the sign of the eigenvectors of the other meshes in such a way as to make the dot product of corresponding eigenvectors form the different meshes positive. Under this assumption, we define $\Phi_0^{\mathscr{S}}$ as the solution to the following minimization problem:

$$\underset{\Phi_0^S \in \mathscr{O}(n)}{\operatorname{argmin}} \quad \sum_i^N d^2(\Phi_i^{\mathscr{S}}, \Phi_0^{\mathscr{S}}) \tag{6.4}$$

where d^2 is the geodesic distance between the eigenvector matrices, while N is the number of meshes representing the same shape $\mathscr{S}$. Given two orthogonal matrices $R_1 \in \mathscr{O}(n)$ and $R_2 \in \mathscr{O}(n)$ belonging to the same connected component of $\mathscr{O}(n)$, we can define the geodesic distance as

$$d_g(R_1, R_2) = \| log(R_1^T R_2) \|_F \tag{6.5}$$

where $\| \cdot \|_F$ is the Frobenius norm. For orthogonal matrices belonging to the same connected component of $\mathscr{O}(n)$, we have

$$d_g^2(R_1, R_2) = \sum_i^n \Theta_i^2 \tag{6.6}$$

where Θ_i are the angles of the rotation $R_1^T R_2 \in \mathscr{SO}(n)$.

Using Taylor's expansion of $\cos\Theta$, we obtain $\Theta^2 = 2 - 2\cos\Theta + O(\Theta^4)$. Recalling that for $A \in \mathscr{SO}(n)$ $\operatorname{Tr}(A) = \operatorname{Tr}(\frac{1}{2}(A + A^T)) = \sum_i^n \cos(\Theta_i)$, we can approximate the geodesic distance as

$$d_g^2(R_1, R_2) = 2n - \operatorname{Tr}\left(R_1^T R_2\right) + O(\Theta_i^4) \tag{6.7}$$

resulting in the following approximation of 6.4:

$$\underset{\phi_0 \in \mathscr{O}(n)}{\operatorname{argmin}} \quad 2Nn - 2\operatorname{Tr}\left(\left(\sum_i^N \phi_i^T\right)\phi_0\right) \tag{6.8}$$

which is equivalent to

$$\underset{\phi_0 \in \mathscr{O}(n)}{\operatorname{argmax}} \quad \operatorname{Tr}\left(\left(\sum_i^N \phi_i^T\right)\phi_0\right) \tag{6.9}$$

For this work, we assume that the geodesic distances between eigenvector matrices of the Laplacian of the meshes that represent the same shape follow a Γ-distribution with shape parameter k and scale parameter θ.

$$\Gamma(x; k, \theta) = \frac{1}{\theta^k \Gamma(k)} x^{k-1} e^{-\frac{x}{\theta}} \tag{6.10}$$

where N is the number of meshes involved in the model construction, x are the geodesic distances between the eigenvectors of the meshes and the eigenvectors centroid, and k and θ are estimated using the maximum likelihood estimation. Since there is no closed form solution for the parameter k, we approximate it using [129] and computing the parameter s as

$$s = \log\left(\frac{1}{N}\sum_{i=1}^{N}\sqrt{x_i}\right) - \frac{1}{N}\sum_{i=1}^{N}\log(x_i) \tag{6.11}$$

where x_i in our instance is the geodesic distance between the eigenvectors of the Laplacian of the mesh i and the eigenvectors centroid of the class. Using s, we can approximate k as

$$k \approx \frac{3 - s + \sqrt{(s-3)^2 + 24s}}{12s} \tag{6.12}$$

Once the shape parameter is computed, we use it to compute the second parameter of the distribution, θ.

$$\theta = \frac{1}{kN}\sum_{i=1}^{N}\sqrt{x_i} \tag{6.13}$$

Finally, thanks to 6.7, we are able to define the geodesic distance between two eigenvector matrices, which is computed as

$$d^2(\phi_i, \phi_0) = 2n - 2\mathrm{Tr}(\phi_i^T \phi_0) \tag{6.14}$$

The second component of the spectral decomposition of our manifold centroid is simply the exponentiation of the average of the logarithm of each j-th eigenvalue

$$\Lambda_{j0} = e^{\frac{1}{N}\sum_{i}^{N}\log\lambda_{ji}} \tag{6.15}$$

We assume that the eigenvalues follow a log-normal distribution. In particular, we assume that each i-th eigenvalue follows the same distribution, *i.e.* shares the same distribution parameters.

$$\log\mathcal{N}(x;\mu,\sigma) = \frac{1}{x\sqrt{2\pi}\sigma}\, e^{-\frac{(\log x - \mu)^2}{2\sigma^2}} \tag{6.16}$$

where x is an eigenvalue, μ is the mean of the distribution and σ is the standard deviation. Let λ_{i0} be the i-th eigenvalue of the spectral decomposition of the manifold centroid. Then, the distribution mean is defined as

$$\mu_i = \log\lambda_{i0} \tag{6.17}$$

Let λ_{ij} be the i-th eigenvalue of the mesh j. We define the standard deviation σ as

$$\sigma_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\log\lambda_{ij} - \mu_i\right)^2} \tag{6.18}$$

Finally, we can define the density of the i-th eigenvalue of the j-th mesh as

$$\lambda_{d_{ij}} = \frac{1}{\lambda_{ij}\sqrt{2\pi}\sigma_i} e^{-\frac{\left(\log(\lambda_{ij})-\mu_i\right)^2}{2\sigma_i}} \tag{6.19}$$

Once both densities are computed, it is possible to compute the density of a particular mesh as

$$p(j) = \left(\prod_i \lambda_{d_{ij}}\right) \Gamma(k,\theta)(g_{d_j}) \tag{6.20}$$

where g_{d_j} is the geodesic distance between the eigenvector matrix of the Laplacian of a mesh and the eigenvector centroid of the shape we are considering.

6.4 Embedding, isometries and lack of correspondences

The use of the whole Laplacian eigendecomposition of a mesh is not necessary, nor encouraged, since we are usually more interested in the smallest eigenvalues and the associated eigenvectors. In fact, most of the structural information is stored in those eigenvalues and eigenvectors. And that is why we embed the Laplacian matrix into a lower dimensional space. Let p be the embedding dimension. In the formulation of the geodesic distances 6.7 and 6.14, the embedding dimension must be taken into account and p should be used in place of n.

The use of spectral shape descriptor entails several advantages like the simple representation, scale invariance and a very good performance for shape retrieval of non-rigid shapes. But there are also several issues that raise from its employment. For example, the isometric embedding of a Riemannian manifold on a lower dimensional space is unique up to isometries. For this reason, we introduce a new rigid transformation $\mathscr{R}_i$ which aligns the eigenvectors of the Laplacian of a mesh i with the eigenvector centroid of a certain shape. Hence, we can rewrite 6.9 introducing the rotation matrix $\mathscr{R}_i$, obtaining

$$\underset{\phi_0, \mathscr{R}_i \in \mathscr{SO}(p)}{\operatorname{argmax}} \quad \operatorname{Tr}\left(\sum_i^N \mathscr{R}_i \phi_i^T \phi_0\right) \tag{6.21}$$

The construction of our invariant representation of a shape is based on the assumption that the correspondences between the vertices of two meshes is known. But usually this kind of information is not available. Thus, the second issue we dealt with is related to the lack of correspondences between two meshes. There are several datasets available that provide the ground-truth for the correspondences (at least of the meshes that represent the same shape, i.e. meshes that belong to the same class), like TOSCA dataset [9] or the kids dataset [153]. But most of the datasets used in shape retrieval task do not provide such information. In order to make our method as flexible as possible, we cast the problem of finding a fine matching between two meshes to a maximum bipartite matching, solving the lack of correspondences as an assignment problem. So let $\phi_i \in R^{n \times p}$ and $\phi_j \in R^{m \times p}$ be the eigenvectors of the embedded Laplacian of two meshes. Let $\mathscr{R} \in \mathscr{SO}(p)$ the rotation matrix which aligns the two meshes. We define a weight matrix W as

$$W = \phi_i \mathscr{R} \phi_j^T \tag{6.22}$$

This matrix is then used as input for the Hungarian algorithm [26]. The Hungarian algorithm yields a permutation matrix P that minimize the assignment cost. Since we want to maximize that cost, the input of the algorithm is the negation of the weight matrix W. Hence, we can define the introduced problem as the maximization problem

$$\underset{\mathscr{R},P}{\operatorname{argmax}} \quad \operatorname{Tr}\left(\phi_j \mathscr{R} \phi_i^T P\right) \tag{6.23}$$

while the geodesic distance between two eigenvector matrices is defined as

$$d^2(\phi_i, \phi_j) = 2n - 2\operatorname{Tr}(\phi_j \mathscr{R} \phi_i^T P) \tag{6.24}$$

6.5 Learning the invariant representation

The proposed method consists of two independent phase. In this section we present the first one, whose main purpose is to learn the invariant representation of each shape of a given dataset. In other words, given a shape S, characterized in the dataset by a set of meshes $\{M_1, M_2, \ldots, M_N\}$, we want to solve the problem 6.9 in order to compute the eigendecomposition of the manifold centroid $\{\Phi_0, \Lambda_0\}$ of the shape. For the eigenvectors component, we employ an iterative optimization process. An iterative approach is mandatory since we have to find the optimum eigenvector centroid Φ_0 while simultaneously optimizing each rotation matrix $\mathscr{R}_i$, one for each of the N meshes of the training set. The adoption of the iterative optimization process allows to split the problem defined in 6.21 in two parts, one for the eigenvector centroid, defined as

$$\Phi_0 = \underset{\Phi_0 \in \mathscr{O}(n)}{\operatorname{argmax}} \quad \operatorname{Tr}\left(\left(\sum_i^N \mathscr{R}_i \Phi_i^T\right) \Phi_0\right) \tag{6.25}$$

and one for the rotation matrix which align the i-th eigenvector matrix with Φ_0.

$$\mathscr{R}_i = \underset{\mathscr{R}_i \in \mathscr{SO}(n)}{\operatorname{argmax}} \quad \operatorname{Tr}\left(\left(\sum_i^N \Phi_i^T \Phi_0\right) \mathscr{R}_i\right) \tag{6.26}$$

So, we define our iterative optimization process as

1. Computation of the mesh Laplacian of each mesh of the training set, $\{\Delta_1, \Delta_2, \ldots, \Delta_N\}$, using Belkin *et al.*formulation [16].

2. Eigendecomposition of the Laplacians, which yields the set $\{(\Phi_1, \Lambda_1), (\Phi_2, \Lambda_2), \ldots, (\Phi_N, \Lambda_N)\}$. The eigenvalues are used in the second part of the learning phase in order to compute the eigenvalues centroid Λ_0.

3. After the application of the rotation matrix $\mathscr{R}_i$ to each Φ_i, we sum the eigenvector matrices together $\Sigma_\Phi = \sum_i^N \mathscr{R}_i \Phi_i$. To estimate the eigenvector centroid Φ_0, we compute the singular value decomposition $svd(\Sigma_\Phi) = ULV^T$. We compute the eigenvector centroid as $\Phi_0 = UV^T$.

4. To estimate the rotation matrix $\mathscr{R}_i$, we compute the sum $\Sigma_{\mathscr{R}} = \sum_i^N \Phi_i^T \Phi_0$. We use again the *svd*, obtaining $svd(\Sigma_{\mathscr{R}}) = ULV^T$. Finally, we compute the rotation matrix as $\mathscr{R}_i = VU^T$.

The steps $3-4$ are repeated till convergence, which is usually reached after a low number of iterations (about 5 iterations).

Once the eigenvector centroid is obtained, we can proceed with the computation of the geodesic distances between each mesh of the training set and the centroid itself, using equation 6.14. This results in the vector $d_g = \{d_1, d_2, \ldots, d_N\}$. As already introduced in section 6.3, we assume that the geodesic distances d_g follow a Γ-distribution. We use 6.12 and 6.13 to compute the distributions parameters k and Θ.

The eigenvalues component of the invariant representation is computed in a more straightforward manner. Λ_0 of a certain class is computed through 6.15. The log-normal distribution parameters μ and σ for the eigenvalues component of the centroid are computed using 6.17 and 6.18.

To summarize, the learning phase of the proposed method applied to a training set with meshes that represent a shape $\mathscr{S}$ produces the manifold centroid $\{\Phi_0, \Lambda_0\}$ and the distribution parameters $\left(\mu_i^{\mathscr{S}}, \sigma_i^{\mathscr{S}}\right)$, $i = 1, \ldots, p$, and $\left(k^{\mathscr{S}}, \Theta^{\mathscr{S}}\right)$. The described learning process is repeated for each class of the dataset. Figure 6.2 shows the distributions yielded by the learning phase introduced in this paragraph.

6.6 Models inference

In section 6.4, we introduced one of the problems which raises from the computation of a distance metric between two meshes, namely the lack of correspondences between these meshes. In the learning phase of the proposed method this was a minor issue, since the meshes that belong to the same class can be assumed to be quite isometric between them self. It is not the case for the meshes that represent different shapes. In this case, we cast the lack of correspondences problem into an assignment problem (see 6.23). In order to solve this problem, we make use of another iterative process where we are looking to optimize the parameters $\mathscr{R}_*$ (a rotation matrix) and P_* (a permutation matrix). Let Φ_* be the eigenvector matrix computed on the Laplacian applied to a mesh M_*. No prior knowledge about the shape represented by M_* is assumed. Let Φ_0 be the eigenvector centroid of a shape $\mathscr{S}$. After a initialization step where we assign the identity matrix to the rotation matrix $\mathscr{R}_*$, we define the optimization process as

1. Computation of the weight matrix W to be used in our assignment problem, $W = \Phi_* \mathscr{R}_* \Phi_0^T$

2. The weight matrix $-W$ is then used as the input of the Hungarian algorithm, whose output is a cost matrix C and a permutation matrix P_*
3. The rotation matrix $\mathscr{R}_*$ is the orthogonal matrix which most closely maps the permuted eigenvector matrix Φ_* to the eigenvector centroid Φ_0, which is the definition of the orthogonal Procrustes problem [162]. So, to estimate the rotation matrix $\mathscr{R}_*$, we use the *svd*, $svd(\Phi_*^T P_* \Phi_0) = ULV^T$. We compute the rotation matrix $\mathscr{R}$ as $\mathscr{R} = VU^T$.

The process converges after a low number of iterations (≈ 3). Once the parameters $\mathscr{R}_*$ and P_* are computed, we use 6.24 to compute the geodesic distance $d_*^{\mathscr{S}}$ between the eigenvector matrix associated to M_* and the eigenvector centroid of the shape $\mathscr{S}$. Let $k^{\mathscr{S}}$ and $\Theta^{\mathscr{S}}$ be the Γ-distribution parameters computed as explained in section 6.5. The probability density of M_* with respect to the eigenvector model of the shape $\mathscr{S}$ is

$$\phi_{d_{(*|\mathscr{S})}} = \Gamma\left(d_*^{\mathscr{S}}; k^{\mathscr{S}}, \Theta^{\mathscr{S}}\right) \tag{6.27}$$

The eigenvalues of the mesh Laplacian are used straightforward after their computation. As introduced in section 6.3, we assume that all the i-th eigenvalues follow a log-normal distribution with parameters μ_i and σ_i, $i = 1, 2, \ldots, p$, with p the number of eigenvalues used (embedding dimension). So, let Λ_* be the eigenvalues of the mesh M_*. Let $\Lambda_0^{\mathscr{S}}$ be the eigenvalue centroid of the shape $\mathscr{S}$. The probability density of each eigenvalue of M_* with respect to the eigenvalues model of the shape $\mathscr{S}$ is $\lambda_{d_{(*|\mathscr{S})}} = \left\{\lambda_{d_{(*|\mathscr{S})_1}}, \lambda_{d_{(*|\mathscr{S})_2}}, \ldots, \lambda_{d_{(*|\mathscr{S})_n}}\right\}$, where $\lambda_{d_{(*|\mathscr{S})_i}}$ is defined as

$$\lambda_{d_{(*|\mathscr{S})_i}} = \log \mathscr{N}\left(\lambda_{*i}; \mu_i^{\mathscr{S}}, \sigma_i^{\mathscr{S}}\right) \tag{6.28}$$

Finally, we combine the afore-computed density using 6.20. For numerical stability, in place of the product of the densities, we compute the sum of the logarithms of the densities. Hence, the probability density of a mesh M_* with respect to the models computed on a shape $\mathscr{S}$ is given by

$$\log p\,(*|\mathscr{S}) = \log \phi_{d_{(*|\mathscr{S})}} + \sum_i^n \log\left(\lambda_{d_{(*|\mathscr{S})_i}}\right) \tag{6.29}$$

Once the combined density is computed, a decision rule is applied in order to assign the mesh to the most probable class.

It is interesting to note that the lack of correspondences does not affect the construction and use of the eigenvalues model. This is due to the fact that we treat each eigenvalue as an independent variable, while each i-th eigenvalue of a mesh belonging to the same shape is an independent observation. The only implicit assumption regards the dimension of the embedding p, which has to be consistent between all the meshes involved both in the learning and in the inference phase. Since the Laplacian eigendecomposition of a mesh is part of the pipeline, the former assumption holds as long as the chosen embedding dimension is smaller than the minimum number of vertex of each mesh involved in the process.

Transformation	VQ [31]	Sup. DL [117]	RMVM
Isometry	98.8	**99.4**	**99.4**
Topology	100	100	100
Isometry+Topology	93.3	95.6	**99.5**
Partiality	94.7	**95.1**	90.0
Triangulation	95.4	95.5	**96.5**

Table 6.1: Comparison of different retrieval methods, in terms of average precision on the SHREC'10 datasets, broken down according to different transformations.

6.7 Experimental results

In this section we evaluate the performance achieved by our method, comparing our results with the current state-of-the-art. Our approach was implemented in MATLAB and the source code is available at the web page of the author [1]. Retrieval performance was evaluated using mean average precision. The proposed method has been applied to several popular datasets in the shape retrieval field.

SHREC'10 ShapeGoogle [31] dataset is the aggregation of three public domain collections: TOSCA shapes [9], Robert Sumner's collection of shapes [173] and Princeton shape repository [8]. It consists of 1184 synthetic meshes, out of which 715 meshes were obtained from 13 shape classes with simulated transformation (55 per shape) used as queries, while the 456 unrelated distractor shapes, which are usually treated as negatives, were not used. The transformations applied to the meshes show different levels of strength. An example of the shapes included in this dataset is shown in figure 6.3 in the first row. The results, which are shown in table 6.1, were obtained using a 10-fold test over 100 iterations.

In particular, we subdivided the meshes belonging to the same class in 10 randomly selected groups, using each one as a test set, while the remaining meshes were used as the training set in order to learn the invariant representation of a particular shape and the associated distribution parameters. This dataset contains a different number of representatives (*i.e.* meshes) for each class, hence the number of positives and negatives differs from class to class. For the smallest subset of meshes (representing the same shape), we had 3 positives and 63 negatives, while for the largest we tested our method against 13 positives and 54 negatives. Like in Litman *et al.*work [117], we removed the 'don't-care' ground-truth labels used in the original benchmark (e.g., male and female shapes were considered the same class). Furthermore, in order to make the dataset more challenging, we re-scaled all the meshes to have the same size. We compared the performance of our method (which takes the name of RMVM, *i.e.* Riemannian Metric Variation Model) with respect to several methods that employ Bag-of-Features descriptors. As the table shows,

[1] http://www.dsi.unive.it/~gasparetto/publications.htm

Transformation	1	≤ 2	≤ 3	≤ 4
Partiality	100	100	97.5	76.7

Table 6.2: Average precision on the SHREC'10 datasets, broken down according to the strength of the transformation applied.

Method	Synthetic	Scanned
ISPM [113]	90.2	25.8
DBN [144]	84.2	30.4
R-BiHDM [200]	64.2	64.0
HAPT [70]	81.7	63.7
ShapeGoogle (VQ) [31]	81.3	51.4
Unsupervised DL [117]	84.2	52.3
Supervised DL [117]	95.4	79.1
RMVM	**96.3**	**79.5**

Table 6.3: Comparison of different retrieval methods in terms of mean average precision on the SHREC'14 Humans datasets.

the proposed method obtains a slight performance improvement (or be at least on par) in 4 out of 5 transformations with respect to the current state-of-the-art approaches. We achieved lower classification accuracy with meshes which represent only partially the underlying shape. This is due to the fact that our approach exploits the prior-knowledge about the correspondences between vertices of different meshes. The removal of some parts of the mesh makes our assumption weaker, and this leads to a lower accuracy in the retrieval task. This issue is clear once we analyse the accuracy variation broken down to the strength of the transformation. Table 6.2 shows the results obtained for each level of strength of the transformation, while figure 6.4 shows a partiality transformation applied to a mesh.

SHREC'14 Humans [144] consists of two different sub-sets. The first part (synthetic) contained 15 different human models created using DAZ Studio, each in 20 different poses (total of 300 models, figure 6.3, middle row). The second one (scanned) contained scans of 40 human subjects, each in 10 different poses (400 shapes in total, figure 6.3, last row). All shapes were down-sampled to have about 6×10^3 triangles. Both datasets are extremely challenging, as they contain geometrically similar human shapes. In particular, the scanned dataset contains human shapes very difficult to distinguish even for a human observer (see figure 6.3, third row). We employed again a 10-fold test over the datasets. After a random permutation of the datasets, we subdivided the meshes into a training set and a test set. For the training set we used 90% of meshes for each class in order to build our shape invariant representation. For the synthetic dataset test set, we employed 2 positives and 28 negatives per query, while for the scanned dataset test set, the number of positives was 1 and the number of negatives was 39. The process was repeated for each possible subset of the datasets and the whole test was run 100 times. In addition to the same Bag-of-Features approaches seen before, we compared our model with the most accurate methods that participate to the SHREC'14 benchmark. In particular, we compared our approach to works based on Histograms of Area Projection Transform (HAPT) [70], Deep Belief Network (DBN) [144], Intrinsic Spatial Pyramid Matching (ISPM) [113], and Reduced Bi-harmonic Distance Matrix (R-BiHDM) [200]. Table 6.3 shows the results yielded by those methods. The proposed method performs slightly better with respect to the current state-of-the-art, achieving high retrieval accuracies in both the synthetic and scanned datasets.

Unfortunately, we are not able to conduct exhaustive experiments with regards to the performance achieved by the compared methods, since the performance analysis of the different approaches is not available. Just to have an insight about the performance achieved by the proposed method, we compare with Litman's approach [117], where the training time was reported to be approximately of 4 hours with a 3.2 Ghz CPU. On our 2.4 Ghz machine, we are able to learn the model of a whole class in about $5s$ (about $80s$ for the whole synthetic dataset). The prediction time is computed in $9s$, which means that the computation of the density of a mesh with respect to a certain class requires about $0.6s$.

6.8 Conclusion

In this work, we proposed a supervised data-driven method for the definition of an invariant representation of an arbitrary shape. Further, we defined a statistical framework based on such representation that can be used to address shape classification tasks. Moreover, we defined an efficient intrinsic metric to approximate the geodesic distance between points over a manifold. One of the limitations of the proposed method involves the knowledge of the correspondence ground-truth between meshes. We tested our method on several standard shape retrieval datasets. Experimental results show that the proposed approach is competitive with the current state-of-the-art for non-rigid 3D shape retrieval.

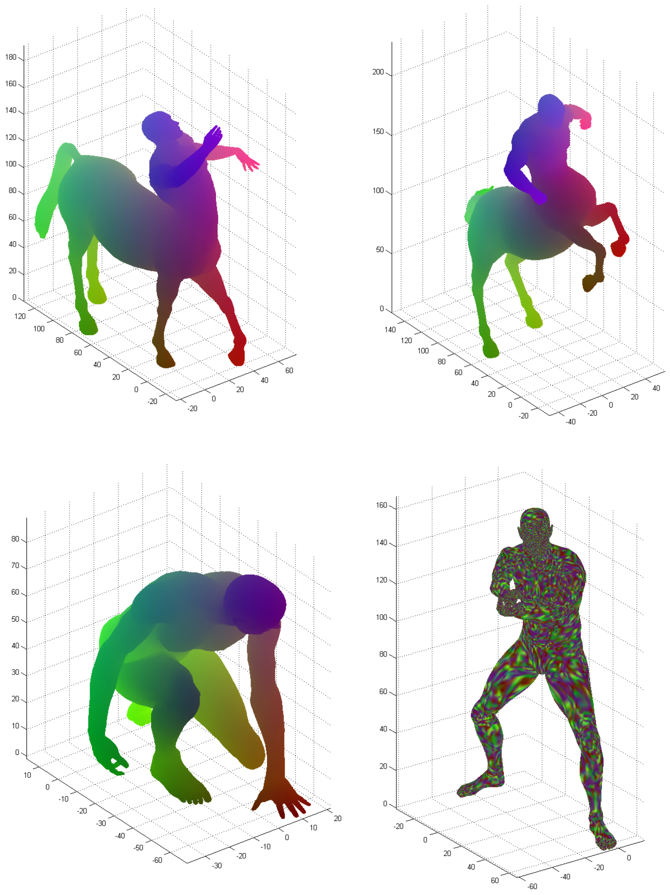


Figure 6.1: Example of known vertex correspondences between meshes which represent the same shape. These meshes, the Centaur (first row) and a human (second row), belong respectively to the SHREC'10 and SHREC'14 dataset. For the former dataset the correspondence ground-truth is known. In particular, we coloured the vertex accordingly to the correspondence map between the meshes. Thus, the same part (*e.g.* the head) takes the same color in both the representations if our map is consistent. The second row shows an example of two meshes which are not in correspondence.

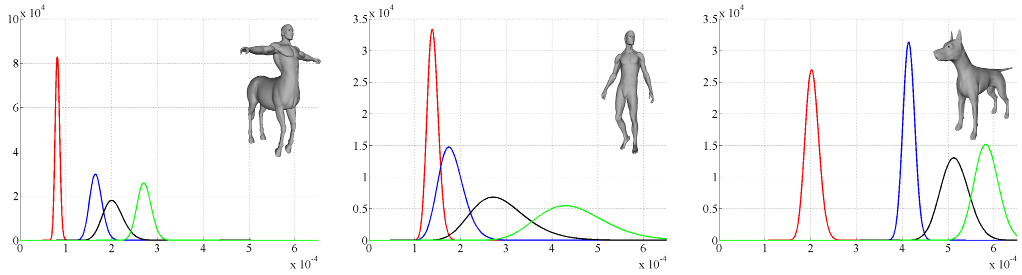


Figure 6.2: Distributions computed on the invariant representation of 3 shapes of the SHREC'10 dataset. The graphs in the second column of the image show the first four distributions computed on the eigenvalues centroid of the shape represented in the first row (see 6.19). In particular, the distributions refer to the first (red), second (blue), third (black) and fourth (green) eigenvalues of the corresponding centroids.

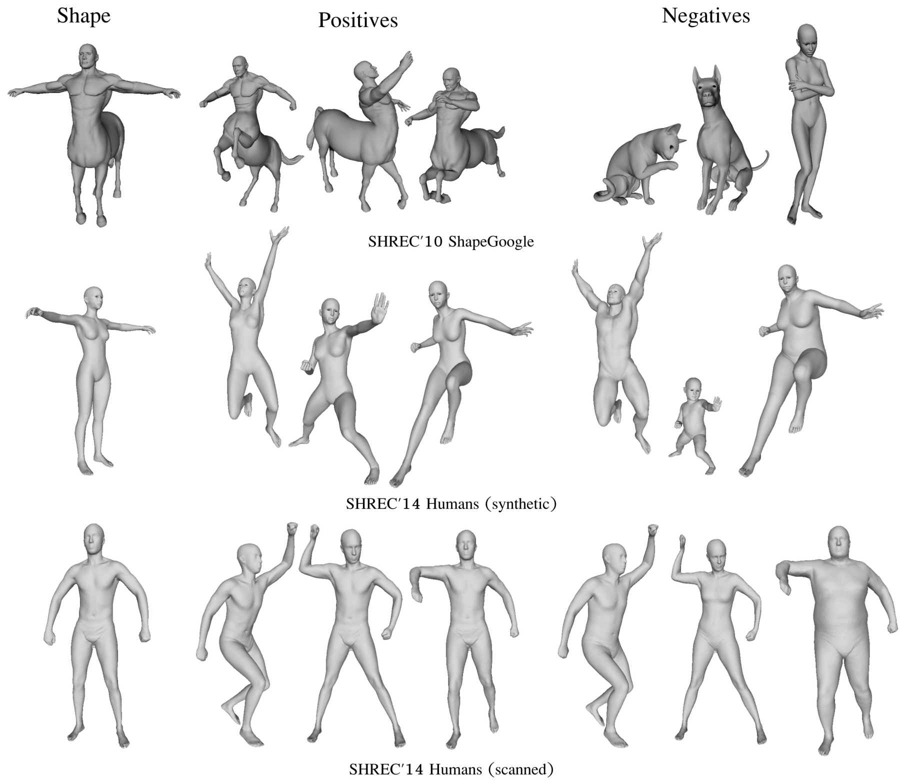


Figure 6.3: Example of the meshes from three datasets used in our experiments, from the easiest to the hardest (top to bottom). SHREC'10 ShapeGoogle dataset contains shapes of different bi and quadrupeds, SHREC'14 Humans (synthetic) dataset contains a collection of human models created through 3D modelling/animation software. The last row shows the "scanned" dataset which contains human models built from the point-clouds contained within the CAESAR using the SCAPE method ([11]). In the first column, a template of the shape is shown. The central part shows meshes representing the same shape but in different poses, while the rightmost part shows some negatives (*i.e.* , meshes representing a different shape).

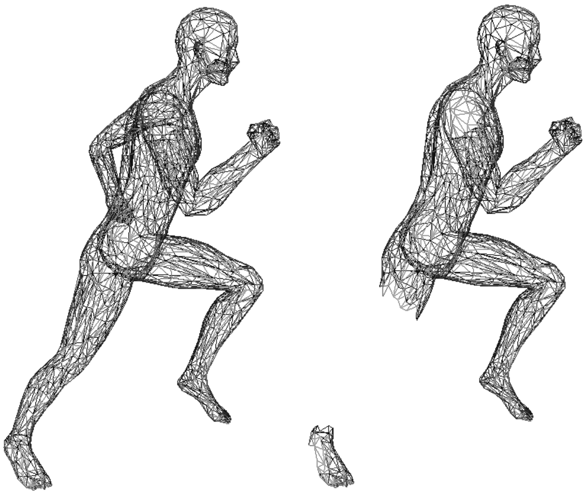


Figure 6.4: Example of partiality transformation on the SHREC'10 dataset. On the left there is the full mesh, on the right the same mesh after the application of a partiality transformation of strength 3.

906. Statistical Model of Riemannian Metric Variation for Deformable Shape Analysis

7

Non-Rigid Dense Bijective Maps

In the previous chapters of this part of the thesis we dealt with the shape retrieval problem devising two generative models both based on spectral theory. While the first attempt avoids the problem of defining a common space between meshes involved in the process, in the second one the existence of such space is assumed (or approximated). Indeed, the proposed approximation is a sub-optimal solution to a problem that is ubiquitous in both the shape retrieval and shape matching research field.

In this chapter we present a novel approach to the computation of dense correspondence maps between shapes in a non-rigid setting. The problem is defined in terms of functional correspondences. We deal with the non-injectivity of the solution of the functional map framework due to the under-determinedness of the original problem. Key to the approach presented in this chapter is the injectivity constraint plugged directly into the problem to optimize, achieved casting it as an assignment problem. This leads to an iterative process which yields a high quality bijective map between the shapes. In the experimental section we present both quantitative and qualitative results, showing that the proposed approach is competitive with the current state-of-the-art on quasi-isometric shape matching benchmarks.

7.1 Introduction

The ubiquitous shape matching problem plays a central role in computer graphics and geometry processing fields. Solutions to this problem have multiple applications, ranging from texture mapping to animation, to name just a few. For instance, in the shape analysis field, these approaches are employed as a pre-processing step in many shape retrieval methods [69, 153]; although it must be said that some try to avoid it completely building data-driven generative models [68].

Several successful approaches have been proposed in the rigid setting, while in the non-rigid one the problem remains difficult, even if we limit the space of deformations to approximate isometries. This is due to the fact that the rigid matching allows to represent the deformations compactly (*i.e.* a rotation and a translation), while in the non-rigid setting the problem is cast in terms of correspondences (of points or regions), rendering the space of the solution intreatable. Nevertheless, several popular approaches have been proposed in the previous ten years. These approaches can be divided into sparse [32,102,142,151] and dense [103, 141]. Sparse approaches strive to extract a few good point correspondences between the shapes under exam and, in general, trade completeness for precision, while dense approaches strive to extract the full bijection between isometric shapes.

A recent groundbreaking approach to the estimation of dense correspondences is represented by the functional maps framework [141], in which shape correspondences are modeled as a linear operator between spaces of functions over the shapes. Such operators can be efficiently represented in the Laplace-Beltrami eigenbasis [105, 141, 146]. Interestingly, the approach has been shown to be able to handle also partial mapping. In particular, Huang and Guibas [83] show that this appraoch can be used to analyze large collections of shapes in which some shapes may be only partially similar, while Rodolà *et al.* [152] generalize the framework to cope with partiality in the shapes. For an analysis and an overview about the current state-of-the-art we refer to [23,27,48].

One problem with the functional map framework is that the correspondences are only intrinsically captured by the map between the functional spaces and the extracted transformation between functions is not guaranteed to correspond to a bijective point-wise mapping between points in the two shapes. A symptom for this problem is given by the fact that impulse functions on one surface are mapped onto diffuse functions over the other shape, resulting in a fuzzy assignment. Further, even selecting the maximum of the mapped function over the target shape as the corresponding point, this results in a non-injective set of correspondences, with several correspondences clustered in relatively few points on the target shape (see Figure 7.1).

In this chapter we address the problem of extracting dense bijective maps between shapes. The problem is cast in terms of functional correspondences, which are constrained in the way they map impulse functions. The bijectivity of the map is then achieved casting the problem onto an equivalent bipartite matching problem.

7.2 Background

In the functional map framework shapes are modelled as compact connected 2-manifolds. Given a manifold $\mathcal{M}$, let $f, g : \mathcal{M} \to \mathbb{R}$, real scalar fields on the manifold. We define the standard inner product $\langle f, g \rangle_{\mathcal{M}} = \int_{\mathcal{M}} f(x)g(x)dx$, where integration is done using the manifold's Riemannian metric. Endowed with this scalar product and the resulting norm, we can define the space of square-integrable functions over $\mathcal{M}$, denoted as

$$L^2(\mathcal{M}) = \{f : \mathcal{M} \to \mathbb{R} \mid \langle f, f \rangle_{\mathcal{M}} < \infty\}. \tag{7.1}$$

This space features the positive semi-definite Laplace-Beltrami operator $\Delta_{\mathcal{M}} : L^2(\mathcal{M}) \to L^2(\mathcal{M})$, generalizing the notion of Laplacian to a non-Euclidean domain. The eigenfunctions of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}\phi_i(x) = \lambda_i\phi_i(x)$ form an orthonormal basis on $L^2(\mathcal{M})$, *i.e.* , $\langle \phi_i, \phi_j \rangle_{\mathcal{M}} = \delta_{ij}$, generalizing the classical Fourier analysis. In fact, a function $f \in L^2(\mathcal{M})$ can be expanded into the *Fourier series* as

$$f(x) = \sum_{i \geq 1} \langle f, \phi_i \rangle_{\mathcal{M}} \phi_i(x). \tag{7.2}$$

Functional correspondence Given two manifolds $\mathcal{N}$ and $\mathcal{M}$, Ovsjanikov *et al.* [141] proposed modelling the correspondences between the spaces of square-integrable functions $L^2(\mathcal{N})$ and $L^2(\mathcal{M})$ in terms of a linear operator $T : L^2(\mathcal{N}) \to L^2(\mathcal{M})$ mapping functions over $\mathcal{N}$ onto functions over $\mathcal{M}$. Clearly this setting generalizes classical vertex-wise correspondences as this can be achieved by a T that maps delta-functions onto delta-functions.

Given bases $\{\phi_i\}_{i\geq 1}$ and $\{\psi_i\}_{i\geq 1}$ on $L^2(\mathcal{N})$ and $L^2(\mathcal{M})$ respectively, the functional correspondence can be expressed w.r.t. to these bases as follows:

$$\begin{aligned} Tf &= T\sum_{i\geq 1} \langle f, \phi_i \rangle_{\mathcal{N}} \phi_i = \sum_{i \geq 1} \langle f, \phi_i \rangle_{\mathcal{N}} T\phi_i \\ &= \sum_{ij \geq 1} \langle f, \phi_i \rangle_{\mathcal{N}} \underbrace{\langle T\phi_i, \psi_j \rangle_{\mathcal{M}}}_{c_{ij}} \psi_j, \end{aligned} \tag{7.3}$$

where the coefficients c_{ij} depend on the choice of the bases. Taking only k elements of each basis, one obtains a rank-k approximation of T as a $k \times k$ matrix $\mathbf{C} = (c_{ij})$.

In order to compute $\mathbf{C}$, Ovsjanikov *et al.* [141] assume to be given a set of q corresponding functions $\{f_1, \ldots, f_q\} \subseteq L^2(\mathcal{N})$ and $\{g_1, \ldots, g_q\} \subseteq L^2(\mathcal{M})$. Denoting by $a_{ij} = \langle f_j, \phi_i \rangle_{\mathcal{N}}$ and $b_{ij} = \langle g_j, \psi_i \rangle_{\mathcal{M}}$ the $k \times q$ matrices of the respective coefficients onto the selected bases, functional correspondence boils down to the linear system

$$\mathbf{CA} = \mathbf{B}. \tag{7.4}$$

If $q \geq k$, the system (7.4) is (over-)determined and is solved in the least squares sense to estimate $\mathbf{C}$.

It is convenient to use the eigenfunctions of the Laplace-Beltrami operators of $\mathcal{N}$ and $\mathcal{M}$ as the bases $\{\phi_i, \psi_i\}_{i\geq 1}$, since under these bases truncating the series at the first k coefficients has the effect of "low-pass" filtering over the functional representations. In fact, while the Laplacian eigenbasis suffers from issues like mixing eigenspaces and sign ambiguity, it has been shown in [97] that the space of functions spanned by the first n eigenfunctions of the Laplace-Beltrami operator are stable under near-isometries as long as the n^{th} and the $(n+1)^{th}$ eigenvalues are well separated.

Furthermore, expressed in the Fourier bases, the matrix C has interesting properties making it more efficient to estimate. In fact, Ovsjanikov *et al.* [141] proved that if the correspondence preserves areas, then the matrix C is orthogonal, *i.e.* , $\mathbf{C}^T\mathbf{C} = \mathbf{I} = \mathbf{C}\mathbf{C}^T$. Further, if it is also an isometry, then we have $c_{ij} = \pm\delta_{ij}$.

More realistically, when the shapes are only approximately isometric, the matrix $\mathbf{C}$ would manifest a funnel-shaped structure, with the majority of elements distant from the diagonal close to zero. This fact has been used as a regularization for (7.4) in [105, 146].

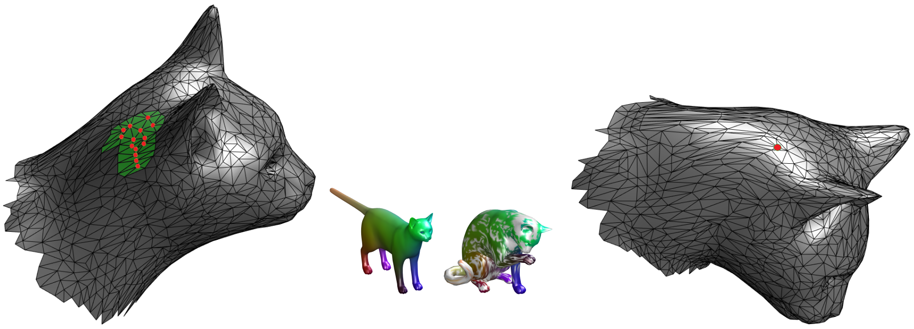


Figure 7.1: Example of non-injectivity of the functional map solution. Here, the maximum of the mapped function over the target shape is selected as the corresponding point, resulting in many-to-one relation between the mesh and the target one.

Discretization. In the discrete setting, the manifold $\mathcal{N}$ is sampled at n points $x_1,\dots,x_n$ which are connected by edges E and faces F, forming a manifold triangular mesh (V,E,F). In this setting, a function on the manifold is represented by an n-dimensional vector $\mathbf{f}=(f(x_1),\dots,f(x_n))^T$. The inner product is discretized as $\langle \mathbf{f},\mathbf{g}\rangle = \mathbf{f}^T\mathbf{S}\mathbf{g}$, where $\mathbf{S} = \text{diag}(s_1,\dots,s_N)$ and $s_i = \frac{1}{3}\sum_{jk:ijk\in F} A_{ijk}$ denotes the local area element. Hence, a basis $\mathbf{\Phi} = (\boldsymbol{\phi}_1,\dots,\boldsymbol{\phi}_k)$ over the vertices of the mesh represents the discretization of an orthogonal functional basis if $\mathbf{\Phi}^T\mathbf{S}\mathbf{\Phi} = \mathbf{I}$.

The discretization of the Laplacian takes the form of an $n\times n$ sparse matrix $\mathbf{L} = -\mathbf{S}^{-1}\mathbf{W}$ using the classical cotangent formula ([52, 125, 145]),

$$w_{ij} \;=\; \begin{cases} (\cot\alpha_{ij}+\cot\beta_{ij})/2 & ij\in E; \\ -\sum_{k\neq i} w_{ik} & i=j; \\ 0 & \text{otherwise.} \end{cases} \tag{7.5}$$

where α_{ij},β_{ij} denote the angles $\angle ikj, \angle jhi$ of the triangles sharing the edge ij.

The discretized Fourier basis $\mathbf{\Phi} = (\boldsymbol{\phi}_1,\dots,\boldsymbol{\phi}_k)$ is obtained by finding k orthogonal functions that satisfy the eigenvalue equation for the laplacian, *i.e.* , $-\mathbf{S}^{-1}\mathbf{W}\mathbf{\Phi} = \mathbf{\Phi}\mathbf{\Lambda}$ with $\mathbf{\Lambda} = \text{diag}(\lambda_1,\dots,\lambda_k)$ the diagonal matrix of the corresponding eigenvalues. This results in the standard generalized eigenvalue problem

$$\begin{aligned} & \mathbf{W}\hat{\mathbf{\Phi}} = -\mathbf{S}\hat{\mathbf{\Phi}}\mathbf{\Lambda} \\ s.t. \quad & \mathbf{\Phi}^T\mathbf{S}\mathbf{\Phi} = \mathbf{I} \end{aligned} \tag{7.6}$$

The Fourier coefficients of a (discretized) function $\mathbf{f}$ are then obtained by taking dot products of the function with the base elements, $a_i = \langle \boldsymbol{\phi}_i, \mathbf{f}\rangle = \boldsymbol{\phi}_i^T\mathbf{S}\mathbf{f}$, resulting in

$$\mathbf{a} = \mathbf{\Phi}^T\mathbf{S}\mathbf{f} \tag{7.7}$$

7.3 Problem statement

In this work, we propose a novel approach to the computation of a dense bijective map between two meshes. This is achieved through an optimization and refinement process based on the well-known functional map approach introduced in section 7.2. In the functional map framework, given a complete basis over the Hilbert space $L^2(\mathscr{M})$ and $L^2(\mathscr{N})$, a non-singular function correspondence operator would indeed result in a bijection between the manifolds. However, after discretization and the adoption of a limited set of basis functions, the bijectivity of the mapping cannot be guaranteed anymore.

In fact, delta functions are transformed into diffused functions by the low-frequency mapping, and a maximum-response reassignment often concentrate mappings onto few points on the target surface. Further, the space of linear functions from $L^2(\mathscr{M})$ to $L^2(\mathscr{N})$ contains the set of induced bijective correspondences, but is in fact larger, containing several transfer functions that cannot be reduced to bijections between points on the two manifolds. In other words, the bijections form a manifold on the space of functional maps which is not identified in the framework, with only the transportation of known functions in (7.4) to help driving the solution. In this chapter we address this issue by introducing additional optimized mapping functions that not only enforce the one-to-one correspondence, but also improve the quality of the map.

Let M and N be the two meshes for which we want to find bijective correspondences, Ψ and Φ be the two Fourier bases for the functional spaces $L^2(M)$ and $L^2(N)$ respectively, and $T : L^2(N) \to L^2(M)$ be the linear operator which models the functional correspondence between shapes. Let D_N and D_M be descriptor matrices used to drive the correspondence. Note that it does not matter which descriptors are employed and several different options are available, however, in this work, we use the Wave Kernel Signature [12] as dense descriptors for the meshes, since it was proven to be locally quite stable, a property we are most interested in.

In addition to the geometrical descriptors, we enforce bijectivity by adding constraints mapping delta functions onto smoothed versions of the delta functions, and we optimize over the location of these delta functions. In other words, to the mapping of D_N onto D_M, we add a mapping of a set of delta functions, expressed in terms of a permutation P over the vertices, onto the kernels K_N and K_M that represent localized descriptors, *i.e.* , smoothed versions of a delta function. This results in the following optimization problem:

$$\begin{aligned}\min_{T,P} \quad & \langle T(D_N|K_N) - (D_M|PK_M), \\ & T(D_N|K_N) - (D_M|PK_M)\rangle_M\end{aligned} \tag{7.8}$$

where $(\cdot|\cdot)$ is the horizontal concatenation operation, P belongs to the permutation group Σ_n, and K_N and K_M are Gaussian kernels built over the geodesic distances between the points of the meshes N and M respectively. In particular, let $i, j = 1, \dots, n$ with n the number of vertices of a mesh, we define a generic K as:

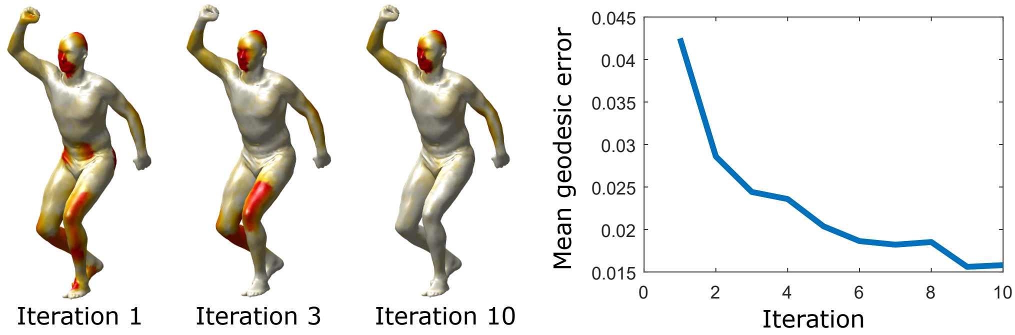


Figure 7.2: On the left, the geodesic error evolution is shown as the colormap on the mesh at a given iteration (red means higher error), while on the right the graph shows the mean geodesic error computed at the first 10 iterations.

$$K_{i,j} = e^{-\frac{1}{2\sigma^2}d^2(i,j)} \tag{7.9}$$

where $d^2(i,j)$ is the geodesic distance between the i^{th} and the j^{th} vertex of the mesh.

Under this formulation, P represents an assignment map of the points of N onto the points of M and we estimate the functional representation of the map T and the permutation matrix P which minimize the Frobenius norm of the difference between the functional mapping of the descriptors D_M augmented with the kernel K_M, and the corresponding functions D_N concatenated with the kernel K_N permuted by P. As a consequence of the fact that $P \in \Sigma_n$, the permutation that minimizes (7.8) will represent a dense one-to-one map between the vertices of the meshes, assuming equal cardinality of the sets of vertices.

7.4 Iterative Optimization

The problem defined in (7.8) is tackled through an iterative optimization process in which, after an initialization step, we firstly optimize over the functional map T, and subsequently over permutation matrix P.

The initialization step is two-fold. Firstly, we need to compute an initial sparse set of matches S^0 between the meshes. This will be used as a sub-sampled permutation matrix in the computation of the first functional correspondence map. To tackle this problem in an unsupervised way, we employ the game theoretic approach [151] which aims to solve the problem of minimum distortion intrinsic correspondences between deformable shapes. However, any method able to retrieve a sparse set of matches can be employed. With S^0 at hand, we augment D_N and D_M with a set of already-in-correspondence descriptors $\hat{D}_N = (D_N|K_N)$ and $\hat{D}_M = (D_M|K_M)$. Then, following the original work, we minimize $\|T\hat{D}_N - \hat{D}_M\|_2^2$ over T. Let T^0 be the first functional map. Note that augmenting the descriptor matrices with correspondent functions has been shown to improve the accuracy of the resulting map (see [141] for more details).

For the sake of clarity, let us unfold (7.8), rewriting it in the equivalent form

$$\begin{aligned}\underset{T,P}{\arg\min} \quad & Tr\big((D_N|K_N)^T T^T \mathbf{S}_M T (D_N|K_N)\big) \\ & + Tr\big((D_M|PK_M)^T \mathbf{S}_M (D_M|PK_M)\big) \\ & -2Tr\big((D_N|K_N)^T T^T \mathbf{S}_M (D_M|PK_M)\big)\end{aligned} \tag{7.10}$$

where $\mathbf{S}_M$ and $\mathbf{S}_N$ (below) are matrices whose diagonal elements are the area elements of the respective shapes. Following straightforwardly from section 7.1, we know that $D_M = TD_N$. The discretization of T using the first k harmonics of the Fourier basis allows to define $T = \Psi C \Phi^T \mathbf{S}_N$. Hence, we can rewrite (7.10) as

$$\begin{aligned}\underset{C,P}{\arg\min} \quad & Tr\big(\Phi^T \mathbf{S}_N (D_N D_N^T + K_N K_N^T) \mathbf{S}_N \Phi\big) \\ & + Tr\big((D_M D_M^T + K_M K_M^T) \mathbf{S}_M\big) \\ & -2Tr\big(C\Phi^T \mathbf{S}_N (D_N D_M^T + K_N K_M^T P^T) \mathbf{S}_M \Psi\big)\end{aligned} \tag{7.11}$$

leaving the unkowns only in the last trace term, resulting in the equivalent problem

$$\underset{C,P}{\arg\max} \quad Tr\big((D_N D_M^T + K_N K_M^T P^T) \mathbf{S}_M \Psi C \Phi^T \mathbf{S}_N\big) \tag{7.12}$$

Our iterative process begins with the computation of the permutation matrix P, which will represent our dense injective solution. Solving (7.12) with respect to P allows to define $W = \mathbf{S}_M \Psi C \Phi^T \mathbf{S}_N K_N K_M^T$, which brings us to the definition of the equivalent problem

$$\underset{P \in \Sigma_n}{\arg\max}\, Tr\left(WP^T\right) \tag{7.13}$$

which is a standard assignment problem that can be solved with one of the several combinatorial optimization algorithms. In this work we use the Auction Algorithm [17], which is known to be very efficient with sparse weight matrix. Which is our case, since W is, practically speaking, a smoothed permutation matrix (smooth because of the intrinsic symmetries).

With P at hand, we can compute the functional correspondences C. Since $C \in \mathcal{O}(k)$, (7.12) can be solved as an orthogonal Procrustes Problem [162] through *singular value decomposition*. Let

$$\begin{aligned} ULV^T &= \Phi^T \mathbf{S}_N (D_N D_M^T + K_N K_M^T P^T) \mathbf{S}_M \Psi \\ &= \big(\Phi^T \mathbf{S}_N (D_N | P^T K_N)\big)\big(\Psi^T \mathbf{S}_M (D_M | K_M^T)\big)^T \end{aligned} \tag{7.14}$$

with ULV^T computed through *svd*. The C which maximizes the objective function can be computed as $C = UV^T$.

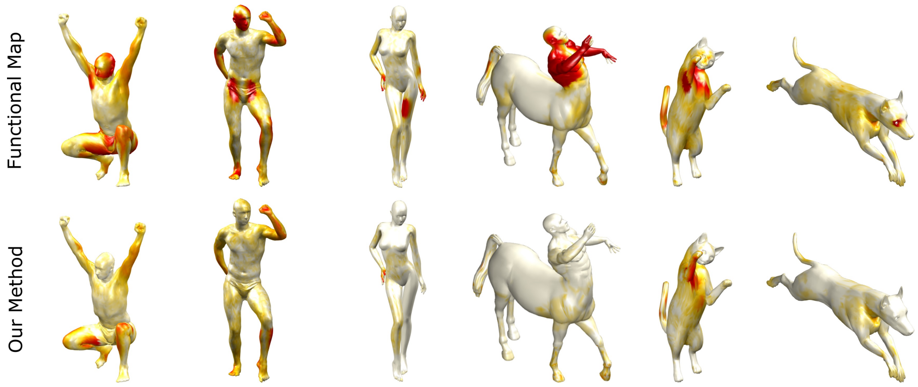


Figure 7.3: Qualitative example of the geodesic errors computed on the meshes by the functional matching approach (first row) and the proposed one (bottom). The red color on the surfaces indicates higher error.

Note that, while *svd* is computed over a manageable $k \times k$ matrix (with k the number of columns in the basis), its computation requires the computation and multiplication of the $n \times n$ matrix $(D_N D_M^T + K_N K_M^T P^T)$. In particular, while the $D_N D_M^T$ is low rank, the $K_N K_M^T P^T$ is full rank, yet sparse. In order to render the computation of the Procrustes matrix computationally feasible, we randomly sample t correspondences from P to use in the estimation of C. This is equivalent to choose t columns of P^T and the same t columns of K_N in (7.14). This, assuming that the descriptors D_N and D_M have dimensionality r, results in the multiplication of the $k \times (r+t)$ matrix $\Psi^T \mathbf{S}_M(D_M|PK_M)$ by the $(r+t) \times k$ matrix $\big(\Phi^T \mathbf{S}_N(D_N|K_N)\big)^T$, resulting in the rather more manageable complexity $O\big(k^2 n(r+t)\big)$.

Once both the permutation matrix P, which represents our injective map, and the functional correspondences C are computed, we check if the new solution reaches the convergence criterion. As a convergence measure, we check how much the map has changed with respect to the previous iteration in terms of geodesic distance. In particular, let $d^2(i^{t-1}, i^t)$ be the squared geodesic distance between the vertices i^{t-1} and i^t both belonging to the mesh N and both set into correspondence with the same vertex j belonging to M at time $t-1$ and t respectively. We define our convergence measure as

$$\frac{1}{n}\sum_i d^2(i^{t-1}, i^t) < \epsilon \tag{7.15}$$

where n is the number of vertices of N while $\epsilon = 10^{-3}$. If the convergence criterion is not satisfied, we take the dense injective map and randomly sub-sample it in order to get a new set of matches S^1 to be used to feed a new iteration of the whole process.

The algorithm of the process is listed in Algorithm 1, while an analysis about the performance achieved is demanded to section 7.5.

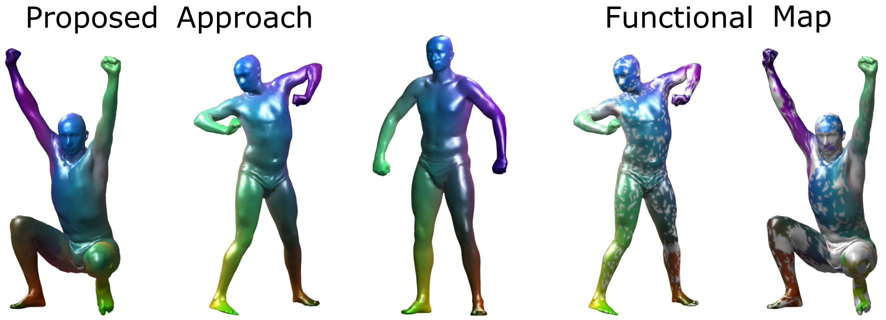


Figure 7.4: Proportion of points matched onto the mesh target. The gray patches represent the areas that have not been mapped. Center: reference mesh. Left: our approach. Right: standard functional maps. The colormap is projected from the reference onto the other meshes.

7.5 Experimental Evaluation

We test our approach on benchmarks built from the TOSCA and SCAPE datasets. We present a qualitative and quantitative analysis of the performance in various settings and compare the results to several state-of-the-art methods for correspondences retrieval between non-rigid meshes. For the qualitative comparison, we use the standard functional map results, since it is the method that performs better among the ones we are comparing against.

Datasets

We selected the TOSCA and SCAPE datasets to assess the performance of the compared methods, since they provide a large variety of objects with ground-truth correspondences.

SCAPE [11] is composed of a set of 71 models representing the same person in different poses. The meshes of the dataset are the result of the SCAPE (*Shape Completion and Animation of People*) method application. It is a data-driven method used to build a model of the human shape which varies both in the shape of the subject and in the pose. The method is applied to a set of scanned human bodies with one-to-one vertex correspondences among all the meshes of the dataset. Each mesh counts $12.5k$ vertices.

TOSCA [29] contains 76 nearly-isometric shapes subdivided in 8 classes (dog, centaur, human males and female, *etc.*). Each class comes with a “null” mesh, which is a representative of the class in a standard pose. Ground-truth correspondences are provided for each mesh that belongs to the same class. Typical vertex count is about $50,000$.

For the sake of efficiency, we have down-sampled the null meshes of each class in TOSCA and the first mesh in SCAPE to approximately $10k$ vertices, while the non-null ones have been re-sampled accordingly. The null meshes of the TOSCA dataset and the first mesh of the SCAPE dataset are both almost extrinsically bilaterally symmetric, which means that we are able to compute the intra-mesh symmetric map and use it in the evaluation of the performance. Indeed, since both symmetric maps are usually allowed as a solution to the correspondence problem between meshes (in particular when intrinsic methods are employed), we use as geodesic error the lower between the two symmetric solutions.

Evaluation Methods

To assess the accuracy of the proposed method we use as measure the geodesic distance of a match with respect to the ground-truth. Let M and N be two meshes. Let $t : N \to M$ be the predicted map, and let $t^* : N \to M$ be the ground-truth map. For every vertex v on N we compute the geodesic distance on $D_N(t(v), t^*(v))$. We aggregate these geodesic distances into the error measure

$$Err(t, t^*) = \sum_{v \in N} \frac{D_N(t(v), t^*(v))}{\sqrt{Area(N)}} \tag{7.16}$$

Note that the geodesic distances are normalized over the square root of the area of the mesh $\sqrt{Area(N)}$. The error $Err(t, t^*)$ is averaged over all the meshes of a class of a dataset. In order to study the distribution of the error, we plot cumulative curves showing the percentage of matches whose errors are smaller than a threshold.

Comparison to other methods

The proposed approach is compared with several state-of-the-art correspondence retrieval methods. In particular, we compare our work with:

- **Blendend map** [103]: Kim *et al.*propose a fully automatic pipeline for finding smooth intrinsic map between non-isometric surfaces by blending a collection of low dimensional maps (which provides a good solution locally) in order to get a better solution globally.

- **Functional map** [141]: Ovjsanikov *et al.*introduce the functional maps framework, in which shape correspondences are modelled as linear operator between spaces of functions on the surfaces. This approach allows to represent the shapes in a very convenient way and one of the main advantage is that the solution to the correspondence problem is reduced to simple algebraic problem. Even if this method allows to retrieve a dense set of matches between meshes, it does not guarantee an injective solution for the reasons explained in section 7.1 and 7.3.

- **Combinatorial Matching** [156]: in this work, Sahillioğlu and Yemez solve the correspondences problem minimizing the isometric distortion in the 3D Euclidean space by using a coarse-to-fine combinatorial search algorithm.

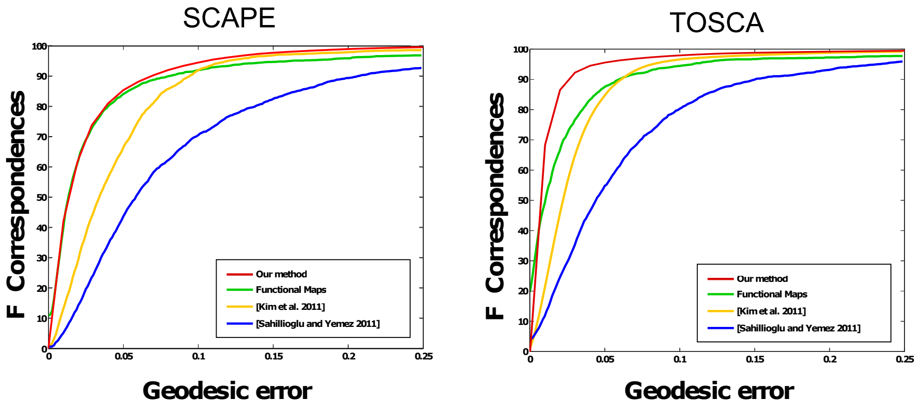


Figure 7.5: Comparison of the normalized error curves obtained on SCAPE and TOSCA datasets using several shape matching algorithms while allowing the symmetric solutions, which are the only meaningful results given the intrinsic nature of the proposed approach.

Results on SCAPE and TOSCA

Before diving into the quantitative results, we show the performance achieved by the proposed method from a qualitative point of view. In Figure 7.3 we compare the maps produced by our approach with those obtained with the original functional map approach. In particular, we present a set of meshes that have been aligned with the proposed method and whose errors computed point-wise are projected directly on the meshes: the redder the surface, the higher the error. The proposed approach greatly reduces the error in areas where the functional map is not able to find the correct correspondences.

In Figure 7.2 we show the mean geodesic error of the maps at 10 iterations of the proposed approach, while on the left side of the image we give a qualitative glimpse of the iterative process. More precisely, we show how the geodesic error (which is used as colormap) changes on the mesh at different iterations.

Figure 7.4 shows the differences between the original functional map and the proposed approach in terms of proportion of matched points on the target mesh. In the images, the gray patches are vertices that are not target of a map. The central part of the image shows the reference mesh (whose colormap is projected onto the other meshes). The two columns on the left show the proportion of surface covered by the dense map yielded by the proposed approach, while the ones on the right side show the results achieved by standard functional maps. It is easy to see that the functional map solution produces a set of matches which are clustered in relatively few points on the target shape, resulting in a wider gray area over the surface with respect to the proposed approach.

Finally, to better characterize the performance of the proposed approach, we performed a set of experiments extracting correspondences between nearly isometric pairs of meshes from the TOSCA and SCAPE datasets. In particular, we randomly selected a mesh as the reference mesh of a certain class. Then, we computed the dense injective map between the reference mesh and all the other meshes belonging to the same class. The error is computed as introduced in section 7.5, averaging all the errors computed according to 7.16 on all the meshes of the datasets. The results are shown in Figure 7.5, where each curve represents the percentage of correspondences found whose normalized geodesic errors are below a certain threshold.

7.6 Conclusion

In this chapter we proposed a novel solution to the problem of extracting bijective maps between shapes. The problem has been cast in terms of functional correspondences while the bijectivity of the map is achieved casting the problem onto an equivalent assignment problem. The experimental session showed that the proposed approach is competitive with the state-of-the-art matching methods.

One of the main issues of our method concerns the focus on particular deformations and cannot, for example, tackle topological changes among the meshes. This is a known limit of spectral methods in general. As a future work, we want to investigate the use of different bases and descriptors, which could lead to an approach which is more robust to a wider range of deformations.

Algorithm 1: Iterative Optimization Process

```
Data: $M, N, S^0$
Result: $C$ and $P$
  1. $S^i = S^0$
  2. Use the sparse set of correspondences $S^i$ to compute the functional map $C^i$
     through (7.14)
  3. Compute the weight matrix $W^i$ and solve (7.13) using the Auction Algorithm [17],
     getting the injective assignment $P^i$
  4. Randomly subsample $P^i$, obtaining $S^{i+1} \subset P^i$
  5. If (7.15) is false and the maximum number of iterations has not been reached,
     increase $i$ and go to 2, otherwise $C = C^i$ and $P = P^i$
```

III

Structure-based Representation

8

Non-Parametric Spectral Model for Graph Classification

Graph-based representations have been used with considerable success in computer vision in the abstraction and recognition of object shape and scene structure. Despite this, the methodology available for learning structural representations from sets of training examples is relatively limited. In this chapter we take a simple yet effective spectral approach to graph learning.

In particular, we define a novel model of structural representation based on the spectral decomposition of graph Laplacian of a set of graphs, but which make away with the need of one-to-one node-correspondences at the base of several previous approaches, and handles directly a set of other invariants of the representation which are often neglected. An experimental evaluation shows that the approach significantly improves over the state of the art.

8.1 Introduction

Graph-based representations have been applied with considerable success to several tasks as convenient means of representing structural patterns. Examples include the arrangement of shape primitives or feature points in images, molecules, and social networks [53]. Their success lies in their ability to concisely capture the relational arrangement of primitives, in a manner which can be invariant to irrelevant transformation such as changes in object viewpoint. Despite their many advantages and attractive features, the methodology available for learning structural representations from sets of training examples is relatively limited, and the process of capturing the modes of structural variation for sets of graphs has proved to be elusive.

Structural representations are widely adopted in the context of Bayesian networks, or general relational models [61], where structural learning processes are used to infer the stochastic dependency between these variables. However, these approaches rely on the availability of correspondence information for the nodes of the different structures used in learning. In many cases the identity of the nodes and their correspondences across samples of training data are not known, rather, the correspondences must be recovered from structure.

In the last few years, there has been some effort aimed at learning structural archetypes and clustering data abstracted in terms of graphs. In this context, spectral approaches have provided simple and effective procedures. For example, Luo and Hancock [124] use graph spectral features to embed graphs in a (low) fixed-dimensional space where standard vectorial analysis can be applied. While embedding approaches like this one preserve the structural information present, they do not provide a means of characterizing the modes of structural variation encountered and are limited by the stability of the graph's spectrum under structural perturbation. Bonev *et al.* [24], and Bunke *et al.* [37] summarize the data by creating super-graph representation from the available samples, while White and Wilson [192] use a probabilistic model over the spectral decomposition of the graphs to produce a generative model of their structure. While these techniques provide a structural model of the samples, the way in which the super-graph is learned or estimated is largely heuristic in nature and is not rooted in a statistical learning framework. Torsello and Hancock [182] define a superstructure called tree-union that captures the relations and observation probabilities of all nodes of all the trees in the training set. The structure is obtained by merging the corresponding nodes and is critically dependent on the order in which trees are merged. Todorovic and Ahuja [176] applied the approach to object recognition based on a hierarchical segmentation of image patches and lifted the order dependence by repeating the merger procedure several times and picking the best model according to an entropic measure. While these approaches do capture the structural variation present in the data, the model structure and model parameter are tightly coupled, which forces the learning process to be approximated through a series of merges, and all the observed nodes must be explicitly represented in the model, which then must specify in the same way proper structural variations and random noise.

In more recent work [178, 185] Torsello and co-workers proposed a generalization for graphs which allowed to decouple structure and model parameters and used a stochastic process to marginalize the set of correspondences. The process however still requires a (stochastic) one-to-one relationship between model and observed nodes and could only deal with size differences in the graphs by explicitly adding a isotropic noise model for the nodes.

In this chapter we aim at defining a novel model of structural representation based on a spectral description of graphs which lifts the one-to-one node-correspondence assumption and is strongly rooted in a statistical learning framework. In particular, we follow White and Wilson [192] in defining separate models for eigenvalues and eigenvectors, but cast the eigenvector model in terms of observation over an implicit density function over the spectral embedding space, and we learn the model through non-parametric density estimation. The eigenvalue model, on the other hand, is assumed to be log-normal, due to consideration similar to [12].

8.2 Spectral Generative Model

Let $G = (V, E)$ be a graph, where V is the set of nodes and $E \subseteq V \times V$ is the set of edges, and let $A = (a_{ij})$ be its adjacency matrix. The degree d of a node is the number of edges incident to the node and it can be represented through the degree matrix $D = (d_{ij})$ which is a diagonal matrix with $d_{ii} = \sum_j a_{ij}$. Starting from these two matrix representations of a graph, it is possible to compute the *Laplacian* matrix, which is defined as the difference between the degree matrix D and the adjacency matrix A:

$$L = D - A$$

The Laplacian is a symmetric positive-definite matrix. Its lower eigenvalue is equal to 0 with multiplicity equal to the number of connected components in G. Further, the Laplacian is associated with random walks over the graph and it has been extensively used to provide spectral representations of structures [118]. The spectral representation of the graph can be obtained from the Laplacian through singular value decomposition. Given a Laplacian L, its decomposition is $L = \Phi\Lambda\Phi^T$, where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_{|V|})$ is the matrix whose diagonal contains the ordered eigenvalues, while $\Phi = (\phi_1|\phi_2|...|\phi_{|V|})$ is the matrix whose columns are the ordered eigenvectors. This decomposition is unique up to a permutation of the nodes of the graph, a change of sign of the eigenvectors, or a change of basis over the eigenspaces associated with a single eigenvalue, *i.e.* , the following properties hold:

$$\begin{aligned} L &\simeq PLP^T = P\Phi\Lambda(P\Phi)^T && (8.1)\\ L &= \Phi\Lambda\Phi^T = \Phi S\Lambda S\Phi^T && (8.2)\\ L &= \Phi\Lambda\Phi^T = \Phi B_\lambda\Lambda B_\lambda\Phi^T && (8.3) \end{aligned}$$

where $\simeq$ indicates isomorphism of the underlying graphs, P is a permutation matrix, S is a diagonal matrix with diagonal entries equal to ± 1, and B_λ is a block-diagonal matrix with the block diagonal corresponding to the eigenvalues equal to λ in Λ and is orthogonal while all the remaining diagonal blocks are equal to the identity matrices.

Our goal is to devise a model for the graph spectra that can capture the main modes of variation present in a set of sample graphs, and that takes into account the invariances of the spectral representation. Following [192] we make two separate and independent models for the eigenvalues and eigenvectors of the Laplacian:

$$P(G|\Theta) = P(\Lambda^G|\Theta^\Lambda)P(\Phi^G|\Theta^\Phi) \qquad (8.4)$$

where Θ is the graph-class model divided into its eigenvalue-model component Θ^Λ and eigenvector-model component Θ^Φ.

For the eigenvalue model we follow [12] and opt to model the observation distribution of a single eigenvalue as a log-normal distribution. In [12] it was shown that this model derived directly from rather straightforward stability considerations derived from matrix perturbation theory. As a result, we model the set of eigenvalues as a series of independent log-normal distribution, one per eigenvalue used, resulting in:

$$P(\Lambda^G|\Theta^\Lambda) = (2\pi)^{\frac{d}{2}} \prod_{i=1}^{d} \frac{1}{\lambda_i \sigma_i} e^{\frac{-(\ln\lambda_i - \mu_i)^2}{2\sigma_i^2}} \tag{8.5}$$

where μ_i and σ_i are model parameters to be learned from data and d is the number of eigenvalues/eigenvectors used in the model.

On the other hand, the eigenvector component is modelled as an unknown distribution $\mathscr{F}$ on the d-dimensional spectral embedding space $\Omega_d \subseteq \mathbb{R}^d$. The d-dimensional spectral embedding of a graph is obtained from the eigenvector matrix Φ^G by taking its first d columns, corresponding to the eigenvectors associated with the d smallest eigenvalues, excluding the trivial constant eigenvector corresponding to a 0 eigenvalue. With the reduced $n \times d$ eigenvector matrix $\hat{\Phi}$ at hand, we take its rows to be points in the d dimensional spectral embedding space Ω_d.

Note that there is a length invariance in the eigenvectors, which are usually assumed to be of unit Euclidean norm. This, however, results in a size compression of the spectral embedding points as the graph size grows. To correct this issue we scale the embedding vectors by multiplying them by the graph size n.

With this model we cast the learning phase into a non-parametric density estimates of the distribution of the spectral embedding points $\phi_1^G, \dots, \phi_n^G$. Under these assumptions, the eigenvector model parameter Θ^Φ is constituted of a collection of N d-dimensional vectors $\theta_1^\Phi, \dots, \theta_N^\Phi$ corresponding to samples from the unknown density function. In the learning phase these are obtained aligning and merging spectral embedding points from the sample graphs belonging to each class.

This per-vertex sample approach takes care of the permutational invariance, but we still need to explicitly deal with the other invariances, i.e., the sign of eigenvectors and choice of an eigenbasis. We solve those invariances by optimizing over the respective transformation groups. Furthermore, we lift the block constraint over the eigenbasis selection, relaxing it to an optimization over the orthogonal group $\mathbb{O}(d)$. This results in the following definition of the eigenvector probability:

$$P(\Phi^G|\Theta^\Phi) = \max_{\mathscr{R}\in\mathbb{O}(d)} \max_{S\in\{\pm 1\}^d} (Nh^d)^{-n} \prod_{i=1}^{n} \sum_{j=1}^{N} e^{-\frac{\|\mathscr{R}S\phi_i^G - \theta_j^\Phi\|^2}{2h^2}} \tag{8.6}$$

which is the product of Parzen-Rosenblatt kernel density estimators. ϕ_i^G is the vector obtained taking the first d elements of the i-th row of the eigenvector matrix Φ^G and θ_j^Φ is the j-th component of the eigenvector model Θ^Φ. Here we assume that the model is simply an array of samples from the graph class.

In this work we use Silverman's rule-of-thumb [171] for the multivariate case to estimate the bandwidth parameter h.

$$h = \left(N\frac{d+2}{4}\right)^{-\frac{1}{d+4}} \sigma \tag{8.7}$$

where σ is computed as the squared root of the trace of the covariance matrix Σ of the eigenvector model divided by the number of nodes of the model

$$\sigma = \sqrt{\frac{1}{n} Tr(\Sigma)} \tag{8.8}$$

8.2.1 Model Learning

The learning process aims to estimate the parameters for the eigenvector and eigenvalue models. Given a set of graphs $G = \{G_1, G_2, \dots, G_m\}$, belonging to the same class $\mathscr{C}$, we firstly compute their spectral decomposition, obtaining the set $\{(\Phi_1^{\mathscr{C}}, \Lambda_1^{\mathscr{C}}), (\Phi_2^{\mathscr{C}}, \Lambda_2^{\mathscr{C}}), \dots, (\Phi_m^{\mathscr{C}}, \Lambda_m^{\mathscr{C}})\}$. In particular, the $\Phi_i^{\mathscr{C}}$s are composed by column vectors which are the first d non-trivial eigenvectors of the Laplacian matrix of the corresponding graph, while the $\Lambda_i^{\mathscr{C}}$s contain the first d non-zero eigenvalues. Hence, d represents our embedding dimension. The eigenvector model of the class $\mathscr{C}$, denoted as $\Phi^{\mathscr{C}}$, is defined as

$$\Phi^{\mathscr{C}} = \begin{bmatrix} \phi_1^1 & \phi_2^1 & \cdots & \phi_d^1 \\ \phi_1^2 & \phi_2^2 & \cdots & \phi_d^2 \\ \vdots & \vdots & \vdots & \vdots \\ \phi_1^m & \phi_2^m & \cdots & \phi_d^m \end{bmatrix}$$

where ϕ_j^i denotes the j-th non-trivial eigenvector (still a column vector) of the i-th graph of the set G. In other word, we perform a vertical concatenation of all the eigenvectors matrices of the graphs that belong to class $\mathscr{C}$. Thus, the dimension of the eigenvector model of the class is $\left(\sum_{i=1}^{m} ||G_i||\right) \times d$.

Estimating the Eigenvector Sign-Flips

The eigenvector matrix produced by the eigendecomposition is unique up to a sign factor. Since our method characterize every node of a graph with a feature vector, a sign disambiguation is mandatory. There are several techniques that allow to detect and solve this ambiguity, like using the correlation between two functions (i.e. probability density functions). If the correlation grows after a flip, then the eigenvector sign should be flipped. Unfortunately, with increasing size, this method becomes computationally heavy.

For such reason, we have to employ an heuristic-based method in order to solve the sign-ambiguity problem. Since it is an heuristic approach, it does not guarantee the discovery of all the correct signs. Given two graphs G_A and G_B, which belong to the same class $\mathscr{C}$, let Φ_j^A and Φ_j^B be the j-th eigenvectors of the spectral representation of the graphs. We assume eigenvectors to be random variables having unknown probability density function. We assume that all the j-th eigenvectors of graphs in the same class share a very similar *pdf* among them, up to the sign. A flipped sign does not influence the shape of a *pdf*, but the peak of the function results shifted. Once a reference graph is selected (for example, A), the sign ambiguity is solved by checking the sign of the peaks of each eigenvector of the reference graph and the others. An eigenvector is flipped when the signs of the peaks are different.

$$\phi_j^B = \begin{cases} \phi_j^B(-1) & \text{if } x_j^{A*} < 0 \text{ and } x_j^{B*} \geq 0\,, \\ \phi_j^B(-1) & \text{if } x_j^{A*} \geq 0 \text{ and } x_j^{B*} < 0\,, \\ \phi_j^B & \text{otherwise.} \end{cases} \tag{8.9}$$

The *pdf*s of each eigenvector are estimated using kernel density estimation. The density estimates are evaluated at 100 points covering the range of the eigenvectors. Those evaluations are then used to find the peaks, more precisely the related independent variables x_j^{A*} and x_j^{B*} of the functions.

Hence, to solve the sign-ambiguity issue, before the construction of $\Phi^{\mathscr{C}}$, we flip each graph according to a reference graph G_f (chosen randomly within G) using (8.9).

The next step involves the rotation of each eigenvectors matrix according to the same reference graph G_f.

Estimating the Eigenvector Orthogonal Transformation

The sign disambiguation process produces a rough rotation which helps to align the eigenvectors of a graph with respect to the eigenvectors of a reference graph. In order to minimize the variance between the eigenvector matrices of a reference graph (one for each class) and the eigenvector matrices of the other graphs, another rotation step is applied. In particular, we are looking for the rotation which minimize the distance between the nodes of two graphs. More formally, we want to maximize the following:

$$\underset{\mathscr{R}\in\mathbb{O}(d)}{\arg\max} \prod_i P(\mathscr{R}x) \tag{8.10}$$

where

$$P(x) \propto \sum_j e^{-\frac{1}{2}\frac{\|x-x_j\|^2}{h^2}} \tag{8.11}$$

The above formulation of the optimization problem is then applied to our definition of probability density applying the constraints to a Parzen-Rosenblatt kernel density estimator, obtaining

$$\underset{\mathscr{R}}{\arg\max} \prod_i \sum_j e^{-\frac{1}{2}\frac{\|\mathscr{R}x_i-y_j\|^2}{h^2}} \tag{8.12}$$

We subdivide our rotation matrix in two rotation matrices, namely $\mathscr{R}$ (the initial rotation) and S (an additive rotation). The log-likelihood obtained after the introduction of the new rotation matrix to equation 8.12 can be written as

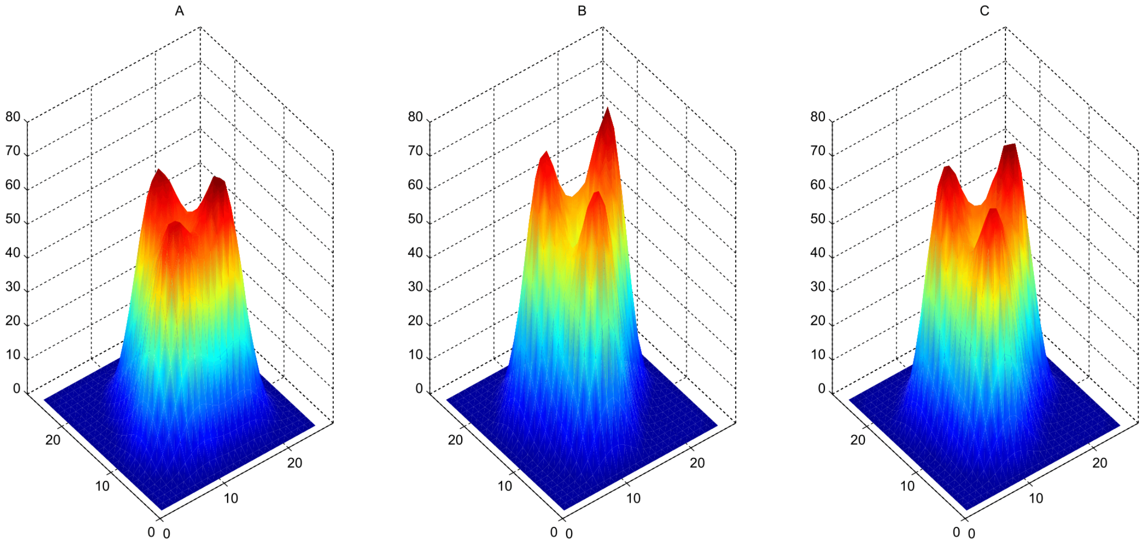


Figure 8.1: Example of the computation of the rotation matrix. A) KDE applied to the eigenvectors matrix of the Laplacian of a graph, B) KDE of a synthetically rotated eigenvectors matrix of the same graph, C) show the KDE of the eigenvectors matrix after the application of the rotation matrix computed using the described method.

$$\mathscr{L} = \sum_i \log\left(\sum_j e^{-\frac{1}{2}\frac{\|S\mathscr{R}x_i - y_j\|^2}{h^2}}\right) \tag{8.13}$$

Let α_{ij} be defined as

$$\alpha_{i,j} = e^{-\frac{1}{2}\frac{\|\mathscr{R}\phi_i - \phi_j^{\mathscr{C}}\|^2}{h^2}} \tag{8.14}$$

In order to solve 8.10, we compute the gradient with respect to the additive rotation matrix S introduced in 8.13.

$$\frac{\partial \mathscr{L}}{\partial S_{hk}} = \sum_i \frac{\sum_j \alpha_{ij}\left(-\frac{1}{2}\frac{\frac{\partial}{\partial S_{hk}}\|S\mathscr{R}x_i - y_j\|^2}{h^2}\right)}{\sum_j \alpha_{ij}} \tag{8.15}$$

where

$$\frac{\partial}{\partial S_{hk}}\|S\mathscr{R}x_i - y_j\|^2 = -2(y_i)_h(\mathscr{R}x_i)_k \tag{8.16}$$

Since they are scalar

$$\partial_S = -2y_j(\mathscr{R}x_i)^T = -2y_j x_i^T \mathscr{R}^T \tag{8.17}$$

We can now rewrite 8.13 as

$$\frac{\partial \mathscr{L}}{\partial S} = \left(\sum_i \frac{\sum_j \alpha_{ij} \frac{1}{h^2} y_j x_i^T}{\sum_j \alpha_{ij}} \right) \mathscr{R}^T \tag{8.18}$$

For the sake of readability, let A be defined as

$$A = \sum_i \frac{\sum_j \alpha_{ij} \frac{1}{h^2} y_j x_i^T}{\sum_j \alpha_{ij}} \tag{8.19}$$

Since S is an orthogonal rotation matrix, it belongs to the Lie group $\mathbb{O}(d)$. The tangent space at the identity element of the Lie group is its Lie algebra, which is the skew-symmetric matrices space. The skew-symmetric component of a matrix M is given by $\frac{M-M^T}{2}$.

In order to project the gradient to the null space (to find the maximum), we have to make $A\mathscr{R}^T$ symmetric. The rotation matrix $\mathscr{R}$ which symmetrizes the previously computed gradient is obtained through the singular value decomposition (SVD) of A, $svd(A) = ULV^T$. In particular, we can compute R as

$$\mathscr{R} = UV^T \tag{8.20}$$

which symmetrize the gradient. Indeed

$$A\mathscr{R}^T = (ULV^T)(VU^T) = ULU^T \tag{8.21}$$

which is symmetric. Refer to figure 8.1 for a graphical example of the described process.

To compute the rotation we used the following algorithm:

1. The initial value of $\mathscr{R}$ is the identity matrix

2. Compute α_{ij} (8.14) for each $i = 1, \dots, n$ (where n is the number of nodes of a graph) and $j = 1, \dots, N$ (where N is the number of nodes of the model).

3. Compute the matrix A (8.19)

4. Compute the singular value decomposition of A, $svd(A) = ULV^T$

5. Compute $\mathscr{R}$ as $\mathscr{R} = UV^T$

6. If the convergence is achieved, i.e. $A = A^T$, or the maximum number of iterations allowed is reached, end the algorithm, otherwise repeat from 2

The maximum number of iterations parameter was set to 10 for the results showed in section 8.3.

Estimating the eigenvalue model

Let $G^{\mathscr{C}} = \{G_1, G_2, \dots, G_m\}$ be a set of graphs belonging to the same class $\mathscr{C}$, and let $\{\Phi_i^{\mathscr{C}}, \Lambda_i^{\mathscr{C}}\}$, $i = 1, \dots, m$, their spectral representation. The diagonal of the eigenvalue matrix $\Lambda_i^{\mathscr{C}}$ contains the eigenvalues $\{\lambda_1^i, \lambda_2^i, \dots, \lambda_d^i\}$ of the i-th graph of the set. Let

$$\Lambda^{\mathscr{C}} = \begin{bmatrix} diag(\lambda_1^{\mathscr{C}}) \\ diag(\lambda_2^{\mathscr{C}}) \\ \vdots \\ diag(\lambda_m^{\mathscr{C}}) \end{bmatrix}$$

be a $m \times d$ matrix containing the firsts d non-zero eigenvalues of the spectral representation. We assume that all the j-th eigenvalues of $\Lambda_i^{\mathscr{C}}$, with $j = 1, \dots, d$, are distributed as a log-normal distribution, as shown in 8.5. We do a maximum likelihood estimate for the model parameters resulting in:

$$\hat{\mu} = \frac{\sum_i \ln x_i}{m}, \quad \hat{\sigma}^2 = \frac{\sum_i (\ln x_i - \hat{\mu})^2}{m} \tag{8.22}$$

8.2.2 Prediction

Once the models are computed, we can combine them in order to classify a graph which does not belong to the training set used to compute $\{\Phi^{\mathscr{C}}, \Lambda^{\mathscr{C}}\}$. Let G^* be such graph. Let Φ^* and Λ^* be the spectral decomposition of the Laplacian of G^*. Thanks to the assumption of independence between the two models, we can define the prediction as the posterior probability

$$P(\mathscr{C} \mid G^*) = P(\Phi^* \mid \Phi^{\mathscr{C}}) P(\Lambda^* \mid \Lambda^{\mathscr{C}}) \tag{8.23}$$

Once both the above mentioned probabilities are computed, i.e. the probabilities with respect to the eigenvector model and to the eigenvalue model, and still assuming the independence between them, we can compute the conditional distribution with respect to the class $\mathscr{C}$ using equation 8.23. But since both $P(\Phi^* \mid \Phi^{\mathscr{C}})$ and $P(\Lambda^* \mid \Lambda^{\mathscr{C}})$ come from a log-derivation (equation 8.25 and 8.26), it can be rewritten as

$$\log P(\mathscr{C} \mid G^*) = \ell_{\mathscr{L}}(\Phi^* \mid \Phi^{\mathscr{C}}) + \ell_{\mathscr{L}}(\Lambda^* \mid \Lambda^{\mathscr{C}}) \tag{8.24}$$

In particular, the eigenvector model log-likelihood is defined as

$$\ell_{\mathscr{L}}(\Phi^* | \Theta^{\Phi}) = \prod_{i=1}^{n} P(x_i) = \sum_{i=1}^{n} \log P(\bar{x}_i | \Theta^{\Phi}) \tag{8.25}$$

where n is the number of nodes of the graph G^*, while $\bar{x}_i$ is the row vector containing all the d coordinates of the eigenvector matrix.

The eigenvalue model log-likelihood is defined as

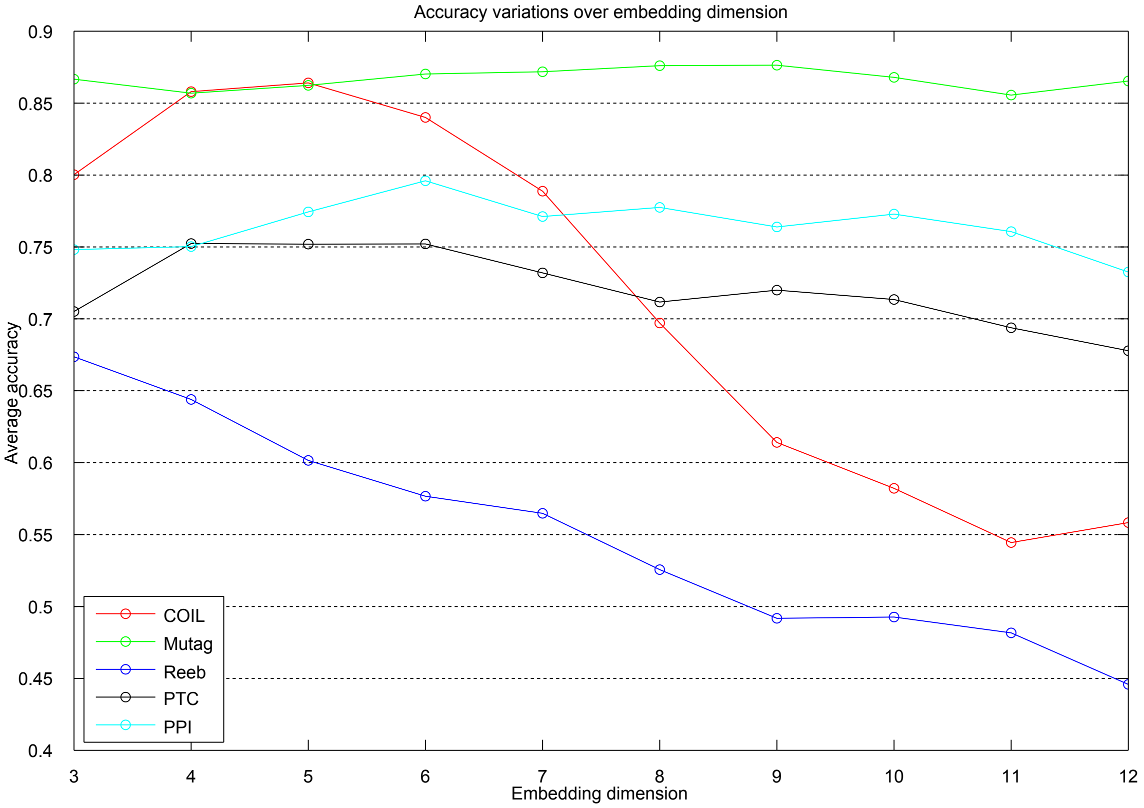


Figure 8.2: Average classification accuracy on all the datasets as we vary the embedding dimension for both the eigenvalues and eigenvectors matrices.

$$\ell_{\mathscr{L}}(\Lambda^* | \mu_i^\Theta, \sigma_i^\Theta) = \prod_{i=1}^{d} P(\lambda_i) = \sum_{i=1}^{d} \log P(\lambda_i) \tag{8.26}$$

with μ_i^Θ and σ_i^Θ which are the parameters estimated using 8.22.

Finally, a decision rule is applied in order to predict the membership of a graph to a certain class. In particular, for this work we classify the graphs assigning them to the most probable class (i.e. the class that yields the higher value).

8.3 Experimental Results

We now evaluate the proposed model comparing it with a number of well-known alternative classification methods. More specifically, we compare our structure-based classifier with some popular graph kernels, like the unaligned QJSD kernel [13], the Weisfeiler-Lehman kernel [164], the graphlet kernel [165], the shortest-path kernel [25], and the random walk kernel [96]. Note that for the Weisfeiler-Lehman we set the number of iterations $h = 3$ and we attribute each node with its degree.

Table 8.1: Classification accuracy (± standard error) on unattributed graph datasets. OUR denotes the proposed model. SA QJSD and QJSU denote the Quantum Jensen-Shannon kernel in the aligned [181] and unaligned [13] version, WL is the Weisfeiler-Lehman kernel [164], GR denotes the graphlet kernel computed using all graphlets of size 3 [164], SP is the shortest-path kernel [25], and RW is the random walk kernel [96]. For each classification method and dataset, the best performance is highlighted in bold.

Datasets	PPI	PTC	COIL5	Reeb	MUTAG
OUR	**79.60 ± 0.86**	**76.80 ± 1.52**	**86.41 ± 0.38**	**67.36 ± 1.52**	**87.74 ± 0.47**
QJSD	68.86 ± 1.00	55.78 ± 0.38	69.83 ± 0.22	35.03 ± 0.26	81.00 ± 0.51
SA QJSD	68.56 ± 0.87	57.07 ± 0.34	69.90 ± 0.22	35.78 ± 0.42	82.11 ± 0.30
WL	79.40 ± 0.83	56.86 ± 0.37	29.08 ± 0.57	50.73 ± 0.39	77.94 ± 0.46
GR	51.06 ± 1.00	55.70 ± 0.18	66.49 ± 0.25	22.90 ± 0.36	81.05 ± 0.41
SP	63.25 ± 0.97	56.32 ± 0.28	69.28 ± 0.42	55.85 ± 0.37	83.36 ± 0.52
RW	49.93 ± 0.83	55.78 ± 0.07	11.83 ± 0.17	15.98 ± 0.42	79.61 ± 0.64

The experiments were run on the following datasets: the **PPI** dataset, which consists of protein-protein interaction (PPIs) networks related to histidine kinase [90] (40 PPIs from *Acidovorax avenae* and 46 PPIs from *Acidobacteria*). The **PTC** (The Predictive Toxicology Challenge) dataset, which records the carcinogenicity of several hundred chemical compounds for male rats (MR), female rats (FR), male mice (MM) and female mice (FM) [115] (here we use the 344 graphs in the MR class). 3) The **COIL** dataset, which consists of 5 objects from [135], each with 72 views obtained from equally spaced viewing directions, where for each view a graph was built by triangulating the extracted Harris corner points. The **Reeb** dataset, which consists of a set of adjacency matrices associated to the computation of reeb graphs of 3D shapes [19]. Finally, the **Mutag** (Mutagenicity) dataset, which consists of graphs representing 188 chemical compounds, and aims to predict whether each compound possesses mutagenicity [164]. Since the vertices and edges of each compound are labeled with a real number, we transform these graphs into unweighted graphs.

We use a binary C-SVM to test the efficacy of the kernels. We perform 10-fold cross validation, where for each sample we independently tune the value of C, the SVM regularizer constant, by considering the training data from that sample. The process is averaged over 100 random partitions of the data, and the results are reported in terms of average accuracy ± standard error. We use a similar approach for the cross validation of our method. We perform a 10-fold cross validation over the datasets, using the proposed model. We tested our method using different numbers of eigenvectors and eigenvalues, which can be seen as one of our free parameter. Furthermore, we tested the model with different levels of sub-sampling, that is, we sub-sampled all the graphs of the datasets (both training and test set) and apply our classification method to it.

Fig. 8.2 shows the average classification accuracy (± standard error) on all the datasets as we vary the number of eigenvectors used. As you can see, every dataset behave differently based on the number of eigenvectors involved. In particular, for the COIL5 dataset,

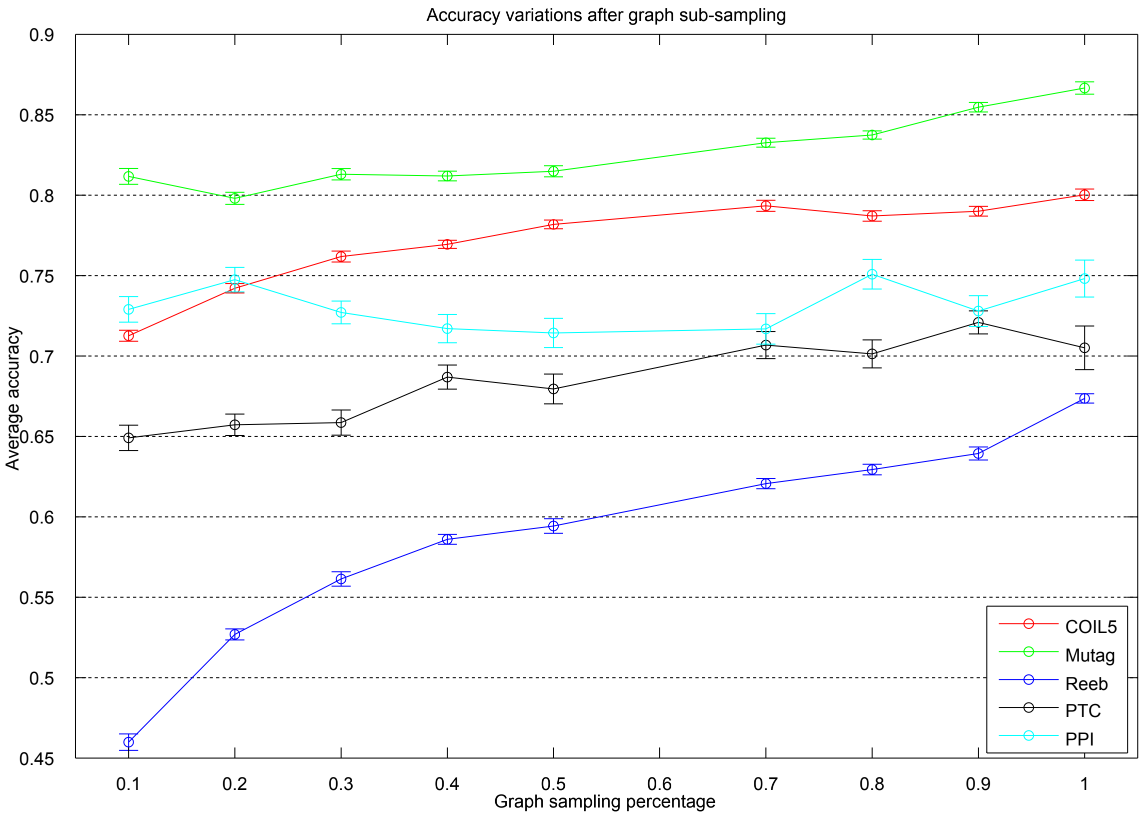


Figure 8.3: Average classification accuracy (with the interval segment representing the ± standard error) on all the datasets as we vary the percentage of sub-sampling applied to each graph of each dataset.

the use of more eigenvectors yields worst results, which means that the eigenvectors associated to the smaller non-zero eigenvalues of the spectra, models the classes better, while the subsequent ones just add noise to our representation. In the contrary, the Mutag dataset benefits from increasing the number of eigenvectors (and eigenvalues) involved in the creation of the class model.

Fig.8.3 shows the average classification accuracy (± standard error) on all the datasets as we vary the percentage of sub-sampling applied to each graph of each dataset. In particular, the first accuracy measure corresponds to the application of our model on the spectral decomposition of the graphs where only 10% of the nodes were preserved. All the datasets (except for Mutag and PPI datasets) reach worse levels of accuracy with a lower number of nodes, meaning that the structural information given by each node of the model is useful for classification purpose. Conversely, the other datasets are more robust to sub-sampling.

Table 9.1 shows the average classification accuracy (± standard error) of the different kernels and of our method on the selected datasets. The proposed model yields an increase of the performance with respect to the confronted graph kernels on all the used datasets. In

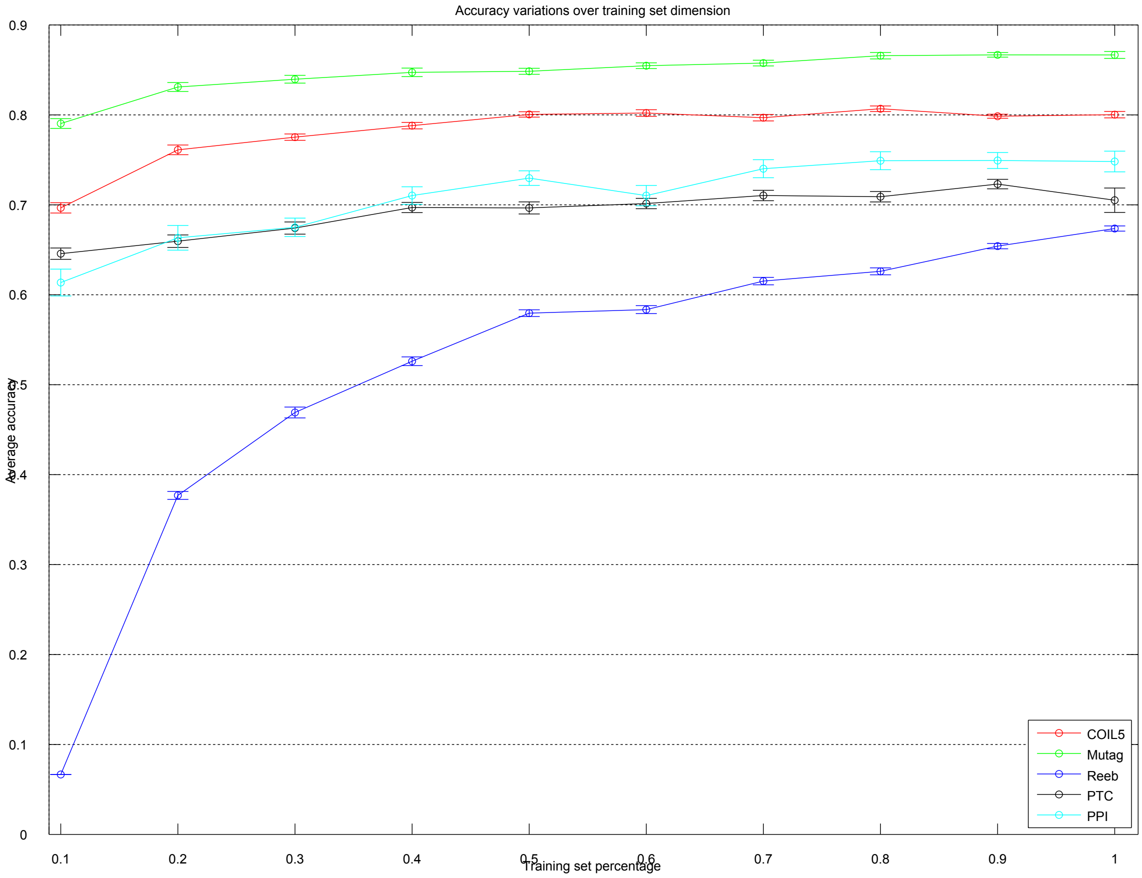


Figure 8.4: Average classification accuracy (with the interval segment representing the ± standard error) on all the datasets as we vary the percentage of graph of the training set used to build the model.

particular, we obtained similar results with respect to the Weisfeiler-Lehman graph kernel on the PPI dataset. This is probably due to the use of the node labels in order to mitigate the localization problem and thus improving node localization in the evaluation process. Even though our model does not exploit node attributes, we were able to outperform all the kernels on all the other datasets.

8.4 Conclusion

In this chapter we have introduced a novel model of structural representation based on a spectral description of graphs which lifts the one-to-one node-correspondence assumption and is strongly rooted in a statistical learning framework. We showed how the defined separate models for eigenvalues and eigenvectors could be used within a statistical framework to address the graphs classification task. We tested the defined method against a number of alternative graph kernels and we showed its effectiveness in a number of structural classification tasks.

9

Transitive State Alignment for the Quantum Jensen-Shannon Kernel

Kernel methods provide a convenient way to apply a wide range of learning techniques to complex and structured data by shifting the representational problem from one of finding an embedding of the data to that of defining a positive semidefinite kernel.

One problem with the most widely used kernels is that they neglect the locational information within the structures, resulting in less discrimination. Correspondence-based kernels, on the other hand, are in general more discriminating, at the cost of sacrificing positive-definiteness due to their inability to guarantee transitivity of the correspondences between multiple graphs.

In this chapter we generalize a recent structural kernel based on the Jensen-Shannon divergence between quantum walks over the structures by introducing a novel alignment step which rather than permuting the nodes of the structures, aligns the quantum states of their walks. This results in a novel kernel that maintains localization within the structures, but still guarantees positive definiteness.

Similarly to the considerations made for chapter 5 and 6, the main difference between the approach proposed in this chapter and the one introduced in chapter 8 is the requirement of a common space in which all the graphs lie. Indeed, the method proposed in the previous chapter avoid it completely through the definition of an alignment invariant statistical model. On the other hand, in the method presented in this chapter we drop the usual structure-permutation approach in favour of the alignment step introduced in section 9.3.

9.1 Introduction

Structural representations have become increasingly popular due to their representational power. However, the descriptiveness comes at the cost of an increased difficulty in applying standard machine learning and pattern recognition techniques to them, as these usually require data that reside in a vector space. The famous kernel trick allows the focus to be shifted from the vectorial representation of data, which now becomes implicit, to a similarity representation. This allows standard learning techniques to be applied to structural data for which no obvious vectorial representation exists.

One of the most influential works on structural kernels was the generic R-convolution kernel proposed by Haussler [79]. Here graph kernels are computed by comparing the similarity of the basic elements for a given decomposition of the two graphs. Depending on the decomposition chosen, we obtain different kernels. Most R-convolution kernels simply count the number of isomorphic substructures in the two graphs. For example, Kashima *et al.* [96] compute the kernel by decomposing the graph into random walks, while Borgwardt *et al.* [25] have proposed a kernel based on shortest paths. Here, the similarity is determined by counting the numbers of pairs of shortest paths of the same length in a pair of graphs. Shervashidze *et al.* [164] have developed a subtree kernel on subtrees of limited size, where the number of subtrees common between two graphs is computed efficiently using the Weisfeiler-Lehman graph invariant.

One drawback of these kernels is that they neglect the locational information for the substructures in a graph. In other words, the similarity does not depend on the relationships between substructures. As a consequence, these kernels cannot establish reliable structural correspondences between the substructures. This limits the precision of the resulting similarity measure. To overcome this problem, Fröhlich *et al.* [62] introduced alternative optimal assignment kernels. Here each pair of structures is aligned before comparison. However, the introduction of the alignment step results in a kernel that is not positive definite in general [189]. The problem results from the fact that alignments are not in general transitive. In other words, if σ is the vertex-alignment between graph A and graph B, and π is the alignment between graph B and graph C, in general we cannot guarantee that the alignment between graph A and graph C is $\pi \circ \sigma$. On the other hand, when the alignments are transitive, there is a common simultaneous alignment of all the graphs. Under this alignment, the optimal assignment kernel is simply the sum over all the vertex/edge kernels, which is positive definite since it is the sum of separate positive definite kernels. While lacking positive definiteness the optimal assignment kernels cannot be guaranteed to represent an implicit embedding into a Hilbert space, they have nonetheless been proven to be very effective in classifying structures.

There has recently been an increasing interest in quantum computing because of the potential speed-ups over classical algorithms. Examples include Grover's polynomially faster search algorithm [75] and Shor's exponentially faster factorization algorithm [168].

Recently Bai *et al.* [13] introduced a graph kernel based on a Quantum analogue of the Jensen-Shannon divergence between average states of continuous-time quantum walks over the structures to be analysed. Being based on the divergence which is conjectured to be negative definite [28], the kernel is thought to be positive definite. However it lacks permutational invariance, thus different permutations of the same graphs result in different values of the kernel. This fact, while mitigated by the long range interactions reinforced by the interference patterns in quantum walks, is a rather undesirable property for a structural kernel. For this reason in this chapter we modify the kernel by adding a novel alignment step that rather than permuting the nodes of the structures, aligns the quantum states of the walks. This results in a novel kernel that is permutationally invariant and maintains similar localization property of the alignment kernels [62]. Further, we prove that the alignment transformations between multiple structures are transitive and that, for this particular alignment, the kernel is always positive definite.

9.2 Quantum Mechanical Background

Quantum walks are the quantum analogue of classical random walks. Given a graph $G = (V, E)$, the state space of the continuous-time quantum walk defined on G is the set of the vertices V of the graph. Unlike the classical case, where the evolution of the walk is governed by a stochastic matrix, in the quantum case the dynamics of the walker is governed by a complex unitary matrix *i.e.* , a matrix that multiplied by its conjugate transpose yields the identity matrix. As a consequence, the evolution of the quantum walk is reversible, which implies that quantum walks are non-ergodic and do not possess a limiting distribution. See [99] for an overview of the properties of quantum walks. Using Dirac notation, we denote the basis state corresponding to the walk being at vertex $u \in V$ as $|u\rangle$. Here a ket $|u\rangle$ is simply representing a unit vector associated with state u, for example, if we use the vertices as the basis for the space, $|u\rangle = \mathbf{e}_u$, i.e., the u-th vector in the canonical basis. Conversely, a bra $\langle u|$ is the co-vector obtained taking the conjugate-transpose of $|u\rangle$. A general state of the walk is a complex linear combination of the basis states, such that the state of the walk at time t is defined as $|\psi_t\rangle = \sum_{u \in V} \alpha_u(t) |u\rangle$ where the amplitude $\alpha_u(t) \in \mathbb{C}$ and $|\psi_t\rangle \in \mathbb{C}^{|V|}$ are both complex.

At each point in time the probability of the walker being at a particular vertex of the graph is given by the square of the norm of the amplitude of the relative state. More formally, let X^t be a random variable giving the location of the walker at time t. Then the probability of the walker being at the vertex u at time t is given by $\Pr(X^t = u) = \alpha_u(t)\alpha^*_u(t)$ where $\alpha^*_u(t)$ is the complex conjugate of $\alpha_u(t)$. Moreover, in a closed system $\sum_{u \in V} \alpha_u(t)\alpha^*_u(t) = 1$.

The evolution of the walk over graph $G = (V, E)$ is governed by Schrödinger equation, where we take the Hamiltonian of the system to be the graph Laplacian L, which, eliminating scaling constants, yields

$$\frac{d}{dt}|\psi_t\rangle = -iL|\psi_t\rangle \tag{9.1}$$

Given an initial state $|\psi_0\rangle$, we can solve Equation 9.1 to determine the state vector at time t $|\psi_t\rangle = e^{-iLt}|\psi_0\rangle = \Phi e^{-i\Lambda t}\Phi^\top |\psi_0\rangle$, where $L = \Phi\Lambda\Phi^\dagger$ is the spectral decomposition of the Laplacian matrix.

While a pure state can be naturally described using a single ket vector, in general a quantum system can be in a *mixed state*, i.e., a statistical ensemble of pure quantum states $|\psi_i\rangle$, each with probability p_i. The *density operator* (or *density matrix*) of such a system is defined as

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|. \tag{9.2}$$

Density operators are positive unit-trace matrices directly linked with the observables of the (mixed) quantum system. Let O be an observable, i.e., an Hermitian operator acting on the quantum states and providing a measurement. The expected value of the measurement O over a mixed state can be calculated from the density matrix ρ: $\langle O \rangle = \mathrm{Tr}\left(\rho O\right)$, where Tr is the trace operator.

The Von Neumann entropy of a density operator ρ is

$$H_N(\rho) = -Tr(\rho \log \rho) = -\sum_j \lambda_j \log \lambda_j, \tag{9.3}$$

where the λ_js are the eigenvalues of ρ. With the Von Neumann entropy to hand, we can define the quantum Jensen-Shannon divergence between two density operators ρ and σ as

$$D_{\mathrm{JS}}(\rho,\sigma) = H_N\left(\frac{\rho+\sigma}{2}\right) - \frac{1}{2}H_N(\rho) - \frac{1}{2}H_N(\sigma) \tag{9.4}$$

This quantity is symmetric, bounded between 0 and 1, and negative definite for pure states and is conjectured with ample experimental evidence to be negative definite for all states [28].

Finally, for a graph $G(V,E)$, let $\left|\psi_t\right\rangle$ denote the state corresponding to a continuous-time quantum walk that has evolved from time $t = 0$ to time $t = T$. We define the time-averaged density matrix $\rho_G^{\dagger}$ for $G(V,E)$

$$\rho_G^{\dagger} = \frac{1}{T}\int_0^T \left|\psi_t\right\rangle\left\langle\psi_t\right| \mathrm{d}t. \tag{9.5}$$

Let ϕ_{xy} denote the (xy)-th element of the matrix of eigenvectors Φ of the Laplacian. Following [154], we compute the (r,c)-th element of ρ_T as follows:

$$\rho_G^{\dagger}(r,c) = \sum_{k=1}^{n}\sum_{y=1}^{n} \phi_{rk}\phi_{cy}\bar{\psi}_k\bar{\psi}_y \frac{1}{T}\int_0^{\dagger} e^{i(\lambda_y-\lambda_k)t}\,\mathrm{d}t. \tag{9.6}$$

If we let $T \to \infty$, Eq.(9.6) further simplifies to

$$\rho_G^{\infty} = \sum_{\lambda\in\tilde{\Lambda}} P_\lambda \rho_0 P_\lambda^{\top} \tag{9.7}$$

where $\tilde{\Lambda}$ is the set of distinct eigenvalues of the Laplacian matrix L and P_λ is the orthogonal projector onto the eigenspace associated with λ.

9.3 State-Aligned QJSD Kernel

In [13] the Bai *et al.*defined a kernel based on the Quantum Jensen Shannon divergence between two continuous-time quantum walks between the graphs. The QJSD kernel was defined as

$$K_{\mathrm{QJSD}}(G_1,G_2) = \exp\left(-\beta D_{\mathrm{JS}}(\rho_1,\rho_2)\right) \tag{9.8}$$

where ρ_1 and ρ_2 are the time-averaged density matrices associated with the quantum walks over G_1 and G_2 respectively, and β is a decay parameter of the kernel. The walks are initialized in the starting state

$$|\Psi_0\rangle = \sum_{u \in V} \sqrt{\frac{d_u}{\sum_{v \in V} d_v}} \, |u\rangle \, . \tag{9.9}$$

The kernel is positive definite under the conjecture that the quantum Jensen-Shannon divergence is negative definite for all states, and exhibited good performance on several graph classification tasks, but its value is dependent on the order under which the nodes are presented due to the mixing term $\frac{\rho+\sigma}{2}$ in the definition of the divergence.

In this chapter we solve the permutational invariance problem of the QJSD kernel by adding an alignment step to the computation of the kernel. In contrast to alternative alignment kernels such as [62], the alignment is not performed over the node permutations Σ_n of the graphs. Rather it is performed over the quantum basis under which the walker can be observed. In classical random walks the nodes of the graph provide a preferred basis for observation as the walker cannot be simultaneously on multiple nodes, thus the only available degree of freedom is in the choice of an order within the basis vectors, *i.e.* , the observation basis is fully defined up to a permutation $\pi \in \Sigma_n$. This is in stark contrast with quantum mechanics where, due to quantum superposition, prior to observation a quantum walker can be simultaneously at multiple nodes, and the observation itself can be performed under any quantum superposition of states. This means that any orthogonal basis is valid for observation and, thus, the basis is defined up to a unitary transformation $O \in U(n)$, where $U(n)$ is the Unitary group over $\mathbb{C}^n$.

Following this property, we define a *State-aligned* QJSD kernel as

$$\begin{aligned} K_{\text{SAQJSD}}(G_1, G_2) &= \max_{O \in U(n)} \exp\left(-\beta D_{\text{JS}}(\rho_1, O\rho_2 O^\dagger)\right) \\ &= \exp\left(-\beta \min_{O \in U(n)} D_{\text{JS}}(\rho_1, O\rho_2 O^\dagger)\right) \end{aligned} \tag{9.10}$$

In the following we will prove some important properties of the state-aligned kernel. Namely we will give a closed form solution to the alignment, prove that the optimal transformation are transitive, and prove that the resulting kernel is positive definite without making use of the negative-definiteness conjecture for the quantum Jensen-Shannon divergence.

9.3.1 Properties of the State-Aligned QJSD Kernel

We start by enunciating a theorem relating the optimal state-alignment to the eigenvectors of the density matrices. For a proof of this result see [179].

Theorem 9.3.1. *Let $\rho 1 = \Phi_1 \Lambda_1 \Phi_1^\dagger$ and $\rho 2 = \Phi_2 \Lambda_2 \Phi_2^\dagger$ be the singular value decompositions of ρ_1 and ρ_2 respectively, with the eigenvalues in descending order in both Λ_1 and Λ_2, then the global minimum of $\bar{H}_N(O)$ is attained by $O^* = \Phi_1 \Phi_2^\dagger$.*

This theorem tells us how to efficiently compute the state alignment. Further, this transformation aligns the eigenvectors resulting in a mixed density matrix $\frac{1}{2}(\rho_1+O^*\rho_2 O^{*\dagger})$ with eigenvalues $\frac{1}{2}(\lambda_1+\mu_i)$ where $\lambda_1,\dots,\lambda_n$ and $\mu_1,\dots,\mu_n$ are the eigenvalues of ρ_1 and ρ_2 respectively taken in descending order with their multiplicity. This means that the aligned Jensen Shannon divergence only needs the eigenvalues of ρ_1 and ρ_2 to be computed, in fact:

$$\min_{O\in U_n} D_{\text{JS}}(\rho_1, O\rho_2 O^\dagger) = \sum_{j=1}^{n} -\frac{\lambda_j+\mu_j}{2}\log\left(\frac{\lambda_j+\mu_j}{2}\right) + \frac{\lambda_j\log(\lambda_j)+\mu_j\log(\mu_j)}{2}. \tag{9.11}$$

This reduces the computational complexity of computing the kernel for all times at which the mixed density matrix is computed, as we do not need to perform the eigendecomposition of the mixed matrix $\frac{1}{2}(\rho_1+\rho_2)$ for each pair of graphs in the kernel. Rather, we only need to compute the eigenvalues (not the eigenvectors) of all the density matrices beforehand. The resulting complexity for the whole kernel computation is $O(Nn^3+N^2n)$ where N is the number of graphs and n their (maximum) size. In contrast, the QJSD kernel has complexity $O(Nn^3+N^2n^2)$ due to the eigenvalue computation for each pair of graphs.

Further, in the case of the infinite-time mixing matrix, we can significantly reduce the computational burden of computing the eigenvalues of the density matrix, by using a result presented in [154]. There it was proven that the infinite-time mixing matrix commuted with the graph Laplacian. As a consequence, ρ^∞ expressed in the eigenbasis of the Laplacian, is a block diagonal matrix where blocks correspond to eigenspaces associated with a single eigenvalue. Let $L=\Phi\Lambda\Phi^\dagger$, be the spectral decomposition of the graph Laplacian, we denote with Φ_j the matrix formed with the columns of Φ corresponding to the eigenvectors associated with the j-th distinct eigenvalue. The j-th diagonal block of ρ^∞ expressed in the eigenbasis Φ is $\Phi_j^\dagger\rho^\infty\Phi_j$. using Eq. (9.7) and recalling that $P_j=\Phi_j^\dagger\Phi_j$, we have

$$\Phi_j^\dagger\rho^\infty\Phi_j = \Phi_j^\dagger\rho^\infty\Phi_j = \Phi_j^\dagger\rho^0\Phi_j = \Phi_j^\dagger\left|\psi_0\right\rangle\left\langle\psi_0\right|\Phi_j = \left|\Phi_j^\dagger\psi_0\right\rangle\left\langle\Phi_j^\dagger\psi_0\right| \tag{9.12}$$

which is a rank 1 matrix with a single non-zero eigenvalue $\lambda_j=\|\Phi_j^\dagger\psi_0\|^2$. Hence, once the singular value decomposition of the graph's Laplacian is to hand, we can compute the eigenvalues of the infinite-time mixing matrix directly, without the need for an additional decomposition. This makes the infinite-time kernel particularly efficient to compute.

It is worth noting that as the graph Laplacian has eigenvalues with higher multiplicity the infinite-time mixing matrix has more zero eigenvalues resulting in a lower Von Neumann entropy. This is particularly interesting since higher multiplicities of the eigenvalues is associated with the presence of symmetries in the graph [133] which, in turn, have been used to characterize the entropy of the structure [132].

We can now prove the following properties for the state-aligned kernel

Theorem 9.3.2. *The Unitary transformations minimizing the quantum Jensen Shannon divergence between pairs of density matrices in a set are transitive,* i.e. *let*

$$O_{i,j} = \operatorname*{argmin}_{O\in U(n)} D_{\text{JS}}(\rho_i, O\rho_j O^\dagger)$$

with $i, j \in \{1,2,3\}$, then

$$D_{\mathrm{JS}}\big(\rho_1, O_{1,2}O_{2,3}\rho_3 O_{2,3}^{\dagger}O_{1,2}^{\dagger}\big) = D_{\mathrm{JS}}\big(\rho_1, O_{1,3}\rho_3 O_{1,3}^{\dagger}\big)$$

Proof The optimal transformation between two density matrices is completely determined by the relation $O_{1,2}^* = \Phi_1\Phi_2^\dagger$ up to a change of sign of the eigenvalue and a change of base for each eignespace associated with an eigenvalue with multiplicity greater than one. In any case these changes do not affect the value of the divergence. However,

$$O_{1,2}^* O_{2,3}^* = \Phi_1\Phi_2^\dagger\Phi_2\Phi_3^\dagger = \Phi_1\Phi_3^\dagger = O_{1,3}^* \tag{9.13}$$

QED.

Theorem 9.3.3. *The quantum aligned QJSD kernel is positive definite.*

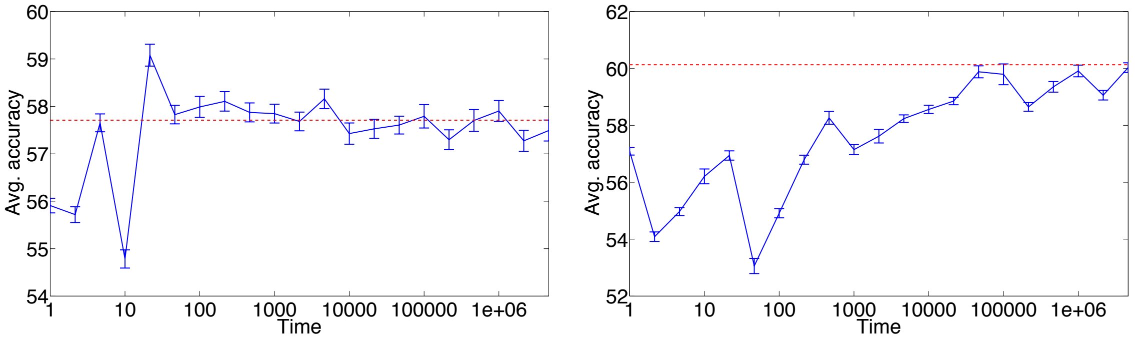
Proof As a consequence of the previous theorems, the value of the quantum Jensen Shannon divergence of the optimally aligned density matrices is equal to the normal Jensen Shannon divergence of the sorted eigenvalues of the density matrices (taken as probability distributions). Since the Jensen Shannon divergence is proven to be negative definite [28] the state-aligned QJSD kernel, being an exponentiation of a negative definite kernel is positive definite [104]. QED.

9.4 Experimental Results

We now evaluate the performance of the State-Aligned (SA) QJSD kernel and we compare it with a number of well-known alternative graph kernels. More specifically, we compare our kernel with the unaligned QJSD kernel [13], the Weisfeiler-Lehman kernel [164], the graphlet kernel [165], the shortest-path kernel [25], and the random walk kernel [96]. Note that for the Weisfeiler-Lehman we set the number of iterations $h = 3$ and we attribute each node with its degree.

We run our experiments on the following datasets: 1) The **PPI** dataset, which consists of protein-protein interaction (PPIs) networks related to histidine kinase [90] (40 PPIs from *Acidovorax avenae* and 46 PPIs from *Acidobacteria*). 2) The **PTC** (The Predictive Toxicology Challenge) dataset, which records the carcinogenicity of several hundred chemical compounds for male rats (MR), female rats (FR), male mice (MM) and female mice (FM) [115] (here we use the 344 graphs in the MR class). 3) The **COIL** dataset, which consists of 5 objects from [135], each with 72 views obtained from equally spaced viewing directions, where for each view a graph was built by triangulating the extracted Harris corner points. 4) The **Reeb** dataset, which consists of a set of adjacency matrices associated to the computation of reeb graphs of 3D shapes [19].

We use a binary C-SVM to test the efficacy of the kernels. We perform 10-fold cross validation, where for each sample we independently tune the value of C, the SVM regularizer constant, by considering the training data from that sample. The process is averaged



(a) Sub-optimal β (b) Optimal β

Figure 9.1: The average classification accuracy as the time parameter of the continuous-time quantum walk varies, for an optimal (left) and sub-optimal value of the decay factor β.

over 100 random partitions of the data, and the results are reported in terms of average accuracy ± standard error.

Fig. 9.1 shows the value of the average classification accuracy (± standard error) on the PTC dataset as we let the time parameter of the continuous-time quantum walk vary. Here the red horizontal line denotes the average accuracy for $T \to \infty$. Note that in Fig. 9.1(a) we set the decay parameter β of the kernel to a sub-optimal value, while in Fig. 9.1(b) we set it to its optimal value, *i.e.* , the value that results in the best classification accuracy. The plot shows that when β is sub-optimal the choice of the time parameter has a clear influence on the performance of our kernel. In fact, we see that the average accuracy reaches a maximum before stabilizing around the asymptotic value. On the other hand, when β is optimized the best classification performance is achieved when $T \to \infty$. Moreover, in the latter case the average classification accuracy is higher than that recorded for smaller values of T and a sub-optimal β.

Table 9.1: Classification accuracy (± standard error) on unattributed graph datasets. SA QJSD and QJSD denote the proposed kernel and its original unaligned version, respectively, WL is the Weisfeiler-Lehman kernel [164], GR denotes the graphlet kernel computed using all graphlets of size 3 [165], SP is the shortest-path kernel [25], and RW is the random walk kernel [96]. For each kernel and dataset, the best performing kernel is highlighted in bold.

Kernel	PPI	PTC	COIL	Reeb
SA QJSD	75.69 ± 0.85	$\mathbf{60.13 \pm 0.51}$	67.84 ± 0.15	38.50 ± 0.26
QJSD	69.12 ± 1.01	56.06 ± 0.45	$\mathbf{69.90 \pm 0.22}$	35.78 ± 0.42
WL	$\mathbf{79.40 \pm 0.96}$	56.95 ± 0.31	29.00 ± 0.57	50.53 ± 0.41
GR	51.94 ± 0.97	55.22 ± 0.19	66.46 ± 0.44	22.80 ± 0.36
SP	63.31 ± 0.80	56.51 ± 0.36	69.68 ± 0.36	$\mathbf{55.93 \pm 0.36}$
RW	50.37 ± 0.78	55.68 ± 0.14	12.18 ± 0.21	16.47 ± 0.43

Table 9.2: Runtime comparison on the four graph datasets.

Kernel	PPI	PTC	COIL	Reeb
SA QJSD	3.68”	13.30”	33.66”	15.35”
QJSD	126.09”	35.28”	2371.17”	544”
WL	4.10”	3.79”	22.52”	11.86”
GR	2.51”	0.73”	9.25”	1.98”
SP	3.85”	0.74”	19.13”	6.15”
RW	11.58”	231”	294.24”	757.67”

Table 9.1 shows the average classification accuracy (± standard error) of the different kernels on the selected datasets. As expected, we see that the state alignment almost invariably yields an increase of the performance with respect to the standard QJSD kernel. Indeed, the localization property of the kernel that results from the quantum state alignment leads to a better discrimination, and thus a higher classification accuracy. Moreover, while the QJSD kernel has not been proven to be positive definite, as the quantum Jensen-Shannon divergence has only been experimentally shown to be negative definite for mixed states, our kernel is indeed positive definite, as proved in the previous Section.

Finally, Table 9.2 shows the runtimes of the different kernels on the four graph datasets. Note that in terms of runtime the SA QJSD kernel easily outperforms the other spectral methods, *i.e.* , the QJSD kernel and the random walk kernel, and it is still competitive when compared with the remaining kernels.

With respect to the other kernels, the SA QJSD kernel achieves the best accuracy on the PTC dataset, and it remains competitive with the best performing ones on the PPI and COIL dataset. On the Reeb dataset, on the other hand, the shortest-path kernel and the Weisfeiler-Lehman kernel outperform our kernel and the remaining ones. Note also that the Weisfeiler-Lehman mitigates the localization problem by making use of the node labels and thus improving node localization in the evaluation of the kernel. On the other hand, our kernel does not take node attributes into account.

9.5 Conclusions

In this chapter we have generalized a recent structural kernel based on the Jensen-Shannon divergence between quantum walks over the graph by introducing a novel alignment step which, rather than permuting the nodes of the structures, aligns the quantum states of their walks. We proved that the resulting kernel maintains the localization within the structures, but still guarantees positive definiteness. We tested our kernel against a number of alternative graph kernels and we showed its effectiveness in a number of structural classification tasks.

10

Transitive Assignment Kernels for Structural Classification

Kernel methods provide a convenient way to apply a wide range of learning techniques to complex and structured data by shifting the representational problem from one of finding an embedding of the data to that of defining a positive semi-definite kernel. One problem with the most widely used kernels is that they neglect the locational information within the structures, resulting in less discrimination. Correspondence-based kernels, on the other hand, are in general more discriminating, at the cost of sacrificing positive-definiteness due to their inability to guarantee transitivity of the correspondences between multiple graphs.

In this chapter we adopt a general framework for the projection of (relaxed) correspondences onto the space of transitive correspondences, thus transforming any given matching algorithm onto a transitive multi-graph matching approach.

The resulting transitive correspondences can then be used to provide a kernel that both maintains locational information and is guaranteed to be positive-definite. Experimental evaluation validates the effectiveness of the kernel for several structural classification tasks.

10.1 Introduction

Graph-based representations have proven invaluable in several application domains due to their ability to characterize complex ensembles in terms of parts and binary relations. Concrete examples include the use of graphs to represent shapes [169], metabolic networks [91], protein structure [85], and road maps [95]. However, the expressive power of graphs comes at the cost of a reduced pattern analysis toolset available to the practitioner. In fact, our ability to analyse data abstracted in terms of graphs is severely limited by the restrictions posed by standard feature-based paradigm dominating pattern recognition techniques, which require data to be representable in a vectorial form.

There are two reasons why graphs are not easily reduced to a vectorial form. First, unlike the components of a vector, there is no canonical ordering for the nodes in a graph, requiring correspondences to be established as a prerequisite for analysis. Second, the variation in the graphs of a particular class may manifest itself as subtle changes in structure. Hence, even if the nodes or the edges of a graph could be encoded in a vectorial manner, the vectors would be of variable length, thus residing in different spaces.

The first 30 years of research in structural pattern recognition have been mostly concerned with the solution of the graph matching problem as the fundamental means of assessing structural similarity [46]. With the correspondences at hand, similarity-based recognition and classification techniques can be used. Alternatively, graphs can be embedded in a low-dimensional pattern space using either multidimensional scaling or non-linear manifold leaning techniques.

Another alternative is to extract feature vectors from the graphs providing a pattern-space representation by extracting structural or topological features. For example, spectral features extracted from the singular value decomposition of the graph Laplacian have been proven effective [67,123,181,194]. For an overall survey about the current state-of-the-art in the graph matching problem, refers to the work by Livi and Lizzi [119].

10.1.1 Graph Kernels

The famous kernel trick [160] has shifted the problem from the vectorial representation of data, which now becomes implicit, to a similarity representation. This has allowed standard learning techniques to be applied to data for which no easy vectorial representation exists. Once we define a positive semi-definite kernel $k: X \times X \to \mathbb{R}$ on a set X, there exists a map $\phi: X \to \mathcal{H}$ into a Hilbert space $\mathcal{H}$, such that $k(x,y) = \phi(x)^\top \phi(y)$ for all $x, y \in X$. Also, given the kernel value between $\phi(x)$ and $\phi(y)$ one can easily compute the distance between them by noting that $||\phi(x)-\phi(y)||^2 = \phi(x)^\top\phi(x)+\phi(y)^\top\phi(y)-2\phi(x)^\top\phi(y)$. Thus, any algorithm that can be formulated in terms of dot products between the input vectors can be applied to the implicitly mapped data points through the direct substitution of the kernel for the dot product. For this reason, in recent years the structural pattern recognition field has witnessed an increasing interest in graph kernels. However, due to the rich expressiveness of graphs, this task has also proven to be difficult, with the problem of defining *complete* kernels, *i.e.*, ones where the implicit map ϕ is injective, sharing the same computational complexity of the graph isomorphism problem [65].

One of the most influential works on structural kernels is the definition of the class of R-convolution kernel proposed by Haussler [79]. Here graph kernels are computed by comparing the similarity of the basic elements for a given decomposition of the two graphs. Depending on the decomposition chosen, we obtain different kernels. Most R-convolution kernels simply count the number of isomorphic substructures in the two graphs. For example, Kashima *et al.* [96] compute the kernel by decomposing the graph into random walks, while Borgwardt et al. [25] have proposed a kernel based on shortest paths. Here, the similarity is determined by counting the numbers of pairs of shortest paths of the same length in a pair of graphs. Shervashidze *et al.* [164] have developed a subtree kernel on subtrees of limited size, where the number of subtrees common between two graphs is computed efficiently using the Weisfeiler-Lehman graph invariant. Recently, Kriege et al [106] proposed that a kernel based on the number of isomorphisms between pairs of subgraphs, while Neumann et al. [138] have introduced the concept of propagation kernels to handle partially labelled graphs through the use of continuous-valued vertex attributes.

10.1.2 Assignment Kernels

One drawback of these kernels is that they neglect the locational information for the substructures in a graph. In other words, the similarity does not depend on the relationships between substructures. As a consequence, these kernels cannot establish reliable structural correspondences. This limits the precision of the resulting similarity measure. Ong *et al.* [139] introduce several kernel methods about indefinite kernel for general structures, while Geibel *et al.* [86, 87] gives a solution to deal with not positive semidefinited kernel based on Schur-Hadamard Inner Product applied on graphs. Further, Schietgat et al. [159] propose a graph metric which is based on the maximum common subgraph, while in [131] the authors exploit indefinte maximum common subgraph kernels using the potential of support vector machine for indefinite matrices, extending the work proposed by Hochreiter and Obermayer [81]. Another interesting solution described by Fröhlich et al. [63] presents alternative optimal assignment kernels. Here each pair of structures is aligned before comparison. Another example of alignment-based kernels are the edit-distance-based kernels introduced by Neuhaus and Bunke [136]. Here the alignments obtained from graph-edit distance are used to guide random walks on the structures being compared.

Unfortunately, the introduction of the alignment step results in a kernel that is not positive definite in general [189]. The problem results from the fact that alignments are not in general transitive. In other words, if σ is the vertex-alignment between graph A and graph B, and π is the alignment between graph B and graph C, in general we cannot guarantee that the optimal alignment between graph A and graph C is $\pi \circ \sigma$. Lacking positive definiteness the optimal assignment kernels cannot be guaranteed to represent an implicit embedding into a Hilbert space. However, they have proven to be very effective in classifying structures.

10.1.3 Multi-Graph Matching

The problem of estimating a transitive set of correspondences between structures, known as the multi-graph matching problem, has received much less attention by the research community than pairwise matching. One of the earliest work, due to Williams et al. [193], imposes the transitive vertex-matching constraint in a softened Bayesian manner, inducing inference triangles by forming fuzzy compositions of pairwise matching functions. Sole-Ribalta and Serratosa [172] extended the Graduated Assignment algorithm [71] to the multi-graph scenario by raising the assignment matrices associated to pair of graphs to assignment hypercube, or tensors, between all the graphs. For computational efficiency, the hypercube is constructed via sequential local pair matching. More recently, Yan et al. [198, 199] proposed a new framework explicitly extending the Integer Quadratic Programming (IQP) formulation of pairwise matching to the multi-graph matching scenario. The resulting IQP is then solved through alternating optimization approach. Pachauri et al. [143], on the other hand, synchronize a given set of assignments through a spectral relaxation.

10.1.4 Contribution

In this chapter we want to investigate the use of multi-graph matching techniques in the context of graph kernels. By forcing the correspondences between the structures under study to satisfy transitivity, we obtain an alignment kernel that, not only is positive definite, but also makes use of more reliable locational information obtained through the enforcement of global consistency constraints. In fact, when the alignments are transitive, there is a common simultaneous alignment of all the graphs. Under this alignment, the kernel is simply the sum over all the vertex/edge kernels, which is positive definite since it is the sum of separate positive definite kernels.

Here we adopt an approach similar to Pachauri et al. [143] in avoiding the definition of a specific multi-graph matching algorithm. Rather, we project a set of (possibly relaxed) assignments to the set of transitive correspondences. Transformation synchronization techniques such as this have been proven effective in several fields due to their effectiveness, their ability to leverage the state of the art in pairwise transformation estimation, and their computational efficiency [77, 184]. The proposed synchronization technique shares some similarities with [143], but we adopt a different relaxation scheme that does not result in a generalized low rank Rayleigh problem, but can however be solved with a projected power method, avoiding the requirement for an eigendecomposition of the matching tensor.

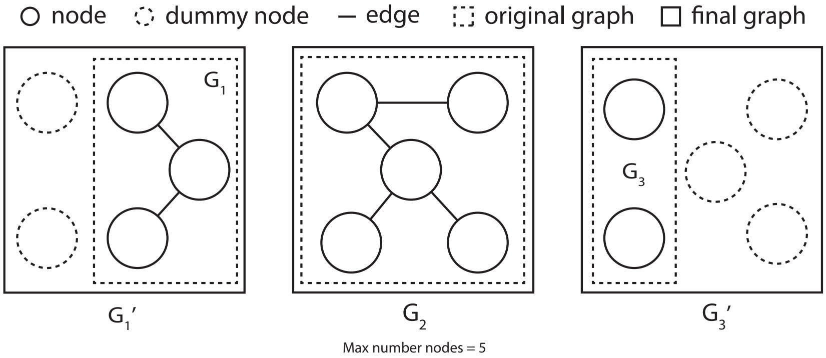
10.2 Projection on the Transitive Alignment Space

Let $G_1, G_2, \ldots, G_N$ be graphs and let P_{ij} for $i, j = 1, \ldots, N$ be a matrix matching vertices in G_i to vertices in G_j obtained with any pairwise matching algorithm. Here we assume that $(P_{ij})_{vw}$ expresses a likelihood that node v in G_i is matched to node w in G_j, but is not required to represent a permutation, and can be in a relaxed space such as the space of doubly stochastic matrices. Our goal is to find a set of permutation matrices $\overline{P}_{ij} \in \Sigma_n$ (with Σ_n the permutation space and $i, j = 1, \ldots, N$) as similar as possible, in the least square sense, to P_{ij}, which satisfy the transitivity constraint. Namely,

$$\forall i, j, k = 1, \ldots, N \qquad \overline{P}_{ij}\overline{P}_{jk} = \overline{P}_{ik}. \tag{10.1}$$

In order to do this first we force the graphs all to the same size n by padding them with dummy disconnected nodes to the maximum size of all the graphs of the set (see figure 10.1).

Figure 10.1: Graphical example about the refinement task of our datasets. In the figure, the set is composed of three graphs G_1, G_2 and G_3. The maximum number of nodes is 5 (the second graph), hence we add two disconnected nodes in G_1 and three in G_3 in order to obtain respectively the extended graphs G_1' and G_2'. The final dataset with the same number of nodes $n = 5$ is composed by the graphs set G_1', G_2 and G_3'.



Once the graphs are all of the same size, we can enforce transitivity through the introduction of an unknown reference canonical ordering and the alignment matrices $Q_i \in \Sigma_n$ $i = 1, \dots, N$ that map vertices in G_i to the reference order. With these matrices to hand we set $\overline{P}_{ij} = Q_i Q_j^T$. Note that there is no lack in representation power, as the transitivity constraint guarantees the existence of such canonical ordering. In fact, let for example $Q_i = \overline{P}_{i1}$. For transitivity, we have

$$\overline{P}_{ij} = \overline{P}_{i1}\overline{P}_{1j} = \overline{P}_{i1}\overline{P}_{j1}^T = Q_i Q_j^T. \tag{10.2}$$

Furthermore, such canonical ordering is not unique, since for any permutation matrix $P \in \Sigma_n$, we have

$$\overline{P}_{ij} = Q_i Q_j^T = (Q_i P)(P^T Q_j^T). \tag{10.3}$$

With the canonical ordering representation the projection on the transitive space of permutations cast as the following minimization process

$$\begin{aligned}
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min} \sum_{i,j=1}^{N} ||P_{ij} - Q_i Q_j^T||_2^2 = \\
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min} \sum_{i,j=1}^{N} \left(||P_{ij}||_2^2 + ||Q_i Q_j^T||_2^2 - 2\mathrm{Tr}(Q_j Q_i^T P_{ij})\right) = \\
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min} 2N^2 n - 2 \sum_{i,j=1}^{N} \mathrm{Tr}(Q_i^T P_{ij} Q_j) = \\
&\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\min} 2N^2 n - 2 \sum_{i,j=1}^{N} \mathrm{vec}(Q_i)^T \left(I \otimes P_{ij}\right) \mathrm{vec}(Q_j),
\end{aligned} \tag{10.4}$$

where $||\cdot||_2$ is the Frobenius matrix norm while Tr is the linear trace operator.

This is equivalent to the following Integer Quadratic Problem

$$\underset{\mathbf{Q}\in(\Sigma_n)^N}{\arg\max} \underbrace{\begin{pmatrix} \mathrm{vec}(Q_1) \\ \mathrm{vec}(Q_2) \\ \vdots \\ \mathrm{vec}(Q_N) \end{pmatrix}^T}_{\mathrm{vec}(\mathbf{Q})^T} \underbrace{\begin{pmatrix} I\otimes P_{11} & I\otimes P_{12} & \dots & I\otimes P_{1N} \\ I\otimes P_{21} & I\otimes P_{22} & \dots & I\otimes P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ I\otimes P_{N1} & I\otimes P_{N2} & \dots & I\otimes P_{NN} \end{pmatrix}}_{\Pi} \underbrace{\begin{pmatrix} \mathrm{vec}(Q_1) \\ \mathrm{vec}(Q_2) \\ \vdots \\ \mathrm{vec}(Q_N) \end{pmatrix}}_{\mathrm{vec}(\mathbf{Q})}. \tag{10.5}$$

where $\otimes$ represents the Kronecker product while I is the identity matrix. Note that if the pairwise matches are symmetric, *i.e.*, $P_{ij} = P_{ji}^T$, then Π is symmetric as well. However, as in all quadratic problem, Π (and thus P_{ij}) can be made symmetric without affecting the result.

Our proposal is to relax this to the problem

$$\begin{array}{ll} \text{maximize} & \mathbf{x}^T\Pi\mathbf{x} \\ \text{s.t.} & \mathbf{x}\in(S_n)^N \end{array} \tag{10.6}$$

where S_n is the unit sphere in $\mathbb{R}^n$, and then project the solution to $(\Sigma_n)^N$ in order to obtain the alignment matrices S_i (which differs from the Q_I seen before since we are working on a relaxed space) and, consequently, the transitive permutation matrices $\overline{P}_{ij} = S_iS_j^T$.

We solve 10.6 efficiently through a power iteration projected to $(S_n)^N$ by noting that the gradient of the quadratic form can be computed in terms of multiplications and additions of the matching and alignment matrices:

$$\begin{aligned} \Pi\mathbf{x} &= \begin{pmatrix} I\otimes P_{11} & I\otimes P_{12} & \dots & I\otimes P_{1N} \\ I\otimes P_{21} & I\otimes P_{22} & \dots & I\otimes P_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ I\otimes P_{N1} & I\otimes P_{N2} & \dots & I\otimes P_{NN} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^N P_{1i}X_i \\ \sum_{i=1}^N P_{2i}X_i \\ \vdots \\ \sum_{i=1}^N P_{Ni}X_i \end{pmatrix}, \end{aligned} \tag{10.7}$$

where $\mathbf{x}^T = (\mathbf{x}_1^T, \dots, \mathbf{x}_N^T)^T$ expresses the N spherical components of $\mathbf{x}$, and X_i is the matrix representing the current relaxation of Q_i, for which we have $\mathbf{x}_i = \mathrm{vec}(X_i)$.

Hence, we can maximize 10.6 by iterating the recurrence

$$X_i^{(t+1)} = \frac{\sum_{j=1}^N P_{ij}X_j^{(t)}}{||\sum_{j=1}^N P_{ij}X_j^{(t)}||_2} \tag{10.8}$$

Once the matrices X_i are at hand, we obtain the closest (in the least square sense) permutations Q_i by solving N maximum bipartite assignment problems.

10.3 Transitive Assignment Kernel

With transitive matches to hand, we follow Fröhlich et al. [63] in the definition of an assignment kernel between graphs $G_i = (V_i, E_i)$ and $G_j = (V_j, E_j)$: we define two sets of kernels, one $K_v : V_i \times V_j \to \mathbb{R}$ for the vertices, and one $K_e : V_i^2 \times V_j^2 \to \mathbb{R}$ for the edges and fuse them with the transitive correspondence $\pi_{ij} : V_i \to V_j$ encoded in $\overline{P}_{ij}$, to obtain the *Transitive Assignment Kernel*:

$$\text{TAK}(G_i, G_j) = \sum_{v \in V_i} K_v\big(v, \pi_{ij}(v)\big) + \sum_{v \in V_i} \sum_{w \in V_i} K_e\big((v,w),(\pi_{ij}(v), \pi_{ij}(w))\big) \tag{10.9}$$

Here, both kernels are assumed to be positive semidefinite and symmetric. In our experiments we used the dot product between Heat Kernel Signatures [174] (HKS) for the vertex kernel K_v. More precisely, given an undirected graph G of n nodes, let $A = (a_{ij})$ the $n \times n$ adjacency matrix (where a_{ij} is the weight of the edge between the nodes i and j in G) and D the degree matrix, we compute the related $n \times n$ Laplacian matrix L as

$$L = D - A$$

Let ϕ_i the i-th eigenvector of L (with $i = 1, \dots, n$) and $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)^T$ the eigenvalues of the Laplacian. Finally, let m be a set of time values $\{t_1, t_2, \dots, t_m\}$. We define the HKS feature vector $f = (f_1, f_2, \dots, f_m)^T$ as

$$f_j = \sum_{k=1}^{n} \exp(-t_j \lambda_k) \phi_k^2$$

Once computed, the feature vectors are collected on a $n \times m$ matrix F as columns

$$F = \big(f_1, f_2, \cdots, f_m\big)$$

Given two graph G^i and G^j (with the same number of nodes n), our HKS kernel is defined as the sum of the dot product between the respective feature matrices $k = F_i \cdot F_j = (k_1, k_2, \dots, k_n)^T$. Hence, the kernel matrix is defined as

$$K_v(G^i, G^j) = \sum_{w=1}^{n} k_w$$

On the other hand, the edge kernel K_e was chosen to be a discrete enforcement of the topological structure:

$$K_e\big((u,v),(a,b)\big) = \begin{cases} 1 & \text{if } \big((u,w) \in E_i \wedge (a,b) \in E_j\big) \vee \big((u,w) \notin E_i \wedge (a,b) \notin E_j\big) \\ 0 & \text{otherwise.} \end{cases} \tag{10.10}$$

The positive semidefiniteness of the proposed kernel can be proved through the closure properties of positive definite functions. The closure under sum states that, given a non-empty set X and two positive semidefinite symmetric kernels $K_A, K_B : X \times X \to \mathbb{R}$, it holds

$$K = K_A + K_B : X \times X \to \mathbb{R} \tag{10.11}$$

Then, K is a positive semidefinite symmetric kernel. In other words, in order to construct a new positive semidefinite kernel as the sum of existing ones (K_v and K_e in our instance), first the kernels need to be positive semidefinite. Second, they all must be defined in the same space. The kernels employed in 10.9 are positive semidefinite by hypothesis. Furthermore, since the projection on the transitive alignment space introduces a reference canonical order (and such canonical ordering is guaranteed by the transitivity constraints, see section 10.2), the space of the kernels is the same. In fact, the kernels defined as the sum of all K_vs (K_A) and the sum of all K_es (K_B) are clearly positive semidefinite since all K_vs and all K_es belong to the same respective spaces. Hence, the kernel defined in 10.9 is positive semidefinite. Note that without the transitive alignment and its induced canonical ordering, the assumption that all K_vs and K_es belong to the same respective spaces would be wrong.

10.4 Experimental Evaluation

We evaluate the performance of the proposed method in terms of classification accuracy and we compare it with a number of well-known kernels, namely the Weisfeiler-Lehman kernel [164] (where the number of iterations parameter was set to $h = 3$ and we used the degree of each node as the node attribute), the graphlet kernel [165], the shortest-path kernel [25], the random walk kernel [96] and an experimental kernel based on the Heat Kernel Signature [174] method. In particular, we employ the Heat Kernel Signature to compute the feature descriptors with respect to $k = 100$ time parameters t uniformly distributed within the range $[1, 10]$ and we build the kernel as described in section 10.3.

Furthermore, we compare the performance of the proposed method with respect to the state-of-the-art of graph matching methods, namely the Spectral Matching (SM) [111] and Reweighted Random Walks Matching (RRWM) [44]. In order to do so, we address the classification task using several popular datasets with and without the permutations computed by the graph matching methods.

Given a pair of graphs (G^p, G^q) with the same number of nodes n, we compute the $n^2 \times n^2$ affinity matrix $M_{pq} = (m_{ia,jb})$ as

$$m_{ia,jb} = \exp\left(-\frac{(a^p_{ij} - a^q_{ab})^2}{\sigma^2}\right)$$

where σ^2 is a scale factor which is experimentally set to 0.15. This affinity matrix is employed as the input of one of the graph matching technique (GM) introduced above (SM and RRWM), obtaining the $n \times n$ weight matrix $W_{pq} = \mathrm{GM}(M_{pq})$. Note that the number of nodes of the graphs G^p and G^q are not required to be same, since if they are different, we will just add some disconnected dummy nodes in order to make the number of the nodes equal, as explained in section 10.2. Finally, we use the real matrix W_{pq} as the input for the Hungarian algorithm, which is a well-known method that performs a combinatorial optimization finding a maximum weight matching in a weighted bipartite graph. This results in a discretized version of the weight matrix, which is, in practice, a permutation matrix. Hence, we define the permutation matrix P_{pq} as

$$P_{pq} = \text{Hungarian}(W_{pq})$$

We run our experiments on the following datasets:

MUTAG dataset [51] was constructed based on data from review of literatures about mutagenicities in *Salmonella Typhimurium* based on 200 aromatic and heteroaromatic nitro compounds. As a result, 188 congeners were extracted together with their structure-activity relationship (SAR) data.

PPI dataset, which consists of protein-protein interaction (PPIs) networks related to histidine kinase [90] (40 PPIs from *Acidovorax avenae* and 46 PPIs from *Acidobacteria*).

PTC (The Predictive Toxicology Challenge) dataset, which records the carcinogenicity of several hundred chemical compounds for male rats (MR), female rats (FR), male mice (MM) and female mice (FM) [115] (here we use the 344 graphs in the MR class).

COIL dataset, which consists of 5 objects from [135], each with 72 views obtained from equally spaced viewing directions, where for each view a graph was built by triangulating the extracted Harris corner points.

Reeb dataset, which consists of a set of adjacency matrices associated to the computation of reeb graphs of 3D shapes [19].

ENZYMES dataset [161] is based on graphs representing protein tertiary structures consisting of 600 enzymes from the BRENDA enzyme database, which are correctly assigned to one of the 6 EC top-level classes.

SHOCK dataset consists of graphs from the Shock 2D shape database. Each graph of the 150 graphs divided into 10 classes is a skeletal-based representation of the differential structure of the boundary of a 2D shape.

For efficiency purposes, the experiments do not involve the whole datasets. In particular, we select a certain number of classes and a certain number of graphs for each class. The selection of these subsets is performed randomly on the original datasets. Table 10.1 shows the number of classes and the number of graphs of each dataset that has been used to compute the results. In order to get an homogeneous number of nodes within the graphs of the same dataset, we add to each graph $n_{MAX} - n_i$ dummy nodes (*i.e.* not connected nodes), where n_{MAX} is the maximum number of nodes among the graphs of a certain dataset, while n_i is the number of nodes of the i-th graph.

We used a binary C-SVM to test the efficacy of the kernels. We performed 10-fold cross validation, where for each sample we independently tune the value of C, the SVM regularizer constant, by considering the training data from that sample. The process was averaged over 100 random partitions of the data, and the results are reported in terms of average accuracy ± standard error. In particular, at each 10-fold cross validation iteration,

Table 10.1: Details of the datasets.

Dataset Name	Classes	Graphs per class	Total Graphs	Graph Nodes
MUTAG	2	≈ 94	188	28
PPI	2	20	40	161
PTC	2	30	60	70
COIL	3	20	60	112
Reeb	3	20	60	86
ENZYMES	3	20	60	26
SHOCK	10	15	150	33

the dataset is randomly permuted and subdivided in 10 folds. Every fold is used as a crossvalidation fold, while the remaining are use to train the SVM. The process is repeated 100 times. Finally, we define the standard error as

$$\hat{\sigma}_X = \sqrt{n} \cdot \sqrt{\frac{\sum_{i=1}^{n} (x - \bar{x})^2}{n}} = \sqrt{\sum_{i=1}^{n} (x - \bar{x})^2}$$

where $\bar{x}$ is the mean accuracy obtained in a crossvalidation iteration with n samples $X = \{x_1, x_2, \dots, x_n\}$.

Table 10.2: Classification accuracy (± standard error) on unattributed graph datasets. Respectively, HKS is the Heat Kernel Signature [174], WL is the Weisfeiler-Lehman kernel [164], GR denotes the graphlet kernel computed using all graphlets of size 3 [165], SP is the shortest-path kernel [25], and RW is the random walk kernel [96]. The second part of the table collects the accuracy of HKS kernel employing the permutations from Spectral Matching (SM) [111] and Reweighted Random Walks Matching (RRWM) [44] with respect to the transitive versions produced by our method (denoted by the prefix T). For each kernel and dataset, the best performing kernel is highlighted in italic, while the bold highlights the maximum just considering data in the second part of the table for each pair of graph matchings (non transitive w.r.t. transitive).

Kernel	MUTAG	PPI	PTC	COIL	Reeb	ENZYMES	SHOCK
HKS	80.5 ± 0.2	63.6 ± 0.7	50.2 ± 0.5	*87.8 ± 0.8*	46.6 ± 0.6	56.9 ± 0.6	*46.8 ± 0.3*
WL	78.3 ± 0.2	*70.4 ± 0.8*	*67.1 ± 0.6*	70.6 ± 0.7	*68.7 ± 0.4*	55.4 ± 0.6	35.0 ± 0.2
SP	*83.3 ± 0.2*	58.5 ± 0.7	50.5 ± 0.6	86.7 ± 0.6	68.1 ± 0.4	52.2 ± 0.5	39.0 ± 0.3
RW	80.1 ± 0.2	48.5 ± 0.8	41.6 ± 0.6	65.2 ± 0.7	49.8 ± 0.6	13.6 ± 0.3	1.7 ± 0.1
GR	81.5 ± 0.2	30.3 ± 0.5	51.6 ± 0.6	87.1 ± 0.5	22.7 ± 0.6	47.0 ± 0.6	26.1 ± 0.3
HKS-SM	69.0 ± 0.3	60.9 ± 0.8	49.4 ± 0.6	84.8 ± 1.0	45.7 ± 0.6	49.1 ± 0.6	39.4 ± 0.4
HKS-TSM	**80.7** ± **0.2**	**64.2** ± **0.8**	**50.1** ± **0.6**	**87.0** ± **0.9**	**46.2** ± **0.5**	***57.2*** ± ***0.7***	**46.7** ± **0.3**
HKS-RRWM	79.8 ± 0.2	60.4 ± 0.9	**52.1** ± **0.5**	**87.3** ± **0.9**	44.5 ± 0.6	44.9 ± 0.6	25.7 ± 0.2
HKS-TRRWM	**80.5** ± **0.2**	**64.3** ± **0.8**	50.9 ± 0.5	86.1 ± 0.9	**44.8** ± **0.6**	**45.5** ± **0.6**	**46.4** ± **0.3**

Table 10.2 shows the average classification accuracy (± standard error) of the different kernels on the selected datasets. The first part of the table shows the accuracy computed using the datasets after the pruning operation mentioned before. The second part of the table (after the double line) shows the classification accuracy achieved after the application

of the permutations yielded by the compared graph matching methods. More precisely, given $\mathbf{P}_{ij}$ the permutation matrix which defines the correspondences of the graph i with respect to graph j, we compute the value of the kernel between the permuted graph i and the graph j. In particular, HKS-SM shows the classification accuracy obtained permuting the graphs using the Spectral Matching results, while HKS-TSM shows the results obtained using the proposed method which has been initialized using Spectral Matching. The results show that the proposed method is competitive and outperform the other graph matching algorithms in almost all the datasets. COIL and PTC datasets are an exception, since HKS-RRWM performs slightly better with respect to our proposal. Note that the first part of the table should be treated by the reader just as a reference of the accuracies that the current state-of-the-art kernel methods achieve. Indeed, these kernels work independently from the alignment of the graphs to be classified. The main goal of the experimental results is the comparison between the proposed alignment method with respect to the compared ones, namely Spectral Matching and Reweighted Random Walks Matching. In particular, we want to show the performance achieved by the current state-of-the-art in graph matching methods with respect to the transitive approach we are presenting.

10.5 Conclusion

In this chapter we investigated the use of multi-graph matching techniques in the context of graph kernels. By forcing the correspondences between the structures under study to satisfy transitivity, we obtain an alignment kernel that, not only is positive definite, but also makes use of more reliable locational information obtained through the enforcement of global consistency constraints. We proposed a general framework for the projection of (relaxed) correspondences onto the space of transitive correspondences, thus transforming any given matching algorithm to a transitive multi-graph matching approach. The resulting transitive correspondences where used to provide an alignment-based kernel that was able to both maintain locational information and guarantee positive-definiteness. Experimental evaluation shows that the projection onto the transitive space almost invariably increases the classification performance of the alignment kernel, often taking it to a performance level that is at least statistically equivalent to the best performing well-tuned graph kernels present in the literature.

IV

Conclusion and Future Works

11

Conclusion

In this thesis we approached the problem of dealing with non-vectorial data from different directions and proposed several methods that try to solve it.

In the second part of the thesis we proposed a brief introduction of the problem and what has already been done in literature. In the second part of the thesis we tackled the non-vectorial data problem in the shape analysis field, in which we proposed two generative models employed in a shape retrieval task. Furthermore, we proposed a shape matching algorithm based of functional maps.

In Chapter 5 we presented a supervised data-driven approach for the construction of a generative model based on the spectral decomposition of the Laplace-Beltrami operator. We defined a statistical framework that models a shape as two independent generative models for the eigenvector and the eigenvalue components of the spectral representation of the Laplacian, making no assumption about the distribution of the spectral embedding space of the eigenvector part and estimating it through kernel density estimation. Moreover, we modelled the eigenvalues as a log-normal distribution. We tested our method on one of the most recent and popular dataset for shape retrieval, obtaining encouraging results.

In Chapter 6 we presented another supervised learning method for the definition of an invariant representation of an arbitrary shape. Further, we defined a statistical framework based on such representation that can be used to address shape classification tasks. One of the main contributions is the definition of an efficient intrinsic metric to approximate the geodesic distance between points over a manifold. Even if the proposed method is limited by the assumption that correspondence maps between involved surfaces are known, we proposed an approximative approach that deal with it. We tested our method on several standard shape retrieval datasets and the experimental results show that the proposed approach is competitive with the compared non-rigid 3D shape retrieval methods.

Finally, in Chapter 7 we presented a novel approach to the computation of dense correspondence maps between shapes in a non-rigid setting. The problem is defined in terms of functional correspondences. We deal with the non-injectivity of the solution of the functional map framework due to the under-determinedness of the original problem. Key to the approach is the injectivity constraint plugged directly into the problem to optimize, achieved by casting it as an assignment problem. This leads to an iterative process which yields a high quality bijective map between the shapes. In the experimental section we present both quantitative and qualitative results, showing that the proposed approach is competitive with the current state-of-the-art on quasi-isometric shape matching benchmarks.

In the third part we showed how to deal with data represented through graphs. In particular, we proposed three methods that deal with graph embedding and transitiveness of graph matching solutions.

In Chapter 8 we have introduced a novel model of structural representation based on a spectral descriptor of graphs which lifts the one-to-one node-correspondence assumption and is strongly rooted in a statistical learning framework. We showed how the defined separate models for eigenvalues and eigenvectors could be used within a statistical framework to address the graphs classification task. We tested the defined method against a number of alternative graph kernels and we showed its effectiveness in a number of structural classification tasks.

In Chapter 9 we have generalized a structural kernel based on the Jensen-Shannon divergence between quantum walks over the graph by introducing a novel alignment step which, rather than permuting the nodes of the structures, aligns the quantum states of their walks. We proved that the resulting kernel maintains the localization within the structures, but still guarantees positive definiteness. We tested our kernel against a number of alternative graph kernels and we showed its effectiveness in a number of structural classification tasks.

Finally, in Chapter 10 we investigated the use of multi-graph matching techniques in the context of graph kernels. By forcing the correspondences between the structures under study to satisfy transitivity, we obtain an alignment kernel that, not only is positive definite, but also makes use of more reliable locational information obtained through the enforcement of global consistency constraints. We proposed a general framework for the projection of (relaxed) correspondences onto the space of transitive correspondences, thus transforming any given matching algorithm to a transitive multi-graph matching approach. The resulting transitive correspondences where used to provide an alignment-based kernel that was able to both maintain locational information and guarantee positive-definiteness. Experimental evaluation shows that the projection onto the transitive space almost invariably increases the classification performance of the alignment kernel, often taking it to a performance level that is at least statistically equivalent to the best performing well-tuned graph kernels present in the literature.

11.1 Future Work

For the work in learning Riemannian variations on deformable shapes we would like to apply the method to structure-based data. In particular, we want to try to use the metric defined over the manifold to compute graph medians of a set of graphs. Since graphs have usually a much lower cardinality with respect to shapes, we would use more sophisticated matching algorithms to bring the graph embeddings in a common space. The deviation from the true generalized median (introduced in Chapter 4) would provide a good indication of the true performance of the approach.

We would like to extend the work on shape matching presented in Chapter 7 lifting the one-to-one correspondence constraint in the optimization formulation and defining it as a transport problem. In this setting, the objective becomes to compute the map that minimize the distortion while preserving the area transfer between the surfaces. Furthermore, a regulation factor could be added to the problem formulation in order to deal with the ubiquitous problem of topological changes between surfaces, a problem that afflicts every approach based on spectral descriptors.

Finally, we would like to employ the shape matching algorithm defined in Chapter 7 as the pre-processing step of a shape retrieval method based on statistical distributions defined directly on special manifolds (*i.e.* Stiefel and Grassmanian manifolds). Hopefully, this could lead to a generative model able to capture and characterize a broader class of transformations between surfaces.

Bibliography

[8] The princeton shape benchmark. In *Proceedings of the Shape Modeling International 2004* (Washington, DC, USA, 2004), SMI '04, IEEE Computer Society, pp. 167–178.

[9] A. M. BRONSTEIN, M. M. BRONSTEIN, R. K. *Numerical geometry of non-rigid shapes*. Springer, 2008.

[10] ALBARELLI, A., RODOLÀ, E., BERGAMASCO, F., AND TORSELLO, A. A non-cooperative game for 3d object recognition in cluttered scenes. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 3DIMPVT 2011* (2011), pp. 252–259.

[11] ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. Scape: shape completion and animation of people. *ACM Trans. Graph 24* (2005), 408–416.

[12] AUBRY, M., SCHLICKEWEI, U., AND CREMERS, D. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (Nov 2011), pp. 1626–1633.

[13] BAI, L., HANCOCK, E. R., TORSELLO, A., AND ROSSI, L. *A Quantum Jensen-Shannon Graph Kernel Using the Continuous-Time Quantum Walk.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 121–131.

[14] BANDELT, H. J., AND CHEPOI, V. Metric graph theory and geometry: a survey. *Contemporary Mathematics 453* (2008), 49–86.

[15] BAYATI, M., GLEICH, D. F., SABERI, A., AND WANG, Y. Message-passing algorithms for sparse network alignment. *ACM Trans. Knowl. Discov. Data 7*, 1 (Mar. 2013), 3:1–3:31.

[16] BELKIN, M., SUN, J., AND WANG, Y. Discrete laplace operator on meshed surfaces. In *Proceedings of the Twenty-fourth Annual Symposium on Computational Geometry* (New York, NY, USA, 2008), SCG '08, ACM, pp. 278–287.

[17] BERTSEKAS, D. P. Auction algorithms for network flow problems: A tutorial introduction. *Computational Optimization and Applications 1*, 1, 7–66.

[18] BESL, P., AND MCKAY, N. D. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 14*, 2 (Feb 1992), 239–256.

[19] BIASOTTI, S., MARINI, S., MORTARA, M., PATANÈ, G., SPAGNUOLO, M., AND FALCIDIENO, B. 3d shape matching through topological structures. In *Discrete Geometry for Computer Imagery*, I. Nyström, G. Sanniti di Baja, and S. Svensson, Eds., vol. 2886 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2003, pp. 194–203.

[20] BIMBO, A. D., AND PALA, P. Content-based retrieval of 3d models. *ACM Trans. Multimedia Comput. Commun. Appl. 2*, 1 (Feb. 2006), 20–43.

[21] BISHOP, C. M. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[22] BLEI, D. M., NG, A. Y., AND JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res. 3* (Mar. 2003), 993–1022.

[23] BOGO, F., ROMERO, J., LOPER, M., AND BLACK, M. J. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (Piscataway, NJ, USA, June 2014), IEEE.

[24] BONEV, B., ESCOLANO, F., LOZANO, M., SUAU, P., CAZORLA, M., AND AGUILAR, W. Constellations and the unsupervised learning of graphs. In *Graph-Based Representations in Pattern Recognition*, F. Escolano and M. Vento, Eds., vol. 4538 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2007, pp. 340–350.

[25] BORGWARDT, K. M., AND KRIEGEL, H.-P. Shortest-path kernels on graphs. In *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM 2005)* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 74–81.

[26] BOURGEOIS, F., AND LASSALLE, J.-C. An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM 14*, 12 (Dec. 1971), 802–804.

[27] BOYER *et al.*, E. Shrec 2011: Robust feature detection and description benchmark. In *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval* (Aire-la-Ville, Switzerland, Switzerland, 2011), 3DOR '11, Eurographics Association, pp. 71–78.

[28] BRIËT, J., AND HARREMOËS, P. Properties of classical and quantum jensen-shannon divergence. *Physical review A 79*, 5 (2009), 052311.

[29] BRONSTEIN, A., BRONSTEIN, M., AND KIMMEL, R. *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008.

[30] BRONSTEIN, A. M., BRONSTEIN, M. M., CASTELLANI, U., DUBROVINA, A., GUIBAS, L. J., HORAUD, R. P., KIMMEL, R., KNOSSOW, D., LAVANTE, E. V.,

MATEUS, D., OVSJANIKOV, M., AND SHARMA, A. SHREC'10 Track: Correspondence Finding. In *Eurographics Workshop on 3D Object Retrieval* (2010), M. Daoudi and T. Schreck, Eds., The Eurographics Association.

[31] BRONSTEIN, A. M., BRONSTEIN, M. M., GUIBAS, L. J., AND OVSJANIKOV, M. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph. 30*, 1 (Feb. 2011), 1:1–1:20.

[32] BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS 103*, 5 (2006), 1168–1172.

[33] BRONSTEIN, M., AND KOKKINOS, I. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (June 2010), pp. 1704–1711.

[34] BUNKE, H. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters 18*, 8 (1997), 689 – 694.

[35] BUNKE, H. Error correcting graph matching: On the influence of the underlying cost function. *IEEE Trans. Pattern Anal. Mach. Intell. 21*, 9 (Sept. 1999), 917–922.

[36] BUNKE, H., AND ALLERMANN, G. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters 1*, 4 (1983), 245 – 253.

[37] BUNKE, H., FOGGIA, P., GUIDOBALDI, C., AND VENTO, M. Graph clustering using the weighted minimum common supergraph. In *Graph Based Representations in Pattern Recognition*, E. Hancock and M. Vento, Eds., vol. 2726 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2003, pp. 235–246.

[38] BUNKE, H., AND GÜNTER, S. Weighted mean of a pair of graphs. *Computing 67*, 3 (Nov. 2001), 209–224.

[39] BUNKE, H., MÜNGER, A., AND JIANG, X. Combinatorial search versus genetic algorithms: A case study based on the generalized median graph problem. *Pattern Recognition Letters 20*, 11–13 (1999), 1271 – 1277.

[40] CAELLI, T., AND KOSINOV, S. An eigenspace projection clustering method for inexact graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 4 (April 2004), 515–519.

[41] CARCASSONI, M., , CARCASSONI, M., AND HANCOCK, E. R. Alignment using spectral clusters. In *Proceedings of the 13th British Machine Vision Conference* (2002), pp. 213–222.

[42] CHEN, D.-Y., TIAN, X.-P., TE SHEN, Y., AND OUHYOUNG, M. On visual similarity based 3d model retrieval, 2003.

[43] CHEN, Q., AND KOLTUN, V. Robust nonrigid registration by convex optimization. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015* (2015), pp. 2039–2047.

[44] CHO, M., LEE, J., AND LEE, K. Reweighted random walks for graph matching. In *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds., vol. 6315 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2010, pp. 492–505.

[45] CHUNG, F. R. K. *Spectral Graph Theory*. American Mathematical Society, 1997.

[46] CONTE, D., FOGGIA, P., SANSONE, C., AND VENTO, M. Thirty years of graph matching in pattern recognition. *IJPRAI 18*, 3 (2004), 265–298.

[47] CORNEA, D. N., SILVER, D., YUAN, X., AND BALASUBRAMANIAN, R. Computing hierarchical curve-skeletons of 3d objects. *The Visual Computer 21*, 11 (2005), 945–955.

[48] COSMO, L., RODOLÀ, E., BRONSTEIN, M. M., TORSELLO, A., CREMERS, D., AND SAHILLIOGLU, Y. Shrec'16: Partial matching of deformable shapes. In *Proc. 3DOR* (2016).

[49] CRIMINISI, A., SHOTTON, J., AND KONUKOGLU, E. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends® in Computer Graphics and Vision 7*, 2–3 (2012), 81–227.

[50] DAROM, T., AND KELLER, Y. Scale-invariant features for 3-d mesh models. *Image Processing, IEEE Transactions on 21*, 5 (May 2012), 2758–2769.

[51] DEBNATH, A. K., LOPEZ DE COMPADRE, R. L., DEBNATH, G., SHUSTERMAN, A. J., AND HANSCH, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. vol. 34, ACS Publications, pp. 786–797.

[52] DUFFIN, R. J. Distributed and lumped networks. *Journal of Mathematics and Mechanics 8*, 5 (1959), 793–826.

[53] ESTRADA, F., AND JEPSON, A. Benchmarking image segmentation algorithms. *International journal of computer vision 85*, 2 (2009), 167–181.

[54] FERRER, M., SERRATOSA, F., AND SANFELIU, A. *Synthesis of Median Spectral Graph*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 139–146.

[55] FERRER, M., SERRATOSA, F., AND VALVENY, E. *On the Relation Between the Median and the Maximum Common Subgraph of a Set of Graphs*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 351–360.

[56] FERRER, M., VALVENY, E., AND SERRATOSA, F. Bounding the size of the median graph. In *Proceedings of the 3rd Iberian Conference on Pattern Recognition and Image Analysis, Part II* (Berlin, Heidelberg, 2007), IbPRIA '07, Springer-Verlag, pp. 491–498.

[57] FERRER, M., VALVENY, E., AND SERRATOSA, F. Median graphs: A genetic approach based on new theoretical properties. *Pattern Recognition 42*, 9 (2009), 2003 – 2012.

[58] FERRER, M., VALVENY, E., SERRATOSA, F., AND BUNKE, H. *Exact Median Graph Computation Via Graph Embedding*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 15–24.

[59] FERRER, M., VALVENY, E., SERRATOSA, F., RIESEN, K., AND BUNKE, H. An approximate algorithm for median graph computation using graph embedding. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (Dec 2008), pp. 1–4.

[60] FOWLKES, C., BELONGIE, S., CHUNG, F., AND MALIK, J. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 2 (Feb 2004), 214–225.

[61] FRIEDMAN, N., AND KOLLER, D. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine Learning 50*, 1-2 (2003), 95–125.

[62] FRÖHLICH, H., WEGNER, J. K., SIEKER, F., AND ZELL, A. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22Nd International Conference on Machine Learning* (New York, NY, USA, 2005), ICML '05, ACM, pp. 225–232.

[63] FRÖHLICH, H., WEGNER, J. K., SIEKER, F., AND ZELL, A. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd International Conference on Machine Learning (ICML 2005)* (Bonn, Germany, August 2005), L. de Raedt and S. Wrobel, Eds., ACM Press, pp. 225–232.

[64] FUNKHOUSER, T., AND KAZHDAN, M. Shape-based retrieval and analysis of 3d models. *Communications of the ACM* (2005), 58–64.

[65] GAERTNER, T., FLACH, P., AND WROBEL, S. On graph kernels: Hardness results and efficient alternatives. In *Proceedings of the 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop* (August 2003), Springer-Verlag, pp. 129–143.

[66] GAO, X., XIAO, B., TAO, D., AND LI, X. A survey of graph edit distance. *Pattern Analysis and Applications 13*, 1 (2010), 113–129.

[67] Gasparetto, A., Minello, G., and Torsello, A. A non-parametric spectral model for graph classification. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods* (2015), pp. 312–319.

[68] Gasparetto, A., Minello, G., and Torsello, A. Non-parametric spectral model for shape retrieval. In *2015 International Conference on 3D Vision, 3DV 2015, Lyon, France, October 19-22, 2015* (2015), pp. 344–352.

[69] Gasparetto, A., and Torsello, A. A statistical model of riemannian metric variation for deformable shape analysis. In *CVPR* (2015), IEEE, pp. 1219–1228.

[70] Giachetti, A., and Lovato, C. Radial symmetry detection and shape characterization with the multiscale area projection transform. *Computer Graphics Forum 31*, 5 (2012), 1669–1678.

[71] Gold, S., and Rangarajan, A. A graduated assignment algorithm for graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 18*, 4 (Apr 1996), 377–388.

[72] Goldberg, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.

[73] Gong, B., Liu, J., Wang, X., and Tang, X. Learning semantic signatures for 3d object retrieval. *Multimedia, IEEE Transactions on 15*, 2 (Feb 2013), 369–377.

[74] Grauman, K., and Darrell, T. The pyramid match kernel: discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Oct 2005), vol. 2, pp. 1458–1465 Vol. 2.

[75] Grover, L. K. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1996), STOC '96, ACM, pp. 212–219.

[76] Haemers, W. H., and Spence, E. Enumeration of cospectral graphs. *European Journal of Combinatorics 25*, 2 (2004), 199 – 211. In memory of Jaap Seidel.

[77] Hartley, R. I., Trumpf, J., Dai, Y., and Li, H. Rotation averaging. *International Journal of Computer Vision 103*, 3 (2013), 267–305.

[78] Haseeb, M., and Hancock, E. 3d shape classification using commute time. In *Structural, Syntactic, and Statistical Pattern Recognition*, G. Gimel'farb, E. Hancock, A. Imiya, A. Kuijper, M. Kudo, S. Omachi, T. Windeatt, and K. Yamada, Eds., vol. 7626 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, pp. 208–215.

[79] Haussler, D. Convolution kernels on discrete structures, 1999.

[80] Hilaga, M., Shinagawa, Y., Kohmura, T., and Kunii, T. L. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* (New York, NY, USA, 2001), SIGGRAPH '01, ACM, pp. 203–212.

[81] Hochreiter, S., and Obermayer, K. Support vector machines for dyadic data. *Neural Comput. 18*, 6 (June 2006), 1472–1510.

[82] Hofmann, T. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn. 42*, 1-2 (Jan. 2001), 177–196.

[83] Huang, Q., Wang, F., and Guibas, L. J. Functional map networks for analyzing and exploring large shape collections. *TOG 33*, 4 (2014), 36.

[84] Huang, Q.-X., Su, H., and Guibas, L. Fine-grained semi-supervised labeling of large shape collections. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 190:1–190:10.

[85] Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proceedings of the National Academy of Sciences 98*, 8 (2001), 4569.

[86] Jain, B. J., and Geibel, Wysotzki, F. Svm learning with the sh inner product. In *12th European Symposium on Artificial Neural Networks* (Bruges, Belgium,).

[87] Jain, B. J., Geibel, P., and Wysotzki, F. Svm learning with the schur-hadamard inner product for graphs. *Neurocomput. 64* (Mar. 2005), 93–105.

[88] Jain, V., and Zhang, H. A spectral approach to shape-based retrieval of articulated 3d models. *Comput. Aided Des. 39*, 5 (May 2007), 398–407.

[89] Jeh, G., and Widom, J. Simrank: A measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2002), KDD '02, ACM, pp. 538–543.

[90] Jensen, L. J., Kuhn, M., Stark, M., Chaffron, S., Creevey, C., Muller, J., Doerks, T., Julien, P., Roth, A., Simonovic, M., et al. String 8—a global view on proteins and their functional interactions in 630 organisms. *Nucleic acids research 37*, suppl 1 (2009), D412–D416.

[91] Jeong, H., Tombor, B., Albert, R., Oltvai, Z., and Barabási, A. The large-scale organization of metabolic networks. *Nature 407*, 6804 (2000), 651–654.

[92] Jiang, X., Munger, A., and Bunke, H. An median graphs: properties, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 10 (Oct 2001), 1144–1151.

[93] JOHNSON, A. E., AND HEBERT, M. Using spin-images for efficient object recognition in cluttered 3-d scenes. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE 21*, 5 (1999), 433–449.

[94] JOSHI, S. H., KLASSEN, E., SRIVASTAVA, A., AND JERMYN, I. *Removing Shape-Preserving Transformations in Square-Root Elastic (SRE) Framework for Shape Analysis of Curves*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 387–398.

[95] KALAPALA, V., SANWALANI, V., AND MOORE, C. The structure of the united states road network. *Preprint, University of New Mexico* (2003).

[96] KASHIMA, H., TSUDA, K., AND INOKUCHI, A. Marginalized kernels between labeled graphs. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA* (2003), pp. 321–328.

[97] KATO, T. *Perturbation Theory for Linear Operators*. Springer-Verlag GmbH, 1995.

[98] KAZHDAN, M., FUNKHOUSER, T., AND RUSINKIEWICZ, S. Rotation invariant spherical harmonic representation of 3d shape descriptors. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), SGP '03, Eurographics Association, pp. 156–164.

[99] KEMPE, J. Quantum random walks: An introductory overview. *Contemporary Physics 44*, 4 (2003), 307–327.

[100] KENDALL, D. G. Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society 16*, 2 (1984), 81–121.

[101] KHOURY, R., VANDEBORRE, J.-P., AND DAOUDI, M. Indexed heat curves for 3d-model retrieval. In *Pattern Recognition (ICPR), 2012 21st International Conference on* (Nov 2012), pp. 1964–1967.

[102] KIM, V. G., LIPMAN, Y., CHEN, X., AND FUNKHOUSER, T. A. Möbius transformations for global intrinsic symmetry analysis. *Comput. Graph. Forum 29*, 5 (2010), 1689–1700.

[103] KIM, V. G., LIPMAN, Y., AND FUNKHOUSER, T. Blended intrinsic maps. 79:1–79:12.

[104] KONDOR, R. I., AND LAFFERTY, J. D. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the Nineteenth International Conference on Machine Learning* (San Francisco, CA, USA, 2002), ICML '02, Morgan Kaufmann Publishers Inc., pp. 315–322.

[105] KOVNATSKY, A., BRONSTEIN, M., BRONSTEIN, A., GLASHOFF, K., AND KIMMEL, R. Coupled quasi-harmonic bases. *Comput. Graph. Forum 32*, 2pt4 (2013), 439–448.

[106] KRIEGE, N., AND MUTZEL, P. Subgraph matching kernels for attributed graphs. In *ICML* (2012), icml.cc / Omnipress.

[107] KUHN, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 1-2 (1955), 83–97.

[108] LATECKI, L. J., WANG, Q., KOKNAR-TEZEL, S., AND MEGALOOIKONOMOU, V. Optimal subsequence bijection. In *Seventh IEEE International Conference on Data Mining (ICDM 2007)* (Oct 2007), pp. 565–570.

[109] LAVOUÉ, G. Combination of bag-of-words descriptors for robust partial shape retrieval. *The Visual Computer 28*, 9 (2012), 931–942.

[110] LAZEBNIK, S., SCHMID, C., AND PONCE, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (2006), vol. 2, pp. 2169–2178.

[111] LEORDEANU, M., AND HEBERT, M. A spectral technique for correspondence problems using pairwise constraints. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (Oct 2005), vol. 2, pp. 1482–1489 Vol. 2.

[112] LI, C., AND BEN HAMZA, A. Skeleton path based approach for nonrigid 3d shape analysis and retrieval. In *Combinatorial Image Analysis*, J. Aggarwal, R. Barneva, V. Brimkov, K. Koroutchev, and E. Korutcheva, Eds., vol. 6636 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, pp. 84–95.

[113] LI, C., AND BEN HAMZA, A. A multiresolution descriptor for deformable 3d shape retrieval. *Vis. Comput. 29*, 6-8 (June 2013), 513–524.

[114] LI, C., AND HAMZA, A. Intrinsic spatial pyramid matching for deformable 3d shape retrieval. *International Journal of Multimedia Information Retrieval 2*, 4 (2013), 261–271.

[115] LI, G., SEMERCI, M., YENER, B., AND ZAKI, M. J. Effective graph classification based on topological and label attributes. *Statistical Analysis and Data Mining 5*, 4 (2012), 265–283.

[116] LIPMAN, Y., AND FUNKHOUSER, T. Möbius voting for surface correspondence. In *ACM SIGGRAPH 2009 Papers* (New York, NY, USA, 2009), SIGGRAPH '09, ACM, pp. 72:1–72:12.

[117] Litman, R., Bronstein, A., Bronstein, M., and Castellani, U. Supervised learning of bag-of-features shape descriptors using sparse coding. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 127–136.

[118] Litman, R., and Bronstein, A. M. Learning spectral descriptors for deformable shape correspondence. *IEEE Trans. Pattern Anal. Mach. Intell. 36*, 1 (2014), 171–180.

[119] Livi, L., and Rizzi, A. The graph matching problem. *Pattern Analysis and Applications 16*, 3, 253–283.

[120] López-Sastre, R., García-Fuertes, A., Redondo-Cabrera, C., Acevedo-Rodríguez, F., and Maldonado-Bascón, S. Evaluating 3d spatial pyramids for classifying 3d shapes. *Computers and Graphics 37*, 5 (2013), 473 – 483.

[121] Luo, B., and Hancock, E. R. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell. 23*, 10 (Oct. 2001), 1120–1136.

[122] Luo, B., Wilson, R., and Hancock, E. *A linear generative model for graph structure*. SPRINGER-VERLAG BERLIN, 2005, pp. 54–62.

[123] Luo, B., Wilson, R. C., and Hancock, E. R. Spectral embedding of graphs. *Pattern Recognition 36*, 10 (2003), 2213 – 2230.

[124] Luo, B., Wilson, R. C., and Hancock, E. R. A spectral approach to learning structural variations in graphs. *Pattern Recognition 39*, 6 (2006), 1188 – 1198.

[125] MacNeal, R. H. *The solution of partial differential equations by means of electrical networks*. PhD thesis, California Institute of Technology, 1949.

[126] Macrini, D., Siddiqi, K., and Dickinson, S. From skeletons to bone graphs: Medial abstraction for object recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), pp. 1–8.

[127] Mademlis, A., Daras, P., Tzovaras, D., and Strintzis, M. G. 3d object retrieval using the 3d shape impact descriptor. *Pattern Recognition 42*, 11 (2009), 2447 – 2459.

[128] Melnik, S., Garcia-Molina, H., and Rahm, E. Similarity flooding: a versatile graph matching algorithm and its application to schema matching. In *Data Engineering, 2002. Proceedings. 18th International Conference on* (2002), pp. 117–128.

[129] Minka, T. P. Estimating a gamma distribution, 2002.

[130] MOHAMED, W., AND BEN HAMZA, A. Reeb graph path dissimilarity for 3d object matching and retrieval. *The Visual Computer 28*, 3 (2012), 305–318.

[131] MOHR, J., JAIN, B. J., SUTTER, A., TER LAAK, A., STEGER-HARTMANN, T., HEINRICH, N., AND OBERMAYER, K. A maximum common subgraph kernel method for predicting the chromosome aberration test. *Journal of Chemical Information and Modeling* (2010), 1821–1838.

[132] MOWSHOWITZ, A. Entropy and the complexity of graphs: I. an index of the relative complexity of a graph. *The bulletin of mathematical biophysics 30*, 1 (1968), 175–204.

[133] MOWSHOWITZ, A. Graphs, groups and matrices. *Proceedings, 25th Summer Meeting Canadian Math. Congress, Congr. Num. IV, Utilitas Math., pp. 509–522* (1971).

[134] MULDER, M. The structure of median graphs. *Discrete Mathematics 24*, 2 (1978), 197 – 204.

[135] NENE, S. A., NAYAR, S. K., AND MURASE, H. Columbia object image library (coil-20). *Dept. Comput. Sci., Columbia Univ., New York.[Online] http://www. cs. columbia. edu/CAVE/coil-20. html 62* (1996).

[136] NEUHAUS, M., AND BUNKE, H. Edit distance-based kernel functions for structural pattern classification. *Pattern Recognition 39*, 10 (2006), 1852–1863.

[137] NEUHAUS, M., AND BUNKE, H. Automatic learning of cost functions for graph edit distance. *Information Sciences 177*, 1 (2007), 239 – 247.

[138] NEUMANN, M., PATRICIA, N., GARNETT, R., AND KERSTING, K. Efficient graph kernels by randomization. In *ECML/PKDD (1)* (2012), P. A. Flach, T. D. Bie, and N. Cristianini, Eds., vol. 7523 of *Lecture Notes in Computer Science*, Springer, pp. 378–393.

[139] ONG, C. S., CANU, S., AND SMOLA, A. J. Learning with non-positive kernels. In *In Proc. of the 21st International Conference on Machine Learning (ICML)* (2004), pp. 639–646.

[140] OSADA, R., FUNKHOUSER, T., CHAZELLE, B., AND DOBKIN, D. Shape distributions. *ACM Trans. Graph. 21*, 4 (Oct. 2002), 807–832.

[141] OVSJANIKOV, M., BEN-CHEN, M., SOLOMON, J., BUTSCHER, A., AND GUIBAS, L. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph. 31*, 4 (July 2012), 30:1–30:11.

[142] OVSJANIKOV, M., MÉRIGOT, Q., MÉMOLI, F., AND GUIBAS, L. One point isometric matching with the heat kernel, 2010.

[143] PACHAURI, D., KONDOR, R., AND SINGH, V. Solving the multi-way matching problem by permutation synchronization. In *NIPS* (2013), C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, Eds., pp. 1860–1868.

[144] PICKUP, D., SUN, X., ROSIN, P. L., MARTIN, R. R., CHENG, Z., LIAN, Z., AONO, M., BEN HAMZA, A., BRONSTEIN, A., BRONSTEIN, M., BU, S., CASTELLANI, U., CHENG, S., GARRO, V., GIACHETTI, A., GODIL, A., HAN, J., JOHAN, H., LAI, L., LI, B., LI, C., LI, H., LITMAN, R., LIU, X., LIU, Z., LU, Y., TATSUMA, A., AND YE, J. SHREC'14 track: Shape retrieval of non-rigid 3d human models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval* (2014), EG 3DOR'14, Eurographics Association.

[145] PINKALL, U., AND POLTHIER, K. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics 2*, 1 (1993), 15–36.

[146] POKRASS, J., BRONSTEIN, A. M., BRONSTEIN, M. M., SPRECHMANN, P., AND SAPIRO, G. Sparse modeling of intrinsic correspondences. *Computer Graphics Forum 32*, 2pt4 (2013), 459–468.

[147] QIN, Z., JIA, J., AND QIN, J. Content based 3d model retrieval: A survey. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on* (June 2008), pp. 249–256.

[148] REUTER, M., WOLTER, F.-E., AND PEINECKE, N. Laplace-spectra as fingerprints for shape matching. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling* (New York, NY, USA, 2005), SPM '05, ACM, pp. 101–106.

[149] REUTER, M., WOLTER, F.-E., AND PEINECKE, N. Laplace-beltrami spectra as 'shape-dna' of surfaces and solids. *Comput. Aided Des. 38*, 4 (Apr. 2006), 342–366.

[150] RIESEN, K., NEUHAUS, M., AND BUNKE, H. *Graph Embedding in Vector Spaces by Means of Prototype Selection*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 383–393.

[151] RODOLÀ, E., BRONSTEIN, A. M., ALBARELLI, A., BERGAMASCO, F., AND TORSELLO, A. A game-theoretic approach to deformable shape matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (June 2012), pp. 182–189.

[152] RODOLÀ, E., COSMO, L., BRONSTEIN, M. M., TORSELLO, A., AND CREMERS, D. Partial functional correspondence. *Computer Graphics Forum* (2016), n/a–n/a.

[153] RODOLÀ, E., ROTA BULO, S., WINDHEUSER, T., VESTNER, M., AND CREMERS, D. Dense non-rigid shape correspondence using random forests. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on* (June 2014), pp. 4177–4184.

[154] ROSSI, L., TORSELLO, A., HANCOCK, E. R., AND WILSON, R. C. Characterizing graph symmetries through quantum jensen-shannon divergence. *Phys. Rev. E 88* (Sep 2013), 032806.

[155] RUSTAMOV, R. M. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), SGP '07, Eurographics Association, pp. 225–233.

[156] SAHILLIOĞLU, Y., AND YEMEZ, Y. Coarse-to-Fine Isometric Shape Correspondence by Tracking Symmetric Flips. *Computer Graphics Forum* (2013).

[157] SALIM, N., JOHN HOLLIDAY, ., , AND WILLETT, P. Combination of fingerprint-based similarity coefficients using data fusion. *Journal of Chemical Information and Computer Sciences 43*, 2 (2003), 435–442. PMID: 12653506.

[158] SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM J. Res. Dev. 3*, 3 (July 1959), 210–229.

[159] SCHIETGAT, L., RAMON, J., BRUYNOOGHE, M., AND BLOCKEEL, H. An efficiently computable graph-based metric for the classification of small molecules. In *Proceedings of the 11th International Conference on Discovery Science* (Berlin, Heidelberg, 2008), DS '08, Springer-Verlag, pp. 197–209.

[160] SCHOLKOPF, B., AND SMOLA, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, Cambridge, MA, USA, 2001.

[161] SCHOMBURG, I., CHANG, A., EBELING, C., GREMSE, M., HELDT, C., HUHN, G., AND SCHOMBURG, D. Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Research 32*, suppl 1 (2004), D431–D433.

[162] SCHÖNEMANN, P. A generalized solution of the orthogonal procrustes problem. *Psychometrika 31*, 1 (1966), 1–10.

[163] SCOTT, G. L., AND LONGUET-HIGGINS, H. C. An algorithm for associating the features of two images. *Proceedings of the Royal Society of London B: Biological Sciences 244*, 1309 (1991), 21–26.

[164] SHERVASHIDZE, N., SCHWEITZER, P., VAN LEEUWEN, E. J., MEHLHORN, K., AND BORGWARDT, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research 12* (2011), 2539–2561.

[165] SHERVASHIDZE, N., VISHWANATHAN, S., PETRI, T., MEHLHORN, K., AND BORGWARDT, K. Efficient graphlet kernels for large graph comparison. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics* (2009).

[166] SHI, J., AND MALIK, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 8 (Aug. 2000), 888–905.

[167] SHIH, J.-L., LEE, C.-H., AND WANG, J. T. A new 3d model retrieval approach based on the elevation descriptor. *Pattern Recognition 40*, 1 (2007), 283 – 295.

[168] SHOR, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput. 26*, 5 (Oct. 1997), 1484–1509.

[169] SIDDIQI, K., SHOKOUFANDEH, A., DICKINSON, S., AND ZUCKER, S. Shock graphs and shape matching. *International Journal of Computer Vision 35*, 1 (1999), 13–32.

[170] SIDDIQI, K., ZHANG, J., MACRINI, D., SHOKOUFANDEH, A., BOUIX, S., AND DICKINSON, S. Retrieving articulated 3-d models using medial surfaces. *Machine Vision and Applications 19*, 4 (2008), 261–275.

[171] SILVERMAN, B. W. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, 1986.

[172] SOLÉ-RIBALTA, A., AND SERRATOSA, F. Models and algorithms for computing the common labelling of a set of attributed graphs. *Computer Vision and Image Understanding 115*, 7 (2011), 929–945.

[173] SUMNER, R. W., AND POPOVIĆ, J. Deformation transfer for triangle meshes. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 399–405.

[174] SUN, J., OVSJANIKOV, M., AND GUIBAS, L. A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, Eurographics Association, pp. 1383–1392.

[175] TANGELDER, J., AND VELTKAMP, R. A survey of content based 3d shape retrieval methods. In *Shape Modeling Applications, 2004. Proceedings* (June 2004), pp. 145–156.

[176] TODOROVIC, S., AND AHUJA, N. Extracting subimages of an unknown category from a set of images. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on* (June 2006), vol. 1, pp. 927–934.

[177] TOLDO, R., CASTELLANI, U., AND FUSIELLO, A. The bag of words approach for retrieval and categorization of 3d objects. *The Visual Computer 26*, 10 (2010), 1257–1268.

[178] TORSELLO, A. An importance sampling approach to learning structural representations of shape. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008), pp. 1–7.

[179] TORSELLO, A. On the state alignment of the quantum jensen shannon divergence. Tech. rep., 2014.

[180] TORSELLO, A., AND DOWE, D. L. *Learning a Generative Model for Structural Representations*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 573–583.

[181] TORSELLO, A., GASPARETTO, A., ROSSI, L., BAI, L., AND HANCOCK, E. Transitive state alignment for the quantum jensen-shannon kernel. In *Structural, Syntactic, and Statistical Pattern Recognition*, P. Fränti, G. Brown, M. Loog, F. Escolano, and M. Pelillo, Eds., vol. 8621 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2014, pp. 22–31.

[182] TORSELLO, A., AND HANCOCK, E. Learning shape-classes using a mixture of tree-unions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 28*, 6 (June 2006), 954–967.

[183] TORSELLO, A., AND HANCOCK, E. R. *Learning Mixtures of Weighted Tree-Unions by Minimizing Description Length*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 13–25.

[184] TORSELLO, A., RODOLÀ, E., AND ALBARELLI, A. Multiview registration via graph diffusion of dual quaternions. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011* (2011), IEEE Computer Society, pp. 2441–2448.

[185] TORSELLO, A., AND ROSSI, L. Supervised learning of graph structure. In *SIMBAD* (2011), M. Pelillo and E. R. Hancock, Eds., vol. 7005 of *Lecture Notes in Computer Science*, Springer, pp. 117–132.

[186] TRUNK, G. V. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-1*, 3 (July 1979), 306–307.

[187] UMEYAMA, S. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell. 10*, 5 (Sept. 1988), 695–703.

[188] VAN WYK, B. J., AND VAN WYK, M. A. A pocs-based graph matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26*, 11 (Nov 2004), 1526–1530.

[189] VERT, J.-P. The optimal assignment kernel is not positive definite. *CoRR abs/0801.4061* (2008).

[190] WANG, Y.-K., FAN, K.-C., AND HORNG, J.-T. Genetic-based search for error-correcting graph isomorphism. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 27*, 4 (Aug 1997), 588–597.

[191] WHITE, D. *Generative Models for Graphs*. PhD Thesis, 2009.

[192] WHITE, D., AND WILSON, R. Spectral generative models for graphs. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on* (Sept 2007), pp. 35–42.

[193] WILLIAMS, M. L., WILSON, R. C., AND HANCOCK, E. R. Multiple graph matching with bayesian inference. *Pattern Recognition Letters 18* (1997), 080.

[194] WILSON, R. C., HANCOCK, E. R., AND LUO, B. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 7 (July 2005), 1112–1124.

[195] WU, Z., AND LEAHY, R. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 15*, 11 (Nov 1993), 1101–1113.

[196] XIAO, B., AND HANCOCK, E. R. *A Spectral Generative Model for Graph Structure*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 173–181.

[197] XU, G. Discrete laplace-beltrami operator on sphere and optimal spherical triangulations. *Int. J. Comput. Geometry Appl 16* (2004), 75–93.

[198] YAN, J., TIAN, Y., ZHA, H., YANG, X., ZHANG, Y., AND CHU, S. M. Joint optimization for consistent multiple graph matching. In *Proceedings of the 2013 IEEE International Conference on Computer Vision* (Washington, DC, USA, 2013), ICCV '13, IEEE Computer Society, pp. 1649–1656.

[199] YAN, J., WANG, J., ZHA, H., YANG, X., AND CHU, S. Consistency-driven alternating optimization for multigraph matching: A unified approach. *Image Processing, IEEE Transactions on 24*, 3 (March 2015), 994–1009.

[200] YE, J., YAN, Z., AND YU, Y. Fast nonrigid 3d retrieval using modal space transform. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval* (New York, NY, USA, 2013), ICMR '13, ACM, pp. 121–126.

[201] YOUNG, G., AND HOUSEHOLDER, A. S. Discussion of a set of points in terms of their mutual distances. *Psychometrika 3*, 1 (1938), 19–22.

[202] YU, M., ATMOSUKARTO, I., LEOW, W. K., HUANG, Z., AND XU, R. 3d model retrieval with morphing-based geometric and topological feature maps. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (2003), pp. 656–661.

[203] ZAGER, L. A., AND VERGHESE, G. C. Graph similarity scoring and matching. *Applied Mathematics Letters 21*, 1 (2008), 86 – 94.

[204] ZHANG, C., , ZHANG, C., AND CHEN, T. Indexing and retrieval of 3d models aided by active learning, 2001.

[205] ZHU, P., AND WILSON, R. C. Stability of the eigenvalues of graphs. In *Computer Analysis of Images and Patterns, 11th International Conference, CAIP 2005, Versailles, France, September 5-8, 2005, Proceedings* (2005), pp. 371–378.