UNIVERSITÀ DEGLI STUDI DI VERONA

Department of Computer Science
Graduate School of Natural Sciences and Engineering
Doctoral Program in Computer Science

Local Geometry Processing for Deformations of Non-Rigid 3D Shapes

Simone Melzi

Tutor: Prof. Umberto Castellani

INF/01, XXX cycle, 2015

UNIVERSITÀ DEGLI STUDI DI VERONA

DEPARTMENT OF
Computer Science

GRADUATE SCHOOL OF
Natural Sciences and Engineering

DOCTORAL PROGRAM IN
Computer Science

Cycle XXX / Year A.A. 2014/2015

TITLE OF THE DOCTORAL THESIS
Local Geometry Processing for Deformations of Non-Rigid 3D Shapes

S.S.D. INF/01

Coordinator: Prof. MASSIMO MERRO

Signature:

Tutor: Prof. UMBERTO CASTELLANI

Signature:

Doctoral Student: Dott. SIMONE MELZI

Signature:

Local Geometry Processing for Deformations of Non-Rigid 3D Shapes – Simone Melzi
PhD Thesis, Verona, June 8, 2018.
ISBN: XXXXXXXXXXXXX

Yet on the other hand it is certain that mathematics generally, and particularly geometry, owes its existence to the need which was felt of learning something about the relations of real things to one another. The very word geometry, which, of course, means earth-measuring, proves this.

Albert Einstein

"Geometry and Experience", 1922

Sommario

Nella vita reale incontriamo spesso oggetti non rigidi che si manifestano con configurazioni molto variabili. Le diverse posizioni assunte da un corpo umano sono un chiaro esempio. Queste deformazioni delle forme 3D non rigide complicano notevolmente l'analisi di superfici geometriche ricavate da oggetti reali. Dovendo affrontare variazioni locali, distorsioni e deformazioni parziali risulta complicato analizzare le forme nella loro globalità. Per questo motivo, negli ultimi decenni, numerosi strumenti sono stati proposti per l'analisi locale di immagini, grafi, forme e superfici. A differenza dei metodi globali l'analisi locale si focalizza sulle proprietà di ciascun punto e del suo intorno locale.

In questo manoscritto proponiamo quattro nuovi approcci locali per affrontare le problematiche generate da tali deformazioni migliorando la descrizione puntuale delle superfici che rappresentano oggetti reali. Partendo da questi metodi di analisi locale, proponiamo una serie di strumenti applicativi e risvolti teorici legati alla geometria locale degli oggetti. Più precisamente, estendiamo la Windowed Fourier Transform *dal caso standard di segnali definiti su domini Euclidei a versioni specificamente costruite per superfici. Inoltre proponiamo una nuova* base locale *per gli spazi funzionali definiti sulle superfici che migliorano la localizzazione dell'analisi di segnali su questi domini. Infine introduciamo i* processi evolutivi a tempo discreto*, uno strumento che caratterizza i punti at-*

traverso le loro relazioni con gli altri punti sulla superficie. Il contributo di questa tesi è costituito dall'insieme di strumenti per l'analisi locale di superfici osservando come l'analisi di forme possa arricchirsi di tale localizzazione.

Abstract

Geometry processing and in particular spectral geometry processing deal with many different deformations that complicate shape analysis problems for non-rigid 3*D objects. Furthermore, pointwise description of surfaces has increased relevance for several applications such as shape correspondences and matching, shape representation, shape modelling and many others. In this thesis we propose four local approaches to face the problems generated by the deformations of real objects and improving the pointwise characterization of surfaces. Differently from global approaches that work simultaneously on the entire shape we focus on the properties of each point and its local neighborhood. Global analysis of shapes is not negative in itself. However, having to deal with local variations, distortions and deformations, it is often challenging to relate two real objects globally. For this reason, in the last decades, several instruments have been introduced for the local analysis of images, graphs, shapes and surfaces. Starting from this idea of localized analysis, we propose both theoretical insights and application tools within the local geometry processing domain. In more detail, we extend the windowed Fourier transform from the standard Euclidean signal processing to different versions specifically designed for spectral geometry processing. Moreover, from the spectral geometry processing perspective, we define a new family of localized basis for the functional space defined on surfaces that improve the spatial lo-*

calization for standard applications in this field. Finally, we introduce the discrete time evolution process as a framework that characterizes a point through its pairwise relationship with the other points on the surface in an increasing scale of locality. The main contribute of this thesis is a set of tools for local geometry processing and local spectral geometry processing that could be used in standard useful applications. The overall observation of our analysis is that localization around points could factually improve the geometry processing in many different applications.

Contents

1 Introduction 15

1.1 Geometry processing 15

1.2 Deal with deformations 16

1.3 Need for locality 17

1.4 Our goals and contributions 19

1.5 Outline 20

2 Background 23

2.1 Signal processing 23

2.1.1 Fourier Transform 24

2.1.2 Properties of the Fourier transform 27

2.1.3 Domain dimensionality and geometry 28

2.2 Differential geometry 29

2.2.1 Manifolds as non euclidean domain 29

2.2.2 Spectral geometry processing 32

2.3 Discrete representation 33

2.4 Point-to-point matching 34

2.5 Data and Datasets 36

3 Diffusion process 41

3.1 Diffusion process 41

3.2 Heat kernel 43

3.3 Wave kernel 44
3.4 spectral diffusion descriptors 45
3.4.1 Heat Kernel Signature (HKS) 45
3.4.2 Wave Kernel Signature (WKS) 46
3.4.3 Signal Processing and Heat/Wave Kernel Signature 47

4 Advances in Functional Maps 51
4.1 Original framework 51
4.1.1 Fourier transform and the functional maps 54
4.1.2 Continuity 56
4.1.3 Linearity of the constraints 57
4.2 Region indicator functions in functional maps 59
4.2.1 Overview of the Proposed Method 61
4.2.2 RBC experiments 63
4.3 Functional maps for brain classification 67
4.3.1 Proposed Method 70
4.3.2 Functional maps kernel 72
4.3.3 Results 73
4.3.4 Limitations and future work 76

5 Localized spectral geometry processing 77
5.1 Need for spectral localization 77
5.2 Windowed Fourier transform 78
5.2.1 Overview 78
5.2.2 Standard Windowed Fourier transform 80
5.2.3 Localized spectral CNN 84
5.2.4 Results 89
5.3 Anisotropic windowed Fourier transform 96
5.3.1 Overview and related work 96
5.3.2 Need for orientation-sensitive 97
5.3.3 Background 98

5.3.4 Proposed method 100
5.3.5 Discretization 102
5.3.6 Applications and results 104
5.3.7 Conclusion and future work 111
5.4 LMH: a localized bases for spectral geometry processing 116
5.4.1 Limits of the standard basis 117
5.4.2 Localized manifold harmonics 120
5.4.3 Properties of LMH 124
5.4.4 Implementation 136
5.4.5 Applications 138
5.4.6 A non-optimal localized basis 150
5.5 Conclusions 157

6 Localized non-spectral geometry processing 159
6.1 Need for spatial localization 159
6.2 Discrete time evolution process 160
6.2.1 Overview of the framework 161
6.2.2 Discrete time evolution process on manifolds 163
6.2.3 Analysis of Higher Order Relations 166
6.2.4 Connection to Continuous process 169
6.2.5 Proposed Descriptor 171
6.2.6 Relation to Heat Kernel Signature 180
6.3 Results 182
6.4 Locality in discrete time evolution process and take home message 202

7 Conclusion 205

References 209

1

Introduction

1.1 Geometry processing

Geometry processing, is a recent field of research that uses concepts from mathematics, computer science and engineering to design efficient algorithms and tools for the processing and the analysis of geometric data.

In the last decades the interest in this discipline has greatly grown. This is due to increased technology and available devices in an ever wider range that allow the acquisition and generation of many geometric data from the reality. Geometric data can represent many different objects. Generally they encode surfaces, as two-dimensional manifolds embedded in a three-dimensional space. Thus we can say that the standard objects of study of geometry processing are all the possible objects that we can meet in the reality of every day.

In this thesis, we will refer to *shapes* as any kind of discrete or digital representation of surfaces. In the era of multimedia and automation, is necessary to allow computers to analyze, explore and learn about the world of real objects.

Applications of geometry processing algorithms already cover a wide range of areas from multimedia, entertainment and classical computer-aided design, to biomedical computing, reverse engineering and scientific computing. Some of these applications are acquisition, reconstruction, analysis, manipulation, simulation and transmission of complex 3D models.

Under this perspective, numerous traditional tools of mathematics and geometry can be adapted in order to investigate properties of shapes. This is the spirit of the work presented in this thesis, and geometry processing is the wide field to which this thesis belongs.

1.2 Deal with deformations

Working with shapes, seen as digital representation of objects from real world we have to deal with many issues. Leaving aside the problems related to representation, which are not addressed in this thesis, we still have to face many issues related to deformations. Indeed in the reality of every day, the same object or the same class of objects could be find in different deformed versions.

As a first example we can find a glass resting on a table or upside down. In the digital representation we would like to be able to understand that they are the same object and recognize the deformation between them. This is an example of *rigid deformation*, which means that the two objects are exactly the same object and they are related by a rigid motion or rigid transformations in $\mathbb{R}^3$. The real world is not rigid, most of the shapes can manifest themselves in different poses that differ for a *non rigid deformation*. An example of this is easily obtained replacing the glass with a cat on the table, and thinking to the multitude of positions that the cat can take. Among the non rigid deformations we can identify two subclasses: isometric and non isometric deformations. With *isometric deformation* or *intrinsic isometry* we indicate a transformation between two shapes that preserves the distances induced by the metrics of the surfaces. With a more mathematical rigour we can say:

Definition 1.1. *Let $\mathcal{M}$ and $\mathcal{N}$ be two compact Riemannian surfaces without boundary, a map $T : \mathcal{M} \to \mathcal{N}$ that is surjective is an* isometric deformation *or* intrinsic isometry *if and only if $\forall x,\ y \in \mathcal{M}$, $d_{\mathcal{M}}(x, y) = d_{\mathcal{N}}(T(x), T(y))$. Where $d_{\mathcal{M}}(\cdot,\cdot)$ and $d_{\mathcal{N}}(\cdot,\cdot)$ are distances induced by the Riemannian metric respectively on $\mathcal{M}$ and on $\mathcal{N}$.*

A non isometric deformation is a deformation which is not an isometry. The distances induced by the Riemannian metric are called *Geodesic distances*. The geodesic distance between two points $x,\ y \in \mathcal{M}$ can be thought as the minimum length of the paths connecting x to y and belonging to $\mathcal{M}$. Intrinsic means that a definition or a quantity can be fully expressed in terms of the metric of the surface, and so it is independent from the embedding. Conversely we indicate with extrinsic all stuff that are related with the embedding. In the next Chapter an introduction to Remannian geometry and some reference about this topic will be given. For simplicity in this thesis we will not worry about the surjectivity of the map T and we will refer to T as an isometry (by omitting intrinsic) when T preserves geodesic distances. In the discrete representation of the real world, we usually deal with *near isometric deformation*, that is a deformation that is close to an isometric deformation but for some points or regions does not completely preserve the isometry. Among the non isometric deformation we have many types of deformations such as topological deformations, scale deformations, and others that it is possible to see in real world or in its discrete representation.

1.3 Need for locality

Due to this large amount of deformations we can meet working with non-rigid 3D real objects, it is not easy to analyze the shape entirely, because a surface and its deformations can be very different when viewed globally.

In the same way limitations in the acquisition of real data often generate different types of noise such as topological errors, partial views, missing parts, widespread noise on the surface just to list some of them. All this errors are usually localized on some regions or around some points. A global approach in the comparison of shapes may be very subject to the presence of these kinds of error and at the same time global approach makes these errors impossible to be localized.

Furthermore in many standard applications of geometry processing the attention for pointwise and localized analysis is increased. Indeed the global analysis of shapes has some clear limitations. As example consider two isometric deformations of a given human shape in which harms and legs are in different positions. In many applications such as animation and anatomical studies, being able to separate these deformations is absolutely desirable. From a global point of view it is impossible to separate the harms from the legs deformations. A correspondence between points, or a subdivision of the shapes in local semantically consistent segments make the distinction between the different deformations possible. The local approach is also related with the Riemannian surfaces that we adopt for representing 3D non-rigid objects. Indeed the definition of Riemannian surface is strictly related to a local characterization of the surface, and it describes each point through its neighborhoods.

Also in discrete settings the localized analysis is used to improve the representation of data, as done as example in [35], where many different advantages given by the localized analysis of discrete data are shown.

Gathering together all these reasons we believe that in the geometry processing there is a clear *need for localization*. With localization we mean a set of theoretical insights and practical methods which reinforces and permits localized analysis of surfaces.

1.4 Our goals and contributions

Localized analysis on shapes is a task of great interest in geometry processing. This interest is mainly due to the importance of a pointwise characterization and description of shapes in several applications and problems. Real objects are subject to a countless number of deformations. It can be not easy to deal with these deformations if we look at the shapes globally. The local analysis can help to face the problems caused by these deformations.

Therefore, the main goal of this thesis is to improve the localized analysis and description of shapes. Our final objective is to produce theoretically sound tools for the localized analysis of deformable surfaces, in order to be more robust with respect to several deformations of the shapes and being more informative with respect to the properties of points and their neighborhoods.

From the theoretical point of view our contributions is fourfold:

1. Given a pair of shapes we propose a new *functional kernel* which allows the comparison of shapes using only their functional representations.
2. We extend the definitions of some standard signal processing tools on surface domain such as the *windowed Fourier transform* and its anisotropic version that is strictly related with *Gabor transform.*
3. We introduce a new family of basis which could be localized on the spatial domain allowing us to obtain a good spectral localization together with a good spatial localization and improving the representation of signals defined on manifolds.
4. We define the *discrete time evolution process* on surfaces.

From the practical side, we investigate how these new methods could be used in different applications. We adopt the functional kernel to the brain surfaces classification in order to detect diseases among the analyzed subjects. We exploit a biclustering technique to find stable regions between shapes. We derive point descriptors for the point-to-point matching from the windowed Fourier transform, its anisotropic version and from the discrete time evolution process framework. From the anisotropic windowed Fourier transform we also obtain a salient point detector and a new stable method for shape segmentation. We show how our localized basis for spectral geometry processing can improve spectral shape processing and the shape correspondence applications. Finally we test how the operator derived from the discrete time evolution process can improve the shape correspondence. We expect that these tools and the related theory can be involved to improve many geometry processing applications such as shape retrieval, shape correspondences, shape segmentation, shape modelling and many others.

1.5 Outline

The rest of this thesis is organized as follows: In Chapter 2 an overview of the main subjects is given. More precisely we focus on the theoretical basis of this thesis: signal processing, differential geometry, discrete representation and spectral differential geometry as the confluence of the three previous ones. At the end of Chapter 2 the point-to-point matching is presented as the main application faced in this thesis together with the datasets used in the experiments.

Chapter 3 is dedicated to diffusion process as an example of local analysis on surfaces. Diffusion process allows us to introduce *heat kernel signature, wave kernel signature* and the *optimal shape descriptor*.

These two Chapters are introductory to our contribution, which is instead completely contained in the following Chapters. An overview of the Functional maps framework is given at the beginning of Chapter 4. The original framework is then improved using the new stable region indicator functions obtained from our biclustering technique. Our functional kernel for brain classification, a new application of the functional maps, is introduced at the end of this Chapter. In Chapter 5 we extend the windowed Fourier transform tools to surfaces allowing the localized spectral geometry processing. The localized manifold harmonics, our localized basis for spectral geometry processing concludes this Chapter. Discrete time evolution process on surfaces is introduced in Chapter 6, accompanied by an analysis on the localization in a non spectral geometry processing approach. Finally Chapter 7 summarized the conclusions of this thesis and possible future works that may arise from it.

2

Background

In Chapter 1 we have already given a general overview of the topics covered in this thesis. In this Chapter we introduce the theoretical background that is required to understand our original contribution. In the last Section 2.5 of this Chapter many notation an data used in this thesis are introduced.

2.1 Signal processing

Signal processing is a technology that covers important roles in many different areas. Signal processing is the representation, the manipulation and the transformation of signals. From an abstract point of view a signal is a function that " conveys information about the behaviour or attributes of some phenomenon ", [123]. From mathematical perspective signals are represented as functions of one or more independent variables. In practice a signal is any physical behaviour or variation of a quantity in some domain, such as in time domain or in space domain. This domain is the set in which the independent variables of the function "signal" could vary. Some example of signal are music, video, image, speech, radar and many others.

Signals are divided in two main categories: continuous and discrete signals. Continuous signals are defined on a continuous domain, i.e., the signal varies continuously with respect to its independent variables. Discrete signals are instead defined on a discrete set or a discrete subset of a continuous space. In this case the independent variables have discrete values and so the signal is a discrete collection of values. Also the independent variables could be continuous or discrete, so as the realized values of the function "signal" could be either continuous or discrete. Signal for which both the independent variables and the realized values are discrete are said digital signals. In this thesis we mainly focus on discrete signal and therefore the independent variables (one or more) belong to a discrete domain. A complete and deep analysis of discrete signal processing can be found in [113].

Signal processing importance is twofold, first due to wide diffusion of signals in many applications, second signal processing solves many and different problems involving signals.

One example of problem is Signal representation, which means the process of transforming the original signal in a new version that possess certain properties. The new signal is a faithful representation of the original one, which contains all the important original information. At the same time the new signal has some properties that make it more useful, such as for example compactness, homogeneity, simplicity and others.

Another important class of problems is Signal interpretation. The goal of Signal interpretation is to obtain a characterization of the signal. The obtained characterization highlights some content or property of the signal that is not so evident in the original signal.

Manipulation and transformation of a signal is another family of problem faced by signal processing. The objective in this case is to create new and unseen signals.

2.1.1 Fourier Transform

A key problem in signal processing is the choice of the representation adopted for signals. The desired representation must be informative, concise and must allow and simplify operations on signals. Several representations have been proposed, but in this work we will limit our analysis to the Fourier Transform. Its name is due to Joseph Fourier, which in 1807 in the paper “Mémoire sur la propagation de la chaleur dans les corps solides” [48], introduced and developed the theory of this transform. The goal of Fourier was to study a way in which general functions (signals) may be represented and approximated as linear combinations of simpler functions. These simpler functions are typically trigonometric such as sines and cosines. Fourier was motivated by the search for a simpler representation of heat transfer. Namely we refer to this representation as the *Fourier transform* also referred as the frequency representation of the original signal. In other words, if we have a signal defined on a certain domain, such as time for sound, the Fourier transform represents the signal into the frequencies that generate it.

More formally, in the continuous and 1-dimension case, given an integrable function $f : \mathbb{R} \longrightarrow \mathbb{C}$ we can define the Fourier transform of f as follows:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \omega x} dx \tag{2.1}$$

The Fourier transform gives us a coefficient $\hat{f}(\omega)$ for every frequency $\omega \in \mathbb{R}$, which corresponds to the amount of that frequency in the analyzed signal. The collection of these values $\hat{\mathbf{f}} = \{\hat{f}_\omega\}_{\omega\in\mathbb{R}}$ is the representation in the frequency domain of the original signal f.

Usually it is possible to define an inverse Fourier transform, which allows the reconstruction of the original signal starting from its representation in the frequency domain.

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{2\pi i x \omega} d\omega \tag{2.2}$$

The inverse Fourier transform is referred as Fourier synthesis.

Thanks to the Fourier transform is possible to find a direct connection between some operations in the original domain and other operations in the frequency domain as we will see later in this Chapter.

Here we clarify how the Fourier transform allows us to represent functions as linear combinations of simpler sines and cosines function. Thanks to the Euler's formula sines and cosines can be rewritten as complex exponential functions:

$$e^{2\pi ix} = cos(2\pi x) + isin(2\pi x),\ \forall x \in \mathbb{R}. \tag{2.3}$$

Replacing the exponential in Equation 2.1 with the last equivalence 2.3 we get the linear combinations of trigonometric functions.

A more general way to refer to this representation of functions is harmonic analysis. The name harmonics is due to the ancient Greek word, *harmonikos*, that means *skilled in music*. Harmonics is uses to indicate waves whose frequencies are integer multiples of a fixed frequency. This is the case of frequencies of music notes, that are indeed harmonics waves. The use of harmonics has been generalized beyond its original meaning in many other contexts.

Harmonics and music are strictly related with Fourier transform thanks to the work of a famous physicist Ernst Chladni. In 1787 Chladni published the book "Discoveries Concerning the Theories of Music", in which he proposed his experiments about the patterns that the sand assumes if it is spread on a thin metal plate putting into vibration using a bow. Briefly these patterns can be mathematically calculated solving the eigendecomposition of a standard operator.

Given an operator **Q** from a domain to itself, the *eigendecomposition* of **Q** is given by the (possibly infinite) set of scalars (namely *eigenvalues*) $\{\lambda_i\}_i$ and corresponding points in the given domain (namely *eigenvectors*) $\left\{\phi_i\right\}_i$ for which the following equation is verified:

$$\mathbf{Q}\phi_i = \lambda_i\phi_i,\ \forall i \tag{2.4}$$

The considered operator is the *Laplace operator*, a classical second order differential operator. The eigenfunctions of the Laplace operator are exactly the harmonics used as basis of the Fourier transform. Just as a historical remark, this relation between sound and Laplace eigenfunctions leads to the question asked by Kac in 1966: *Can we hear the shape of a drum?* [72].

2.1.2 Properties of the Fourier transform

In general several hard operations in the original domain of a signal can be easily performed in the frequency domain, and the frequency representation has many interesting properties. In the following we resume some of these properties omitting proofs that go beyond the aims of this thesis. A function f is a square-integrable function if $\int_{-\infty}^{\infty} |f(x)|^2 dx < \infty$. For every square-integrable functions $f,g : \mathbb{R} \longrightarrow \mathbb{C}$, we define the convolution between f and g as:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(x)g(x-t)dt\,. \tag{2.5}$$

Convolution can be viewed as a modified version of one of the original functions, defined as a weighted average of the first function weighted by a reversed and shifted version of the second function. For every square-integrable functions $f,g : \mathbb{R} \longrightarrow \mathbb{C}$, and their Fourier transform $\hat{f}$ and $\hat{g}$ we have:

Linearity: if $h(x) = af(x) + bg(x)$ then $(\hat{h})(\omega) = a\hat{f}(\omega) + b\hat{g}(\omega)$, $\forall a,\ b \in \mathbb{C}$;

Translation shifting: if $h(x) = f(x - x_0)$ then $\hat{h}(\omega) = \hat{f}(\omega)e^{-2\pi i\omega x_0}$, $\forall x_0 \in \mathbb{R}$;

Modulation shifting: if $h(x) = f(x)e^{2\pi i\omega_0 x}$ then $\hat{h}(\omega) = \hat{f}(\omega - \omega_0)$, $\forall \omega_0 \in \mathbb{R}$;

Integration: $\hat{f}(0) = \int_{-\infty}^{\infty} f(x)e^{2\pi i 0x} dx = \int_{-\infty}^{\infty} f(x)dx$, i.e. the value of $\hat{f}(0)$ is equal to the integral of f on its domain;

Parseval's formula: $\int_{-\infty}^{\infty} f(x)\overline{g(x)}dx = \int_{-\infty}^{\infty} \hat{f}(\omega)\overline{\hat{g}(\omega)}d\omega$;

Plancherel's theorem: $\int_{-\infty}^{\infty} |f(x)|^2 dx = \int_{-\infty}^{\infty} |\hat{f}(\omega)|^2 d\omega$, that follows directly from the Parseval's formula. From Plancherel's theorem we can said that the Fourier transform preserves the energy of the original signal;

Convolution theorem: if $h(x) = f(x) * g(x)$ then $\hat{h}(\omega) = \hat{f}(\omega)\hat{g}(\omega)$.

This latter property is a clear case in which a complicated operation such as convolution is solved in the frequency domain as a simple product. The convolution theorem will be very important in this thesis as we will use this theorem to extend convolution from the classical case to more generalized versions.

Another key property of the Fourier transform is the **uncertainty principle**. Without going into details this principle states that for a signal it is not possible to have in the same time a localization in the original domain and a localization in the frequency domain. In other words if we have a signal that is squeezed around some point in the original domain then we have something more spread in the frequency domain. It is thus impossible to arbitrarily concentrate both a function in its original domain and its Fourier transform in the frequencies domain.

2.1.3 Domain dimensionality and geometry

Signal processing is classically defined for one-dimensional signals, or rather for functions of one independent variable with value in $\mathbb{R}$. Standard examples of one-dimensional signal are sound that varies in time, electric potential that varies along the points of a circuit. Most of the signal processing theory and definitions are born in the context of one-dimensional signals, that are more common and easy to control. Anyway signal processing problems are not confined to one dimensional signals. Although there are some fundamental differences in the theory between one-dimensional and multi-dimensional signals most of the concepts and techniques can be extended directly to multi-dimensional signal processing. Another important extension is related to the geometry of the domain on which signal are defined. In the classical case we have a one or multi-dimensional euclidean domain on which operations and measures are well defined. If we want to study signals on some non euclidean domain, we have to face some issues generated by the different geometry of the domain.

2.2 Differential geometry

2.2.1 Manifolds as non euclidean domain

A particular class of non euclidean domains is the one faced in this thesis: 2-dimensional surfaces embedded in $\mathbb{R}^3$. As we saw in Chapter 1 these surfaces represent real objects and for simplicity we will refer to these objects as *shapes*. We approximate and model a shape as a smooth compact connected Riemannian surface $\mathcal{M}$ (possibly with a boundary $\partial\mathcal{M}$) embedded into $\mathbb{R}^3$. A complete and detailed introduction to manifolds is out of the scope of this thesis. In the following we will introduce some basic concepts and definitions that will be useful in our analysis. For a deep introduction to Riemannian manifolds we refer the reader to [41].
Let $\mathcal{M}$ be a Riemannian manifold as previously described. We consider a generic point $x \in \mathcal{M}$. Locally around the point x, the manifold is homeomorphic to the *tangent space* (or *plane*) $T_x\mathcal{M}$. The disjoint union of all the tangent spaces is the *tangent bundle* $T\mathcal{M}$. We further equip the manifold $\mathcal{M}$ with a *Riemannian metric*, defined as an inner product $\langle\cdot,\cdot\rangle_{T_x\mathcal{M}} : T_x\mathcal{M} \times T_x\mathcal{M} \to \mathbb{R}$ on the tangent space depending smoothly on x. Properties expressed solely in terms of the metric are called *intrinsic*. In particular, *isometric* (metric-preserving) deformations of the embedded manifold preserve all intrinsic structures.

Let $f : \mathcal{M} \to \mathbb{R}$ and $F : \mathcal{M} \to T\mathcal{M}$ denote real *scalar* and *tangent vector fields* on the manifold, respectively. We can define inner products as:

$$\langle f, g\rangle_{L^2(\mathcal{M})} = \int_{\mathcal{M}} f(x)g(x)\mathrm{d}x; \tag{2.6}$$

$$\langle F, G\rangle_{L^2(T\mathcal{M})} = \int_{\mathcal{M}} \langle F(x), G(x)\rangle_{T_x\mathcal{M}}\mathrm{d}x; \tag{2.7}$$

where, $\mathrm{d}x$ is the area element induced by the metric. We denote by $L^2(\mathcal{M}) = \{f : \mathcal{M} \to \mathbb{R} \mid \langle f, f\rangle^2_L(\mathcal{M}) < \infty\}$ the space of square-integrable real functions on $\mathcal{M}$.

Laplace-Beltrami operator.

In classical calculus, the notion of derivative describes how the value of a function f changes with an infinitesimal change of its argument x. Due to the lack of vector space structure on the manifold (meaning that we cannot add two points, $x + dx$), we need to define the *differential* of f as an operator $\mathrm{d}f : T\mathcal{M} \to \mathbb{R}$ acting on tangent vector fields. At each point x, the differential is a linear functional $\mathrm{d}f(x) = \langle \nabla f(x), \cdot\rangle_{T_x\mathcal{M}}$ acting on tangent vectors $F(x) \in T_x\mathcal{M}$, which model a small displacement around x. The change of the function value as the result of this displacement is given by applying the differential to the tangent vector, $\mathrm{d}f(x)F(x) = \langle \nabla_{\mathcal{M}} f(x), F(x)\rangle_{T_x\mathcal{M}}$, and can be thought as an extension of the notion of the classical directional derivative. The operator $\nabla_{\mathcal{M}} f : L^2(\mathcal{M}) \to L^2(T\mathcal{M})$ in the above definition is called the *intrinsic gradient*, and is similar to the classical notion of the gradient defining the direction of the steepest change of the function at a point.

The *intrinsic divergence* $\mathrm{div}_{\mathcal{M}} : L^2(T\mathcal{M}) \to L^2(\mathcal{M})$ is defined as an operator adjoint to the intrinsic gradient, $\langle F, \nabla_{\mathcal{M}} f\rangle_{L^2(T\mathcal{M})} = \langle -\mathrm{div}_{\mathcal{M}} F, f\rangle_{L^2(\mathcal{M})}$, where $f \in L^2(\mathcal{M})$ and $F \in L^2(T\mathcal{M})$ are some scalar and vector fields, respectively.

The positive semi-definite *Laplace-Beltrami operator* (LBO) is defined as:

$$\Delta_{\mathcal{M}} f = -\mathrm{div}_{\mathcal{M}}(\nabla_{\mathcal{M}} f) \ ,$$

generalizing the corresponding notion of Laplace operator from Euclidean spaces to manifolds. The Laplacian is self-adjoint,

$$\langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} g \rangle_{L^2(T\mathcal{M})} = \langle \Delta_{\mathcal{M}} f, g \rangle_{L^2(\mathcal{M})} = \langle f, \Delta_{\mathcal{M}} g \rangle_{L^2(\mathcal{M})}. \tag{2.8}$$

Geometrically, the Laplace-Beltrami operator can be interpreted as the (normalized) difference between the average of a function on an infinitesimal sphere around a point and the value of the function at the point itself.

Spectral analysis on manifolds.

Given a compact manifold $\mathcal{M}$ with boundary $\partial\mathcal{M}$, the LBO as self-adjoint and positive semi-definite operator, admits an orthonormal eigendecomposition [29]:

$$\Delta_{\mathcal{M}} \phi_i(x) = \lambda_i \phi_i(x) \quad x \in \mathrm{int}(\mathcal{M}) \tag{2.9}$$

$$\langle \nabla_{\mathcal{M}} \phi_i(x), \hat{n}(x) \rangle = 0 \quad x \in \partial\mathcal{M}, \tag{2.10}$$

with Neumann boundary conditions (2.10), where $\hat{n}$ is the normal vector to the boundary. Here, $0 = \lambda_1 \leq \lambda_2 \leq \ldots$ is a countable set of non-negative real eigenvalues and $\phi_1, \phi_2, \ldots$ are the corresponding orthonormal eigenfunctions satisfying $\langle \phi_i, \phi_j \rangle_{L^2(\mathcal{M})} = \delta_{ij}$.
In analogy to the Euclidean case, the LBO $\Delta_{\mathcal{M}}$ endows us with the Fourier analysis on surfaces. The Laplacian eigenfunctions form indeed an orthonormal basis for $L^2(\mathcal{M})$ that are thus referred as *manifold harmonics (MH)*. A function $f \in L^2(\mathcal{M})$ can therefore be expressed as the *Fourier series*:

$$f(x) = \sum_{i \geq 1} \underbrace{\langle \phi_i, f \rangle_{L^2(\mathcal{M})}}_{\hat{f}_i} \phi_i(x), \tag{2.11}$$

where $\hat{f}_i$ are the Fourier coefficients of the Fourier transform). The synthesis $f = \sum_{i \geq 1} \hat{f}_i \phi_i(x)$ is the inverse Fourier transform. The eigenvalues λ_i can be interpreted as frequencies in the classical harmonic analysis. Truncating the series (2.11) to the first k terms will result in a band-limited (with bandwidth λ_k) representation of f.
Thanks to this extension of the Fourier transform to surfaces and starting

from the convolution theorem and equation 2.5, we can generalize the definition of the *convolution* between two functions f and $g \in L^2(\mathcal{M})$ as:

$$f * g = \sum_{i \geq 1} \hat{f}_i \hat{g}_i \phi_i(x), \tag{2.12}$$

where $\{\hat{f}_i \hat{g}_i\}_{i \geq 1}$ are the Fourier coefficients of the convolution $f * g$.

Dirichlet energy

Given a function $f \in L^2(\mathcal{M})$, the *Dirichlet energy* of f is defined as:

$$\mathscr{E}_S(f) := \langle \nabla_{\mathcal{M}} f, \nabla_{\mathcal{M}} f \rangle_{L^2(T\mathcal{M})} = \langle f, \Delta f \rangle_{L^2(\mathcal{M})} \tag{2.13}$$

measures how 'smooth' the function is. It is possible to show that the Laplacian eigenbasis is the solution to the optimization problem

$$\min_{\psi_1,\dots,\psi_k} \sum_{i=1}^{k} \mathscr{E}_S(\psi_i) \text{ s.t. } \langle \psi_i, \psi_j \rangle_{L^2(\mathcal{M})} = \delta_{ij} \tag{2.14}$$

and thus it can be considered as the smoothest possible orthonormal eigenbasis. Furthermore, the eigenvalues can be obtained as the values of the Dirichlet energy, $\mathscr{E}_S(\phi_i) = \lambda_i$.

2.2.2 Spectral geometry processing

Spectral geometry processing was proposed at the beginning of the 90's as the attempt to move tools from signal processing to the 2-dimensional surfaces setting. This vision takes the surfaces as the domain on which signals are defined, and replace the analysis of the points seen as their embedding in $\mathbb{R}^3$ with the analysis of signals defined on the surface. There are many classical contexts in which signal processing gives a new solution for many complicated problems. Spectral geometry processing aims to find the same advantage in Geometry processing. As we saw above Fourier transform is one of the milestone in classical signal processing and it is at the base of many standard techniques, such as manipulation, filtering and compression of signal. In analogy with standard signal processing also in spectral geometry processing the definition of the Fourier analysis on surfaces allows to extend many techniques on shape. We refer the reader to [87] for a complete of definition and basic constructions in spectral geometry processing. In particular for the parallel with Fourier analysis please refer to [146]. Many other paper could be cited for spectral geometry processing, we propose just some of them, [86], [143], [142].

2.3 Discrete representation

In the discrete setting, a surface $\mathscr{M}$ is represented as discrete samplings from $\mathscr{M}$ itself, that is a collection of points in $\mathbb{R}^3$. Mostly, two different discrete representations are used:

point clouds: just the collections of points sampled from the surface given as coordinates in the Euclidean space $\mathbb{R}^3$.

polygonal meshes: in addition to points is also given a connectivity among points that improve the representation of the surface.

In this thesis, except for special cases, the manifold $\mathscr{M}$ is sampled at N points $x_1, \dots, x_N$ and is approximated by a triangular mesh (V, E, F) constructed upon these points, where $V = \{1, \dots, n\}$, $E = E_\mathrm{i} \cup E_\mathrm{b}$ and F are the vertices, edges, and faces of the mesh, respectively (E_i and E_b denote the interior and boundary edges, respectively). The discretization of the Laplace-Beltrami operator $\Delta_{\mathscr{M}}$ takes the form of an $N \times N$ sparse matrix $\mathbf{L} = -\mathbf{A}^{-1}\mathbf{W}$ according to the standard lumped linear FEM [94]. The *mass matrix* $\mathbf{A}$ is a diagonal matrix of area elements $\mathbf{A}_i = \frac{1}{3}\sum_{jk:ijk \in F} \mathbf{A}_{ijk}$, where $\mathbf{A}_{ijk}$ denotes the area of triangle ijk. The *stiffness matrix* $\mathbf{W}$ contains the *cotangent weights* [120]:

$$w_{ij} = \begin{cases} (\cot \alpha_{ij} + \cot \beta_{ij})/2 & ij \in E_{\mathrm{i}}; \\ (\cot \alpha_{ij})/2 & ij \in E_{\mathrm{b}}; \\ -\sum_{k \neq i} w_{ik} & i = j; \\ 0 & \text{else;} \end{cases} \tag{2.15}$$

where α_{ij}, β_{ij} denote the angles $\angle ikj, \angle jhi$ of the triangles sharing the edge ij. Note that as in the continuous case the eigenvectors are **A**-orthonormal, i.e. $\Phi^\top \mathbf{A} \Phi = \mathbf{I}$.
Functions $f \in L^2(\mathcal{M})$ are represented as N-dimensional vectors

$$\mathbf{f} = (f(x_1), \ldots, f(x_N))^\top ,$$

the inner products $\langle f, g \rangle_{L^2(\mathcal{M})}$ are discretized by area-weighted dot products $\mathbf{f}^\top \mathbf{A} \mathbf{g}$.

2.4 Point-to-point matching

In this Section we would like to briefly introduce one of the most well-known geometry processing application: *point-to-point matching*. This Section does not give a complete overview of all the methods related to this application, but rather outline a classical application that will be faced in this thesis. This allows the reader to address most of the experimental part of this thesis.

Point based matching plays an important role in shape analysis and geometry processing. The key approach consists of defining an effective point *signature* (or descriptor) able to capture the most notable local shape characteristic or *feature* in the neighborhood of the point. Shape descriptors are commonly used in a wide range of geometry processing applications, such as correspondence, segmentation, labelling, and retrieval. A shape descriptor is a method for describing the local behaviour of the surface around some point, which is captured by a multi-dimensional vector. The set of descriptors for all the points of the surface can be thought of as a vector field thereon. Typically, one wishes a descriptor that is *discriminative* (highlighting distinctive attributes), *robust* (invariant with respect to noise and deformations) *compact* (using a small number of dimensions), and *computationally-efficient*. There is a plethora of literature on geometric shape descriptors, and we refer the reader to a recent survey for a comprehensive overview [88]. Descriptors like spin images [70], shape distributions [114], integral volume descriptors [98], and multi-scale features [119] are based on extrinsic structure of the shape and therefore invariant under Euclidean transformations, but not under non-rigid deformations. One of the first works to deal with deformations was Elad and Kimmel [46] employing multidimensional scaling to represent the geodesic distance metric in the Euclidean space. Other descriptors based on geodesic were proposed in [59], while [8] used conformal factors. Spectral descriptors try to exploit the geometry arising from the eigenfunctions and eigenvalues of the Laplace-Beltrami operator of the surface [9,35,86]; popular methods include shapeDNA [125], global point signature (GPS) [133], heat kernel signatures (HKS) [50, 140], wave kernel signatures (WKS) [5], and heat kernel maps [117]. HKS ans WKS will be widely discussed in the following Chapter 3, as a derivation of diffusion processes. Another class of approaches try to bring successful models like SIFT [93] or shape context, [7] from images to surfaces [79,139]. Following the recent image processing trend of learning invariant structure rather than trying to hand-craft them, several learning frameworks have been proposed in the geometry processing community as well, for applications such as correspondence [131], retrieval [89], labelling and segmentation [63, 73]. Several methods for learning descriptors have appeared very recently [36,90,99]. The main advantage of learning methods is, instead of trying to create a generic descriptor that would work well for all kinds of shapes, to design a *class-specific* descriptor that would address fine-grained differences between shapes in the class, and particular type of noise or deformations.

Global Point Signature (GPS)

To conclude this subsection we would like to give an example of spectral descriptor that will be important for our further discussion. This example also makes us possible to anticipate some limits of the LBO and its eigendecompositions which will be widely discussed in the following chapters.
Rustamov [133] proposed the *global point signature (GPS)* embedding, a dense shape descriptor constructed using scaled LBO eigenfunctions,

$$\mathbf{f}(x) = (\lambda_1^{-1/2}\phi_1(x), \ldots, \lambda_Q^{-1/2}\phi_Q(x))^\top, \tag{2.16}$$

thus associating each point x with a Q-dimensional descriptor (see [9,35] for earlier constructions in the theoretical math community).

Due to an inherent ambiguity in the definition of the LBO eigenbasis, GPS descriptors cannot be matched in a simple-minded manner. First, an eigenfunction is defined up to sign, $\Delta_{\mathcal{M}}(\pm\phi_i) = \lambda_i(\pm\phi_i)$. Second, if an eigenvalue with non-trivial multiplicity is present in the spectrum of $\Delta_{\mathcal{M}}$, any rotation in the corresponding subspace produces valid eigenfunctions. Third, noise and non-isometric deformations may alter the eigenvalues and eigenfunctions of the LBO. Trying to cope with these ambiguities, several techniques have been proposed trying to match GPS descriptors (see, e.g. [100]). This overview is not exhaustive, and we refer the reader to the cited related works and references therein for a more complete picture.

2.5 Data and Datasets

This Section contains references and informations about some of the data and datasets used in this thesis. With dataset we mean a collection of shapes represented as triangular meshes that are available for research activity. Many different datasets are available for geometry processing, each of which is characterized by the class or classes of shape contained, the type of problems that they present, and different ground-truth informations that they may possess.

FAUST [13] is a recent dataset of scanned human shapes in different poses. The dataset is challenging due to the significant variability between different human subjects. Ground truth point-wise correspondence between the shapes is available for all points. All of these meshes have the same connectivity. We use the whole FAUST dataset consisting of 100 shapes, 10 poses of 10 different subjects respectively, along with additional shapes that have been edited by adding different types of noise: Gaussian noise, heavy subsampling, voxelization noise, topological noise (glued fingers and missing parts). In addition are also available the real scans of humans from which the dataset is generated. These real scans are high resolution non-watertight mesh. These meshes are more noisy, without registration and without ground truth correspondences available.

Princeton Segmentation Benchmark (PSB) [31] is a large dataset that contains 3PSB80 triangular meshes equally divided in 19 categories, so for every category there are (PSB)20 elements. A common meaningful segmentation is given as ground truth for each category, this segmentation was defined from the manually segmentations of the surface meshes dividing them into functional parts, done by eighty people and yielding an average of 11 human-generated segmentations. We limit ourself to a subset of 8 categories for reasons linked to the memory and due to the segmentation method used.

SCAPE [4] similarly to FAUST is a dataset containing a scanned human model in different poses. SCAPE datasets contains 71 registered triangular meshes of a particular person in different poses, with more than 10K vertices. A ground truth point-wise correspondence between the shapes is known.

TOSCA dataset [19], comprises 7 different shape classes (centaur, horse, two male subjects, female, cat, and dog). All these shapes are synthetic models. In each such class, a “null” shape and some different near-isometric deformations are available. Despite the deformations this dataset has a strong isometry between elements of the same class. The number of vertices of these shapes is different for each class and goes from around 10K to around 30k. A ground truth point-wise correspondence between the shapes in the same class is available for all points.

CAESAR is a human shapes dataset recovered from MPII Human Shape [121], a family of expressive 3D human body shape models learned from CAESAR dataset [126] the largest commercially available dataset that contains 3D scans of over 4500 subjects in a standard pose. We use a random selected subset of 21 shapes from the *CAESAR-fitted meshes* collection in which a template is fitted. For every shape we have around 6k vertices with 1:1 ground-truth correspondence.

KIDS [131] consists of a collection of 3D shapes undergoing nearly-isometric and intraclass deformations. In this dataset we find two different shape classes (*kid* and *fat kid*) in 16 different poses. The same poses are applied to both classes. The authors provide all shapes with consistent triangulations using around 60k vertices consistently ordered giving the ground-truth correspondence as the identity map.

MISC dataset is composed of pairs of highly non-isometric shapes such as a horse and an elephant. Therefore, this dataset is particularly challenging since the usual hypothesis of isometric relations between shapes is totally violated. Manually generated ground truth point-wise correspondences are available for a dense subset of points in this dataset.

SHREC'11 benchmark [17]. The class includes one full human shape (i.e., the *null* shape) and 5 versions of its simulated transformations of pose deformation with strong deformations of different types.

A Benchmark for 3D Interest Point Detection Algorithms, this dataset [43] was created specifically for the salient point detection application. Meshes are chosen from the Stanford 3D Scanning Repository and some others from the SHREC'2007. There are two dataset available. Dataset A consists of 24 surfaces which were hand-marked by 23 human subjects. Dataset B with 43 models, that are marked at least by 16 subjects. On these dataset is thus provided a human-generated ground truth of salient points that is obtained from the hand-marked selection made by several subjects. The authors define also a new evaluation criterion (WME) based on importance of the selected points, where importance is based on the same hand-marked selections. The output of six state of art methods are provided but not for all the models. For this reason we limit our tests on the 21 available models in the dataset A.

3

Diffusion process

In this Chapter we briefly introduce diffusion process on shapes. We present two alternatives that have been proposed during the last decades. At the end of this chapter we describe some descriptors that derive from this kind of process, and that are strictly related with spectral geometry processing. In our idea this chapter should be a first step in the geometry processing word. We would like to use these discussion in order to make the reader more familiar with tools, properties of shapes and their spectral representation.

3.1 Diffusion process

We introduce the reader to the diffusion process using the point of view proposed by the authors of [35]. In [35] diffusion process is seen as a tool "for finding meaningful geometric descriptions of data sets". The diffusion process theory is general and applicable to many different domains as done in [35]. In this thesis our interest is limited to the case where the domain $\mathcal{M}$ is a compact Riemannian manifold. Consistently with Chapter 2, in the discrete setting we represent $\mathcal{M}$ as a triangular mesh, as is done also in [18] that will be the main reference of this Chapter. Let $\mathrm{d}x$ the standard area measure on $\mathcal{M}$. A function $k : \mathcal{M} \times \mathcal{M} \to \mathcal{M}$ is called *diffusion kernel* if it satisfies the following properties [18]:

$(K1)$ *Non-negativity*: $k(x,x) \geq 0$.
$(K2)$ *Symmetry*: $k(x,y) = k(y,x)$.
$(K3)$ *Positive-semidefiniteness*: for every bounded f,

$$\iint k(x,y) f(x) f(y) \, \mathrm{d}x(x) \, \mathrm{d}x(y) \geq 0.$$

$(K4)$ *Square integrability*: $\iint k^2(x,y) \, \mathrm{d}x(x) \, \mathrm{d}x(y) < \infty$.
$(K5)$ *Conservation*: $\int k(x,y) \, \mathrm{d}x(y) = 1$.

Starting from a diffusion kernel it is possible to define a linear operator, namely a *diffusion operator*:

$$\mathbf{K}f = \int k(x,y) f(y) \, \mathrm{d}x(y).$$

The properties that define k guarantee us that the diffusion operator $\mathbf{K}$ admits a an eigendecomposition with eigenvalues $\{\alpha_l\}_{l=0}^{\infty}$ s.t. $0 \leq \alpha_l \leq 1$, and their associated eigenvectors $\{\phi_l\}_{l=0}^{\infty}$, which form an orthonormal basis of $L^2(\mathcal{M})$. Thanks to the *spectral theorem* we can give the following spectral representation of the diffusion kernel k:

$$k(x,y) = \sum_{l=0}^{\infty} \alpha_l \phi_l(x) \phi_l(y). \tag{3.1}$$

Moreover, it is easy to show that for every $t \in \mathbb{R}$ and $t \geq 0$, $\mathbf{K}^t$ is again a diffusion operator which is the diffusion operator associated to the diffusion kernel $k_t(x,y)$ defined similarly to 3.1, as $\mathbf{K}^t$ inherits the eigenfunctions from $\mathbf{K}$, and its eigenvalues correspond to $\{\alpha_l^t\}_{l=0}^{\infty}$.
From a graph-based point of view $k(x,y)$ can be interpreted as the one step transition probability of a random walk on $\mathcal{M}$ from x to y. In this perspective the value $k_t(x,y)$ obtained from $\mathbf{K}^t$ represents the t steps transition probability of a random walk on $\mathcal{M}$ from x to y. Looking at t as a scale parameter (usually the time), it is now possible to define a

scale space of diffusion kernels $\{k_t(x,y)\}_{t\in T}$ for a given set of scale parameters T. In the discrete case the diffusion operator becomes a $N \times N$ matrix $\mathbf{K}$, where N is the number of vertices of the mesh that represent $\mathcal{M}$.

3.2 Heat kernel

There are several diffusion process that could be analyzed using the theory exposed in the previous section. We focus on the *heat diffusion process*, that is one of the most used in literature. The heat diffusion process on a surface $\mathcal{M}$ is governed by the *heat equation*, a differential equation of the following form:

$$-\mathbf{\Delta}_{\mathcal{M}} u(x,t) = \frac{\partial u(x,t)}{\partial t}, \tag{3.2}$$

where $\mathbf{\Delta}_{\mathcal{M}}$ is the Laplace-Beltrami operator on $\mathcal{M}$. Usually a function u_0 is given as the initial heat distribution on the surface, or in other words $u_0(x) = u(x,0)$, $\forall x \in \mathcal{M}$ at time $t = 0$. The solution of the differential equation 3.2 for the initial condition $u(x,0) = u_0$ is a function $u : \mathcal{M} \times \mathbb{R} \to \mathbb{R}$ that solve 3.2 and s.t. $u(x,0) = u_0$.

The solutions of this differential equation are widely studied and can be written in a form that is coherent to the previous Chapter. Indeed it is possible to define a diffusion kernel, the *heat kernel* $h(x,y)$, and a diffusion operator, the *heat operator* $\mathbf{H}$ that play exactly the same roles as k and $\mathbf{K}$ in the general version. As done previously for $\mathbf{K}$ also for $\mathbf{H}$ we can consider the scale space of heat operator $\mathbf{H}_t$ for a given real parameter $t \geq 0$.

$\mathbf{H}_t$ can be viewed as an operator that, fixing an initial heat distribution u_0 produces as output the heat distribution after a time t.

Furthermore it is possible to see that the heat operator $\mathbf{H}$ is strongly related with the LBO by the following equivalence:

$$\mathbf{H}_t = e^{-t\mathbf{\Delta}_{\mathcal{M}}}. \tag{3.3}$$

Thanks to this equivalence we have that $\mathbf{H}_t$ and $\mathbf{\Delta}_{\mathcal{M}}$ have exactly the same eigenvectors as shortly proven. Lets ϕ be an eigenvector of $\mathbf{\Delta}_{\mathcal{M}}$ such that $\mathbf{\Delta}_{\mathcal{M}}\phi = \lambda\phi$ for a given eigenvalue λ than:

$$\begin{aligned}\mathbf{H}_t\phi = e^{-t\mathbf{\Delta}_M}\phi = \sum_{j=0}^{\inf}\frac{(-1)^j(t)^j}{j!}\mathbf{\Delta}_M^j\phi = \sum_{j=0}^{\inf}\frac{(-1)^j(t)^j}{j!}\lambda^j\phi \\ = \sum_{j=0}^{\inf}\frac{(-1)^j(\lambda t)^j}{j!}\phi = e^{-\lambda t}\phi,\end{aligned} \tag{3.4}$$

From the last equivalence we obtain that ϕ is an eigenvector of $\mathbf{H}_t$ and its corresponding eigenvalue is $e^{-\lambda t}$.

We denote with $\mathbf{H}_t(u_0)$ the heat distribution at time t for a given initial distribution u_0. Thanks to the spectral theorem, in the spirit of equation 3.1, we can write the heat kernel for a time t using the Fourier basis:

$$h_t(x, y) = \sum_{l=0}^{\infty} e^{-\lambda_l t}\phi_l(x)\phi_l(y). \tag{3.5}$$

3.3 Wave kernel

A second remarkable diffusion process is the *wave diffusion process*. The wave diffusion is based on a different physical model that represents the motion of a quantum particle on the manifold, whose behaviour is governed by the *Schrödinger equation*:

$$-i\mathbf{\Delta}_{\mathcal{M}}u(x, t) = \frac{\partial u(x, t)}{\partial t} \tag{3.6}$$

where $u(x, t)$ is the complex wave function capturing the particle behaviour.

Despite an apparent similarity to the heat equation (3.2), because of the complex exponent, the behaviour of the solution of the Schrödinger equation is oscillatory. Assuming that a quantum particle oscillates with unknown position on $\mathcal{M}$.

In [5] it is shown how to approximate an initial energy E for the particle. From this approximation is also possible to obtain an energy probability distribution f_E^2 with expectation value E. This energy probability distribution is defined in [5] as a log-normally distribution dependent on the eigenvalues of the LBO eigendecomposition of the surface. Thanks to this definition the energy is explicitly related to frequencies expressed as eigenvalues $\{\lambda_l\}_i$. Assuming that the particle oscillates at frequency λ drawn from a probability distribution $f_E^2(\lambda)$, the solution of (3.6) can be expressed in the Fourier domain as

$$u(x,t) = \sum_{l=0}^{\infty} e^{i\lambda_l t} f_E(\lambda_l)\phi_l(x). \tag{3.7}$$

It is also appreciable to highlight the link between the the wave kernel and the probability of finding the particle at point x that can be computed as:

$$p(x) = \lim_{T\to\infty} \int_0^T |u(x,t)|^2 dt = \sum_{l=0}^{\infty} f_E^2(\lambda_l)\phi_l^2(x), \tag{3.8}$$

and depends on the initial energy distribution f_E.

3.4 spectral diffusion descriptors

3.4.1 Heat Kernel Signature (HKS)

From the heat diffusion on shape a widely used point descriptor is obtained: the *Heat Kernel Signature* (HKS).

The component at time t of the *Heat Kernel Signature* (HKS) [140] [50] for a point $x \in \mathcal{M}$ is defined as:

$$h_t(x,x) = \sum_{l=0}^{k} e^{-\lambda_l t}\phi_l^2(x) \tag{3.9}$$

where λ_l, ϕ_l are eigenvalues and eigenfunctions given by the LBO eigendecomposition and k is the number of selected eigenfunctions. In the continuous case k could be equal to ∞. If $k = \infty$ the heat kernel is exactly computed. In the discrete case $k \leq N$ (N is the number of vertices of $\mathcal{M}$), usually $k = 100$ or 200. In this case the heat kernel is approximated due to the truncation in the set of eigenvalues and eigenvectors.

In [140] it is shown how the HKS ensures the so called *informative theorem* which states that if $\mathcal{M}$ and $\mathcal{N}$ are two compact manifold and the eigenvalues of the respective Laplace-Beltrami operators are not repeated, then the heat HKS is preserved for every isometry T between the two manifold, i.e. $h_t^{\mathcal{M}}(x, x) = h_t^{\mathcal{N}}(T(x), T(x))$, $\forall x \in \mathcal{M}$.

As we already highlighted in Chapter 2, some shapes could have some eigenvalues that are very close each others by leading to a switch in the order. In our practical experience the HKS descriptors are quite robust with respect to this non optimal situation.

The HKS is also known as the *autodiffusivity function*. The physical interpretation of autodiffusivity is the amount of heat remaining at point x after time t. Geometrically, autodiffusivity is related to the Gaussian curvature $K(x)$ by virtue of the Taylor expansion $h_t(x, x) = \frac{1}{4\pi t} + \frac{G(x)}{12\pi} + \mathcal{O}(t)$ where $G(x)$ denotes the Gaussian curvature at point x. Sun et al. [140] defined the *heat kernel signature* (HKS) of dimension Q at point x by sampling the autodiffusivity function at some fixed times $t_1, \ldots, t_Q$,

$$\mathbf{f}(x) = (h_{t_1}(x, x), \ldots, h_{t_Q}(x, x))^\top. \tag{3.10}$$

3.4.2 Wave Kernel Signature (WKS)

While resembling the HKS in its construction and computation, WKS is based on log-normal transfer functions that act as band-pass filters and thus exhibits better spatial localization. In the same way, for a given approximation of the energy expected value E and a energy probability distribution f_E^2 , we can define the component for the energy E of the *Wave*

Kernel Signature (WKS) [5], as

$$w(x,E) = \sum_{l=0}^{k} f_E(\lambda_l)^2 \phi_l(x)^2. \tag{3.11}$$

The WKS can be seen as the approximation of the probability of finding the particle with energy probability distribution f_E^2 at point x and $k \leq N$ is the number of selected eigenfunctions. As we already said Aubry et al. [5] considered a log-normal frequency distribution defined as

$$f_E(\lambda_l) = \exp\left(\frac{\log E - \log \lambda_l}{2\sigma^2}\right), \tag{3.12}$$

with mean frequency E and standard deviation σ. They defined the Q-dimensional *wave kernel signature* (WKS) as the collection

$$\mathbf{f}(x) = (w(x,E_1),\dots,w(x,E_Q))^\top, \tag{3.13}$$

where $E_1,\dots,E_Q$ are Q logarithmically-sampled energies. In [5] all the fixed parameters are explained in details. The WKS is also *intrinsic* and *informative,* i.e. once again for every isometry T between $\mathscr{M}$ and $\mathscr{N}$, we have that $w_{\mathscr{M}}(E,x) = w_{\mathscr{N}}(E,T(x))$, $\forall x \in \mathscr{M}$ and for every $E \in \mathbb{R}$.

3.4.3 Signal Processing and Heat/Wave Kernel Signature

Several popular spectral shape descriptor take a generic form of the diagonal of a parametric kernel diagonalized by the LBO eigenbasis. Notable examples include the *heat kernel signature (HKS)* [50, 140] and the *wave kernel signature (WKS)* [5] presented just before. More specifically, such methods construct at each point $x \in \mathscr{M}$ a Q-dimensional descriptor $\mathbf{f}$ defined as:

$$\mathbf{f}(x) = \sum_{i=0}^{\infty} \boldsymbol{\tau}(\lambda_i)\phi_i^2(x) \tag{3.14}$$

expressed by a bank of transfer functions $\boldsymbol{\tau}(\lambda) = (\tau_1(\lambda), \dots, \tau_Q(\lambda))^\top$. This formula can be seen as a filtering operation on some signal represented as a linear combination of the square of the eigenfunctions of the LBO.

Such descriptors have several appealing properties making their use popular in numerous applications. First, they are intrinsic and hence invariant to isometric deformations of the manifold by construction. Second, they are computable at every point of the manifold. Third, (3.14) can be efficiently computed using the first few eigenvectors and eigenvalues of the Laplace-Beltrami operator.

HKS uses low-pass transfer functions $\tau_t(\lambda) = e^{-t\lambda}$ for various values of the parameter $t \in \{t_1, \dots, t_Q\}$, giving rise to the *autodiffusivity function* $h_t(x,x)$, whose physical interpretation is the amount of heat remaining at point x after time t. A notable drawback of HKS is poor spatial localization, which is a consequence of the uncertainty principle: good localization in the spectral domain (large value of t) results in a bad localization in the spatial domain.

WKS uses band-bass transfer functions $\tau_E(\lambda) = \exp\left(\frac{\log E - \log\lambda}{2\sigma^2}\right)$ for various values of the parameter $E \in \{E_1, \dots, E_Q\}$. The physical interpretation of WKS is the probability to find a quantum particle at point x, given that it has an initial log-normal energy distribution with mean value E and variance σ. Typically, WKS exhibits oscillatory behaviour and has a better localization compared to HKS.

Optimal spectral descriptors (OSD)

Looking at HKS and WKS as descriptors obtained by a filtering operation gives us the possibility to introduce a learned descriptor that is based on the same type of construction, namely the *Optimal Shape Descriptor (OSD)*. Litman and Bronstein [90] used parametric transfer functions expressed as

$$\tau_q(\lambda) = \sum_{m=1}^{M} a_{qm}\beta_m(\lambda) \tag{3.15}$$

in some fixed (e.g. B-spline) basis $\beta_1(\lambda),\ldots,\beta_M(\lambda)$, where a_{qm} ($q = 1,\ldots,Q$, $m = 1,\ldots,M$) are the parametrization coefficients. Plugging (3.15) into (3.14) one can express the qth component of the spectral descriptor as

$$f_q(x) = \sum_{k\geq 1} \tau_q(\lambda_k)\phi_k^2(x) = \sum_{m=1}^{M} a_{qm} \underbrace{\sum_{k\geq 1} \beta_m(\lambda_k)\phi_k^2(x)}_{g_m(x)}, \tag{3.16}$$

where $\mathbf{g}(x) = (g_1(x),\ldots,g_M(x))^\top$ is a vector-valued function referred to as *geometry vector*, dependent only on the intrinsic geometry of the shape. Thus, (3.14) is parametrized by the $Q \times M$ matrix $\mathbf{A} = (a_{lm})$ and can be written in matrix form as $\mathbf{f}(x) = \mathbf{Ag}(x)$. The main idea of [90] is to *learn* the optimal parameters $\mathbf{A}$ by minimizing a task-specific loss. Given a training set consisting of a pair of geometry vectors $\mathbf{g}, \mathbf{g}^+$ representing knowingly similar points (*positives*), and $\mathbf{g}, \mathbf{g}^-$ representing knowingly dissimilar points (*negatives*), one tries to find $\mathbf{A}$ such that $\|\mathbf{f}-\mathbf{f}^+\| = \|\mathbf{A}(\mathbf{g}-\mathbf{g}^+)\|$ is as small as possible and $\|\mathbf{f}-\mathbf{f}^-\| = \|\mathbf{A}(\mathbf{g}-\mathbf{g}^-)\|$ is as large as possible. The authors show that the problem boils down to a simple Mahalanobis-type metric learning.

The geometry vectors are introduced here, because they will be used in some of the next Chapters.

4 Advances in Functional Maps

In this Chapter we introduce a new set of data and a new application related to a specific framework for geometry processing: Functional Maps. Functional Maps is a recently proposed framework for inference and manipulation of maps between shapes. Our contribution in this Chapter is twofold. First, we propose a new method to provide a set of indicator functions that is used effectively to improve the original functional maps estimation. Second, we introduce our functional kernel for brain classification as a new application of functional maps. We start the Chapter with an overview of the theory on which the functional maps is based. As we show the mathematical construction, that makes the functional maps representation efficient for maps between shapes, is strongly related to the Fourier transform. For this reason, for its great potential in applications, and for the wide use of this framework in our work we dedicate the whole Chapter 4 to functional maps.

4.1 Original framework

Considering a pair of Riemannian surfaces $\mathscr{M}$ and $\mathscr{N}$, to these surfaces we can associate a functional space. As example we can consider the space of real value continuous functions defined on the surface $\mathscr{M}$: $\mathscr{F}(\mathscr{M},\mathbb{R}) = \{f : \mathscr{M} \longrightarrow \mathbb{R} \text{ s.t. } f \text{ is continuous }\}$.

The main idea behind functional maps is to move the analysis of the correspondence between two shapes from the set of points on the surfaces to their functional spaces. In the geometry processing context functional maps was firstly introduced by Ovsjanikov et al. in [115], for more details we refer the reader to this paper.

Let's start with a correspondence Π between our pair of surfaces $\Pi : \mathscr{N} \longrightarrow \mathscr{M}$, expressed as a point-to-point map. At this level we can consider Π as a bijective map for simplicity and $\mathscr{M}$ and $\mathscr{N}$ could be continuous or discrete surfaces without any variations in the definitions. Now if we consider $\mathscr{F}(\mathscr{M},\mathbb{R})$ and $\mathscr{F}(\mathscr{N},\mathbb{R})$ the functional spaces defined over $\mathscr{M}$ and $\mathscr{N}$ respectively, then Π induces a map T between $\mathscr{F}(\mathscr{M},\mathbb{R})$ and $\mathscr{F}(\mathscr{N},\mathbb{R})$ in the following way:

$$\begin{gathered} T : \mathscr{F}(\mathscr{M},\mathbb{R}) \longrightarrow \mathscr{F}(\mathscr{N},\mathbb{R}) \\ f \longmapsto g = f \circ \Pi \end{gathered}$$

From the definition of g it is clear that $g \in \mathscr{F}(\mathscr{N},\mathbb{R})$, indeed for every point $y \in \mathscr{N}$ we have that $g(y) = f(\Pi(y)) \in \mathbb{R}$, because $\Pi(y) \in \mathscr{M}$ and f is well defined for every point on $\mathscr{M}$. We assert here that knowledge of T is equivalent to knowledge of Π, and that T is a linear map between functional spaces while Π is a more sophisticated map between surfaces. The linearity of the map T is with respect to the three operations that make the space $\mathscr{F}(\mathscr{M},\mathbb{R})$ an algebra, that is $\forall\, f, f_1, f_2 \in \mathscr{F}(\mathscr{M},\mathbb{R})$ and $\forall\, a \in \mathbb{R}$:

scalar product $T(af) = aT(f)$;
sum of functions $T(f_1 + f_2) = T(f)$;
product of functions $T(f_1 \odot f_2) = T(f_1) \odot T(f_2)$;

where $\odot$ is the point-wise product. For more details and the proofs of these facts we refer the reader to [115], [111].

Now suppose that the functional spaces are equipped with basis, $\phi_1^{\mathscr{M}}, \ldots, \phi_i^{\mathscr{M}}, \ldots$ and $\phi_1^{\mathscr{N}}, \ldots, \phi_j^{\mathscr{N}}, \ldots$ respectively for $\mathscr{M}$ and $\mathscr{N}$, such that we can write every function in $\mathscr{F}(\mathscr{M},\mathbb{R})$ as a linear combination of the basis $f =$

$\sum_{i=1} a_l \phi_i^{\mathscr{M}}$, $\forall f \in \mathscr{F}(\mathscr{M},\mathbb{R})$, with coefficients $a_i \in \mathbb{R}$, similarly for $\mathscr{F}(\mathscr{N},\mathbb{R})$. With these basis and thanks to the linearity of the map T we can write:

$$T(f) = T(\sum_{i=1} a_i \phi_i^{\mathscr{M}}) = \sum_{i=1} T(a_i \phi_i^{\mathscr{M}}) = \sum_{i=1} a_i T(\phi_i^{\mathscr{M}}). \tag{4.1}$$

Here, thanks to the map T, we have that the functions $T(\phi_i^{\mathscr{M}}) \in \mathscr{F}(\mathscr{N},\mathbb{R})$ and so we can write them as:

$$T(\phi_i^{\mathscr{M}}) = \sum_{j=1} c_{ij} \phi_j^{\mathscr{N}} \ \forall i \in \mathbb{N}. \tag{4.2}$$

Putting together 4.1 and 4.2 we obtain:

$$T(f) = \sum_{i=1} a_i T(\phi_i^{\mathscr{M}}) = \sum_{i=1} a_i \sum_{j=1} c_{ij} \phi_j^{\mathscr{N}} = \sum_{j=1} \sum_{i=1} a_i c_{ij} \phi_j^{\mathscr{N}}. \tag{4.3}$$

From the last equation we can compute the coefficients b_j of $T(f)$ in the basis $\phi_1^{\mathscr{N}}, \ldots, \phi_j^{\mathscr{N}}, \ldots$ as $b_j = \sum_{i=1} a_i c_{ij}$, and so $T(f) = \sum_j b_j \phi_j^{\mathscr{N}}$. Note that each c_{ij} does not depend from a_i and b_j, $\forall i, j \in \mathbb{N}$ as shown in equation 4.2 where all the c_{ij} are defined, so they are independent from the coefficients a_i and b_J and conversely they only depend from the selection of the basis functions $\{\phi_i^{\mathscr{M}}\}$ and $\{\phi_j^{\mathscr{N}}\}$.
Using these basis, if we represent the function $f \in \mathscr{F}(\mathscr{M},\mathbb{R})$ as its coefficients $\mathbf{a} = (a_1, \ldots, a_i, \ldots)$ and similarly $T(f) \in \mathscr{F}(\mathscr{N},\mathbb{R})$ as $\mathbf{b} = (b_1, \ldots, b_j, \ldots)$, starting from the previously definition of $b_j = \sum_{i=1} a_i c_{ij}$, we can fully represent the map T as a matrix $\mathbf{C} = (c_{ij})$ (possibly infinite dimensional) such that:

$$\mathbf{b} = \mathbf{C}\mathbf{a}, \tag{4.4}$$

Regarding the selection of the basis, a particular case is observed when the orthonormal basis, composed by the indicator functions for each vertex is chosen. Indeed with this choice in discrete setting the matrix $\mathbf{C}$ coincides with the permutation matrix associated with the correspondence map Π. Furthermore no dimensionality reduction can be gained with this choice.

As stressed in the previous paragraph the matrix representation **C** of the functional map depends only on the choice of the basis for the two functional spaces. It is important to highlight the connection between the functional maps framework and signal processing. As previously introduced in Chapter 2, from Fourier transform we receive a canonical basis for the square-integrable functions on a surface. In Chapter 2 we also introduced the Laplace Beltrami operator (LBO) defined on surface, and we observed the connection between its eigenfunctions and the standard Fourier basis in the Euclidean case. If we limit the functional spaces to the square-integrable real value functions

$$\mathscr{F}(\mathscr{M},\mathbb{R}) = L^2(\mathscr{M}) = \{f : \mathscr{M} \longrightarrow \mathbb{R} \; s.t \int_{\mathscr{M}} f^2(x)dx < \infty\},$$

$$\text{and similarly } \mathscr{F}(\mathscr{N},\mathbb{R}) = L^2(\mathscr{N}) = \{f : \mathscr{N} \longrightarrow \mathbb{R} \; s.t \int_{\mathscr{N}} f^2(y)dy < \infty\},$$

then we can use the LBO eigenfunctions as basis for these two spaces. As done in the Chapter 2 we will refer to the LBO eigenfunctions as Manifold Harmonics (MH). Here we resume once again the properties of the MH, which have already been shown in Chapter 2. First, as we saw in Chapter 2, the MH are a set of orthonormal functions, that is:

$$\langle \phi_i^{\mathscr{M}}, \phi_l^{\mathscr{M}} \rangle_{\mathscr{M}} = \begin{cases} 1 & \text{if } i = l \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

where $\langle \cdot, \cdot \rangle_{\mathscr{M}}$ is the standard inner product on the Riemannian manifold $\mathscr{M}$. The orthonormality allows us to compute in an easy way both for the coefficients of functions and for the elements of the matrix **C**. For the coefficients of a function $f \in L^2(\mathscr{M})$ we have:

$$a_i = \langle \phi_i^{\mathscr{M}}, f \rangle_{\mathscr{M}}, \; \forall i = 1, \dots , \tag{4.6}$$

where the orthonormality guarantees the uniqueness of the coefficients $\{a_i\}$. These coefficients $\{a_i\}$ are exactly the Fourier coefficients of the Fourier transform of f. Regarding the elements $c_i j$ of the matrix $\mathbf{C}$ starting from equation 4.2 we can obtain the following equation:

$$c_i j = \langle \phi_j^{\mathcal{N}}, T(\phi_i^{\mathcal{M}}) \rangle_{\mathcal{N}}, \; \forall i, j = 1, \dots , \tag{4.7}$$

since the function $T(\phi_i^{\mathcal{M}}) \in L^2(\mathcal{N})$ can be written as linear combination of the basis $\{\phi_j^{\mathcal{N}}\}$ with coefficients defined as in Equation 4.6.
A second important property of the MH is that we can consider only a subset of the basis $\{\phi_i\}_{i=0}^{k_{\mathcal{M}}}$ for some $k_{\mathcal{M}} \in \mathbb{N}$ obtaining a low-frequencies approximation of the functional space $L^2(\mathcal{M})$. Indeed if the MH are ordered with increasing eigenvalues (or frequencies), representing f as the truncated linear combination $f = \sum_{i=0}^{k_{\mathcal{M}}} a_i \phi_i^{\mathcal{M}}$ corresponds to a low-pass filtering of the function f. In the same way for $L^2(\mathcal{N})$ we can select a subset $\{\phi_j\}_{j=0}^{k_{\mathcal{N}}}$ for some $k_{\mathcal{N}} \in \mathbb{N}$ which can be different from $k_{\mathcal{M}}$. Using only a finite (and preferably small) number of basis functions we can obtain a compact representation of the functional space. This is true also for the functional map, with a finite matrix $\bar{\mathbf{C}}$ which can be seen as the top-left submatrix of the original matrix $\mathbf{C}$ with dimension $k_{\mathcal{N}} + 1 \times k_{\mathcal{M}} + 1$. The order of the MH makes it possible to obtain different levels of representation of the functional spaces in a trade-off between compactness (obtained with small values of $k_{\mathcal{M}}$) and accuracy of the approximation (given instead by large values of $k_{\mathcal{M}}$). The matrix $\mathbf{C}$ is strongly dependent from this choice inheriting compactness and accuracy directly from the level of the approximation used for the functional spaces. A third notable property of the MH is that ordered collections of MH spanned subspace of $L^2(\mathcal{M})$ which is stable with respect to the isometric deformations. Although as discussed in Chapters 2 and 3 the MH of two different (also isometric) shapes may differ for the sign and for some switches in the order, the subspace spanned by the fist $k_{\mathcal{M}} + 1$ MH is known to be stable to near-isometric deformations unless the $k_{\mathcal{M}} + 1$-th and the $k_{\mathcal{M}} + 2$-

th eigenfunctions belong to the same eigenspaces. The stability to near-isometric deformations ensures that the functions generated by the MH are also invariant to isometric deformations. If the correspondence Π is also an isometry than the matric $\mathbf{C}$ assumes a very particular structure. In fact if Π is an isometry it preserves the area elements and if we suppose that the MH on $\mathscr{M}$ and on $\mathscr{N}$ share the order, and that there are not eigenvalues with multiplicity greater than one, then the matrix $\mathbf{C}$ is a diagonal matrix. The diagonal structure of $\mathbf{C}$ means that the functional map T maps every eigenfunction $\phi_i^{\mathscr{M}}$ to $\phi_i^{\mathscr{N}}$, less than a scalar. Although this diagonal structure is guaranteed only by two strong assumptions, in general for an isometry Π is true that the $\mathbf{C}$ obtained from the MH is a sparse matrix close to diagonal with some non-diagonal elements only in correspondence with the repeated eigenvalues or switch in the order of eigenfunctions on the two shapes.

4.1.2 Continuity

The functional representation of a point-to-point map makes easier to analyze the continuous structure of the surfaces. The point-to-point maps are by themselves discrete in the sense that they consider as unit of information points that are sampled on the surfaces. Functions that can be represented by MH are continuous functions defined on the continuous domain of the surface. So with functional representation of correspondences and using the MH as basis we obtain:

- a continuous map $\mathbf{C}$ with respect to continuous variations of the input function f (its coefficients $\mathbf{a}$);
- a continuous representation of the functional spaces with $\mathbf{C}$ that maps continuous functions on $\mathscr{M}$ to continuous functions on $\mathscr{N}$ due to the fact that the MH can only represent continuous functions on the surfaces.
- a continuous representation also of the space of maps between shapes, in the sense that continuous variations in the elements of the matrix $\mathbf{C}$

gives continuous variations of the resulting meaningful maps between shapes.

The continuous representation of the spaces of maps makes possible all the operations as sum difference composition among set of maps. As we already highlighted, the choice of the MH as basis allows us to ensure the continuity of the functions considered on the two shapes and in the same time the continuity of the map. This property is definitely desirable if we are trying to represent a correspondence between pairs of isometric shapes. The correspondence has to be continuous in the domain and in the image of the map in the sense that near points from one shape has to be mapped in near points on the second shape in a continuous way.

4.1.3 Linearity of the constraints

Generally for a given pair of shapes $\mathcal{M}$ and $\mathcal{N}$ the point-to-point correspondence Π is unknown, and it is therefore impossible to obtain the map T and its matrix representation $\mathbf{C}$. The matrix $\mathbf{C}$ can be estimated solving a constrained optimization problem, that will be shortly defined. One of the main contributions of the functional representation is that it allows operations and to impose a set of constraints linearly, simplifying their application naturally and improving the final results of this optimization problem.

Functional constraints

Let $f \in L^2(\mathcal{M})$ and $g \in L^2(\mathcal{N})$ be a pair of corresponding functions respectively defined on $\mathcal{M}$ and $\mathcal{N}$. Here for corresponding functions we mean that f and g are the same function but defined on the two different surfaces. Let $\mathbf{a}$ and $\mathbf{b}$ be the representations respectively of f and g in the basis functions $\{\phi_i^{\mathcal{M}}\}$ and $\{\phi_j^{\mathcal{N}}\}$. Since these are corresponding functions we can impose on $\mathbf{C}$ a function preservation constraint defined as:

$$\mathbf{Ca} - \mathbf{b} = 0, \tag{4.8}$$

here we stressed again that the linearity of this constraint is allowed by the functional representation. This constraint only depend on the function f and g and is independent from **C** and from the basis selected for the functional spaces.
In order to select functions for this constraint we have different possible choices. We can use as function a set of point descriptors, and looking for a **C** that approximately maps a descriptor computed on $\mathscr{M}$ in the same descriptor computed on $\mathscr{N}$. As example in [115], the already presented HKS and WKS descriptors are used.
A different function that could be used are landmark point correspondences. If we know that $y \in \mathscr{N}$ is the corresponding point of a point $x \in \mathscr{M}$ than we can use as function f and g the indicator function of these points. For indicator function of a point x we mean a function which is zero everywhere but equal to 1 in x. Any other kind of function derived from the knowledge of the correspondence between these two points can be used.
In the same way if instead of correspondence between two points we know the correspondence between two regions or segments on the two shapes we can use the indicator of these regions as function for the functional constraint. For an example of how this kind of functions can be obtained for a pair of shapes we refer to Section 4.2. Usually we refer to the ordered set of functions selected for this type of constraint as *probe functions* denoted as **F** and **G** respectively for the functions on $\mathscr{M}$ and on $\mathscr{N}$.

Operator commutativity constraints

A different linear constraint that could be considered is the commutativity with linear operators defined on $\mathscr{M}$ and $\mathscr{N}$. These operator could be quite general and in [115] some examples are given. In particular if we are looking for a map **C** induced by an isometry Π then as shown in [115], the functional maps has to commute with the LBO of the two surfaces.

In particular if we are using the MH as basis of the functional space this constraint can be written as:

$$\mathbf{\Lambda}_{\mathcal{N}}\mathbf{C} - \mathbf{C}\mathbf{\Lambda}_{\mathcal{M}} = 0. \tag{4.9}$$

where $\mathbf{\Lambda}_{\mathcal{M}}$ and $\mathbf{\Lambda}_{\mathcal{N}}$ are diagonal matrices of eigenvalues of the LBO respectively of $\mathcal{M}$ and $\mathcal{N}$.

Optimization problem

Finally putting together the constraints defined just above we can give a complete formulation of the optimization problem that we can solve in order to estimate **C**. Given the sets of probe functions **F** and **G**, the estimation of **C** can be obtained as the solution of the following minimization problem:

$$C = \underset{\mathbf{Q}}{\operatorname{argmin}} ||\mathbf{QF} - \mathbf{G}||_F^2 + \alpha ||\mathbf{\Lambda}_{\mathcal{N}}\mathbf{Q} - \mathbf{Q}\mathbf{\Lambda}_{\mathcal{M}}||_F^2, \tag{4.10}$$

where $\alpha \in [0,1]$ is a real parameter that weighs the contribution of the operator commutativity constraint.

4.2 Region indicator functions in functional maps

In this Section we introduce a new method for finding indicator functions of corresponding regions between two shapes. The main application of the proposed method is the improvement obtained in functional maps when we use these functions as input in the optimization process.

As we saw in the previous Section in the functional maps optimization is strongly recommended to use some indicator as probe functions in the functional constraints. As indicator functions one can use some point landmarks, such as some salient points or some corresponding points that could be selected using some preliminary point-to-point correspondence pipeline. Often it is not easy to find this kind of data. Alternatively, some indicators of corresponding regions may be used. The problem of finding corresponding regions is widely addressed in literature. A solution for finding corresponding regions is the region-based correspondence (RBC). Region-based correspondence (RBC) is a highly relevant and non-trivial computer vision problem. Given two 3D shapes, RBC seeks segments/regions on these shapes that can be reliably put in correspondence. The problem thus consists both in finding the regions and determining the correspondences between them. In other words the problem can be reformulated as that of finding regions on the shapes that behave similarly and can thus be easily put in correspondence. This problem is different from *shape co-segmentation*, a different possible solution to this problem, since in RBC the goal is not to find meaningful semantic segments in various shapes (*e.g.*, limbs in animal shapes), which is the goal of shape co-segmentation, but rather to determine regions in the two shapes that are in correspondence, as doing for example in [49]. As recently mentioned in [49] this problem statement is closely related to that of "biclustering", implying that RBC can be cast as a *biclustering problem* [95], in particular if the points on the shapes can be endowed with a similarity measure (*e.g.*, based on some descriptor). Given such a measure, a similarity matrix may be built and the goal of biclustering is to simultaneously cluster both the rows and columns of this matrix [95]. Given a data matrix, biclustering aims at retrieving sub-matrices (*i.e.*, *biclusters*), in each of which a certain subset of rows exhibits a "coherent behaviour" (in some sense) in a certain subset of columns. Biclustering differs from standard clustering, which treats whole rows/columns, in that the former is able to focus on local information in portions of rows and portions of columns. Many biclustering methods have been proposed, differing in the type of biclusters that can be retrieved, as well as in the adopted criteria and algorithms, as comprehensively reviewed in [95, 112]. Biclustering is a well-studied problem, with applications to gene expression data, recommender systems, market segmentation, and other areas [39, 42, 67, 75, 107, 136]. However, maybe surprisingly, biclustering has not been used for shape correspondence, with the notable exception of [49]. This is arguably due to the two following challenges: Typical 3D shapes in computer vision and graphics contain considerable *geometric* information, which is typically not taken into account in biclustering methods. Perhaps more fundamentally, in the context of RBC, an important property is required of the retrieved bi-clusters: spatial coherence, *i.e.*, nearby points should be grouped to nearby regions. This property is neither present, nor is easy to encourage, by using standard biclustering methods.

In this Section, we exploit this implication by tackling RBC via a novel biclustering approach, called S^4B (**s**patially **s**mooth **s**pike and **s**lab **b**iclustering), which: (i) casts the problem in a probabilistic low-rank matrix factorization perspective; (ii) uses a spike and slab prior to induce sparsity; (iii) is enriched with a spatial smoothness prior, based on geodesic distances, encouraging nearby vertices to belong to the same bicluster. This type of spatial prior cannot be used in classical biclustering techniques. We test the proposed approach on the FAUST dataset, introduced in Chapter 2, outperforming both state-of-the-art RBC techniques and classical biclustering methods. For more details about the method proposed in this Section, please refer to [40].

4.2.1 Overview of the Proposed Method

Here we introduce a novel algorithm, where the rationale is to decompose the data matrix into levels, each corresponding to a different bicluster (as in [24, 61]), thus allowing to obtain non-exhaustive and possibly overlapping biclusters. In this class of approaches, sparsity plays a crucial role. In fact, the data matrices to which biclustering is typically applied have large numbers of rows and columns (*e.g.*, thousands by hundreds, in gene expression data), but the biclusters often involve only small portions thereof. The proposed method results from combining a probabilistic low-rank matrix factorization criterion with a spike and slab prior

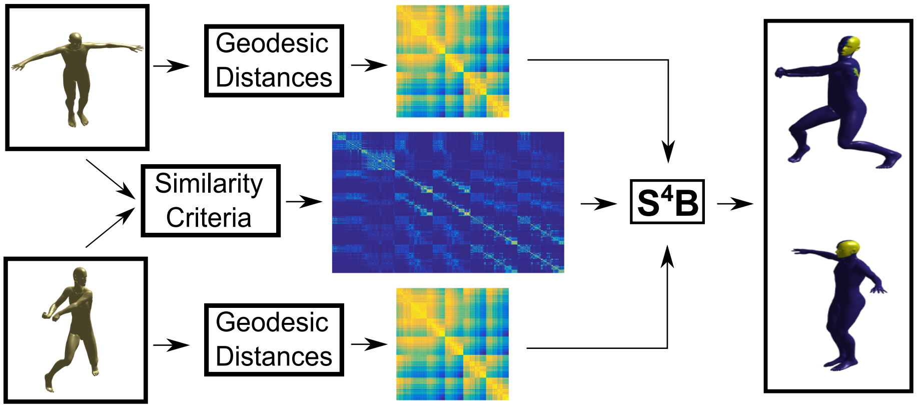


Fig. 4.1: The framework of the the **s**patially **s**mooth **s**pike and **s**lab **b**iclustering (S^4B) for the RBC problem

to encourage sparsity. Spike and slab priors were proposed in [106] for variable selection in linear regression, and later generalized and adopted by many authors as general-purpose sparsity-inducing priors [68]. The other main novel ingredient herein proposed is a spatial smoothness prior, to improve the bicluster coherence/quality. Similarly to what has been proposed for clustering [32], we exploit known pair-wise relations to encourage certain rows (and/or columns) to belong to the same bicluster. This is obtained by combining the spike and slab prior with two pair-wise priors, one for the rows and one for the columns, encouraging pairs of nearby rows/columns to be grouped together. We call our method **s**patially **s**mooth **s**pike and **s**lab **b**iclustering (S^4B).

The spatial priors are derived from two similarity matrices (one for the rows and one for the columns), based on the geodesic distances between the shapes' vertices. Those matrices control the strength with which each pair of vertices (on each shape) is encouraged to belong to the same bicluster. Consequently, the proposed S^4B method combining two types of information (as depicted in Fig. 4.1): (i) the affinity between pairs of vertices of different shapes; (ii) the neighbourliness between vertices on the same shape.

Complexity:

The leading term concerning space complexity is $\mathcal{O}(nk)$ (or $\mathcal{O}(km)$, which is the space needed to store the $A(z)$ (or $B(v)$) matrix. Thus, an adequate sparse representation can overcome this possible drawback. Regarding time complexity (for each iteration), the leading term is $\mathcal{O}(n^3k^3)$ (or $\mathcal{O}(m^3k^3)$) which is the worst case scenario for matrix multiplication/inversion of a $\mathcal{O}(nk)$ (or $\mathcal{O}(mk)$) matrix.

4.2.2 RBC experiments

This subsection describes the experiments carried out to compare S^{4}B with the relevant state-of-the-art, including specific RBC and general biclustering techniques. Regarding RBC-specific techniques, the most relevant work is the recent *stable region correspondences* (SRC) approach [49], which uses a power iteration scheme. We can directly compare the corresponding regions obtained by S^4B and SRC with respect to the ground-truth mapping between the two shapes. Other methods produce point-to-point correspondences based on geometric features of the shapes. We compare our corresponding regions to *blended intrinsic maps* (BIM) [76], a popular point-to-point correspondence method. To evaluate BIM in the context of corresponding regions, we follow [49] and use the point-to-point mapping to transport the segmentation computed on one shape to the other. We also provide a comparison between S^4B and its version without the spatial smoothness prior, to show that this prior is crucial to obtain high quality results.
All the experiments use FAUST dataset [13]. As we saw before this dataset presents both near-isometric (different poses of the same subject) and non-isometric deformations (due to the significant variability between different subjects). All of the shapes have the same number of vertices, and the ground-truth one-to-one correspondence (or map) between each pair of shapes is available. We measure the quality of the results as the

global labelling accuracy with respect to the ground-truth map. More precisely, since every method assigns a label to each vertex on the two shapes, we compute a score that sums the influence area of vertices in one shape that are given the same label as their mapping in the second shape. This provides the percentage of the shape's area that has a correct correspondence. Then, we compute the same score by inverting the role of first and second shape, and consider the mean of these two scores as the final score. Formally,

$$\text{score}(L_{\mathscr{M}}, L_{\mathscr{N}}) = \sum_{i=1}^{N} \Big(L_{\mathscr{M}}(i) = L_{\mathscr{N}}\big(f(i)\big)\Big) A_{\mathscr{M}}(f(i)) \tag{4.11}$$

$$\text{quality} = \frac{1}{2}\Big(\frac{\text{score}(L_{\mathscr{M}}, L_{\mathscr{N}})}{\sum(A_{\mathscr{M}})} + \frac{\text{score}(L_{\mathscr{N}}, L_{\mathscr{M}})}{\sum(A_{\mathscr{N}})}\Big), \tag{4.12}$$

where $\mathscr{M}$ and $\mathscr{N}$ are the shapes, $L_{\mathscr{M}}$ and $L_{\mathscr{N}}$ are the given labels, $A_{\mathscr{M}}$ and $A_{\mathscr{N}}$ indicate the influence area of each vertex, and f is the ground-truth point-to-point mapping.

To evaluate the SRC method, we followed the guidelines provided by its authors [49]. Concerning BIM, since its performance is highly influenced by the starting segmentation, we evaluate the point-to-point mapping using two possible segmentations: (i) based on geodesic Voronoi cells around a *farthest point sampling* [141], which provides segments of uniform size; (ii) based on the output labels of S^4B . This gives us a starting segmentation, that we transfer to the second shape using the correspondences provided by BIM.

Implementation details.

For S^4B , we fix 8 possible biclusters, and the affinity matrix is computed as described in [49], with the same descriptors HKS and WKS. For the geodesic similarities, we considered as "near" only distances below 5% of the maximum. Notice that if we turn off the spatial smoothness prior, we obtain a standard biclustering algorithm, hereafter referred to as *spike*

	Stable Region	**BIM Voronoi**	**BIM S^{4}B**	**SSBi**	**S^{4}B**
scenario1	95.37	95.87	**97.98**	29.91	97.36
scenario2	85.34	95.35	94.21	30.39	**95.73**
scenario3	85.39	92.51	92.5	32.32	**94.25**
global	86.58	93.26	93.36	31.8	**94.8**

	Stable Region	**BIM Voronoi**	**BIM S^{4}B**	**SSBi**	**S^{4}B**
scenario1	94.95	96.76	97.84	30.07	**97.98**
scenario2	87.42	96.17	93.82	30.95	**96.63**
scenario3	87.63	92.82	92.55	33.77	**94.96**
global	89.33	93.1	93.15	31.26	**95.52**

Table 4.1: Results on the FAUST dataset using *SRC, BIM, SSBi*, and the proposed S^4B . Scenario1: pairs of shapes of the same subject in different poses. Scenario2: pairs of different subjects in the same pose. Scenario3: pairs of different subjects in different poses. The left/right tables show mean/median scores for each scenario, and the global mean/median score.

and slab biclustering (SSBi). We used parameters that lead to similar numbers of segments for each of the methods tested, for fairness of comparison. Once the method converges, we obtain the probability that each vertex belongs to the retrieved biclusters. Finally, we assign each vertex to the bicluster (and hence the label) maximizing that probability (discarding labels with probability below 0.75).

Evaluation.

We randomly selected 50 pairs of shapes from the FAUST dataset and applied each of the previously mentioned methods. The results reported in Table 4.1 show that S^4B performs better than SRC, BIM, and SSBi. Particularly, S^4B outperforms both the SRC and BIM-Voronoi approaches. By comparing S^4B and SSBi, we can state that the spatial prior is crucial to obtain high quality results. All of these results have been statistically evaluated with a paired T-test with significance level equal to 5%.

Figure 4.2 shows some of the results obtained: the first row shows results by the SRC method [49]; the second row shows results of SSBi; the third row presents results of the new S^4B . Clearly, the S4B results present high coherence among the different pairs of shapes (although they have

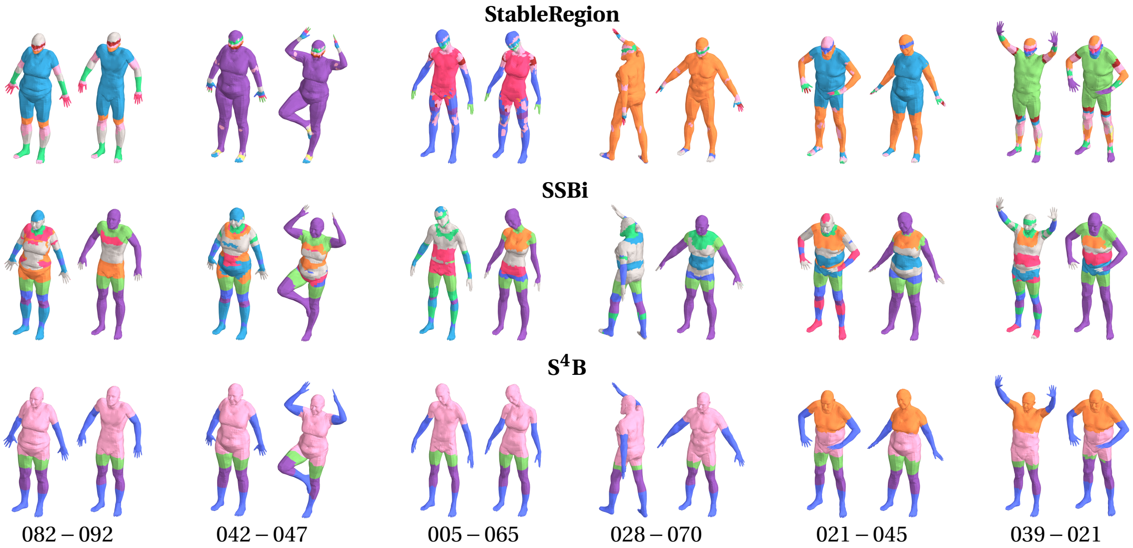


Fig. 4.2: Qualitative results of *SRC*, *SSBi* and *S^4B* on the FAUST dataset.

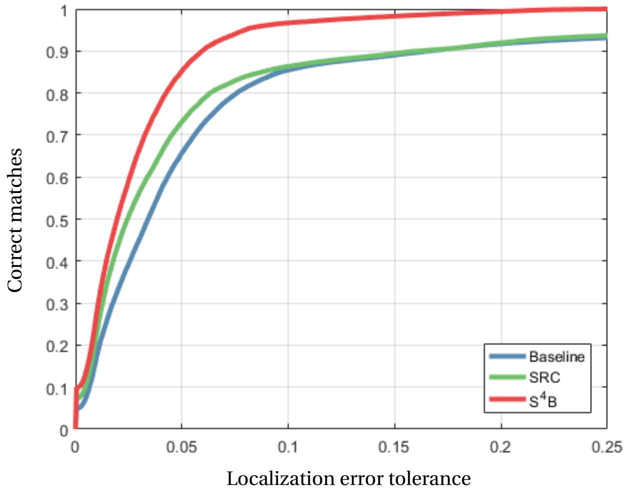


Fig. 4.3: Performances of S^{4}B and SRC results when adopted to create a point-wise estimation map.

obviously been analysed independently), moreover they produce more connected regions, unlike those obtained by SRC and SSBi.

S^4B in the functional maps

The final goal for which we present this method is to improve the functional maps framework using some stable region indicator functions in the functional constraints that define the optimization problem. S^4B gives us a possible set of indicator functions that could be used in the functional maps framework. We thus compare the standard baseline [115] (without region correspondences) with the regions provided by both S^4B and SRC. Figure 4.3 plots the performance of S^4B and SRC, using the standard *correspondence quality characteristic* [76], as a function of the radius r, with each point in the curves representing the matching percentage considering only points that are at distant less than r from their ground truth correspondence. This results show that the performances clearly increase by adding the region indicators obtained from S^4B . Note that adding region indicators from SRC improves the baseline results. However also the stability of our regions are clearly more informative since they produce a stronger improvement on the quality of the functional maps.

4.3 Functional maps for brain classification

In this Section we exploit our functional kernel, a new functional maps approach for brain classification. The functional representation of brain shapes, or their subparts, enables us to improve the detection of morphological abnormalities associated with the analyzed disease. The proposed method is based on the spectral shape paradigm that is largely used for generic geometric processing but still few exploited in the medical context. Moreover, we propose a new kernel, called the Functional maps kernel (FM-kernel) for the Support Vector Machine (SVM) classification that is specifically designed to work on the functional space. The obtained results for bipolar disorder detection on the putamen *regions are promising in comparison with other spectral-based approaches. More details about this method can be found in [102].*

With respect to the other parts of this thesis in this section we move from the general geometry processing task to a specific applications field, the medical imaging. Automatically detection of abnormal anatomical shapes derived from diseased subjects is a fundamental goal in medical imaging. This task is typically formulated as a two-class classification problem, assigning to each shape a healthy or diseased label [147] [148]. In particular, thanks to the increased amount of data available, the attention of researchers is often focused on advanced learning-by-example methods [54] [57] [6] [25] [145] [27]. These tools require good shape representation and measure that encodes the relationship between the shapes. The desired representation should be informative, concise and efficient in computational terms. In order to capture possible brain deformations due to the disease, it is convenient to exploit geometry and topology properties of the anatomical parts as shape representation [71] [53] [52].To this aim, new spectral shape descriptors and methods have been adopted in this area, aiming at investigating advanced shape analysis approaches for the characterization of brain structures. A first method based on spectral properties was proposed in [51], where spherical harmonic descriptors (SPHARM) are computed on brain surfaces after a shapes registration step. In [125] Reuter et al. introduced a spectral global descriptor, namely *Shape-DNA*. This signature is defined as the increasing ordered sequence of the first Laplace-Beltrami operator (LBO) eigenvalues. The Shape-DNA is invariant to the isometric deformations and by neglecting higher frequencies of the shape it is also robust to noise. This descriptor is proposed for two different versions: the external surfaces and the entire volume. The two surface-based and volume-based versions are also introduced by Castellani et al. in [26] where the HKS point signature, introduced in Chapter 3, has been extended to describe the entire shape by leading to the so called *Global Heat Kernel Signature*

(GHKS). Differently from *Shape-DNA* this approach is based on a point signature that encodes local information. Furthermore the GHKS allows a multi-scale analysis that enhances the discriminative properties of the signature. Note that both the approaches [125] and [26] do not require an explicit registration phase for shape comparison. In [96], a collection of three well known spectral descriptors, HKS, WKS and the *Scale Invariant Heat Kernel Signature* SI-HKS, [22] are computed at every vertex of the mesh and then used in a Bag of features framework for spectral shape analysis of brain structures in order to detect the Alzheimer's Disease. The multiscale analysis is instead the basic idea of [150]. This approach encodes the volumetric geometry information starting from the volumetric LBO and obtaining a multi-scale volumetric morphology signature which describes the transition probability by random walk between the point pairs and depends on heat transmission time. Starting again from the LBO eigendecomposition an interesting technique is recently presented by Rabiei et al. in [124]. In this work the *Graph Windowed Fourier* [137] is exploited to encode the geometric properties of the brain cortex. More specifically, a *Gyrification Index* is introduced to represents at every point how much the surface is folded. Shifting the focus on functional spaces can be effective and productive as for example in [92]. This work proposes a spectral framework namely *Brain Transfer* to transfer functions between different shapes, in order to explore the shape and functional variability of retinotopy. We propose a new method for shape classification based on the Functional maps framework [115] that was introduced in the previous Sections. The contribution of the proposed method is two-fold:

- Firstly we extend the use of Functional map to the medical domain, to improve the encoding of morphological relations between pairs of brain-shapes.
- Secondly we propose a new dissimilarity measure properly designed for the functional space. In particular, from this dissimilarity measure

we derived a well defined new kernel, namely the Functional maps kernel (FM-Kernel) that is effective and theoretically founded.

We evaluated our method for the characterization of brain abnormalities in the context of mental health research. In particular, we propose a brain classification study on a dataset of patients affected by bipolar disorder and healthy controls. We focused on the *putamen* region, which is a deep gray matter brain structure, part of the basal ganglia, a functional and anatomical heterogeneous region which is thought to be affected, particularly in shape, by bipolar disorder [66]. In order to check the actual effectiveness of the proposed method and the richness added by the Functional maps framework in this context, we compared our method with more classical shape analysis methods based on a spectral approach.

4.3.1 Proposed Method

In this subsection we show the main contributions of our paper that are: i) the design of a Functional maps framework on the spectral domain for brain comparison, and ii) the customized Functional maps kernel for brain classification.

Computing Functional Maps

As mentioned in Section 4.1, we can approximate the Functional map C taking into account two sets of linear constraints. In particular, such constraints are defined by pairs of corresponding functions and by operators that satisfy the commutativity property with respect to C. In this work we assume that in the absence of disease and disorders the brain surfaces are closer to isometric shapes with respect to the variations caused by the presence of disturbances. Therefore, for shapes belonging to the same class it is possible to find a map Π that can be approximated by an isometry. Thus, a good approximation of the Functional maps, in order to detect disorders, can be computed starting from isometry invariant descriptors and operators. This is the motivation that has driven

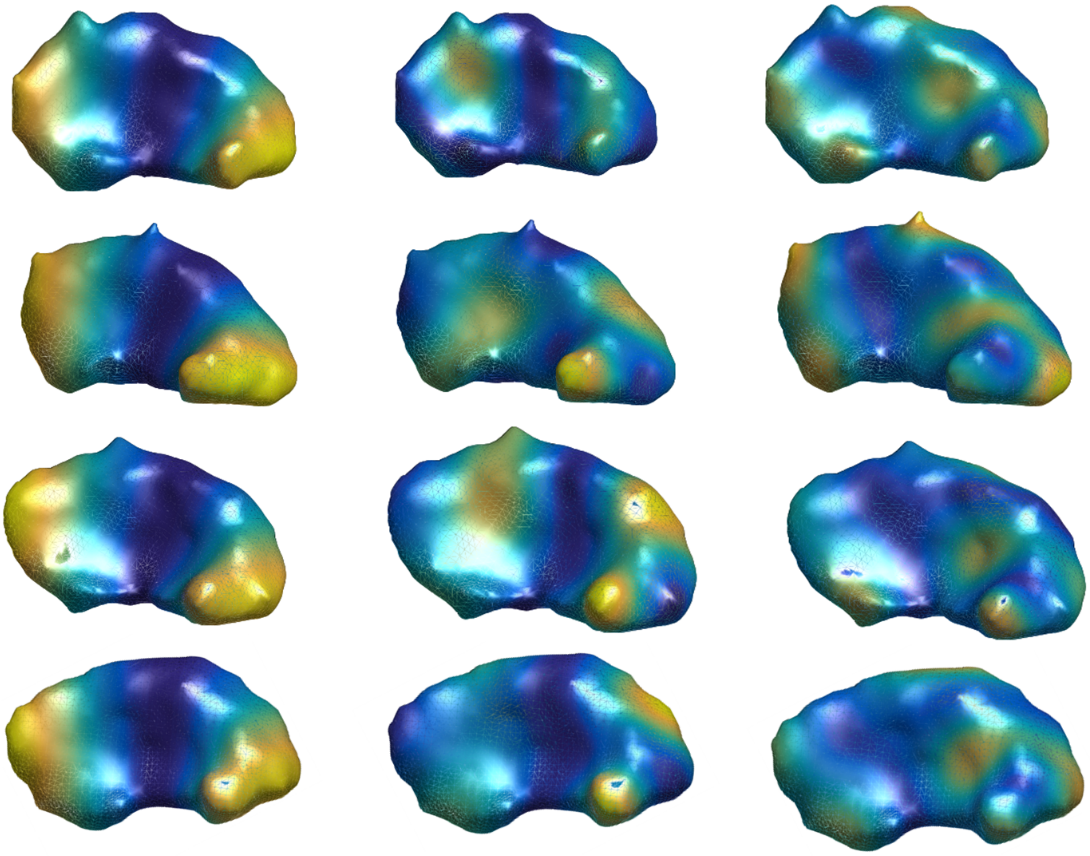


Fig. 4.4: Distribution of WKS values for several shapes of putamen region and energy values. From left to right $E = 10, 140, 180$. Lines 1 and 2 subjects with bipolar disorder, lines 3 and 4 normal controls.

our choices of function and operator constraints. We adopt the original constraints proposed by the authors in [115]. They adopted the LBO for the operator commutativity and as functions constraint the two spectral point descriptors HKS and WKS, introduced in Chapter 3, that are known to be stable and invariant to isometries. Some of these signatures are shown in Figure 4.4.

As shown in Section 4.1, we can estimate the map C for every pair of surfaces $(\mathcal{M},\mathcal{N})$, fixing as basis the LBO eigenfunctions of these surfaces. For the sake of clarity we denote with $C_{\mathcal{M},\mathcal{N}}$ the map between $\mathcal{M}$ and $\mathcal{N}$. Now, we need a specific kernel based on this map to perform our classification task. Given the the pair $(\mathcal{M},\mathcal{N})$, we compute two maps: i) $C_{\mathcal{M},\mathcal{N}}$ defined from $L^2(\mathcal{M})$ to $L^2(\mathcal{N})$ and, ii) the inverse $C_{\mathcal{N},\mathcal{M}}$. Clearly the exact Functional map from a functional space $L^2(\mathcal{M})$ to itself is the identity map $Id_{\mathcal{M}}$.

If the estimated maps are correct we can draw the following commutative diagram:

$$\begin{tikzcd}
L^2(\mathcal{M}) \arrow[r, "C_{\mathcal{M},\mathcal{N}}"] \arrow[dr, "Id_{\mathcal{M}}"'] & L^2(\mathcal{N}) \arrow[d, "C_{\mathcal{N},\mathcal{M}}"] \\
& L^2(\mathcal{M})
\end{tikzcd}$$

This diagram shows that a function should remain the same when it is moved from shape $\mathcal{M}$ to $\mathcal{N}$, and than put it back to $\mathcal{M}$ again.

In order to quantify how well the maps $C_{\mathcal{M},\mathcal{N}}$ and $C_{\mathcal{N},\mathcal{M}}$ have been calculated, we can define the following measure:

$$||C_{\mathcal{N},\mathcal{M}}C_{\mathcal{M},\mathcal{N}} - Id_{\mathcal{M}}||_F, \tag{4.13}$$

which tells us how much the previous diagram is actually commutative. Now we infer that if two surfaces are in the same class, i.e. they do not differ sensibly, we can compute $C_{\mathcal{M},\mathcal{N}}$ and $C_{\mathcal{N},\mathcal{M}}$ in a sufficiently exact way, such that $C_{\mathcal{M},\mathcal{N}}C_{\mathcal{N},\mathcal{M}} \approx Id_{\mathcal{M}}$. Thus for surfaces that belongs to the same class we obtain small value in the equation 4.13, conversely these score will be higher if the surfaces come from different classes. At this point we can advisedly define the following distance function:

$$d(\mathscr{M},\mathscr{N}) = \frac{1}{2}(||C_{\mathscr{N},\mathscr{M}}C_{\mathscr{M},\mathscr{N}} - Id_{\mathscr{M}}||_F + ||C_{\mathscr{M},\mathscr{N}}C_{\mathscr{N},\mathscr{M}} - Id_{\mathscr{N}}||_F). \quad (4.14)$$

This distance function has the following nice properties:

- *Symmetry:* $d(\mathscr{M},\mathscr{N}) = d(\mathscr{N},\mathscr{M}), \forall \mathscr{M},\mathscr{N}$.
- *Zero diagonal:* $d(\mathscr{M},\mathscr{M}) = 0, \forall \mathscr{M}$.
- *Nonnegativity:* $d(\mathscr{M},\mathscr{N}) \geq 0, \forall \mathscr{M},\mathscr{N}$.

Thanks to this properties and referring to [58] we can define a *distance substitution kernel* on the distance d, that we will call *Functional maps kernel* (FM-kernel). Given a collection of surfaces $\{\mathscr{M}_i\}_{i\in I}$ we define the FM-kernel as:

$$K(i,j) = e^{-\gamma d(\mathscr{M}_i,\mathscr{M}_j)^2}, \forall i,j \in I. \quad (4.15)$$

As shown in [58] the obtained kernel can be successfully applied in SVM for classification.

4.3.3 Results

In this Section we show how the Functional maps framework together with our new FM-kernel improve the brain classification performance on the spectral domain. With this aims we explore the comparison with all the spectral methods that are more related to our framework. We also report the results obtained using different classifiers, namely the Support Vector Machines (SVM) and the Nearest Neighbour (NN) classifier.

Materials

We analyze a dataset of patients affected by bipolar disorder and healthy control subjects. More precisely, 34 control subjects (22 males, 29 +/- 5 years old (y.o.)), 34 patients affected by bipolar disorder (15 males, 45 +/- 13 y.o.) underwent an MRI session. MRI data were obtained using a Siemens 3.0 T Magnetom Allegra MRI scanner (Siemens Ag). The following parameters were used for T1-weighted images: 256 x 256 x 256 voxels, $1 \times 1 \times 1\ mm^3$, TR 2060 ms, TE 3.93 ms, flip angle 15 degree. Cortical and

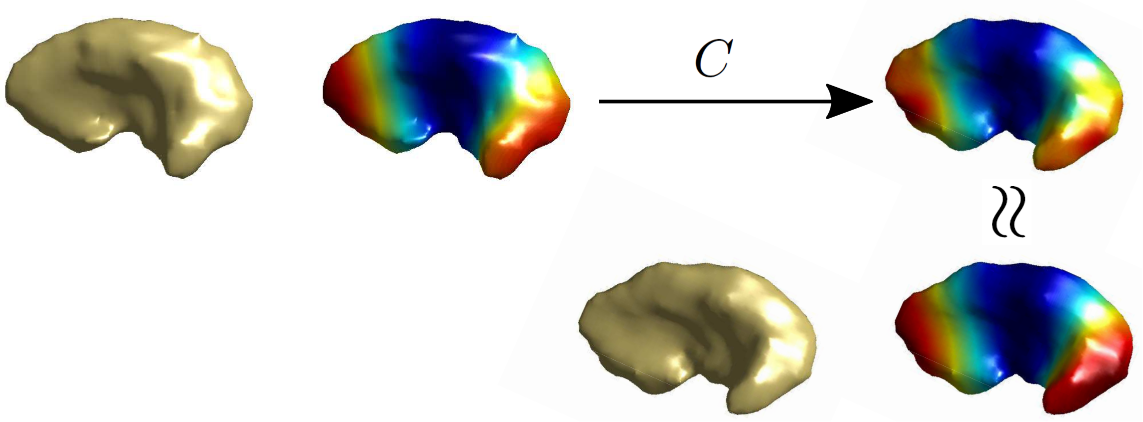


Fig. 4.5: A couple of putamen surfaces and two WKS descriptors computed on them. Through the C map calculated using the Functional maps framework, we transport the function defined on the first shape to the second one and as shown we achieve a good approximation of the desired function on the second shape.

subcortical surfaces were obtained using FreeSurfer version 4.3.1[1] [38]. First, non-brain tissues were excluded, then images were segmented into white and gray matter (WM and GM respectively), and then, meshes of the boundaries between WM and GM and between GM and CSF were estimated. We focused on the *putamen*, a deep gray matter brain structure, which is thought to be modified in the shape in subjects that are affected by bipolar disorder [66]. The process encoded by the functional map framework is shown in Figure 4.5. The function defined on the first shape is represented by the WKS descriptor. Such function is map to the second shape by using C by showing that the transported WKS values are very similar to the original one.

Comparison with other methods

We compare our method with the state of the art spectral methods. In order to establish how much the Functional maps framework and the proposed FM-kernel improve the classification results, we select meth-

[1] http://surfer.nmr.mgh.harvard.edu/

ods that encode the same information used in our Functional maps construction. We consider the *Shape-DNA* (S-DNA) descriptor [125], i.e., the increasing ordered sequence of the first LBO eigenvalues. We evaluate also the so called *Global Heat Kernel Signature* (GHKS) [26], a multiscale histogram representation of the pointwise HKS. Similarly, we define the *Global Wave Kernel Signature* (GWKS) based on the WKS. Finally, since in our method the information coming from LBO, HKS, and WKS is integrated into the same framework we carried out a further evaluation with descriptors obtained by the concatenation of GHKS and GWKS (GHKS+GWKS), or GHKS, WKS and S-DNA (ALL3desc).

Comparison with different classifiers

we show the results obtained by different choices of classifiers. Specifically, we adopt a K-Nearest Neighbor (KNN) classifier, with $k = 6$ and the standard SVM classifier using LIBSVM [28]. A cross-validation scheme is introduced to estimate the SVM parameters as suggested in [28].

Classification Results						
Method:	**Ours**	**S-DNA**	**GHKS**	**GWKS**	**GHKS+GWKS**	**ALL3desc**
SVM	**72.06**	70.59	67.65	69.12	69.12	70.59
KNN	**64.71**	60.29	63.24	61.76	60.29	63.24

Table 4.2: Results in classification for the bipolar disorder on the putamen shapes. The evaluated methods are SVM and KNN classifiers

Table 4.2 shows the results. Our proposed approach outperforms all the other methods, also in their joined version. This confirm our claim that performing the classification on the functional space improves the results. Nevertheless, our method performed at best for both SVM and KNN showing its independence from the choice of classifier. Since the proposed FM-kernel is designed specifically for the SVM classifier it does not surprise that the best performance was obtained with this classifier.

4.3.4 Limitations and future work

In this application the Functional maps approach for brain classification in the spectral domain is proposed. We introduced a specific kernel for SVM classification, namely the FM-kernel, based on the integration among different spectral shape analysis operators and descriptors. We evaluated our new classification method for bipolar disorder detection on the putamen regions by showing very promising results in comparison with other spectral-based approaches. As future works we consider to learn more suitable spectral descriptors for specific tasks as suggested in [36]. In particular, we will focus on the reduction of the importance of the isometry constraint between shapes that is difficult to justify from the clinical point of view even if it is working well in practice. We could also include further information related to the anatomical structure as additional constraints for the Functional maps framework such as the parts of a prior available shape segmentation procedure. Moreover, we will consider that the major variability is for the non-healthy subjects and therefore a new classifier based on the training of a single class (the healthy one) will be considered. Finally, a more exhaustive clinical evaluation will be carried out by exploring other brain regions and by enlarging the cohort of available subjects.

5

Localized spectral geometry processing

The advantages of spectral frameworks is that they are completely intrinsic by construction and work only with eigenvalues and eigenfunctions of the LBO, thus naturally allowing to deal with shapes in any representation, e.g. meshes or point clouds. The main idea of this Chapter is to formulate the classical time-frequency analysis to signals lying on a 3D surface. Classically the goal of the time-frequency analysis is to obtain a good localization in the time together with a good localization in the frequency domain. In this thesis instead of the localization in time we consider the spatial localization on the surface. In this fashion we obtain a local frequency representation which encodes effectively and efficiently the local "context" of a given point. The goal of the work collected in this Chapter is to propose methods to face the uncertainty principle *introduced in Chapter 2. In other words we would like to be able to analyze the signal defined on the surface with a good localization both in the spatial and in the spectral domain.*

5.1 Need for spectral localization

In this Chapter we briefly point out the motivations for the *windowed Fourier transform* (WFT) in the classic case. As we saw in the Chapter 2 the Fourier transform suffers from the so-called uncertainty principle. In a nutshell the Fourier transform is not able to produce an analysis localized simultaneously in the time and in the frequency domain. The main idea of WFT is to first isolate a local part of the signal (in the classic case using a short temporal window), then analyze only the spectral representation of the localized signal instead of considering the entire original signal. By isolating a part of the signal, the temporal (spatial) localization is achieved while the spectral localization guaranteed by the used basis is maintained.

5.2 Windowed Fourier transform

In this Section, we propose a new framework for constructing class-specific dense intrinsic shape descriptors based on vertex-frequency analysis [138]. The core of our construction is the windowed Fourier transform, allowing to capture local context around a point on a surface and represent it in the frequency domain. Combined with a learning framework reminiscent of convolutional neural networks, we are able to learn discriminative and robust descriptors that are specific to a given class of shapes. There are different important factors that must be defined to design the descriptor: i) the initial signal, ii) the window function, iii) the range of frequencies.

5.2.1 Overview

As we saw before the Fourier basis functions used to compute frequency representations of signals, are the eigenfunctions of the Laplace operator. This interpretation allows to naturally generalize Fourier analysis to non-Euclidean domains (manifolds or graphs) by considering the respective Laplacian of these domains [87]. Signal processing on geometric data is an active field of research since the seminal work of Taubin [143], introducing a Laplacian-based approach for mesh smoothing. His approach found extensive application in several other mesh processing tasks, including mesh parametrization [108] and progressive meshes

[65], and mesh compression [74] to name a few (we refer to [87] for an excellent survey). In the area of shape analysis, recent efforts concentrated on the definition of spectral isometry-invariant shape descriptors for matching and retrieval purposes [21, 125, 133]. In [125] a *global* shape descriptor was proposed (*Shape-DNA*). In [133] the so-called global point signature (GPS) was introduced to encode the relation between a single point and all the others on the surface. Of particular interest is the family of spectral methods that build upon the ideas of diffusion geometry [35] in order to encode local geometric structures. Examples of such methods include the heat kernel signature (HKS) [50, 140] and the wave kernel signature (WKS) [5]. Another interesting approach was proposed in [79] where the intrinsic local “context” has been defined for 3D points. It is worth noting that the majority of spectral methods rely on a *global* frequency analysis, considering the Laplacian eigenfunctions as the basis. *Localized* Fourier analysis (a standard construction in signal processing known as the short-time or windowed Fourier transform) has recently been done on graphs [138]. The Fourier approach has been successfully applied for *localized* frequency analysis for 2D image recognition tasks [12], where the spectral information is used to define effective *local* image descriptors. For all these reasons we focus on the definition of a localized Fourier analysis on surfaces extending the windowed Fourier framework on 3D shapes for point-to-point matching. In this Section we explore the use of localized frequency analysis (a generalization of the windowed Fourier transform to manifolds) for the design of intrinsic shape descriptors. Applying the windowed Fourier transform to some dense intrinsic descriptor adds a local "context" capturing richer geometric structures. The resulting local frequency representations are then passed through a bank of filters whose coefficient are determined by a learning procedure minimizing a task-specific cost. Conceptually, our approach is reminiscent of convolutional neural networks (CNN), and also generalizes several previous methods such as spectral CNN and

GPS embeddings. Our experimental results show that the proposed approach allows learning class-specific shape descriptors significantly outperforming recent state-of-the-art methods on standard benchmarks.

5.2.2 Standard Windowed Fourier transform

A central piece to our construction of shape descriptors is the notion of *windowed Fourier transform (WFT)*, generalizing this construction from classical signal processing to non-Euclidean domains. Here, we follow the approach of Shumann et al. [138] for the generalization of the WFT in the spectral domain. We note that in principle other methods for hierarchical and local frequency analysis on graphs or manifolds can be used instead of the presented construction, including wavelets [35] or compressed modes [110].

Classical WFT

The main idea of classical WFT is to analyze the frequency content of a signal that is localized by means of multiplication by a window. Given a function $f \in L^2(\mathbb{R})$ and some 'mother window' g localized at zero, one computes the WFT as

$$(Sf)_{x,\omega} = \int_{\mathbb{R}} f(x')g(x'-x)e^{-ix'\omega}dx'. \tag{5.1}$$

Note that the WFT has two indices: spatial location x of the window and frequency ω of the signal in that window. Alternatively, it can be presented as an inner product with a translated an modulated window, $(Sf)_{x,\omega} = \langle f, M_\omega T_x g\rangle_{L^2(\mathbb{R})}$, where T_x and M_ω denote the translation and modulation operators, respectively.

Translation operator

In the Euclidean setting translation is simply $(T_{x'}f)(x) = f(x-x')$. In order to generalize it to manifolds, translation to point $x' \in \mathscr{M}$ can be replaced by convolution with a delta-function centered at x', yielding

$$
\begin{aligned}
(T_{x'}f)(x) &= (f * \delta_{x'})(x) \\
&= \sum_{k\geq 1} \langle f, \phi_k \rangle_{L^2(\mathscr{M})} \langle \delta_{x'}, \phi_k \rangle_{L^2(\mathscr{M})} \phi_k(x) \\
&= \sum_{k\geq 1} \hat{f}_k \phi_k(x') \phi_k(x),
\end{aligned} \tag{5.2}
$$

where convolution is understood in the generalized sense of equation 2.12 and from the standard definition in equation 2.5. Note that such a translation is not shift-invariant in general, i.e., the window would change when moved around the manifold (see Figure 5.1).

Modulation operator

Modulation in the classical case is a multiplication by a basis function $(M_\omega f)(x) = e^{i\omega x} f(x)$. In the Fourier domain, the action of modulation amounts to translation $(\widehat{M_{\omega'} f})(\omega) = \hat{f}(\omega - \omega')$. In the generalized case, the modulation is defined in exactly the same way,

$$
(M_k f)(x) = \phi_k(x) f(x), \tag{5.3}
$$

where the eigenvalue λ_k corresponding to the eigenfunction ϕ_k plays the role of 'frequency'.

Manifold WFT

Combining the two operators together, we have the modulated and translated window (transform 'atom'; see examples in Figure 5.1) expressed as

$$
g_{x',k}(x) = (M_k T_{x'} g)(x) = \phi_k(x) \sum_{l\geq 1} \hat{g}_l \phi_l(x') \phi_l(x). \tag{5.4}
$$

Note that the 'mother window' is defined here in the frequency domain by the coefficients $\hat{g}_l$. We thus readily have the WFT of a signal $f \in L^2(\mathscr{M})$

$$
(Sf)_{x,k} = \langle f, g_{x,k} \rangle_{L^2(\mathscr{M})} = \sum_{l\geq 1} \hat{g}_l \phi_l(x) \langle f, \phi_l \phi_k \rangle_{L^2(\mathscr{M})}, \tag{5.5}
$$

with equation 5.5 the WFT is extended to surfaces.

The WFT can be regarded as a meta-descriptor: given some dense descriptor f (e.g. one of the components of HKS, WKS, or a geometry vector (3.16)), we construct $D(x)f = ((Sf)_{x,1},\dots,(Sf)_{x,K})^\top$ taking the first K frequencies of the WFT. The WFT allows to capture the local context of a signal on the manifold, making it roughly analogous to taking the values of the signal in a small “patch”; here $D(x)$ acts as a position-dependent “patch operator” representing the local structure of f around point x in the frequency domain.

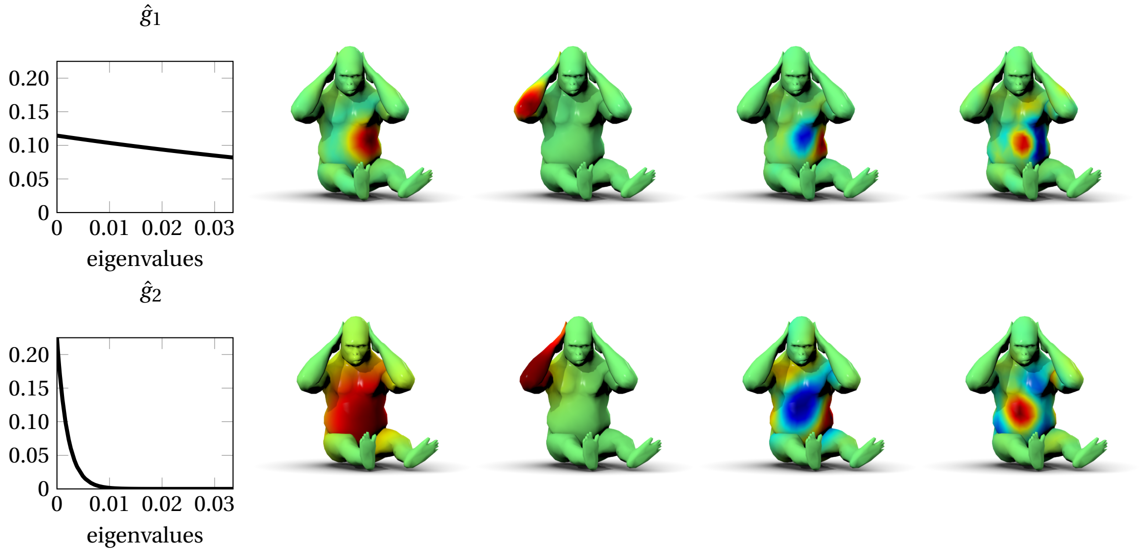


Fig. 5.1: Examples of different WFT atoms $g_{x,k}$ using different windows (top and bottom rows; window Fourier coefficients are shown on the left), shown in different localizations (second and third columns) and modulations (fourth and fifth columns).

Special cases

We would like to point out the following special cases of the WFT, which show that this framework can be considered as a generalization of several previous approaches.

Case I: when $\hat{g}_k = \delta_{k1}$, we simply have $g_{x',k}(x) = \phi_k(x)\phi_1(x')\phi_1(x)$. Since the first LBO eigenvector is constant, the atom (up to scaling) is $g_{x',k}(x) \propto \phi_k(x)$, i.e., the standard LBO eigenbasis element independent on the location x'. The WFT thus reduces to a simple Fourier transform (2.11). This result is an intuitive consequence of the uncertainty principle: when the window is perfectly localized in the frequency domain, it is perfectly delocalized in the spatial domain .

Case II: when $f \equiv 1$, the WFT contains information only about the geometric structure of the manifold. In this setting,

$$(S1)_{x,k} = \sum_{l \geq 1} \hat{g}_l \phi_l(x) \underbrace{\langle \phi_k, \phi_l \rangle_{L^2(\mathcal{M})}}_{\delta_{kl}} = \hat{g}_k \phi_k(x), \tag{5.6}$$

and for a particular choice of $\hat{g}_k = \lambda_k^{-1/2}$ we get Rustamov's GPS descriptor (2.16). *Case III:* when $f = \delta_x$, the DC frequency of the WFT has the form of (3.14),

$$(S\delta_x)_{x,1} = \sum_{l \geq 1} \hat{g}_l \phi_l^2(x), \tag{5.7}$$

and in particular for $\hat{g}_l = \exp\left(\frac{\log \nu - \log \lambda_l}{2\sigma^2}\right)$ we obtain the WKS and for $\hat{g}_l = e^{-t\lambda_l}$ the HKS at point x, respectively.

Discretization

The discretized WFT is computed as

$$S\boldsymbol{f} = (\boldsymbol{f} \boxdot \boldsymbol{\Phi})^\top \, \boldsymbol{A} \, \boldsymbol{\Phi}(\hat{\mathbf{g}} \boxdot \boldsymbol{\Phi}^\top), \tag{5.8}$$

where $\hat{\mathbf{g}}$ is the K-dimensional vector representing the window in the frequency domain, $\mathbf{f}$ is the N-dimensional vector representing the input function, and $(\mathbf{a} \boxdot \mathbf{B})_{ij} = a_i b_{ij}$ denotes element-wise multiplication of a vector and matrix, replicating the vector along the second dimension (`repmat` in MATLAB). The resulting WFT is a matrix of size $K \times N$.

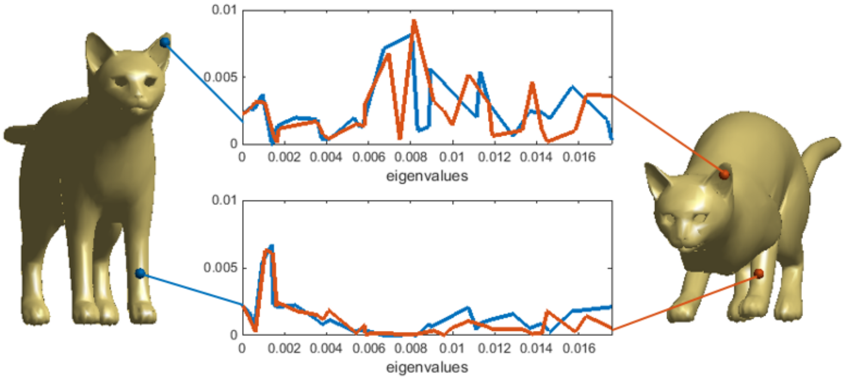


Fig. 5.2: WFTs computed on the constant function on two different poses of the same shape at two points marked in orange and blue.

5.2.3 Localized spectral CNN

Note that the definition of the WFT atoms involves the basis functions $\{\phi_i\}_{i\geq 1}$, therefore, in principle, they are susceptible to change if the Laplacian eigenbasis changes. In particular, eigenfunctions sign flips can change the sign of the WFT. However, at the same time it is known that Laplacian eigenfunctions tend to change according to certain patterns, which can be modelled and taken advantage of [122]. In general, the WFT is not completely invariant, i.e., the descriptors at corresponding points of isometric shapes may differ as can be seen in Figure 5.2. In this section we describe a learning framework that allows to account for this variability. The architecture of our descriptor bears similarity with convolutional neural networks (CNN) [23,83,99] that are now very popular in computer vision applications, and has a layered structure, wherein the output on one layer is used as the input of the next one.

Convolutional neural networks [82] are hierarchical architectures built of alternating convolutional, pooling (non-linear averaging), and fully connected layers. The parameters of different layers are learned by minimizing some task-specific cost function. In image analysis applications, the input into the CNN is a function representing pixel values given on a Euclidean domain (plane); due to shift-invariance the convolution can be thought of as passing a template across the plane and recording the correlation of the template with the function at that location. One of the major problems in applying the CNN paradigm to non-Euclidean domains is the lack of shift-invariance, making it impossible to think of convolution as correlation with a fixed template: the template now has to be location-dependent. Here, we propose using the WFT as a mechanism for extracting local 'patches' from functions defined on manifolds. We can think of $D(x)f = ((Sf)_{x,1}, \ldots, (Sf)_{x,K})$ as a position-dependent "patch operator" representing the local structure of f around point x in the frequency domain. The spatial support of the 'patch' depends on the choice of the window g. Note that in the definition of the WFT the geometric structure of the manifold is captured by the Laplace-Beltrami eigenfunctions. As a result, the same framework can be used for any shape representation (e.g. mesh, point cloud, etc.): the specific representation of the shape influences only the construction of the Laplace-Beltrami operator.

We refer to our approach as *localized spectral CNN* (LSCNN). For the sake of simplicity, the neural network architecture considered in the following consists of only two layers (comparable with the SN1 architecture in [99]). The first layer is a *fully connected layer*, producing outputs as weighted sums of the inputs, followed by a non-linear function. The second layer applies the WFT to extract the local structure of the input around each point. Since each input dimension might contain features of different scale, we employ a different window for each input dimension. The WFTs are then passed through a bank of filters applied in the frequency domain, producing the outputs used as the descriptor dimensions. As the input to the first layer, any intrinsic descriptor can be used (specifically, we use geometry vectors defined in equation (3.16)). All the parameters of the layers (weights, windows coefficients, and filters) are variables that are found by means of supervised learning.

Fully connected layer

Let us be given a P-dimensional input $\mathbf{f}^{\text{in}}(x) = (f_1^{\text{in}}(x),\dots,f_P^{\text{in}}(x))$. The fully connected layer produces a Q-dimensional output defined as

$$f_q^{\text{out}}(x) = \xi\left(\sum_{p=1}^{P}\sum_{k=1}^{K} w_{qp} f_p^{\text{in}}(x)\right), \quad q = 1,\dots,Q, \tag{5.9}$$

where $\xi(t) = \max(0,t)$ is the *ReLU activation function*. Note that without ReLU, if the inputs are geometry vectors, learning the weights of the fully connected layer is equivalent to the OSD [90]. Fixing weights corresponding to low- or band-pass filters, the fully connected layer implements the HKS and WKS, respectively.

Convolutional layer

Next, the output of the fully connected layer acts as the input into the convolutional layer; we denote the input again by $\mathbf{f}^{\text{in}}(x)$ and its dimension by P. For each input dimension, we use a different window. The family of P windows is parametrized in some fixed interpolation basis in the frequency domain as in (3.15),

$$\gamma_p(\lambda) = \sum_{m=1}^{M} b_{pm}\beta_m(\lambda), \qquad p = 1,\dots,P, \tag{5.10}$$

where the $P \times M$ matrix (b_{pm}) of weights defines the windows. Figure 5.3 shows an example of the estimated windows after the learning. The WFT of the pth input dimension uses the respective pth window,

$$(Sf_p^{\text{in}})_{x,k} = \sum_{l\geq 1}\gamma_p(\lambda_l)\phi_l(x)\langle f_p^{\text{in}}, \phi_l\phi_k\rangle_{L^2(\mathcal{M})}, \tag{5.11}$$

producing at each point a K-dimensional vector for each of the P input dimensions. Our goal is to produce a Q-dimensional output, and for this purpose, the WFTs are passed through a bank of filters. The qth dimension of the output is given by

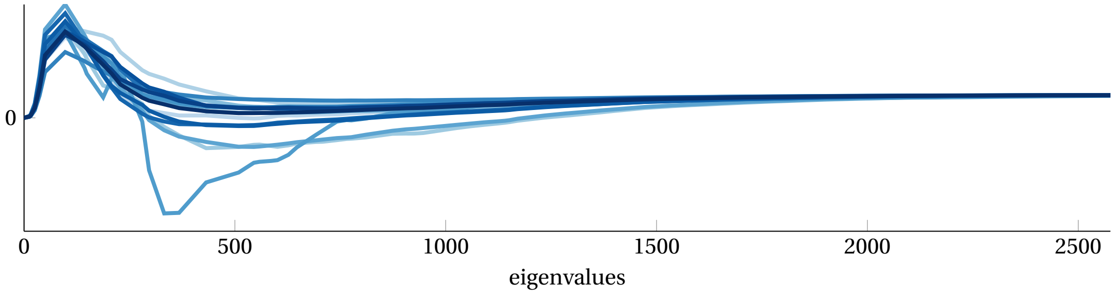


Fig. 5.3: Example of a family of windows $\hat{g}_1, \dots, \hat{g}_P$ learned by the LSCNN on the FAUST dataset.

$$f_q^{\text{out}}(x) = \sum_{p=1}^{P} \sum_{k=1}^{K} a_{qpk} |(S f_p^{\text{in}})_{x,k}|, \quad q = 1, \dots, Q. \tag{5.12}$$

The output of the convolutional layer is used as our final LSCNN descriptor.

Loss function

The LSCNN comprising the fully-connected and convolutional layer is a parametric hierarchical system $\mathbf{f}_\Theta(x)$ producing a Q-dimensional descriptor at each point x (here $\Theta = \{(w_{qp}), (b_{pm}), (a_{qpk})\}$ denotes the set of learnable parameters). Given a training set of knowingly similar and dissimilar pairs of points on pairs of shapes, respectively positives $\mathcal{T}^+ = \{(x, x^+)\}$ and negatives $\mathcal{T}^- = \{(x, x^-)\}$, we aim at estimating the optimal task-specific parameters of the descriptor minimizing the aggregate loss

$$\ell(\Theta) = (1-\mu)\ell_+(\Theta) + \mu\, \ell_-(\Theta) \tag{5.13}$$

where

$$\ell_+(\Theta) = \frac{1}{|\mathcal{T}^+|} \sum_{(x,x^+)\in\mathcal{T}^+} \|\mathbf{f}_\Theta(x) - \mathbf{f}_\Theta(x^+)\|_2^2, \tag{5.14}$$

$$\ell_-(\Theta) = \frac{1}{|\mathcal{T}^-|} \sum_{(x,x^-)\in\mathcal{T}^-} \max\{0, M - \|\mathbf{f}_\Theta(x) - \mathbf{f}_\Theta(x^-)\|^2\},$$

are the positive and negative losses, respectively, μ is a parameter governing their trade-off, and M is a margin mapping the negatives apart.

We stress that HKS, WKS, and OSD descriptors are obtained by a particular choice of the parameters Θ. Thus, if the training set is designed well and training is performed correctly, our descriptor can perform only better than the above.

Comparison to ShapeNet

Masci et al. [99] introduced ShapeNet, a generalization of CNN to triangular meshes based on geodesic local patches. The core of this method is the construction of local geodesic polar coordinates using a procedure previously employed for intrinsic shape context descriptors [79]. The patch operator $(D(x)f)(\theta,\rho)$ in ShapeNet maps the values of the function f around vertex x into the local polar coordinates θ, ρ, leading to the definition of the *geodesic convolution*

$$(f * a)(x) = \sum_{\rho,\theta} a(\theta + \Delta\theta)(D(x)f)(\theta,\rho), \tag{5.15}$$

which follows the idea of multiplication by template, but is defined up to arbitrary rotation $\Delta\theta \in [0, 2\pi)$ due to the ambiguity in the selection of the origin of the angular coordinate. In the ShapeNet convolutional layer, the outputs corresponding to all the rotations of the templates are produced and then a maximum is taken,

$$f_q^{\text{out}} = \max_{\Delta\theta} \sum_{p=1}^{P} f_p^{\text{in}} * a_{\Delta\theta,qp}, \tag{5.16}$$

where $a_{\Delta\theta}(\theta,\rho) = a(\theta+\Delta\theta,\rho)$ denotes the coefficients of the template rotated by $\Delta\theta$, and the convolution is in the sense of equation (5.15).

We note the following main drawbacks of this construction. First, the charting method relies on a fast marching-like procedure requiring a triangular mesh. The method is relatively insensitive to the triangulation, but may fail if the mesh is very irregular. Second, the radius of the geodesic patches must be sufficiently small compared to the convexity radius of the shape, otherwise the resulting patch is not guaranteed to be a topological disk. In practice, this limits the size of the patches one can safely use, or requires an adaptive radius selection mechanism. In contrast, the proposed localized spectral CNN is free of these limitations: it can work with any shape representation, provided one can compute the discretized Laplace-Beltrami operator and its eigenfunctions and eigenvalues for this representation; since the patch operator is constructed in the frequency domain using the WFT, there is also no issue related to the topology of the patch.

5.2.4 Results

Datasets

We used two datasets of scanned human shapes in different poses: SCAPE [4] and FAUST [13] already introduced in Chapter 2. The meshes in SCAPE were resampled to 12.5K vertices, whereas for FAUST we used the registration meshes without further pre-processing. In addition we scaled all shapes to have unit geodesic diameter.

Methods and Settings

In all our experiments, we used $K = 300$ LBO eigenfunctions and eigenvalues computed using MATLAB *eigs* function. For OSD and our descriptor, we used $M = 150$-dimensional geometry vectors as inputs, computed according to (3.15)–(3.16) using B-spline bases [90]. We compared the performance of the proposed approach to HKS [140], WKS [5], OSD [90], spectral CNN (SCNN) [23], and ShapeNet (SN1) [99] using the code and settings provided by the respective authors. To make the comparison fair, all the descriptors were $Q = 16$-dimensional as in [90]. Our descriptor was tested in two configurations. *LSCNN1*, consisting of a fully connected layer (reducing the dimensionality of the 150-dimensional input to 16 dimensions), followed by a convolutional layer using a fixed WFT

Gaussian window $\gamma(\lambda) = e^{-\frac{\lambda^2}{2\sigma^2}}$ with $\sigma = 10^{-5}$. In this configuration the parameters of the network that are learned are $\Theta = \{(w_{qp}), (a_{qpk})\}$. *LSCNN2* is similar to LSCNN1, with the difference that now the WFT windows are also learned. We use 16 filters (one per dimension), each represented by the B-spline coefficients. In this configuration, the free parameters are $\Theta = \{(w_{qp}), (b_{pm}), (a_{qpk})\}$. Furthermore, as a 'sanity check', we also used a configuration without the convolutional layer, comprising only a fully connected+ReLU layer (referred to as NN1). This architecture is compatible with the OSD, with the addition of a non-linearity at the output.

Training

Each dataset was split into disjoint training, validation, and test sets. On the FAUST dataset subjects 1–7 were used for training (10 poses per subject, a total of 70 shapes), subject 8 (10 shapes) for validation, and subject 9–10 for testing (total of 20 shapes). On SCAPE, we used shapes 20–29 and 50–70 for training (total 31 shapes), five different shapes for validation, and 40 remaining shapes for testing. The positive and negative sets of vertex pairs required for training were generated on the fly, to keep the storage requirements for the training algorithm, via uniform stochastic sampling. Each point on the first shape has only a single ground truth match (given by the one-to-one correspondence) and is assigned to one out of $N - 1$ possible negatives: first, sample two shapes, then form the positive set with all corresponding points, and finally, form the negative set with first shape vertices and a random permutation of the ones of the second shape. This strategy differs from [90] who considered only points on the same shape. The advantage of our sampling strategy is that it allows learning invariance also across several poses and subjects. LSCNN was implemented in Theano [11] and trained until convergence using Adadelta [153], a stochastic first order method with automatic adjustment of the learning rate (step size). Training was performed for 250 epochs, each epoch consisting of 100 updates (stochastic gradient descent steps). In each update of the training, we used N positive and negative pairs, where N is the number of shape vertices.

Timing

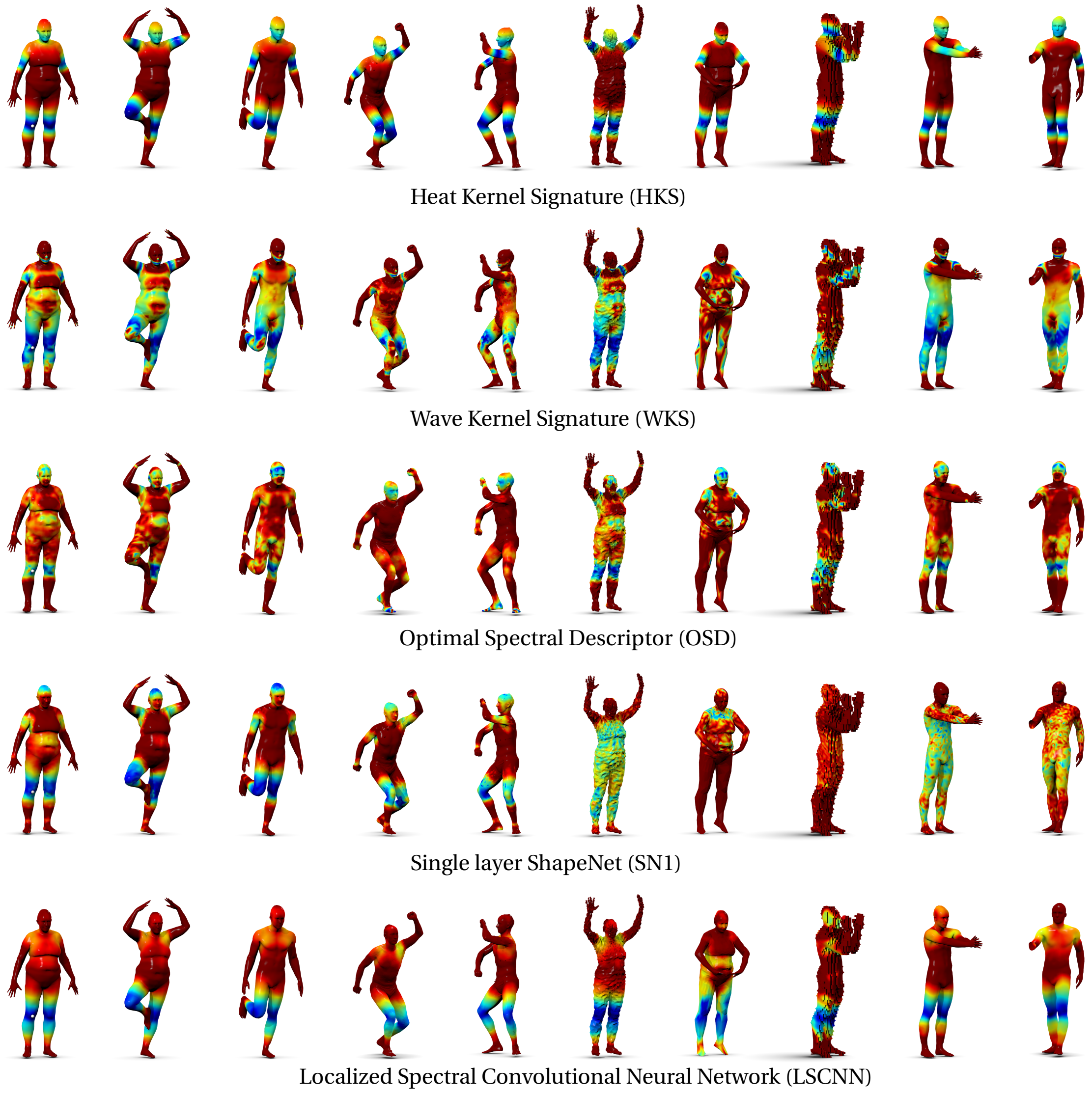
Typical training times for the more complex descriptor (LSCNN2) are around two hours on a NVIDIA TITAN Black GPU board and, at test time, the system is able to produce a throughput of approximately 30K vertices per second. The pre-computation of the LB operator and its eigendecomposition takes around 10s for a shape with 7K vertices.

Similarity map

Figures 5.4 (compare to Figure 2 in [90] and Figures 5–6 in [99]) depicts the Euclidean distance in the descriptor space between the descriptor at a selected point and the rest of the points on the same shape as well as its transformations. Figure 5.5 shows another example of LSCNN on point clouds, where the WFT was computed using the graph Laplacian. Our approach shows a good tradeoff between localization (similar to HKS) and accuracy (less spurious minima than WKS and OSD), as well as robustness to different kinds of noise.

Descriptor evaluation

We evaluated the descriptor performance using the *cumulative match characteristic* (CMC) and the *receiver operator characteristic* (ROC). The CMC evaluates the probability of a correct correspondence among the k nearest neighbors in the descriptor space. The ROC measures the percentage of positives and negatives pairs falling below various thresholds of their distance in the descriptor space (*true positive* and *negative rates*, respectively). The correspondence quality possible with our descriptors was evaluated using the Princeton protocol [76], plotting the percentage of nearest-neighbor matches that are at most r-geodesically distant from the ground truth correspondence.



Heat Kernel Signature (HKS)

Wave Kernel Signature (WKS)

Optimal Spectral Descriptor (OSD)

Single layer ShapeNet (SN1)

Localized Spectral Convolutional Neural Network (LSCNN)

Fig. 5.4: Distance map in the descriptor space. A point on the reference shape (leftmost) is compared to all other points on the same and on other shapes. Shown left-to-right: reference shape from FAUST dataset, different pose of the same shape, different subject in the same dataset, two shapes from SCAPE dataset, Gaussian noise, heavy subsampling, voxelization noise, topological noise (glued fingers and missing parts). Small distances in the descriptor space correspond to cold colors.

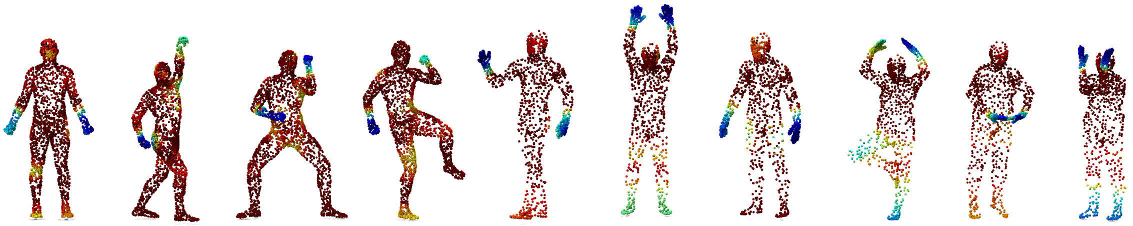


Fig. 5.5: Distance map in the descriptor space computed using LSCNN on point clouds. A point on the reference shape (leftmost) is compared to all other points on the same and on other shapes (four from SCAPE and four from FAUST datasets). Small distances in the descriptor space correspond to cold colors.

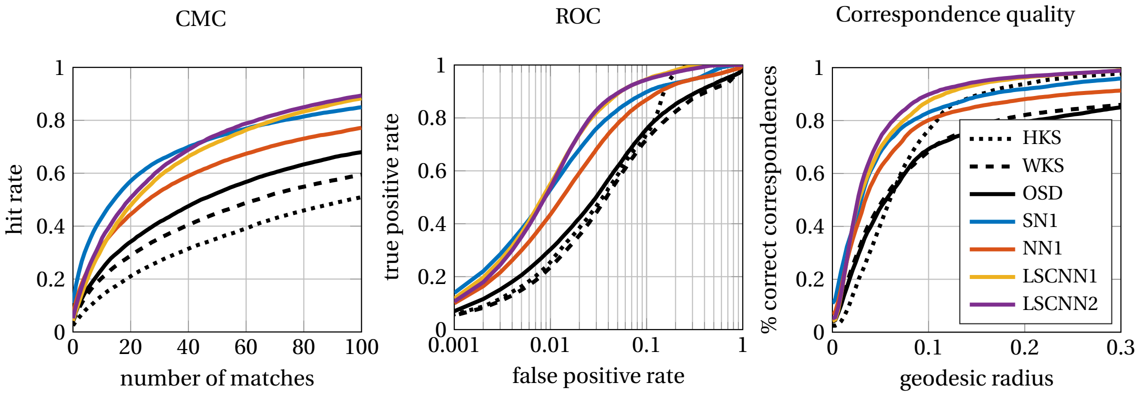


Fig. 5.6: Performance of descriptors trained on a subset of FAUST dataset and tested on a disjoint subset of FAUST dataset.

The performance evaluation is depicted in Figures 5.6–5.8. We observe that NN1 (fully connected layer+ReLU) outperforms the OSD, which we attribute to the non-linearity. We see further significant improvement from using a convolutional layer (LSCNN1 and LSCNN2). Furthermore, we observe that LSCNN generalizes better to data from a different dataset (transfer learning from FAUST to SCAPE) compared to ShapeNet. In this Section we propose a new shape descriptor for local frequency analysis on the 3D surface domain. We show that the Windowed Fourier Transform can be extended to discretized 3D manifold by leading to a very expressive and well localized intrinsic feature point descriptor. In partic-

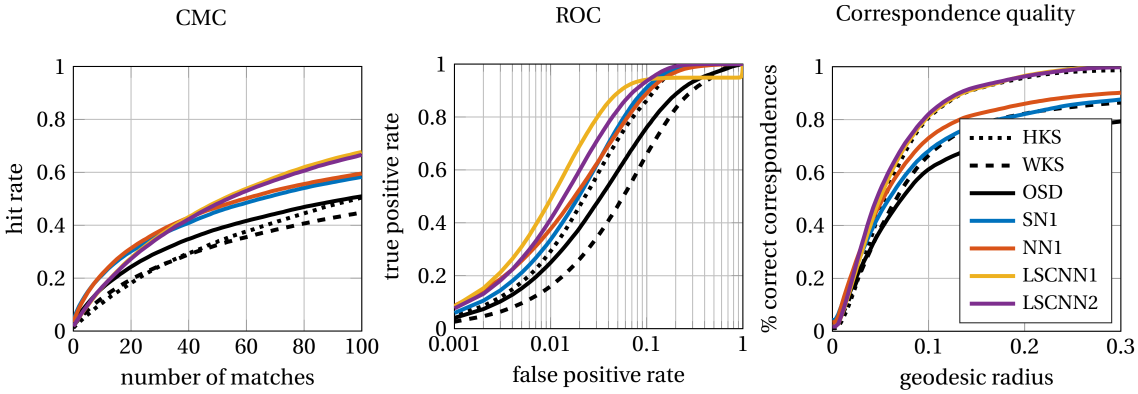


Fig. 5.7: Performance of descriptors trained on a subset of FAUST dataset and tested on SCAPE dataset.

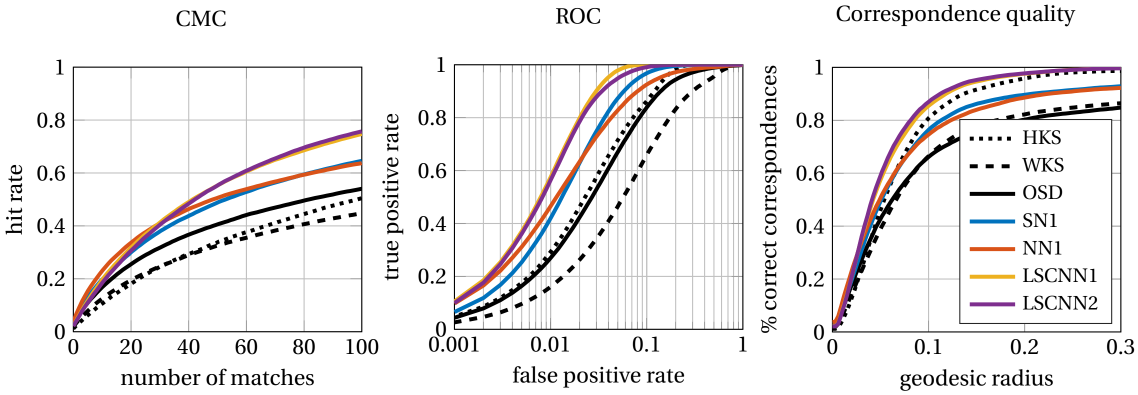


Fig. 5.8: Performance of descriptors trained on a of SCAPE dataset and tested on a disjoint subset of SCAPE dataset.

ular, the exploration of a learning based framework has demonstrated its strength and versatility in designing the desired behaviour of the proposed descriptor. Broadly speaking we believe that our approach will open new perspective of the use of advanced signal processing techniques on non Euclidean domain like the 3D shapes. For more details, please refer to [14].

Limitations

The construction of class-specific descriptors tacitly assumes that all shapes in the class share some common geometric structure, and their Laplacian eigenbasis, up to known ambiguities, do not differ arbitrarily. We hypothesize that if one tries to deal with a class that is too broad (e.g. all mechanical objects, or all living things), the performance advantage of our method over 'hand-crafted' descriptors such as HKS and WKS will diminish, and it is likely that we will learn these descriptors (as they are a particular configuration of our network). The learning based framework being supervised, it depends on the data available.

Extensions

The spectral formulation of our framework allows application to a broad range of geometric structures, such as point clouds or even abstract graphs and networks. Constructing an analogy of successful convolutional neural networks on such domains has been elusive so far, as there is no clear notion of a local 'patch' and its representation. We believe that our approach could be the right path towards this goal.

This work highlights how important is the effect of LBO eigenbasis ambiguity in the spectral shape decompositions, (an example can be seen in Figure 5.2). Our class-specific learning approach is able to overcome this limit but still some open issues remain, especially for high generalization purposes. Generally the only reliable eigenbasis are those associated to the first eigenvalues by leading to a simplified encoding of the shape composed of low frequencies (i.e., mean shape). Some method to include a matching of eigenbasis in the learning framework will be addressed as future work. Alternatively we might look for other methods to avoid this heavy dependence on the LBO eigenbasis ambiguity. A first alternative is proposed in the following Section.

5.3 Anisotropic windowed Fourier transform

In the spirit of the previous WFT method, we proposed a new version of the windowed Fourier transform on surface, that could work without the learning based framework defined for the previous LSCNN method. In order to enrich the content of the obtained analysis we add an orientation sensitive property to the WFT tool. We propose the Anisotropic Windowed Fourier Transform (AWFT), a framework for localized and orientation-sensitive space-frequency analysis *of deformable 3D shapes. With AWFT, we are able to extract meaningful intrinsic localized orientation-sensitive structures on surfaces, and use them in applications such as shape segmentation, salient point detection, feature point description, and matching. We also propose an aggregation strategy to address the problem of the LBO eigenbasis ambiguity, that was solved with a learning strategy in LSCNN. Our method outperforms previous approaches in the considered applications. With respect to the previous LSCNN, AWFT is therefore more flexible in different applicative domains.*

5.3.1 Overview and related work

As we saw in the Chapter 2 and in the previous Section, Fourier analysis is a tool ubiquitously used in a wide range of problems in mathematics and engineering and is the pillar of classical signal processing [97]. In the previous Section we propose to extend the Windowed Fourier Transform (WFT), a standard signal processing tool, on surfaces. In the previous Section, following a recent promising trend, the obtained descriptor is based on the use of machine learning methods to learn optimal task-specific feature descriptors from examples. To do that the convolutional neural networks (CNN) are effectively exploited for 3D meshes and point clouds. The use of machine learning methods to learn optimal task-specific feature descriptors from examples is a growing field of research [15, 89, 90, 131, 151]. In [90] the optimal point descriptor is proposed to learn point-to-point matching. In [89] a supervised version of the bag-of-words approach is introduced for 3D shape retrieval. Finally, in [15] the idea of *anisotropic* diffusion was defined to learn direction-sensitive feature descriptors. In this Section, we propose the Anisotropic Windowed Fourier Transform (AWFT) for localized and orientation-sensitive analysis on manifolds. Our method extends the previous ones of [138] and [14] by adding directional information, resulting in a construction similar in its spirit to the classical Gabor transform. We highlight that in AWFT, differently from the previous Section, we accurately justify the WFT parameters by avoiding to delegate this task to the learning procedure. We show that in many cases the additional information captured by the AWFT is very beneficial. We demonstrate the utility of the proposed method on three common shape analysis tasks: segmentation, salient point detection, and the construction of intrinsic feature descriptors. Our results in these applications compare favorably to the state of the art on standard benchmarks.

5.3.2 Need for orientation-sensitive

As we said just above the AWFT is an *orientation-sensitive* tool for the spectral analysis on manifold. Here we give just an example in order to clarify why the orientation-sensitive property could improve the spectral analysis of signal defined on surface. Lets start with the first row of Figure 5.9. On the left we can see two signals that differ for a rigid rotation in the 2D domain. Applying the rotation the two signal (first row on the right) are exactly the same. Now on the left of the second row we consider the two signals embedded inside a 2-dimensional cat shape. In this case applying the same rigid rotation the two signal continue to be different because of their relation with the cat shape. With AWFT we would like to locally analyze signals that belong to some shape domain. We would be able to recognize two signals that differ in the orientation

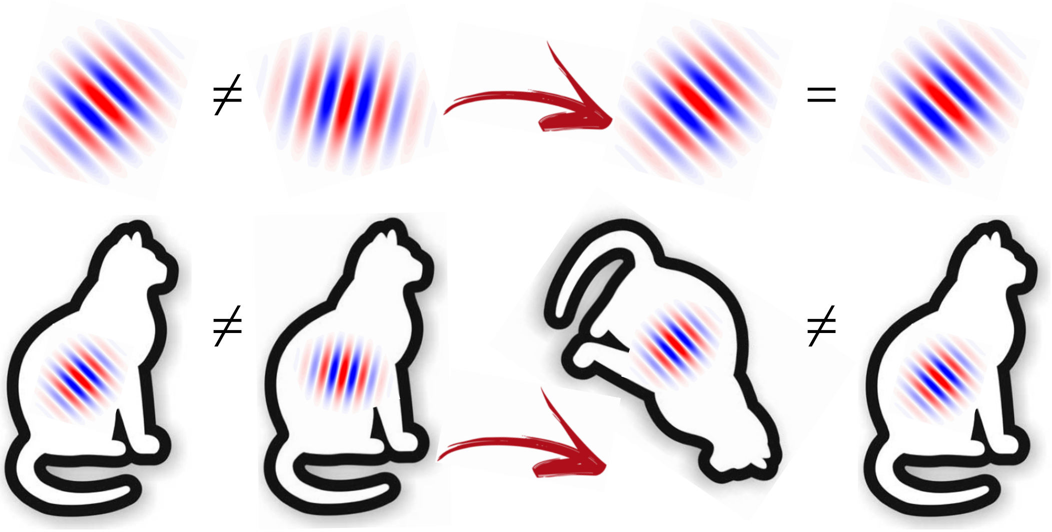


Fig. 5.9: An example of signal for which is important the orientation-sensitive property. First row two signal without any embedding that differ for a rigid rotation in the 2D domain for example. Second row the same example of signals embedded inside a cat shape. The two signals do not differ for a rigid rotation in this case due to the relation with the shape on which the signals belong.

with which they appear on the surface. For this reason the orientation-sensitive property could be very important.

5.3.3 Background

We start from the definition of the *Laplace-Beltrami operator* (or the *Laplacian*) that is introduced in the Chapter 2:

$$\Delta_{\mathcal{M}} f(x) = -\mathrm{div}_{\mathcal{M}}(\nabla_{\mathcal{M}} f(x)). \tag{5.17}$$

Anisotropic Laplacian.

Andreux et al. [3] considered an *anisotropic* Laplace-Beltrami operator of the form

$$\Delta_{\mathcal{M}} f(x) = -\mathrm{div}_{\mathcal{M}}(\mathbf{D}(x)\nabla_{\mathcal{M}} f(x)), \tag{5.18}$$

where $\mathbf{D}(x)$ acts on the intrinsic gradient direction in the tangent space, represented in the orthogonal basis $\mathbf{v}_M(x), \mathbf{v}_m(x)$ of principal curvature

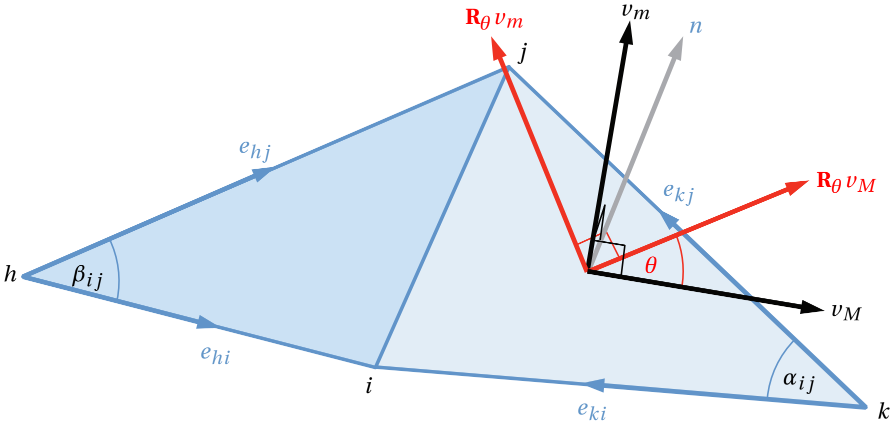


Fig. 5.10: Triangular mesh discretization. The orthogonal basis vectors $\mathbf{v}_M, \mathbf{v}_m$, as well as their rotated counterparts (in red), lie in the plane of the respective triangle (reproduced from [15]).

directions. In particular, the authors considered anisotropy along the maximum curvature direction,

$$\mathbf{D}_\alpha(x) = \begin{bmatrix} \frac{1}{1+\alpha} & \\ & 1 \end{bmatrix}, \tag{5.19}$$

where parameter $\alpha > 0$ controls the level of anisotropy and is defined as the ratio between $\mathbf{v}_M$ and $\mathbf{v}_m$ directions. Boscaini et al. [15] considered anisotropic Laplacians with anisotropy at angle θ w.r.t. the maximum curvature direction,

$$\mathbf{D}_{\alpha\theta}(x) = \mathbf{R}_\theta \mathbf{D}_\alpha(x) \mathbf{R}_\theta^\top, \tag{5.20}$$

where $\mathbf{R}_\theta$ is a rotation by θ in the tangent plane. The resulting Laplacian operator

$$\Delta_{\alpha\theta} f(x) = -\mathrm{div}_{\mathcal{M}}(\mathbf{R}_\theta \mathbf{D}_\alpha(x) \mathbf{R}_\theta^\top \nabla_{\mathcal{M}} f(x)) \tag{5.21}$$

is the centerpiece of the construction proposed in this paper. Note that these operators are not intrinsic, as they depend on the principal curvatures. However, if we consider all the possible angles $\theta \in [0, 2\pi)$, up to the choice of the origin of θ, the result becomes intrinsic.

5.3.4 Proposed method

The main drawback of standard Fourier analysis is that the basis functions are globally supported. As a result, it is practically impossible to localize a small spatial feature in the frequency domain. A common technique in signal processing, referred to as the Windowed Fourier Transform (WFT, also known as Short-Time Fourier Transform or spectrogram) is to *localize* frequency analysis to a window, considering the Fourier coefficients of a function for each window location. The result is a combined space-frequency representation.

WFT on manifolds

Shumann et al. [138] and our previous Section generalized this construction to graphs and manifolds, respectively. The window g is defined in the frequency domain and the WFT is computed as

$$\begin{aligned}(Sf)_{\xi i} &= \langle f, \underbrace{\rho\phi_i \sum_{j\geq 0} \hat{g}_j \phi_j(\xi)\phi_j}_{g_{\xi i}(x)} \rangle_{L^2(\mathcal{M})} \\ &= \rho \sum_{j\geq 0} \hat{g}_j \phi_j(\xi) \langle f, \phi_i \phi_j \rangle_{L^2(\mathcal{M})} ,\end{aligned} \tag{5.22}$$

where $g_{\xi i}(x)$ is the window at position ξ modulated with the ith frequency, referred to as an *atom*. We can think of it as a translated and modulated version of g,

$$g_{\xi i}(x) = \rho M_i T_\xi g(x) = \rho\phi_i(x)(g * \delta_\xi)(x) .$$

$\rho = (\text{area}(\mathcal{M}))^{1/2}$ is a normalization constant ensuring that the modulation with DC is norm-preserving,

$$\rho M_i f(x) = \rho\phi_0(x) f(x) = \rho\frac{1}{\rho} f(x) = f(x).$$

Note that $(Sf)_{\xi i}$ has two indices: spatial location ξ and frequency i.

AWFT

While allowing for a localized frequency analysis, the WFT atoms are agnostic to directional information (see Figure 5.11, left). Such information can be introduced using the anisotropic Laplacian. Let

$$\Delta_{\alpha\theta}\phi_{\alpha\theta,i}(x) = \lambda_{\alpha\theta,i}\phi_{\alpha\theta,i}(x)$$

be the eigendecomposition of the anisotropic Laplacian with orthogonal eigenfunctions $\phi_{\alpha\theta,0}, \phi_{\alpha\theta,1}, \ldots$ and corresponding non-negative eigenvalues $0 = \lambda_{\alpha\theta,0} \leq \lambda_{\alpha\theta,1} \leq \ldots$. We define the *anisotropic WFT* as

$$(Sf)_{\xi\alpha\theta i} = \sum_{j\geq 0} \hat{g}_{\alpha\theta,j}\phi_{\alpha\theta,j}(\xi)\langle f, \phi_{\alpha\theta,i}\phi_{\alpha\theta,j}\rangle_{L^2(\mathscr{M})}. \tag{5.23}$$

The atoms $g_{\xi\theta i}(x)$ (Figure 5.11, right) are direction-aware. One of the important consequences of direction-awareness is the fact that AWFT is capable of disambiguating intrinsic reflection symmetries, as will be discussed in the following.

Choice of the window.

The choice of the window allows for a tradeoff between spatial and frequency localization (by virtue of the uncertainty principle, it is impossible to achieve a perfect localization in both): a narrow window in the frequency domain (rapidly decaying Fourier coefficients $\hat{g}_i$) results in a wide window in the spatial domain, and vice versa. We use a decaying window

$$\hat{g}_{\alpha\theta,i} = e^{-\tau(\lambda_{\alpha\theta,i} - \lambda_{\alpha\theta,1})}, \tag{5.24}$$

where the parameter τ controls the decay rate (larger values of τ produce windows with poorer spatial localization). In order to make this parameter scale-invariant, we recall that scaling the coordinates of the shape uniformly by a factor β scales its area by β^2 and Laplacian eigenvalues by a factor of β^{-2}. We therefore multiply τ by area($\mathscr{M}$).

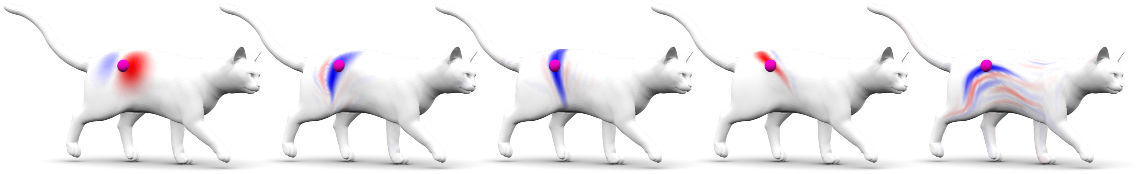


Fig. 5.11: Some atoms of the AWFT on the point marked with small pink spheres on the cat shape. These are obtained with $\tau = 0.002$, the first on the left is the isotropic window while the others are obtained with $\alpha = 300$ and different θ, from left to right 45, 90, 135, 180.

Total Weighted Power

Note that the definition of the AWFT coefficients depending on the basis functions $\{\phi_k^{\alpha\theta}\}_{k\geq 1}$, they suffer not only the changes of sign of the eigenfunctions but also the flips in the order due to the instability of the eigenfunctions of the ALBO (or LBO). Rabiei et al. [124] used the *total weighted power* (TWP) as an aggregate of all frequency information weighted by the normalized corresponding eigenvalue,

$$(S_{\text{TWP}}f)_{\xi\alpha\theta} = \sum_{k\geq 0} \frac{\lambda^2_{\alpha\theta,k}}{\|\Lambda_{\alpha\theta}\|_2^2}(Sf)^2_{\xi\alpha\theta k}, \tag{5.25}$$

where $\|\Lambda\|_2^2 = \sum_{k\geq 0}\lambda^2_{\alpha\theta,k}$ is the norm of the set of eigenvalues. Such an aggregation allows removing the potential ambiguities due to different signs and ordering of the eigenfunctions. Attributing greater value to high frequencies the *TWP* can be seen as a filter that emphasizes localized (high-frequency) properties.

5.3.5 Discretization

In the discrete setting, the surface $\mathscr{M}$ is represented by a triangular mesh, and functions, inner products, LBO as introduced in Chapter 2. $\mathbf{A} = \text{diag}(a_1, \ldots, a_N)$ is the *mass matrix*, and a_i denotes the local area element at vertex i. To each triangle ijk of the mesh, we attach an orthonormal reference frame $\mathbf{U}_{ijk} = [\mathbf{v}_M, \mathbf{v}_m, \hat{\mathbf{n}}] \in \mathbb{R}^{3\times 3}$, where $\hat{\mathbf{n}}$ is the unit normal vector to the triangle and $\mathbf{v}_m$ and $\mathbf{v}_M$ are the directions of principal curvature, computed using the method of [34]. The tensor $\mathbf{D}_\alpha$ for the triangle ijk operating on tangent vectors is expressed w.r.t. $\mathbf{U}_{ijk}$ as:

$$\mathbf{D}_\alpha = \begin{bmatrix} \frac{1}{1+\alpha} & & \\ & 1 & \\ & & 1 \end{bmatrix}.$$

In the case $\theta = 0$, let $\mathbf{e}_{ij} \in \mathbb{R}^3$ denote the oriented edge pointing from vertex i to vertex j, normalized to unit length, and consider the triangle ijk. We define the $\mathbf{H}$-weighted inner product between edges $\mathbf{e}_{kj}$ and $\mathbf{e}_{ki}$ as

$$\left\langle \mathbf{e}_{kj}, \mathbf{e}_{ki} \right\rangle_{\mathbf{H}} = \mathbf{e}_{kj}^T \underbrace{\mathbf{U}_{ijk} \mathbf{D}_\alpha \mathbf{U}_{ijk}^T}_{\mathbf{H}} \mathbf{e}_{ki}, \tag{5.26}$$

where the *shear matrix* $\mathbf{H}$ encodes the anisotropic scaling up to an orthogonal basis change. Note that in the isotropic case ($\alpha = 0$) we have $\mathbf{H} = \mathbf{I}$, such that the $\mathbf{H}$-weighted inner product simplifies to the standard inner product $\langle e_{kj}, e_{ki} \rangle_{\mathbf{H}} = \cos\alpha_{ij}$.

The discretization of the anisotropic Laplacian takes the form of an $n \times n$ sparse matrix $\Delta_{\alpha\theta} = -\mathbf{A}^{-1}\mathbf{W}$. The *stiffness matrix* $\mathbf{W}$ is composed of weights

$$w_{ij} = \begin{cases} -\frac{1}{2}\left(\frac{\langle \mathbf{e}_{kj}, \mathbf{e}_{ki} \rangle_{\mathbf{H}}}{\sin\alpha_{ij}} + \frac{\langle \mathbf{e}_{hj}, \mathbf{e}_{hi} \rangle_{\mathbf{H}}}{\sin\beta_{ij}} \right) & (i,j) \in E \\ -\sum_{k \neq i} w_{ik} & i = j \\ 0 & \text{else} \end{cases} \tag{5.27}$$

where the notation is according to Figure 5.10. In the isotropic case, $\frac{\langle \mathbf{e}_{kj}, \mathbf{e}_{ki} \rangle_{\mathbf{H}}}{\sin\alpha_{ij}} = \frac{\cos\alpha_{ij}}{\sin\alpha_{ij}} = \cot\alpha_{ij}$, thus reducing equation (5.27) to the classical cotangent formula [120] as we saw in Chapter 2. To obtain the general case $\theta \neq 0$, it is sufficient to rotate the basis vectors $\mathbf{U}_{ijk}$ on each triangle around the respective normal n by the angle θ, equal for all triangles (see Figure 5.10, red). Denoting by $\mathbf{R}_\theta$ the corresponding 3×3 rotation matrix, this is equivalent to modifying the $\mathbf{H}$-weighted inner product with the directed shear matrix $\mathbf{H}_\theta = \mathbf{R}_\theta \mathbf{H} \mathbf{R}_\theta^T$. The resulting weights w_{ij} in equation (5.27) are thus obtained by using the inner products $\langle \mathbf{e}_{kj}, \mathbf{e}_{ki} \rangle_{\mathbf{H}_\theta} = \mathbf{e}_{kj}^T \mathbf{H}_\theta \mathbf{e}_{ki}$. The computation of the Laplacian eigenvectors is posed as a generalized eigenproblem

$$\mathbf{W}\Phi = \mathbf{A}\Phi\Lambda\,,$$

where $\Phi = (\phi_1, \ldots, \phi_K)$ is an $N \times K$ matrix containing the first K eigenvectors, and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_K)$ is a diagonal matrix containing the corresponding eigenvalues. For the anisotropic Laplacian, we compute a set of K eigenvectors $\Phi_{\alpha\theta}$ and eigenvalues $\Lambda_{\alpha\theta}$ for each θ and α. The AWFT is computed as

$$(S\mathbf{f})_{\alpha\theta} = (\mathbf{f} \boxdot \Phi_{\alpha\theta})^\top \mathbf{A} \Phi_{\alpha\theta} (\hat{\mathbf{g}}_{\alpha\theta} \boxdot \Phi_{\alpha\theta}^\top),$$

where the result is an $N \times K$-dimensional matrix for θ and α; $\mathbf{f}$ is the input function represented as an N-dimensional vector. $(\mathbf{a} \boxdot \mathbf{B})_{ij} = a_i b_{ij}$ denotes a $K \times N$ matrix obtained by element-wise multiplication of a K-dimensional column vector replicated N times along the second dimension with a $K \times N$ matrix as in the previous Section.

5.3.6 Applications and results

Finally we show the application of AWFT to three standard problems in geometry processing: *shape segmentation*, *salient point detection* and design of *local feature descriptors* for point to point matching. In each application, one has the freedom to define the four main ingredients of AWFT. First, the function f to be analyzed, which represents the information we wish to encode. Second, the size of the window, determined by the parameter τ. We typically use a set of different values $\tau_1,\ldots,\tau_t$ to perform a multi-scale analysis. Third, the window orientations $\theta_1 = 0, \theta_2 = \frac{\pi}{h}, \ldots, \theta_h = \frac{(h-1)\pi}{h}$. Fourth, the anisotropy parameter α. We typically use a set of different values $\alpha_1,\ldots,\alpha_c$ to capture oriented structures of different width.

Computation

We used up to $K = 200$ Laplacian eigenvectors and eigenvalues computed using MATLAB *eigs* function. The computation of AWFT with the settings used in our experiments takes on average less than 5 seconds on a mesh with around 7000 vertices on a machine with 32GB of RAM and an Intel 3,6 GHz Core i7 cpu.

Shape Segmentation

Applying the AWFT to the constant function $f(x) = 1$ gives rise to a reweighted version of the GPS descriptor [14,133] that is direction-aware. We apply on the squared AWFT features $(Sf)^2_{\xi\alpha\theta i}$ the segmentation method of Rodolà et al. [132] (which takes the classical GPS as input) using the settings proposed by the authors.

Settings. We use $f = 1$, single window with $\tau = 0.002$, two levels of anisotropy $\alpha_1 = 100, \alpha_2 = 300$, $\mathscr{F} = f$ the constant function, and orientations $\theta_1 = 0, \theta_2 = \frac{\pi}{4}, \ldots, \theta_4 = \frac{3\pi}{4}$. We use only the first $K = 50$ frequencies.

Data and Evaluation. We evaluate the segmentation results according to the Princeton Segmentation Benchmark [31], consisting of 380 meshes from 19 object classes (20 shapes per class). A common meaningful segmentation is given as ground truth for each category. The ground truth assigns the same label to semantically similar segments (the two arms of a man).

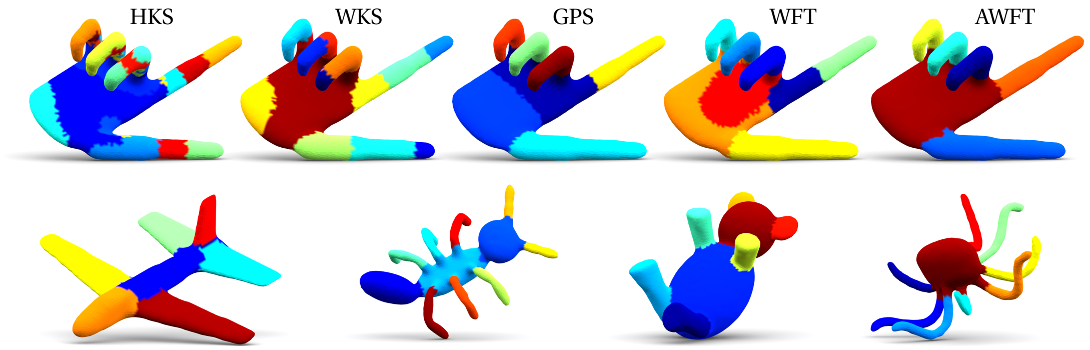


Fig. 5.12: First row: segmentation of the hand shape obtained using (left to right) HKS, WKS, GPS, WFT and AWFT. Second row: a few segmentation examples obtained with AWFT.

We use two different evaluation criteria, *Fidelity* and *Goodness*. For every segment in the ground-truth, Fidelity is the average ratio between the number of the maximum subset of points that is segmented together by the method and the number of points that compose that ground-truth segment. Goodness is in some sense the inverse. For every segment obtained by the method, it is defined as the average ratio between the maximum number of points that are segmented together in the ground truth and the number of points that compose that segment provided by the method.

Results Table 5.1 summarizes the segmentation results in terms of Goodness and Fidelity. For comparison, we show segmentation results obtained with HKS, WKS, GPS and isotropic WFT features. Our AWFT produces the best score on average. Segmentation examples in Figure 5.12 show that AWFT is able to perform a segmentation that is both semantically meaningful and geometrically consistent.

Salient Point Detection

We use the AWFT to construct a saliency map that allows to detect key points on surfaces. We use the logarithm of the mean curvature as an

	HKS	WKS	GPS	WFT	AWFT
airplane	0.41 / 0.82	0.38 / 0.84	0.47 / 0.80	0.42 / 0.85	**0.48** / **0.91**
ant	0.57 / **0.97**	0.61 / 0.90	**0.64** / 0.84	0.63 / 0.90	0.62 / 0.90
bird	0.44 / 0.84	0.45 / **0.89**	**0.52** / 0.86	0.44 / 0.86	**0.52** / **0.89**
fish	**0.62** / **0.95**	0.43 / 0.92	0.43 / 0.86	0.37 / 0.86	0.39 / 0.91
hand	0.52 / 0.91	0.55 / 0.92	0.61 / 0.88	0.61 / 0.88	**0.75** / **0.94**
octopus	0.35 / **0.98**	0.35 / 0.95	**0.44** / 0.87	0.32 / 0.90	0.40 / 0.95
plier	0.35 / 0.90	0.36 / **0.92**	0.43 / 0.83	0.39 / 0.88	**0.51** / 0.91
teddy	0.48 / 0.90	0.51 / 0.88	0.63 / 0.75	0.53 / 0.83	**0.68** / **0.93**
mean	0.47 / 0.91	0.46 / 0.90	0.52 / 0.84	0.46 / 0.87	**0.54** / **0.92**

Table 5.1: Performance in *Fidelity* / *Goodness* on 8 categories of the Princeton Segmentation Benchmark. Best performance is highlighted in bold.

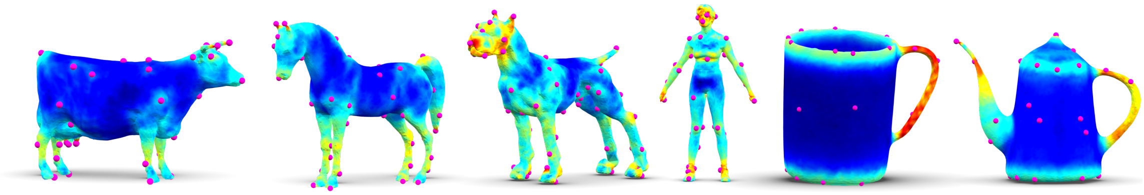


Fig. 5.13: Saliency maps computed with AWFT on some shapes from the evaluation benchmark. The areas considered as more salient are in red while the less important ones tend to blue. The final selected points are highlighted by small pink spheres.

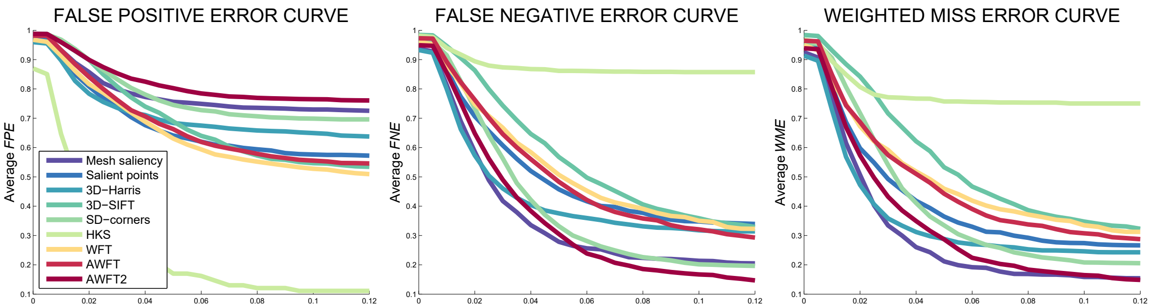


Fig. 5.14: The False Negative, False Positive and Weighted Miss error curves for the proposed methods (in warm colors) and the methods presented in the used benchmark (in cold colors).

input function (capturing large curvature variations) and compute the TWP of the AWFT. This way, we obtain a set of saliency maps for each value of α, θ and τ. For every such saliency map, we compute local maxima as follows: For each point ξ we take the corresponding window $T_\xi g_{\alpha\theta}$ (computed using the same τ used for the map) and locate its maximum. We then perform a non-maximum suppression. Each map is then normalized as proposed in [69]. A single saliency map is obtained by summing up the maps for different α, θ and τ. Finally, the maximum detection and non-maximum suppression is performed again, producing the salient points.

Settings. We use $f = \log(H)$, various window sizes $\tau = 0.0002, 0.0007, 0.001, 0.0015, 0.0055$, a single level of anisotropy $\alpha = 300$, and angles $\theta_1 = 0, \theta_2 = \frac{\pi}{12}, \ldots, \theta_{12} = \frac{11\pi}{12}$.

Data and Evaluation. We follow the test proposed in [43] on two different datasets. Dataset A consists of 24 objects hand-marked by 23 human subjects. Dataset B contains 43 models marked by at least 16 subjects. The human annotations are used as the ground-truth for both datasets. The adopted evaluation criterion (WME) is based on importance of the selected points, where importance is based on these hand-marked selections.

Results. Figure 5.13 shows a few saliency maps and the corresponding selected salient points. Note how the saliency maps identify semantically coherent parts among the animal and cup shape classes. Figure 5.14 evaluates the performance of various feature detection methods. We use two non-maximum suppression settings (denoted AWFT1 and AWFT2) to tradeoff between false positives and negatives. Table 5.2 shows the average area-under-the-curve (AUC) for different methods.

Mean	Mesh saliency	Salient points	3D-Harris	3D-SIFT	SD-corners	WFT	AWFT	AWFT2
0.59	0.57	0.59	0.57	0.63	0.59	0.59	0.59	0.59

Table 5.2: The mean AUC computed on the False Negative and the False Positive error curves for all the comparable methods tested in these curves (the proposed ones and all the others presented in the benchmark, not HKS that is not compatible with other methods).

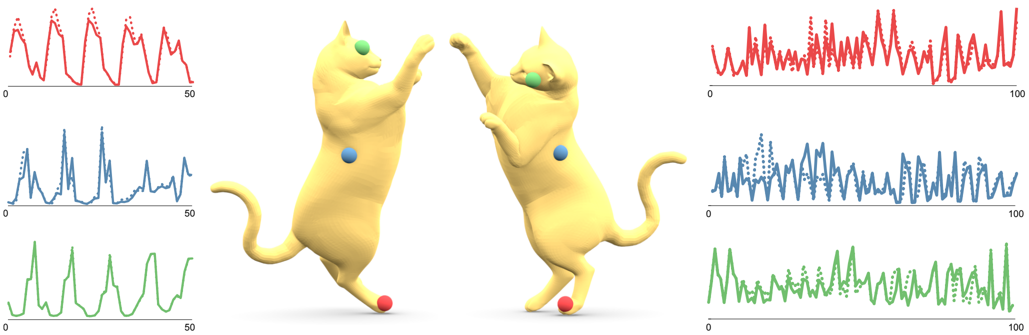


Fig. 5.15: Comparison between WFT (left) and AWFT (right) descriptor computed at three symmetric points of the cat shape (points and corresponding curves are color-coded). Solid and dotted curves represent descriptors of points from the left and right side of the cat, respectively.

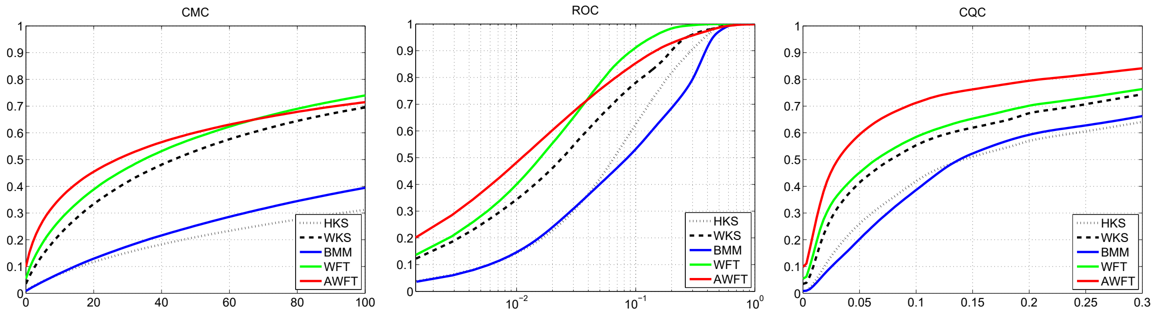


Fig. 5.16: Performance evaluation on FAUST dataset.

Descriptors for Point to Point Matching

In order to obtain a concise and informative descriptor for every point on the surface, we use the TWP to reduce the dimension of the output of

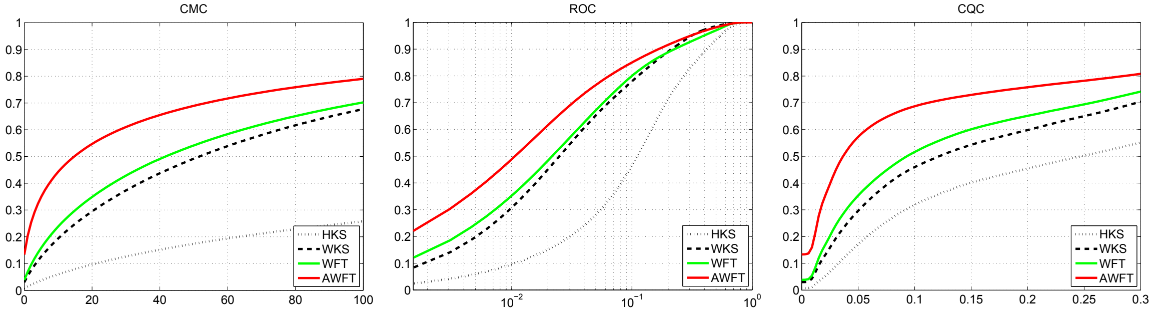


Fig. 5.17: Performance evaluation on CAESAR dataset.

AWFT on some geometrically meaningful functions. One of the key deficencies of standard descriptors such as HKS, WKS or WFT is their invariance to intrinsic symmetries. This is visualized in Figure 5.15, where we show a bilaterally intrinsically symmetric shape (cat). The WFT at symmetric points (leftmost plot in solid and dotted) are nearly identical. On the other hand, orientation is not preserved by the intrinsic symmetry (in fact, if $\sigma : \mathcal{M} \to \mathcal{M}$ is a bilateral symmetry, then $(Sf)_{\xi\alpha\theta i} = (Sf)_{\sigma(\xi)\alpha,-\theta,i})$. As a result, the AWFT descriptor can distinguish intrinsically symmetric points, as we experimentally show in the following.

Settings. We used five input features: f_1, f_2 are the third and fourth *geometry vectors* [90] encoding some of the spectral geometry of the shape; $f_3 = 1$; $f_4 = \phi_1$ is the (isotropic) Fiedler vector providing a consistent ordering on the mesh vertices [87]; and f_5 is the ShapeIndex [78] encoding the curvatures of the surface. We used $\tau = 0.002, 0.05$, $\alpha_1 = 100, \alpha_2 = 300$, and $\theta_1 = 0, \theta_2 = \frac{\pi}{4}, \ldots, \theta_4 = \frac{3\pi}{4}$.

Data and Evaluation. We used two public-domain datasets of scanned human shapes in different poses: FAUST [13] and CAESAR [126]. We select a random set of 60 shapes from the *fitted-meshes* subset of CAESAR, where each shape has ~6K vertices. Ground truth point-wise correspondence is available in both datasets.

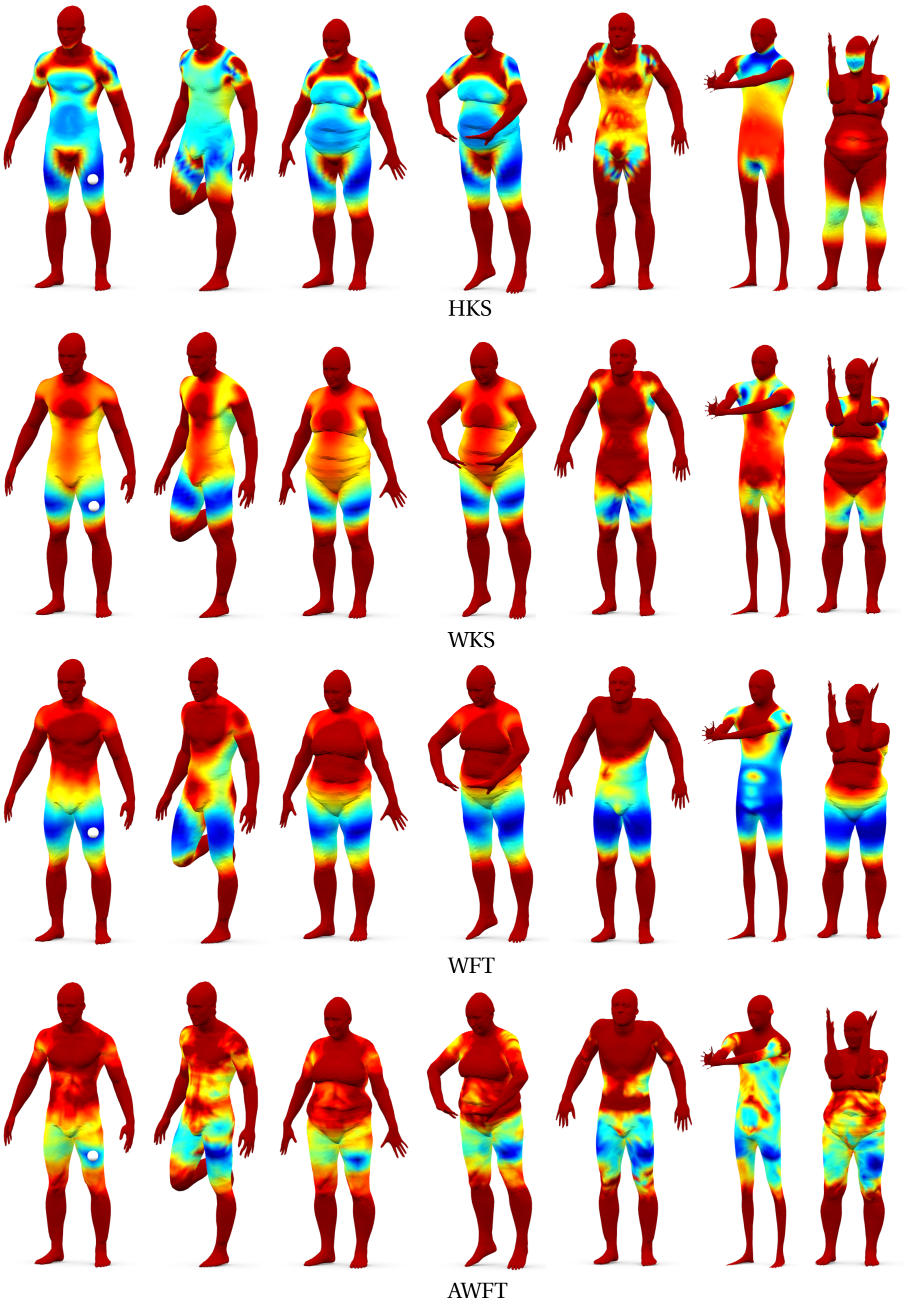
We evaluate descriptor performance using *cumulative match characteristic* (CMC), *receiver operator characteristic* (ROC) and *correspondence quality characteristic* (CQC), already used and defined in the previous experiments on LSCNN.

Results. The evaluation results are summarized in Figures 5.16–5.17, which clearly demonstrate that AWFT outperforms the classical methods. In Figure 5.16, we also compare to the descriptor proposed in the previous version (and in [14]) without the learning step (referred to as BMM). We use the first 20 coefficients of the WFT computed on 5 geometry vectors as in [14], with the same two τ used in AWFT. Despite the double dimensionality BMM does not offer comparable performance without the class-specific learning step. In Figure 5.18, we show the distance between the descriptor at a point (indicated with the white sphere on the first shape on the left) and the rest of the points on the same shape as well as other shapes. AWFT descriptors are the most localized and discriminative, and correctly disambiguate symmetries. We additionally tested our descriptor on the TOSCA dataset, comprising 7 different shape classes with near-isometric deformations within each class. As we show in Figure 5.19, AWFT clearly outperforms the classical methods also on non-human shapes (e.g., cats). These results show that the proposed descriptor is not class-specific and achieves the best performances on different shapes. In a second experiment, we apply the WFT and AWFT on 16-dimensional HKS and WKS descriptors. This was done to show how already good descriptors can be enriched by the proposed tools. The results are shown in Figures 5.20, 5.21, 5.22.

5.3.7 Conclusion and future work

In this Section, we proposed an anisotropic windowed Fourier transform on manifolds. The proposed tool enables to perform local directional frequency analysis and improve the study of geometry in several applications in shape analysis. Its power and versatility were tested on classical datasets for different tasks as shape segmentation, salient points detection and point to point matching. In future work, we will study additional applications of AWFT, as well as its computation on other types of data such as point clouds.

WFT and AWFT methods proposed in the previous two Sections allow us to move from a global spectral analysis to a local spectral analysis on surfaces. From the theoretical point of view the proposed tools solve the uncertainty principle generated by the global support of the Fourier basis functions. The lack of localization in the standard Fourier basis is solved in these methods using a window to localize the signal before the analysis. What we obtain is a localized signal that could be analyzed in its spectral representation. These approaches reach good localizations simultaneously in the space domain and in the frequency domain, and they are promising in different applications.



HKS

WKS

WFT

AWFT

Fig. 5.18: Normalized Euclidean distance between the descriptor at a reference point on the leg (white sphere) and the descriptors computed at the rest of the points for different transformations (from left to right: near isometric deformations, non-isometric deformations, subsampling, smoothing and remeshing). Cold and hot colors represent small and large distances, respectively. For visualization clarity, distances are saturated at 30% of the maximum.

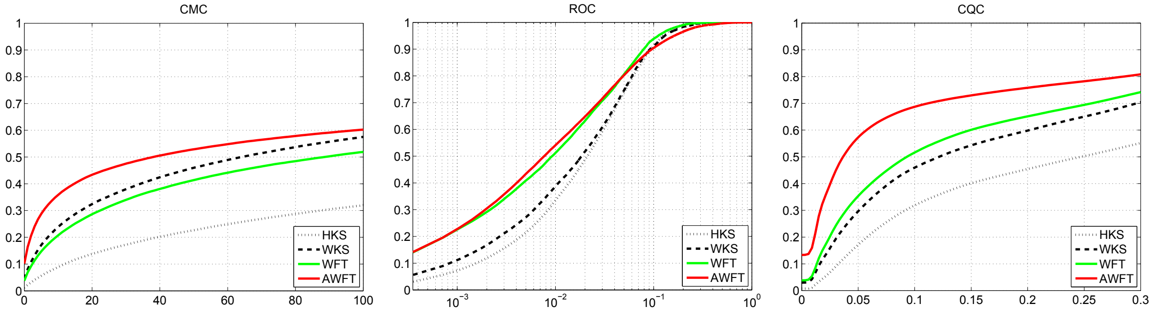


Fig. 5.19: Performance evaluation on the CAT class from TOSCA dataset (11 meshes).

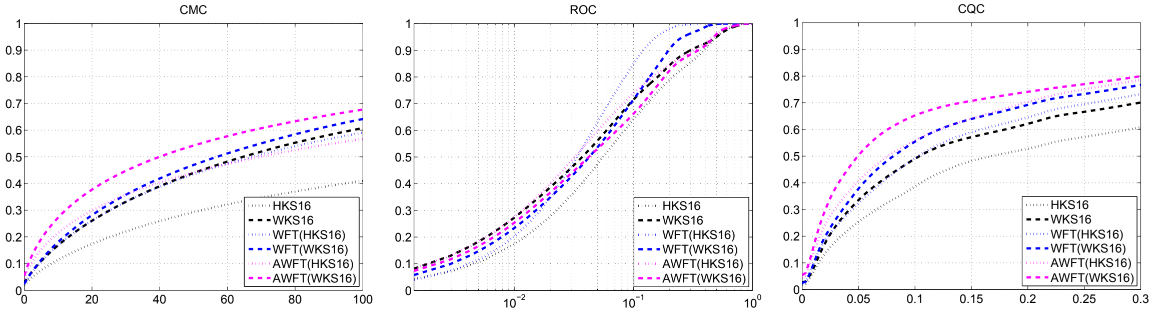


Fig. 5.20: Performance evaluation on FAUST dataset (100 meshes).

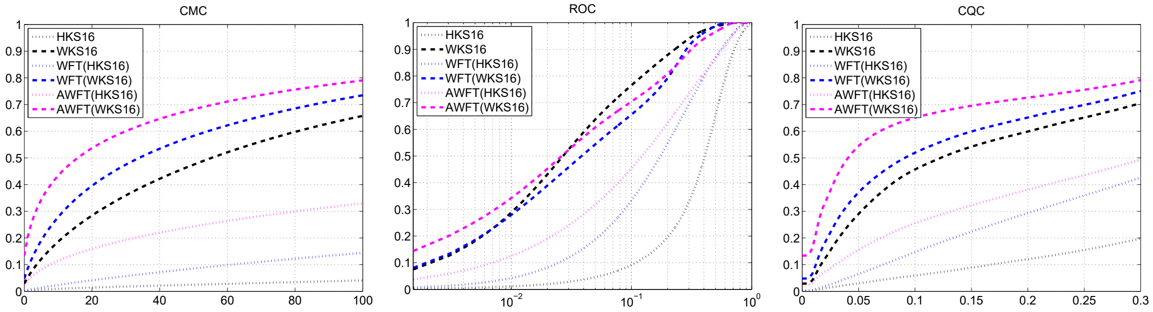


Fig. 5.21: Performance evaluation on CAESAR dataset (60 meshes).

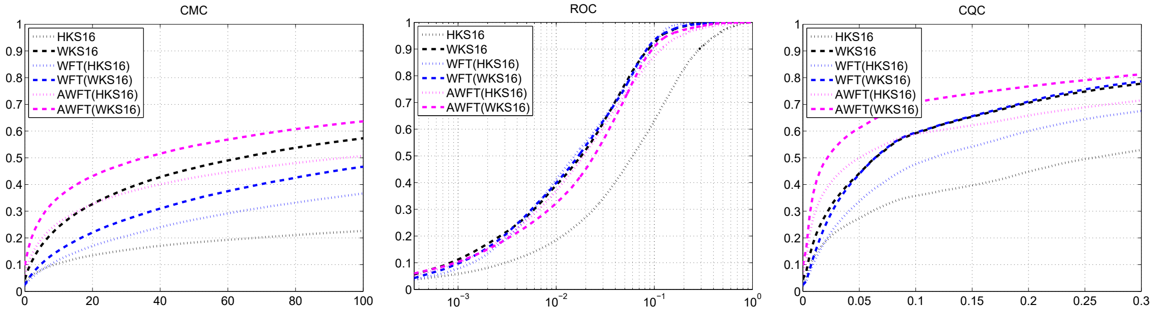


Fig. 5.22: Performance evaluation on the CAT class from TOSCA dataset (11 meshes).

5.4 LMH: a localized bases for spectral geometry processing

In the previous Sections of this Chapter we introduce two different versions of the WFT defined on manifolds. With these methods we face one of the main drawback of the LBO eigenfunctions (or MH) and the Fourier analysis built on this basis that is the lack of spatial localization. Despite the LBO itself is a local (differential) operator, its eigenfunctions and eigenvalues carry geometric and topological information about the entire shape. As a practical consequence, operations that should be local by design are often interested by non-local effects due to the global awareness of these quantities. In other words a limiting aspect of classical spectral analysis on manifolds lies in its inherently "global" nature. To face this problem in WFT and AWFT we localized the signal analyzed by conveying the signal with a window. In both these cases we focus the localization problem on the signals.

An alternative approach could be to consider a basis with different local properties with respect to the MH. In this spirit we propose the Localized Manifold Harmonics *namely LMH, a new localized basis. With LMH we introduce a new framework for local spectral shape analysis, which can solve the lack of spatial localization. In the Following Section we show how to efficiently construct a localized orthogonal bases by solving an optimization problem that can be posed as the eigendecomposition of a new operator obtained by a modification of the standard Laplacian. We study the theoretical and computational aspects of the proposed framework and showcase our new construction on the classical problems of shape approximation and correspondence. We obtain significant improvement compared to classical Laplacian eigenbases as well as other alternatives for constructing localized bases. For more details on the LMH, please refer to [101].*

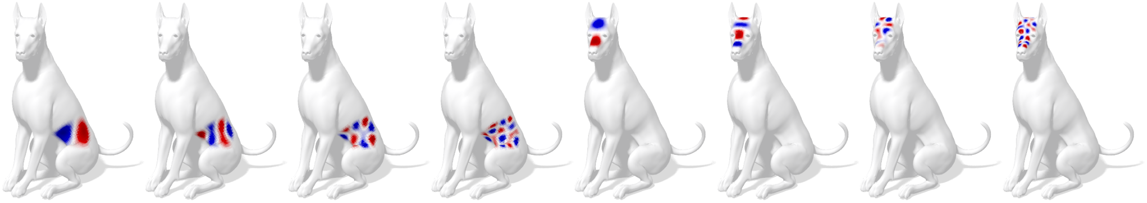


Fig. 5.23: A few Localized Manifold Harmonics (LMH) on two different regions of the dog shape. By changing the region location on the surface, our model provides an ordered set of *localized* harmonic functions (i.e., defined on the entire surface, but strongly concentrated on the selected region). In this figure the localized harmonics are clearly visible across different frequencies. The LMH constitute a valid alternative to the classical manifold harmonics and can be used in conjunction with those, or as a drop-in replacement in typical spectral shape analysis tasks.

5.4.1 Limits of the standard basis

As we already seen spectral methods are ubiquitously used in 3D shape analysis and geometry processing communities for a wide range of applications. The centerpiece of such methods is the construction of an orthogonal basis for the space of functions defined on a manifold, allowing to generalize classical Fourier analysis to non-Euclidean domains. Typically, such bases are constructed by the diagonalization of the Laplace-Beltrami operator [86]. The choice of Laplacian eigenbasis is convenient for several reasons. First, it is intrinsic and thus invariant to manifold parametrization and its isometric deformations [87]. Second, it allows to be agnostic to a specific shape representation, as the Laplace-Beltrami operator can be discretized on meshes, point clouds, volumes, etc. Third, Laplacian eigenbasis turns to be optimal for approximating functions with bounded variation [1] and in many applications only the first few eigenfunctions are sufficient to achieve a good approximation. Finally, in the discrete setting, the computation of the Laplacian eigenbasis has relatively low complexity due to the sparse structure of the Laplacian matrix.

One of the key disadvantages of the Laplacian eigenbases is their *global* support. Thus, representing local structures requires using (potentially, infinitely) many basis functions. In many applications, one wishes to have a local basis that allows to limit the analysis to specific parts of the shape. The recently proposed *compressed manifold harmonics* [20, 81, 110, 118] attempt to construct local orthogonal bases that approximately diagonalize the Laplace-Beltrami operator. The main disadvantage of this framework is the inability to explicitly control the localization of the basis functions. Moreover, the basis is computed by solving an optimization problem on the Stiefel manifold of orthogonal matrices which does not guarantee a global solution. We proposed a new type of intrinsic operators whose spectral decomposition provides a local basis the *Localized Manifold Harmonics* (LMH). Similarly to related constructions like [33], the new basis is smooth, local, and orthogonal; it is localized at specified regions of the shape, explicitly controllable; and it is efficiently computed by solving a standard eigendecomposition, thus coming with global optimality guarantees. The key novelty of our approach comes from its capability to integrate the *global* information obtained by the Laplacian eigenfunctions with *local* details given by our new basis. To this end, the localized basis is constructed in an incremental way, such that the new functions are orthogonal to some given set of functions (e.g., standard Laplacian eigenfunctions). Due to the aforementioned properties, we name our new basis *Localized Manifold Harmonics (LMH).* At the end the set of basis functions that we obtain has the following properties:

- smooth, local, and orthogonal.
- localized at specified regions of the shape, explicitly controllable;
- efficiently computed by solving a standard eigendecomposition, thus coming with global optimality guarantees.

The global structure of the Laplacian eigenbasis has adverse effects in numerous applications. In spectral shape deformation, it is hard to concentrate the analysis on local parts of the shape. In shape correspondence, the dependence on the Laplacian eigenfunctions on the global structure of the shape makes it hard to cope with topological noise and missing parts. In the methods proposed in the previous Sections local analysis is introduced on non-Euclidean manifolds using the Windowed Fourier Transform (WFT) that has been proposed for graphs [138] and by our works on shapes [14, 104]. Although these methods improved the encoding of local parts, we still adopted Laplacian eigenfunctions as a basis to compute the spectral components, and therefore it was still hard to perform well in the challenging scenarios mentioned above.

As a possible remedy, Ozoliņš et al. [118] introduced *compressed modes*, a construction of local orthogonal bases that approximately diagonalize the Laplacian. The key idea of this method is the addition of a sparsity-promoting L_1-norm to the Dirichlet energy (the combined effect of s-moothness and sparsity results in localization of the basis functions). Rustamov [134] previously used a similar regularization to construct local biharmonic kernels for function interpolation. Neumann et al. [110] applied the approach of [118] to problems in computer graphics. Kovnatsky et al. [81] showed an efficient way of computing compressed manifold modes, while Bronstein et al. [20] proposed a more theoretically sound approach for the computation of L_1-norm on manifolds.

Closely related to our method is the recent approach of Choukroun et al. [33], who considered the spectral decomposition of an elliptic operator realized as a diagonal update to the standard Laplacian. Differently from [33], our solutions are simultaneously *localized* and *orthogonal* to the globally-supported Laplacian eigenbasis, leading to important practical consequences in several applications.

The key idea of the LMH is the construction of localized bases by spectral decomposition of a modified Laplacian operator, crafted especially to provide eigenfunctions with local support. Our new operator inherits the important properties of the original Laplacian such as isometry invariance. In particular, it has a clear Fourier-like meaning that makes its use well interpretable. Differently from [20, 81, 110, 134] which impose locality through an L_1 constraint, we allow an explicit indication of the local support of each function. This improves the versatility in controlling the local analysis, especially for semantically-guided interventions. Another important difference from other methods is that our new basis is computed by solving a standard eigendecomposition problem avoiding the need for more complex optimization methods.

5.4.2 Localized manifold harmonics

With LMH we introduce a new framework for spectral shape analysis that is designed to be at the same time *local* and *compatible* with the existing spectral constructions. In practice, our approach boils down to the computation of the eigenfunctions of a new operator, which is realized as a simple update to the classical manifold Laplacian – thus fully retaining the computational efficiency and theoretical guarantees of the resulting optimization process.

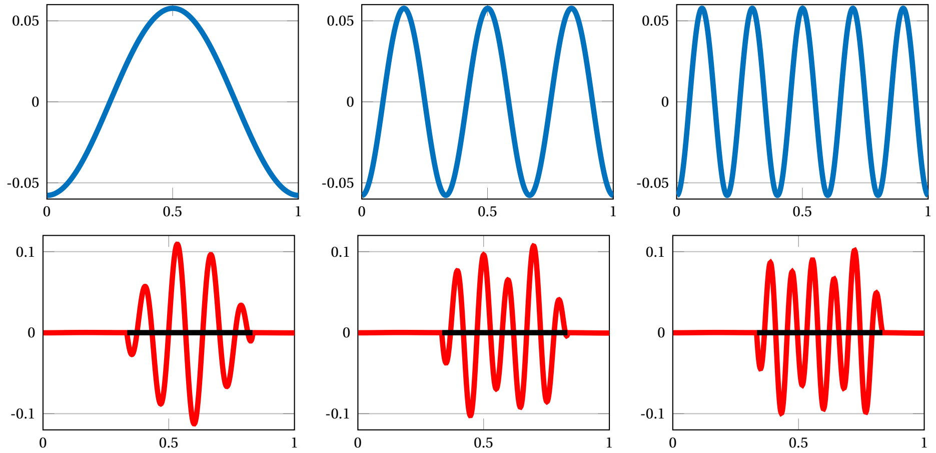


Fig. 5.24: Classical (top row) and localized (bottom row) harmonics in 1D under Neumann boundary conditions. Note that the localized harmonics are orthogonal to those in the first row. The selected region $R \subset [0,1]$ is marked as a black segment.

Definition

Let us be given a manifold $\mathcal{M}$, a region $R \subseteq \mathcal{M}$ thereof, a set of orthonormal functions $\phi_1, \ldots, \phi_{k'}$ (e.g. the first k' Laplacian eigenfunctions), and an integer k. We seek a new set $\psi_1, \ldots, \psi_k$ of functions that are *smooth, orthonormal,* and *localized* on R, as the solution to the following optimization problem:

$$\min_{\psi_1,\ldots,\psi_k} \sum_{j=1}^{k} \mathscr{E}_S(\psi_j) + \mu_R \mathscr{E}_R(\psi_j) \tag{5.28}$$

$$\text{s.t. } \langle \psi_i, \psi_j \rangle_{L^2(\mathcal{M})} = \delta_{ij} \quad i, j = 1, \ldots, k \tag{5.29}$$

$$\langle \psi_i, \phi_j \rangle_{L^2(\mathcal{M})} = 0 \quad i = 1, \ldots, k;\ j = 1 \ldots, k' \tag{5.30}$$

where the constraints (5.30) demand the basis functions to be *orthogonal* to the subspace span$\{\phi_1, \ldots, \phi_{k'}\}$. As we will see in what follows, it allows constructing an incremental set of functions that are orthogonal to a given set of standard Laplacian eigenfunctions.

The first term $\mathscr{E}_S$ is the Dirichlet functional (2.13) promoting the *smoothness* of the new basis. The term

$$\mathscr{E}_R(f) := \int_{\mathcal{M}} (f(x)(1 - u(x)))^2 \, \mathrm{d}x, \tag{5.31}$$

is a quadratic penalty promoting the localization of the basis functions on the given region $R \subseteq \mathcal{M}$. Here $u : \mathcal{M} \to [0,1]$ is a membership function such that $u(x) = 1$ for $x \in R$ and $u(x) = 0$ otherwise. Note that we let function u assume a continuum of values in $[0,1]$, implementing the notion of "soft" membership (the choice between binary and soft u is application-dependent).

We refer to the solutions of problem (5.28) as *localized manifold harmonics (LMH).* Figure 5.24 provides an illustration of LMH in the $[0,1]$ interval, while Figures 5.23 and 5.25 depict a few examples of such bases on 2D manifolds.

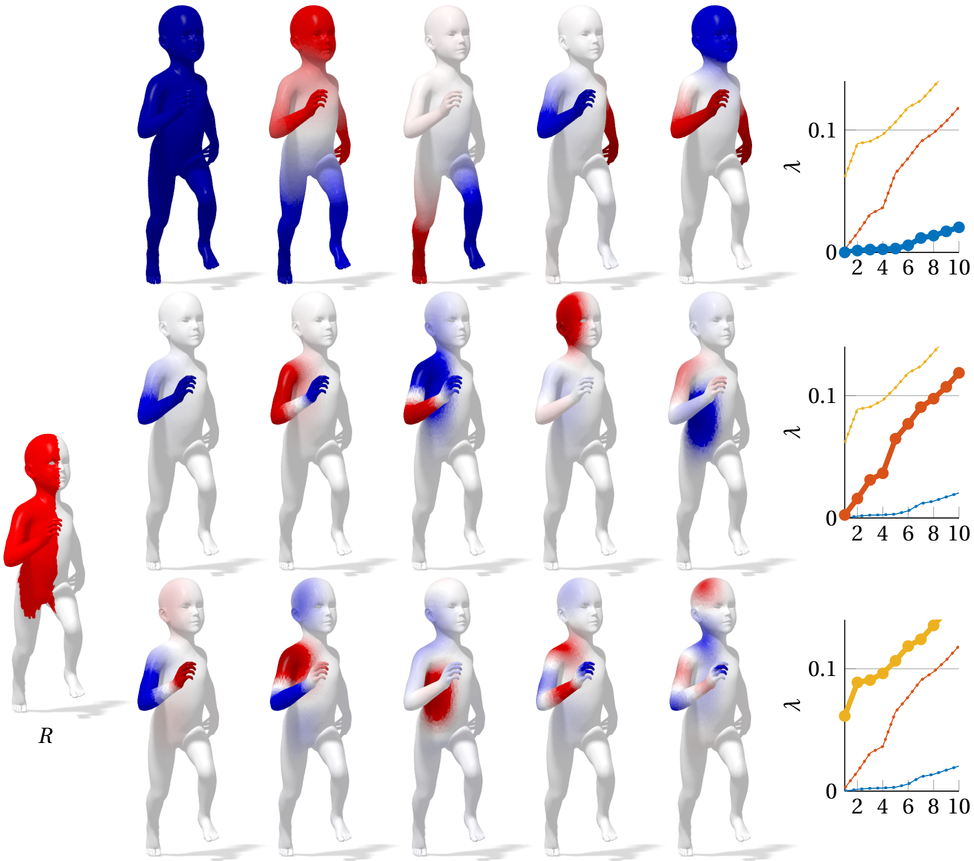


Fig. 5.25: Localized manifold harmonics for the red region shown on the left. We show the first $k' = 5$ standard Laplacian eigenfunctions (top row), the first $k = 5$ LMH with the orthogonality term disabled ($\mu_\perp = 0$, middle row), and the first $k = 5$ LMH obtained by optimizing the full energy (5.33) (bottom row). Note that the latter harmonics are orthogonal to the first k' Laplacian eigenfunctions. We also show the generalized eigenvalues associated with each of the three cases.

Relaxed problem

For practical reasons, in what follows we will consider a relaxed variant of (5.28), in which we replace the hard constraints (5.30) by a large penalty:

$$\min_{\psi_1,\ldots,\psi_k} \sum_{j=1}^{k} \mathscr{E}(\psi_j) \quad \text{s.t.} \ \langle \psi_i, \psi_j \rangle_{L^2(\mathscr{M})} = \delta_{ij}, \tag{5.32}$$

where

$$\mathcal{E}(\psi_j) = \mathcal{E}_S(\psi_j) + \mu_R \mathcal{E}_R(\psi_j) + \mu_\perp \mathcal{E}_\perp(\psi_j)\,, \tag{5.33}$$

$$\mathcal{E}_\perp(f) := \sum_{i=1}^{k'} |\langle \phi_i, f \rangle_{L^2(\mathcal{M})}|^2\,. \tag{5.34}$$

Note that problems (5.32) and (5.28) are equivalent as $\mu_\perp \to \infty$. An empirical evaluation of the equivalence of the two formulations will be provided in Section 5.4.4.

Discretization

Here we refer to Chapter 2 for the overview of the discrete representation of manifold $\mathcal{M}$. We just remember that if N is the number ov vertices of the mesh representing $\mathcal{M}$ than the discretization of the LBO $\Delta_\mathcal{M}$ takes the form of an $N \times N$ sparse matrix $\mathbf{L} = -\mathbf{A}^{-1}\mathbf{W}$, where the *mass matrix* $\mathbf{A}$ is a diagonal matrix of area elements and the *stiffness matrix* $\mathbf{W}$ contains the *cotangent weights*.

We now turn to the discretization of problem (5.32). Let $\mathbf{\Psi} \in \mathbb{R}^{n\times k}$ be a matrix containing our discretized basis functions $\psi_1, \ldots, \psi_k$ as its columns, and same way, let $\mathbf{\Phi} \in \mathbb{R}^{n\times k'}$ be a matrix of the first k' Laplacian eigenfunctions $\phi_1, \ldots, \phi_{k'}$. The total energy is discretized as $\sum_{j=1}^{k} \mathcal{E}(\psi_j) = \mathcal{E}(\mathbf{\Psi})$, comprising purely quadratic terms

$$\mathcal{E}_S(\mathbf{\Psi}) = \mathrm{tr}(\mathbf{\Psi}^\top \mathbf{W} \mathbf{\Psi}) \tag{5.35}$$

$$\mathcal{E}_R(\mathbf{\Psi}) = \mathrm{tr}(\mathbf{\Psi}^\top \mathbf{A}\,\mathrm{diag}(\mathbf{v}) \mathbf{\Psi}) \tag{5.36}$$

$$\mathcal{E}_\perp(\mathbf{\Psi}) = \mathrm{tr}(\mathbf{\Psi}^\top \mathbf{A} \underbrace{\mathbf{\Phi}\mathbf{\Phi}^\top \mathbf{A}}_{\mathbf{P}_{k'}} \mathbf{\Psi}) \tag{5.37}$$

where $\mathbf{v}$ denotes the discrete version of $v(x) \equiv (1-u(x))^2$. In other words, the locality penalty (5.31) is implemented as a diagonal update to the standard Laplacian; while the term promoting orthogonality to $\mathbf{\Phi}$ (5.34) is realized as a rank-k' projector $\mathbf{P}_{k'}$.

Due to the linearity of the trace, the discrete version of problem (5.32) can be expressed as

$$\min_{\mathbf{\Psi}\in\mathbb{R}^{n\times k}} \operatorname{tr}(\mathbf{\Psi}^\top \mathbf{Q}_{v,k'}\mathbf{\Psi}) \quad \text{s.t. } \mathbf{\Psi}^\top \mathbf{A}\mathbf{\Psi} = \mathbf{I}, \tag{5.38}$$

where the matrix

$$\mathbf{Q}_{v,k'} = \mathbf{W} + \mu_R \mathbf{A}\operatorname{diag}(\mathbf{v}) + \mu_\perp \mathbf{A}\mathbf{P}_{k'} \tag{5.39}$$

is symmetric and positive semi-definite (we make the dependency on v, k' explicit as a subscript). Problem (5.38) is equivalent to the *generalized eigenvalue problem*

$$\mathbf{Q}_{v,k'}\mathbf{\Psi} = \mathbf{A}\mathbf{\Psi}\mathbf{\Lambda}, \tag{5.40}$$

(see Theorem 1.2 of [135]). We stress that the new operator $\mathbf{Q}_{v,k'}$ is intrinsic, and so are its eigenfunctions.

As shown later in Subsection 5.4.4, a *global* optimum of this problem can be found by classical Arnoldi-like methods. Note that global solutions to the original constrained problem (5.28) can also be easily computed, however throughout this paper we favor the relaxed formulation for computational efficiency reasons. We refer to Subsection 5.4.4 for comparisons.

5.4.3 Properties of LMH

Here we discuss the main theoretical properties and computational aspects of our framework.

Basis functions

As mentioned before, our localized basis functions are orthonormal eigenfunctions of a matrix (5.40) obtained by modification of the Laplacian. Indeed, by setting $\mu_R = \mu_\perp = 0$, the solution of (5.38) is attained by the first k standard Laplacian eigenfunctions. Similarly, by setting $\mu_R = 0$ (no locality) and for $\mu_\perp \to \infty$, problem (5.38) can be equivalently rewritten as

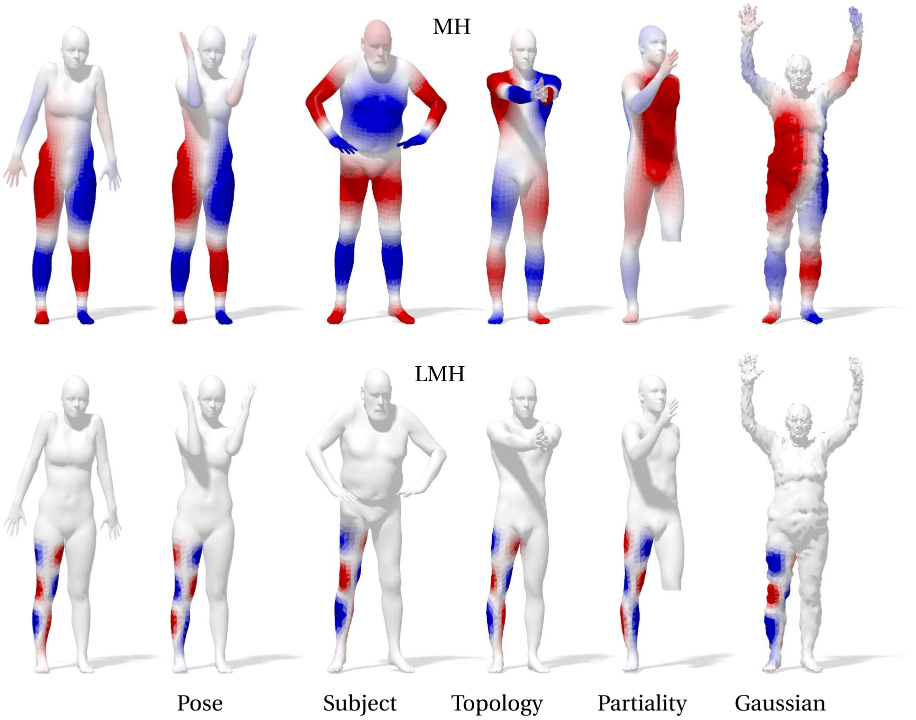


Fig. 5.26: Laplacian eigenfunction ϕ_{12} (top) and localized manifold harmonic ψ_{12} (bottom) under different shape transformations. From left to right: near-isometry (different pose), non-isometric deformation (different subject), topological noise (glued hands), missing part, and geometric noise. LMH is more stable under such deformations compared to the standard MH.

$$\min_{\mathbf{\Psi}\in\mathbb{R}^{n\times k}} \operatorname{tr}(\mathbf{\Psi}^\top \mathbf{W}\mathbf{\Psi}) \quad \text{s.t. } (\mathbf{\Psi}\,\mathbf{\Phi})^\top \mathbf{A}(\mathbf{\Psi}\,\mathbf{\Phi}) = \mathbf{I}, \tag{5.41}$$

whose minimizers are the standard Laplacian eigenfunctions $\psi_1 = \phi_{k'+1}$, $\ldots, \psi_k = \phi_{k'+k}$.

For $\mu_R > 0$ and $\mu_\perp > 0$, we obtain a new set of k functions $\psi_1, \ldots, \psi_k$ localized to a given region $R \subseteq \mathcal{M}$. These functions effectively *extend* the Laplacian eigenbasis, in the sense that the new set $\phi_1, \ldots, \phi_{k'}, \psi_1, \ldots, \psi_k$ forms an *orthonormal basis* for a $k + k'$-dimensional subspace of $L^2(\mathcal{M})$. Importantly, the new basis is still isometry-invariant, and is designed to effectively represent functions with support restricted to the given region (see Figure 5.26). Compared to only using the 'global' Laplacian eigenbasis, the new representation provides a more parsimonious model: fewer localized harmonics are needed to capture the high-frequency content within R, than the number of harmonics that would be needed in the global basis. The localized nature of this construction allows to mitigate considerably the non-local effects associated with the adoption of the global basis (influence of topological noise, etc).

Finally, disabling the orthogonal penalty ($\mu_R > 0$, $\mu_\perp = 0$) would lead to the set of (now possibly linearly dependent) $k + k'$ functions spanning a $k'' \leq k + k'$-dimensional subspace of $L^2(\mathscr{M})$. This may result in a redundant representation of functions supported on $R \subseteq \mathscr{M}$, e.g., whenever a standard Laplacian eigenfunction has also support in R. We refer to the experimental results for a deeper analysis of the effect of orthogonality on the representation quality.

Spectrum

By the interpretation of (5.38) as a generalized eigenvalue problem (5.40), we obtain a natural notion of *spectrum* associated with the LMH, namely given by $\lambda_j(\mathbf{Q}) = \boldsymbol{\psi}_j^\top \mathbf{Q} \boldsymbol{\psi}_j$, $j = 1, \ldots, k$. Indeed, since at the optimum $\mathscr{E}_R(\boldsymbol{\psi}_j) \approx \mathscr{E}_\perp(\boldsymbol{\psi}_j) \approx 0$ for all j, we have $\boldsymbol{\psi}_j^\top \mathbf{Q} \boldsymbol{\psi}_j \approx \boldsymbol{\psi}_j^\top \mathbf{W} \boldsymbol{\psi}_j$, i.e., the Dirichlet energy of $\boldsymbol{\psi}_j$ as in the classical setting (2.13). This provides us with a natural ordering of the basis functions; it remains to see in what measure do the localized harmonics bring additional (higher frequency) information to the global basis formed by the first k' Laplacian eigenfunctions $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{k'}$. A first answer is provided by the following

Theorem 5.1. ***(spectral gap).*** *Let* $(\phi_i, \lambda_i(\mathbf{W}))_{i=1}^n$ *be the eigenpairs of the standard Laplacian, and let* $\mathbf{Q}_{v,k'}$ *be defined as in* (5.39). *Then, for large enough* $\mu_\perp$*, any non-negative* v *and any choice of* $k' \leq n - 1$*, we have* $\lambda_{k'}(\mathbf{W}) \leq \lambda_1(\mathbf{Q}_{v,k'})$*, with equality holding iff* $\phi_{k'+1}(x) = 0$ *whenever* $v(x) \neq 0$.

Proof. Let $\mathbf{W}$, $\mu_\perp \mathbf{A} \mathbf{P}_{k'}$ and $\mu_R \mathbf{A} \mathrm{diag}(\mathbf{v})$ be real symmetric positive semidefinite matrices of dimension $n \times n$, and define

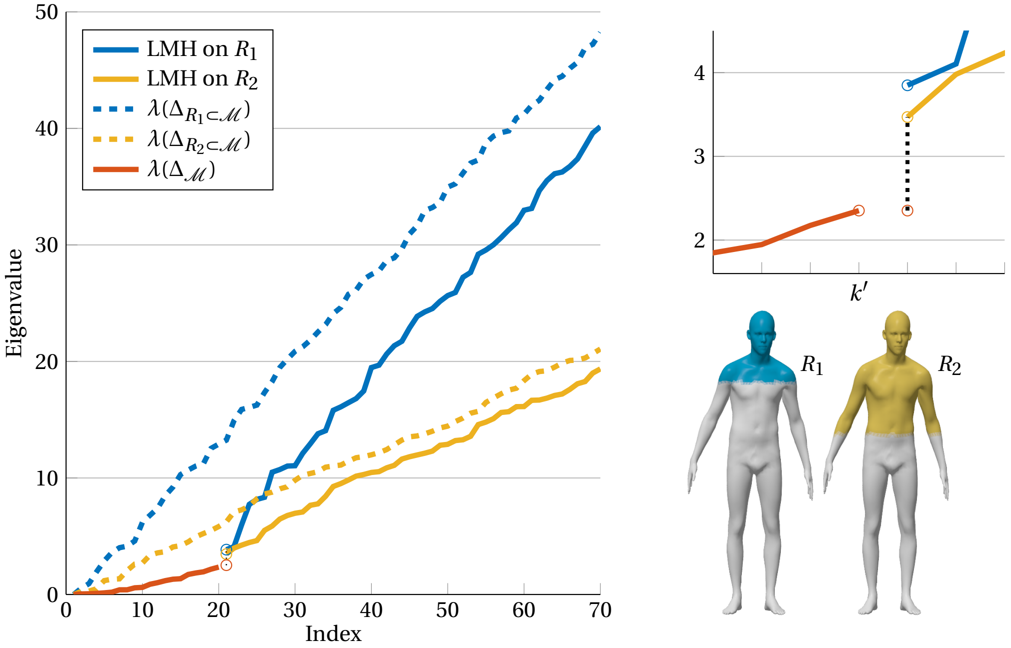


Fig. 5.27: *Left*: LMH spectra grow linearly with rate inversely proportional to the area of the region, and are bounded from above by the standard Laplacian eigenvalues of submanifold $R \subseteq \mathcal{M}$. *Top-right*: Enlargement of the left plot around the index $k' = 20$. We illustrate the spectral gap (dotted) between $\lambda_{k'}(\mathbf{W})$ and $\lambda_1(\mathbf{Q}_{v,k'})$; note that the gap is different among the two regions.

$$\mathbf{Q}_{v,k'} = \mathbf{W} + \mu_\perp \mathbf{A}\mathbf{P}_{k'} + \mu_R \mathbf{A}\mathrm{diag}(\mathbf{v}).$$

Let $0 = \lambda_1(\mathbf{W}) \leq \ldots \leq \lambda_n(\mathbf{W})$ be the eigenvalues for the generalized eigenvalue problem of $\mathbf{W}$ and $\lambda_1(\mathbf{W} + \mu_\perp \mathbf{A}\mathbf{P}_{k'}) \leq \ldots \leq \lambda_n(\mathbf{W} + \mu_\perp \mathbf{A}\mathbf{P}_{k'})$ and $\lambda_1(\mathbf{Q}_{v,k'}) \leq \ldots \leq \lambda_n(\mathbf{Q}_{v,k'})$ be the generalized eigenvalues of $\mathbf{W} + \mu_\perp \mathbf{A}\mathbf{P}_{k'}$ and $\mathbf{Q}_{v,k'}$ respectively. We aim to prove that

$$\lambda_{k'}(\mathbf{W}) \leq \lambda_1(\mathbf{Q}_{v,k'})\,, \tag{5.42}$$

for some $\mu_\perp, \mu_R \in \mathbb{R}$ and for every $k' \in \{0, \ldots, n-1\}$.

We start by observing that

$$\lambda_{k'}(\mathbf{W}) \leq \lambda_{k'+1}(\mathbf{W}) = \lambda_1(\mathbf{W} + \mu_\perp \mathbf{A}\mathbf{P}_{k'})\,, \tag{5.43}$$

where the first inequality is given by the non-decreasing ordering of the eigenvalues, and the equality on the right follows from the fact that for some choice of $\mu_\perp > \lambda_{k'+1}(\mathbf{W})$, $\phi_{k'+1}$ is the minimizer of $\mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x}$ under the orthogonality conditions $\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1$ and $\langle\phi_l,\mathbf{x}\rangle_{L^2(\mathcal{M})}=0$, $\forall\, l\in\{1,\dots,k'\}$, i.e., $(\mu_\perp\mathbf{AP}_{k'})\mathbf{x}=\mathbf{0}$.

Invoking a special case of Corollary 4.3.4b in [62] and using the fact that $\mu_R\mathbf{A}\text{diag}(\mathbf{v})$ only has non-negative eigenvalues (being a diagonal matrix with non-negative entries), we obtain the following inequality:

$$\lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'}) \le \lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})+\mu_R\mathbf{A}\text{diag}(\mathbf{v})) = \lambda_1(\mathbf{Q}_{v,k'})\,. \tag{5.44}$$

Furthermore, this inequality is an equality if and only if $\exists\mathbf{x}\in\mathbb{R}^n$ s.t. $\mathbf{x}\neq 0$ and the following three conditions are satisfied:

1. $(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x}=\lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x}$;
2. $(\mathbf{Q}_{v,k'})\mathbf{x}=\lambda_1(\mathbf{Q}_{v,k'})\mathbf{x}$;
3. $(\mu_R\mathbf{A}\text{diag}(\mathbf{v}))\mathbf{x}=0$.

Putting together (5.43) and (5.44) we can conclude that:

$$\lambda_{k'}(\mathbf{W}) \le \lambda_{k'+1}(\mathbf{W}) \le \lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'}) \le \lambda_1(\mathbf{Q}_{v,k'})\,. \tag{5.45}$$

Note that the existence of a gap is given either by the violation of any of the three conditions above, or in the presence of simple spectra, i.e., whenever $\lambda_{k'}(\mathbf{W})\neq\lambda_{k'+1}(\mathbf{W})$.

Choice of $\mu_\perp$.

We aim to prove that for every $\mu_\perp>\gamma$ for some $\gamma\in\mathbb{R}^+$ we have:

$$\lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'}) \ge \lambda_{k'+1}(\mathbf{W})\,. \tag{5.46}$$

We can rewrite the two terms of this inequality as:

$$\lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'}) = \min_{\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1} \mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x} \tag{5.47}$$

$$\lambda_{k'+1}(\mathbf{W}) = \min_{\substack{\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1 \\ \left\langle\phi_i,\mathbf{x}\right\rangle_{L^2(\mathcal{M})}=0,\ \forall i=1,\dots,k'}} \mathbf{x}^\top\mathbf{W}\mathbf{x}. \tag{5.48}$$

The objective in (5.47) can be rewritten as:

$$\mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{A}\mathbf{P}_{k'})\mathbf{x}=\mathbf{x}^\top\mathbf{W}\mathbf{x}+\mathbf{x}^\top(\mu_\perp\mathbf{A}\mathbf{P}_{k'})\mathbf{x}. \tag{5.49}$$

We now express our vectors as the Fourier series $\mathbf{x}=\sum_{i=1}^n \alpha_i\phi_i$, where $\alpha_i=\langle\phi_i,\mathbf{x}\rangle_{L^2(\mathcal{M})}$. Noting that $\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1$ implies $\sum_{i=1}^n\alpha_i^2=1$, we can write:

$$\begin{aligned}\mathbf{x}^\top\mathbf{W}\mathbf{x}&=(\sum_{i=1}^n\alpha_i\phi_i)^\top\mathbf{W}(\sum_{i=1}^n\alpha_i\phi_i)\\&=(\sum_{i=1}^n\alpha_i\phi_i)^\top(\sum_{i=1}^n\lambda_i(\mathbf{W})\alpha_i\mathbf{A}\phi_i)=\sum_{i=1}^n\lambda_i(\mathbf{W})\alpha_i^2.\end{aligned} \tag{5.50}$$

Similarly, we can rewrite the second summand in (5.49) as:

$$\begin{aligned}\mathbf{x}^\top(\mu_\perp\mathbf{A}\mathbf{P}_{k'})\mathbf{x}&=(\sum_{i=1}^n\alpha_i\phi_i)^\top(\mu_\perp\mathbf{A}\mathbf{P}_{k'})(\sum_{i=1}^n\alpha_i\phi_i) &(5.51)\\&=\mu_\perp(\sum_{i=1}^n\alpha_i\phi_i)^\top(\mathbf{A}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\mathbf{A})(\sum_{i=1}^n\alpha_i\phi_i) &(5.52)\\&=\mu_\perp\Big((\sum_{i=1}^n\alpha_i\phi_i)^\top\mathbf{A}\boldsymbol{\Phi}\Big)\Big(\boldsymbol{\Phi}^\top\mathbf{A}(\sum_{i=1}^n\alpha_i\phi_i)\Big) &(5.53)\\&=\mu_\perp\,[\alpha_1,\ldots,\alpha_{k'}]\,[\alpha_1,\ldots,\alpha_{k'}]^\top &(5.54)\\&=\mu_\perp\sum_{i=1}^{k'}\alpha_i^2. &(5.55)\end{aligned}$$

From (5.50) and (5.55) we can conclude:

$$\mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{A}\mathbf{P}_{k'})\mathbf{x}=\mathbf{x}^\top\mathbf{W}\mathbf{x}+\mathbf{x}^\top(\mu_\perp\mathbf{A}\mathbf{P}_{k'})\mathbf{x}=\sum_{i=1}^n\lambda_i(\mathbf{W})\alpha_i^2+\mu_\perp\sum_{i=1}^{k'}\alpha_i^2. \tag{5.56}$$

At this point we split the proof in three different cases:

1. $\langle\phi_i,\mathbf{x}\rangle_{L^2(\mathcal{M})}=0,\ \forall i=1,\ldots,k'$, that is equivalent to ask that $\mathbf{P}_{k'}\mathbf{x}=\mathbf{0}$. In this case we have:

$$\lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'}) = \min_{\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1} \mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x} \tag{5.57}$$

$$= \min_{\substack{\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1 \\ \langle\phi_i,\mathbf{x}\rangle_{L^2(\mathcal{M})}=0,\ \forall i=1,\dots,k'}} (\mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x}) \tag{5.58}$$

$$= \min_{\substack{\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1 \\ \langle\phi_i,\mathbf{x}\rangle_{L^2(\mathcal{M})}=0,\ \forall i=1,\dots,k'}} \mathbf{x}^\top\mathbf{W}\mathbf{x} = \lambda_{k'+1}(\mathbf{W})\,. \tag{5.59}$$

2. $\mathbf{x} \in span(\phi_1,\dots,\phi_{k'})$, implying that $\alpha_i = 0\ \forall i > k'$ and hence $\mathbf{x} = \sum_{i=1}^{k'} \alpha_i\phi_i$. We get:

$$\mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x} = \sum_{i=1}^{k'} \lambda_i(\mathbf{W})\alpha_i^2 + \mu_\perp\sum_{i=1}^{k'}\alpha_i^2\,. \tag{5.60}$$

Since we take the minimum over the $\mathbf{x}$ s.t. $\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})} = 1$ we have $\sum_{i=1}^{k'}\alpha_i^2 = 1$ and:

$$\mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x} = \sum_{i=1}^{k'} \lambda_i(\mathbf{W})\alpha_i^2 + \mu_\perp \geq \mu_\perp\,, \tag{5.61}$$

where the equality is realized for $\mathbf{x} = \phi_1$ since $\lambda_1(\mathbf{W}) = 0$, and all other cases yield $\mu_\perp$ plus some non-negative quantity. We get to:

$$\lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'}) = \min_{\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1} \mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x} = \mu_\perp\,. \tag{5.62}$$

3. For the last case we have $\langle\phi_i,\mathbf{x}\rangle_{L^2(\mathcal{M})} \neq 0$ for at least one $i = 1,\dots,k'$ and for at least one $i > k'$ at the same time.

$$\mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x} = \sum_{i=1}^{n} \lambda_i(\mathbf{W})\alpha_i^2 + \mu_\perp\sum_{i=1}^{k'}\alpha_i^2 \tag{5.63}$$

$$= \sum_{i=1}^{k'} \lambda_i(\mathbf{W})\alpha_i^2 + \sum_{i=k'+1}^{n} \lambda_i(\mathbf{W})\alpha_i^2 + \mu_\perp\sum_{i=1}^{k'}\alpha_i^2 \tag{5.64}$$

$$= \sum_{i=1}^{k'} (\lambda_i(\mathbf{W})+\mu_\perp)\alpha_i^2 + \sum_{i=k'+1}^{n} \lambda_i(\mathbf{W})\alpha_i^2\,. \tag{5.65}$$

Since $\lambda_i(\mathbf{W}) \geq \lambda_{k'+1}(\mathbf{W})$, $\forall i \geq k'+1$ we can write:

$$\mathbf{x}^\top(\mathbf{W}+\mu_\perp \mathbf{AP}_{k'})\mathbf{x} = \sum_{i=1}^{k'}(\lambda_i(\mathbf{W})+\mu_\perp)\alpha_i^2 + \sum_{i=k'+1}^{n}\lambda_i(\mathbf{W})\alpha_i^2 \tag{5.66}$$

$$\geq \sum_{i=1}^{k'}(\lambda_i(\mathbf{W})+\mu_\perp)\alpha_i^2 + \lambda_{k'+1}(\mathbf{W})\sum_{i=k'+1}^{n}\alpha_i^2 \tag{5.67}$$

$$\geq \sum_{i=1}^{k'}\mu_\perp\alpha_i^2 + \lambda_{k'+1}(\mathbf{W})\sum_{i=k'+1}^{n}\alpha_i^2 \,. \tag{5.68}$$

If we take $\mu_\perp > \lambda_{k'+1}(\mathbf{W})$ in order to satisfy the condition imposed by case 2, we get:

$$\mathbf{x}^\top(\mathbf{W}+\mu_\perp \mathbf{AP}_{k'})\mathbf{x} \geq \sum_{i=1}^{k'}\mu_\perp\alpha_i^2 + \lambda_{k'+1}(\mathbf{W})\sum_{i=k'+1}^{n}\alpha_i^2 \tag{5.69}$$

$$> \lambda_{k'+1}(\mathbf{W})\sum_{i=1}^{k'}\alpha_i^2 + \lambda_{k'+1}(\mathbf{W})\sum_{i=k'+1}^{n}\alpha_i^2 \tag{5.70}$$

$$= \lambda_{k'+1}(\mathbf{W})\sum_{i=1}^{n}\alpha_i^2 \tag{5.71}$$

$$= \lambda_{k'+1}(\mathbf{W})\,. \tag{5.72}$$

We can therefore conclude that

$$\lambda_1(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'}) = \min_{\langle\mathbf{x},\mathbf{x}\rangle_{L^2(\mathcal{M})}=1} \mathbf{x}^\top(\mathbf{W}+\mu_\perp\mathbf{AP}_{k'})\mathbf{x} > \lambda_{k'+1}(\mathbf{W}) \textit{ if } \mu_\perp > \lambda_{k'+1}(\mathbf{W})\,. \tag{5.73}$$

In Figure 5.28 we show an empirical evaluation across several choices of $\mu_\perp$.

This theorem ensures the existence of a non-negative gap between the two spectra, i.e., the new basis functions do not introduce any redundancy, and the gap is the smallest possible by the global optimality of (5.38). In other words, the localized basis “picks up” where the global basis “left off” (see Figure 5.27 for examples). Note that the last condition

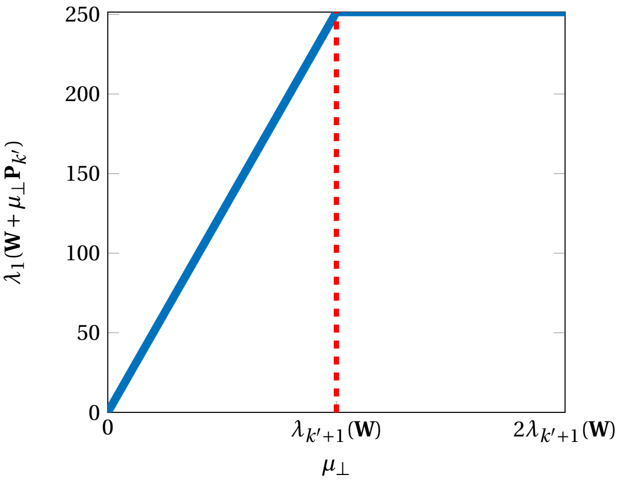


Fig. 5.28: Plot of $\lambda_1(\mathbf{W}+\mu_\perp \mathbf{A}\mathbf{P}_{k'})$ at increasing $\mu_\perp$. Note how for every $\mu_\perp \leq \lambda_{k'+1}(\mathbf{W})$ the frequency (y-axis) increases, converging at $\mu_\perp > \lambda_{k'+1}(\mathbf{W})$. At convergence, the orthogonality constraint (encoded in the penalty term $\mathscr{E}_\perp(\psi)$ in the LMH formulation) is satisfied.

on v in Theorem 5.1 is almost never realized in practice: equality is obtained only when the Laplacian eigenfunction $\phi_{k'+1}$ is localized to the same region indicated by v.

Interestingly, for a special class of functions v the spectrum $\lambda_1(\mathbf{Q}_{v,k'}) \leq \lambda_2(\mathbf{Q}_{v,k'}) \leq \ldots$ follows a well-defined behaviour, as remarked below.

Observation: Let v be a *binary* indicator function supported on some (possibly disconnected) region $R \subseteq \mathscr{M}$.

Then, $\lambda_i(\mathbf{Q}_{v,k'}) - \lambda_1(\mathbf{Q}_{v,k'}) \propto i/\sqrt{\text{Area}(R)}$ as $i \to \infty$.

The observation above can be thought of as a generalization of Weyl's asymptotic law [29] to sub-regions of $\mathscr{M}$ (see Figure 5.27).

Comparison to standard Laplacian on parts

Perhaps the most direct way to achieve locality is to consider the given region $R \subseteq \mathscr{M}$ as a separate manifold with boundary ∂R and Laplacian Δ_R, and then compute the eigen-decomposition $\mathbf{W}^R\mathbf{\Psi}^R = \mathbf{A}^R\mathbf{\Psi}^R\mathbf{\Lambda}^R$ of Δ_R (note that $\mathbf{W}^R, \mathbf{A}^R$ can be obtained as submatrices of $\mathbf{W}, \mathbf{A}$ followed by nor-

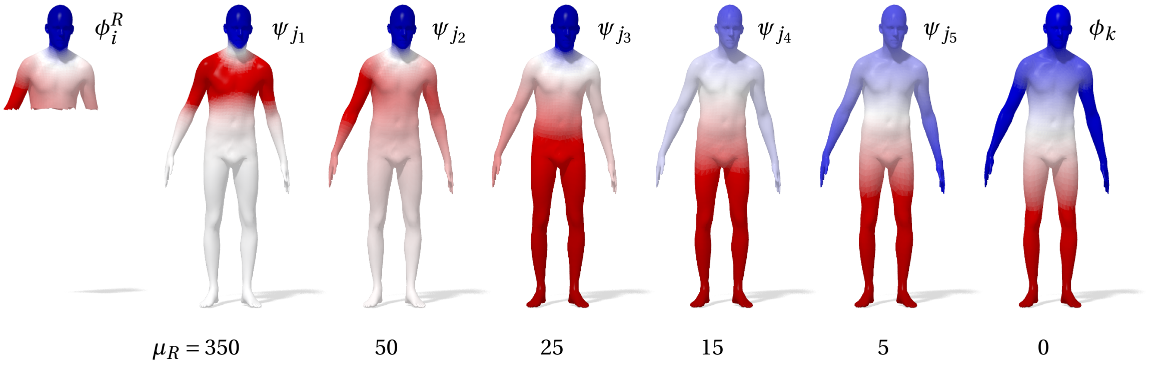


Fig. 5.29: Our model allows to smoothly transition from a localized solution equivalent to a standard Laplacian eigenfunction ϕ_i^R on a *partial* shape with Neumann boundary conditions (first column), to a globally supported solution equivalent to a standard Laplacian eigenfunction ϕ_k on the *full* shape (last column). Each ψ is obtained by solving a different problem with a different μ_R and $\mu_\perp = 0$. Note that the resulting "interpolating" harmonics $\psi_{j_{1,\dots,5}}$ do not necessarily correspond to the same eigenvalue.

malization and by fixing the weights along ∂R). The eigenfunctions ψ_i^R can then be extended to the entire $\mathscr{M}$ by means of zero-padding,

$$\tilde{\psi}_i^R(x) = \begin{cases} \psi_i^R(x) & x \in R \\ 0 & \text{else} \end{cases}$$

A first difference between this and our approach lies in the fact that $\langle \tilde{\psi}_i^R, \phi_j \rangle_{L^2(\mathscr{M})} \neq \delta_{ij}$ in general, i.e., the extended partial eigenfunctions do not "complete" the global basis and there is no separation of spectra (guaranteed in our case by Theorem 5.1), leading in turn to a redundant representation.

Secondly, our approach is more general in that we allow "soft" regions represented by allowing v to obtain values in the interval $[0,1]$, which is obviously not achievable by extracting sub-regions. This latter property is especially important in applications where a sharp (binary) selection would lead to undesirable boundary effects around the region of interest (see Subsection 5.4.5 for examples).

Finally, we stress that the standard Laplacian $\Delta_{R\subseteq\mathcal{M}}$ may have an eigenspace in common with our operator with $\mu_\perp = 0$, a binary v on R, and large enough μ_R. In turn, the full Laplacian $\Delta_{\mathcal{M}}$ is always obtained for $\mu_R = 0$. A remarkable manifestation of this fact is given by the "interpolation effect" shown in Figure 5.29. Note that the observation above is not true in general, since we do not impose any boundary conditions w.r.t. R in our problem (indeed, we allow R to be soft), while all eigenfunctions of Δ_R always satisfy specific boundary conditions such as (2.10). Despite the loose connection, this observation allows us to complement the lower bound of Theorem 5.1 by the following

Theorem 5.2. ***(upper bound).*** *Let* $\mathbf{W}^R$ *be the stiffness matrix associated with the submanifold* $R \subseteq \mathcal{M}$ *and define* v *as the binary indicator function of* $\mathcal{M} \setminus R$. *Then,* $\lambda_i(\mathbf{Q}_{v,k'}) \leq \lambda_{i+k'}(\mathbf{W}^R)$ *for any* $k \leq n$.

Proof. We want to show that $\forall k \in \{1, 2, \ldots, n\}$ we have the following upper bound:

$$\lambda_i(\mathbf{Q}_{v,k'}) \leq \lambda_{i+k'}(\mathbf{W}^R)\,.$$

Similarly to Theorem 1, the proof follows directly from Corollary 4.3.4b in [62], which specialized to our case reads:

$$\lambda_i(\mathbf{W}^R + \mu_\perp \mathbf{A}\mathbf{P}_{k'}) \leq \lambda_{i+\pi}(\mathbf{W}^R)\,, \tag{5.74}$$

where π is the number of positive eigenvalues of $\mu_\perp \mathbf{A}\mathbf{P}_{k'}$. Since $\mathbf{Q}_{v,k'} = \mathbf{W}^R + \mu_\perp \mathbf{A}\mathbf{P}_{k'}$ and using the fact that $\mu_\perp \mathbf{A}\mathbf{P}_{k'}$ is a positive semidefinite matrix with rank k', we have $\pi = k'$, leading to:

$$\lambda_i(\mathbf{Q}_{v,k'}) = \lambda_i(\mathbf{W}^R + \mu_\perp \mathbf{A}\mathbf{P}_{k'}) \leq \lambda_{i+\pi}(\mathbf{W}^R) = \lambda_{i+k'}(\mathbf{W}^R)\,. \tag{5.75}$$

See Figure 5.27 for an example.

Comparison to compressed manifold modes

Ozoliņš et al. [118] proposed computing *compressed manifold modes (CMM)* as solutions to

$$\min_{\boldsymbol{\Psi}\in\mathbb{R}^{n\times k}} \operatorname{tr}(\boldsymbol{\Psi}^\top \mathbf{W}\boldsymbol{\Psi}) + \mu\|\boldsymbol{\Psi}\|_1 \quad \text{s.t.} \;\; \boldsymbol{\Psi}^\top \mathbf{A}\boldsymbol{\Psi} = \mathbf{I}. \tag{5.76}$$

Problem (5.76) makes use of a sparsity-inducing L_1 prior which, together with the smoothness promoted by the Dirichlet term, leads to the resulting functions having compact support controlled by parameter μ. It is important to note that this model does not allow to explicitly control the modes location. As shown in [110], these functions tend to concentrate around areas like shape protrusions and ridges. While different in its nature, the CMM model (5.76) admits a computational procedure which shares some similarities with ours. Assume that the solution of (5.76) for a given μ is a set of k functions supported on regions $R_1,\ldots,R_k$ (obtained *a posteriori*), represented by the soft indicators $1-v_1,\ldots,1-v_k$, and let $\mu_\perp = 0$. Then, the application of our framework using the matrices $\mathbf{Q}_{v_i,0}$ corresponds to one iteration of the iterative reweighting scheme proposed for the efficient computation of CMMs in [20].

Comparison to elliptic operator.

Concurrently with our work, Choukroun et al. [33] considered a family of elliptic operators of the form $\mathbf{H} = \mathbf{W} + \mathbf{V}$, where the *potential* $\mathbf{V}$ is a diagonal operator akin to our localization term (5.36). The same approach was recently followed in [91] to obtain localized basis functions around points of interest on the surface.

Differently from these approaches, we seek for localized basis functions that *simultaneously* lie in a subspace orthogonal to span$\{\phi_1,\ldots,\phi_{k'}\}$, where ϕ_i are the first k' standard Laplacian eigenfunctions. In other words, we seek to "augment" the global basis by introducing a local refinement, while the aforementioned works attempt to construct a complete basis in agreement with the input potential. This is a crucial difference that has noticeable effects in practice, as we will demonstrate in Subsection 5.4.5.

5.4.4 Implementation

Optimization

As shown in Subsection 5.4.2, computing our localized basis functions boils down to solving a generalized eigenvalue problem $\mathbf{Q\Psi} = \mathbf{A\Psi\Lambda}$, with $\mathbf{Q} = \mathbf{W} + \mu_R \mathbf{A}\mathrm{diag}(\mathbf{v}) + \mu_\perp \mathbf{A\Phi\Phi}^\top \mathbf{A}$. We note that computing $\mathbf{Q}$ explicitly involves the construction of a dense $n \times n$ matrix $\mathbf{A\Phi\Phi}^\top\mathbf{A}$, which may become prohibitive for large meshes. However, we avoid this computation altogether by noticing that the $\mu_\perp$-term has very low rank $k' \ll n$. This condition allows the application of exact update formulas throughout the optimization of problem (5.38). We show how to *efficiently* compute a global solution to the generalized eigenvalue problem:

$$\mathbf{Q\Psi} = \mathbf{A\Psi\Lambda}, \tag{5.77}$$

with $\mathbf{Q} = \mathbf{W} + \mu_R \mathbf{A}\mathrm{diag}(\mathbf{v}) + \mu_\perp (\mathbf{A\Phi})(\mathbf{A\Phi})^\top$.

Since the operator $\mathbf{Q}$ is real and symmetric w.r.t. the positive semi-definite mass matrix $\mathbf{A}$, we employ the (globally optimal) implicitly restarted Arnoldi method (IRAM) [85] (as implemented in the ARPACK suite [84]) for computing its first k eigenpairs. The application of IRAM involves iteratively solving linear systems of the form:

$$\mathbf{Qx}^{(t)} = \mathbf{Ab}^{(t)} \tag{5.78}$$

for some given $\mathbf{b}^{(t)} \in \mathbb{R}^n$. Expressing $\mathbf{Q}$ in terms of the matrices $\mathbf{Z} = \mathbf{W} + \mu_R \mathbf{A}\mathrm{diag}(\mathbf{v})$ and $\mathbf{B} = \mathbf{A\Phi} \in \mathbb{R}^{n\times k'}$, we come to:

$$(\mathbf{Z} + \mu_\perp \mathbf{BB}^\top)\mathbf{x}^{(t)} = \mathbf{Ab}^{(t)}. \tag{5.79}$$

Note that matrix $\mathbf{BB}^\top$ can be interpreted as a rank-k' update to $\mathbf{Z}$ (with $k' \ll n$), allowing us to apply the Sherman-Morrison-Woodbury identity [152]:

$$(\mathbf{Z} + \mu_\perp \mathbf{BB}^\top)^{-1} = \mathbf{Z}^{-1} - \mu_\perp \mathbf{Z}^{-1}\mathbf{B}\underbrace{(\mathbf{I} + \mu_\perp \mathbf{B}^\top\mathbf{Z}^{-1}\mathbf{B})}_{\mathbf{Y}}{}^{-1}\mathbf{B}^\top\mathbf{Z}^{-1}.$$

It is important to notice that the rhs does not involve the computation of $\mathbf{B}\mathbf{B}^\top$, and only involves efficient operations with a sparse $n \times n$ matrix $\mathbf{Z}$ and a dense $k' \times k'$ matrix $\mathbf{Y}$. The application of this formula for the solution of problem (5.79), and in turn (5.77) via IRAM, is illustrated for clarity in Algorithm 1. A similar procedure was followed in [20] for the computation of CMM.

ALGORITHM 1: Efficient solution of problem (5.79).

```
Solve sparse linear system $\mathbf{Z}\boldsymbol{\xi} = \mathbf{A}\mathbf{b}^{(t)}$ for $\boldsymbol{\xi} \in \mathbb{R}^{n}$;
Solve sparse linear system $\mathbf{Z}\boldsymbol{\Gamma} = \mu_{\perp}\mathbf{B}$ for $\boldsymbol{\Gamma} \in \mathbb{R}^{n \times k'}$;
Solve dense linear system $(\mathbf{I}_{k'} + \mathbf{B}^{\top}\boldsymbol{\Gamma})\boldsymbol{\eta} = \mathbf{B}^{\top}\boldsymbol{\xi}$ for $\boldsymbol{\eta} \in \mathbb{R}^{k'}$;
Compute final solution $\mathbf{x}^{(t)} = \boldsymbol{\xi} - \boldsymbol{\Gamma}\boldsymbol{\eta}$.
```

Timing

In Table 5.3 we report the runtime (in seconds) required by our method as executed on an Intel 3.6 GHz Core i7 cpu with 16GB ram. We compare the execution time for the exact problem with hard orthogonality constraints (5.28) and the relaxed problem (5.32). Note that while the latter relaxation is significantly more efficient, the two formulations yielded numerically close solutions in all our experiments (see Figure 5.30). We show how to compute a global solution to problem (5.28). We start by observing that the hard constraints (5.30) require the desired basis functions $\boldsymbol{\Psi}$ to lie in the null space of the linear map $\mathbf{P}_{k'} := \boldsymbol{\Phi}\boldsymbol{\Phi}^\top\mathbf{A}$ (i.e., the projector onto $\mathrm{Im}(\boldsymbol{\Phi})$), or equivalently to lie in the range of $\mathbf{I} - \mathbf{P}_{k'}$ (i.e. the projector onto the orthogonal subspace). This is easily achieved by letting $\boldsymbol{\Psi} = (\mathbf{I} - \mathbf{P}_{k'})\mathbf{Y}$, and solving the generalized eigenvalue problem:

$$\tilde{\mathbf{Q}}\mathbf{Y} = \tilde{\mathbf{A}}\mathbf{Y}\boldsymbol{\Lambda}, \tag{5.80}$$

where $\tilde{\mathbf{Q}} = (\mathbf{I} - \mathbf{P}_{k'})^\top(\mathbf{W} + \mu_R\mathbf{A}\mathrm{diag}(\mathbf{v}))(\mathbf{I} - \mathbf{P}_{k'})$ and $\tilde{\mathbf{A}} = (\mathbf{I} - \mathbf{P}_{k'})^\top\mathbf{A}(\mathbf{I} - \mathbf{P}_{k'}) = \mathbf{A}(\mathbf{I} - \mathbf{P}_{k'})$. A similar trick was recently used in [44] for computing maximum magnitude eigenvalues of a large matrix. Note that solving prob-

	k = 100		**k = 200**		**k = 300**	
	hard	**soft**	**hard**	**soft**	**hard**	**soft**
∼ 120K	-	112.3s	-	200.5s	-	304.9s
∼ 12K	40.8s	7.6s	66.7s	16.4s	79.3s	25.0s
∼ 1.2K	1.1s	0.7s	2.2s	1.0s	4.6s	1.4s

Table 5.3: Runtime comparison for global optimization of our problem under hard (5.28) and soft constraints (5.32) across different mesh resolutions (number of faces) and basis size k. Tests denoted by '-' could not run due to memory limitations.

lem (5.80) involves the explicit construction of a dense $n \times n$ matrix $\mathbf{P}_{k'}$, becoming prohibitive for large meshes.

Choice of parameters

Parameters μ_R and k' control the locality and the number of global harmonics to use, respectively, and are application-dependent (see Subsection 5.4.5). Parameter $\mu_\perp$ enforces orthogonality w.r.t. the standard Laplacian eigenfunctions, and should be chosen large enough so that the orthogonality constraints are satisfied. In our experiments we used $\mu_R \approx 10^2$ and $\mu_\perp \approx 10^5$; see Figure 5.30 for a quantitative evaluation on the choice of $\mu_\perp$.

5.4.5 Applications

Localized manifold harmonics are a general tool that can be employed as a drop-in replacement for, or in conjunction with the classical manifold harmonics ubiquitous in spectral shape analysis. We showcase their application in two broad tasks in graphics: spectral shape processing and shape correspondence.

Spectral shape processing

In this context, the surface $\mathscr{M}$ is represented as a vector-valued function $\mathbf{x} : \mathscr{M} \to \mathbb{R}^3$, encoding the spatial coordinates of its embedding in $\mathbb{R}^3$; transformations to the surface geometry are then phrased as filtering operations applied to the coordinate functions. Vallet and Lévy [146]

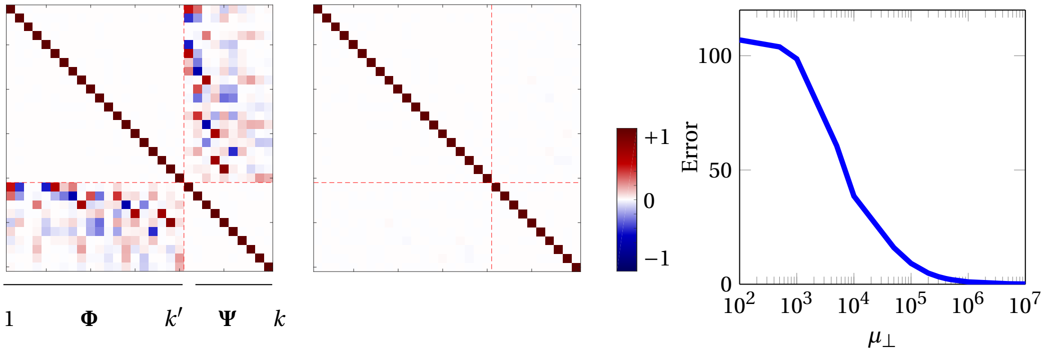


Fig. 5.30: *Left*: Matrix of inner products $(\langle b_i, b_j\rangle_{L^2(\mathcal{M})})$, where $b_i, b_j \in \{\phi_1, \ldots, \phi_{k'}, \psi_1, \ldots, \psi_k\}$. Here ψ_i are the optimal localized basis functions computed with $\mu_\perp = 10^{-1}$ (first column) and $\mu_\perp = 10^5$ (second column). *Right*: We plot the discrepancy between solutions to the exact (5.28) and relaxed (5.32) problems as a function of $\mu_\perp$, measured as the L_2 distance between the resulting spectra.

proposed to perform such filtering in the Fourier domain, where the coordinates $\mathbf{x}$ are expressed as linear combinations of Laplacian eigenfunctions,

$$\mathbf{x} = \sum_{i\geq 1} \langle \phi_i, \mathbf{x}\rangle_{L^2(\mathcal{M})} \phi_i \,, \tag{5.81}$$

where, with some abuse of notation we denote, $\langle \phi_i, \mathbf{x}\rangle_{L^2(\mathcal{M})} = (\langle \phi_i, x_1\rangle_{L^2(\mathcal{M})}$ $, \ldots, \langle \phi_i, x_3\rangle_{L^2(\mathcal{M})})$. By truncating the summation to the first k' terms, one obtains a band-limited representation of the surface. The representation is coarse for small k', while finer details are captured for large values of k'; see Figure 5.31 (top row) for an example. The expression in (5.81) provides an effective way for representing and manipulating simple shapes with smoothly varying coordinate functions, which can be compactly represented in the first few harmonics. Conversely, this representation is much less efficient for surfaces having details at smaller scales.

Assume a given set of regions, identifying areas of the shape with geometric detail. By computing localized harmonics $\{\psi_j\}_j$ on the given regions, we obtain a representation of the surface geometry:

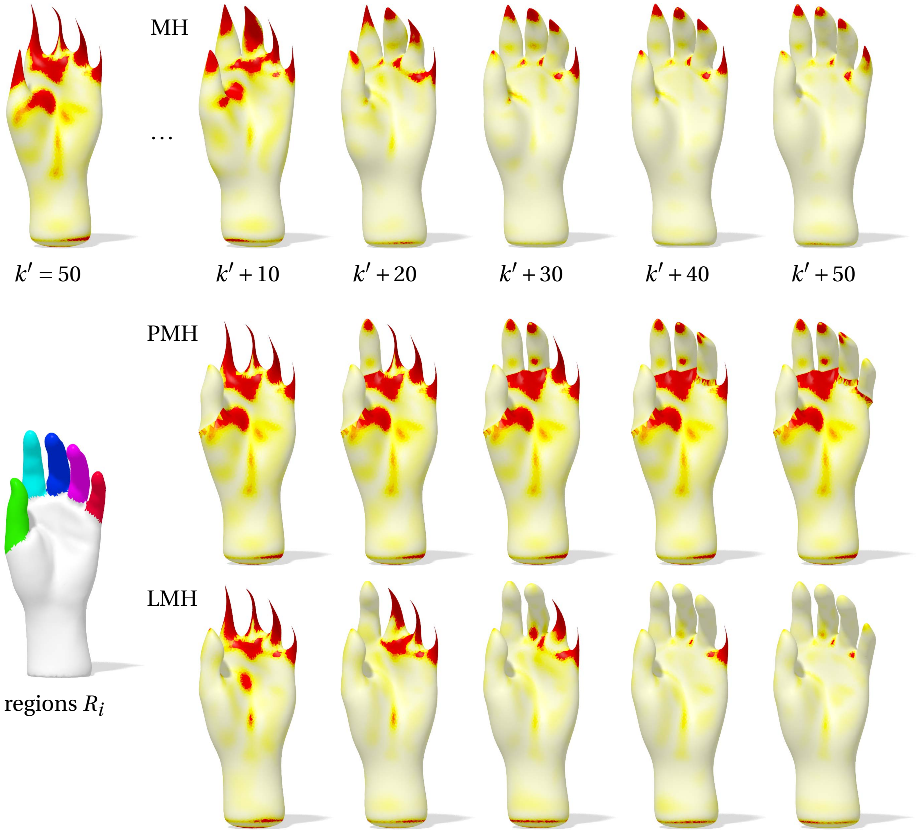


Fig. 5.31: *First row*: Surface reconstruction via (5.81) using the first k' to $k'+50$ Laplacian eigenfunctions. *Second row*: Each finger $R_i \subset \mathscr{M}$ is treated as a separate sub-manifold, and the eigenfunctions of the "partial" Laplacians Δ_{R_i} are used to update the initial reconstruction by adding 10 harmonics per finger. *Third row*: Reconstruction via (5.82) with 10 localized harmonics per finger. Note the significantly higher accuracy of LMH despite using the same number of harmonics as MH and PMH. The heatmap encodes reconstruction error, growing from white to dark red.

$$\mathbf{x} \approx \sum_{i=1}^{k'} \langle \phi_i, \mathbf{x} \rangle_{L^2(\mathscr{M})} \phi_i + \sum_{j=1}^{k} \langle \psi_j, \mathbf{x} \rangle_{L^2(\mathscr{M})} \psi_j \,, \tag{5.82}$$

where $\phi_1, \ldots, \phi_{k'}$ are the standard Laplacian eigenfunctions and $\langle \phi_i, \psi_j \rangle_{L^2(\mathscr{M})} \approx 0$ for all $i = 1, \ldots, k'$ and $j = 1, \ldots, k$. For a fixed number of terms in the series, the expression (5.82) yields a more accurate approximation of the original surface than (5.81), since the localized ba-

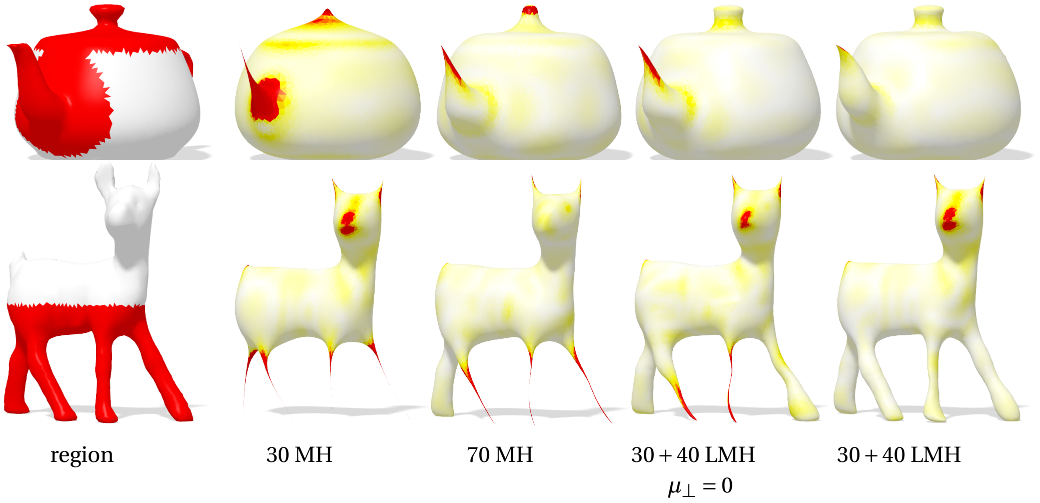


Fig. 5.32: Comparison between manifold harmonics (MH) and localized manifold harmonics (LMH) without and with the orthogonality term (5.34) using $k' = 30$ and $k = 40$.

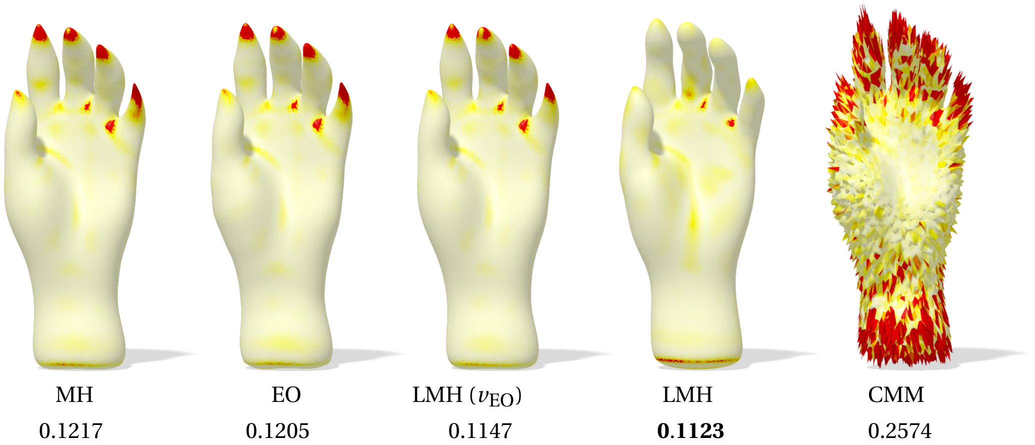


Fig. 5.33: Comparison between different pipelines using a fixed number of basis functions (equal to 100 for all methods – LMH uses 50 global and 50 localized harmonics in both experiments). We report the reconstruction error below each method.

sis functions capture the high-frequency content more quickly (as also manifested in the rapid growth of the spectrum, see Figure 5.27). We refer to Figure 5.31 for a detailed illustration of this behaviour. In Figure 5.32, we demonstrate the effect of the lack of orthogonality ($\mu_\perp = 0$) on the reconstruction quality, and in Figure 5.34 provide a quantitative

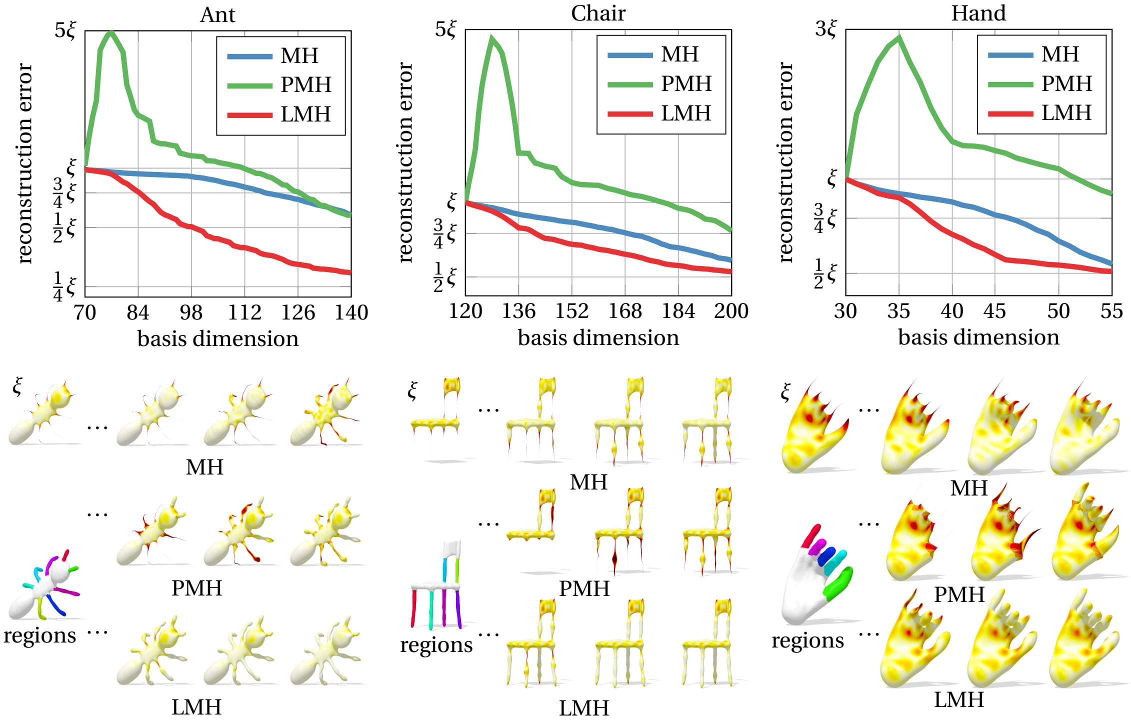


Fig. 5.34: Comparisons among MH, PMH and LMH in surface representation. For each class we report the average reconstruction error at increasing number of basis functions (x-axis). Here, ξ denotes the error obtained by MH with $k' = 50$. The heatmaps encode reconstruction error, growing from white to dark red.

evaluation on pre-segmented meshes from the Princeton segmentation benchmark [31]. In these tests, we compare with standard manifold harmonics (MH) and "partial" manifold harmonics (PMH). The latter approach consists in reconstructing the surface indicated by each region $R_i \subset \mathscr{M}$ separately by using the eigenfunctions of the Laplacian Δ_{R_i}; the reconstructed part is then "glued" back to the full shape. We measure the point-wise reconstruction error by the Euclidean distance between each reconstructed vertex and its corresponding point in the original surface.

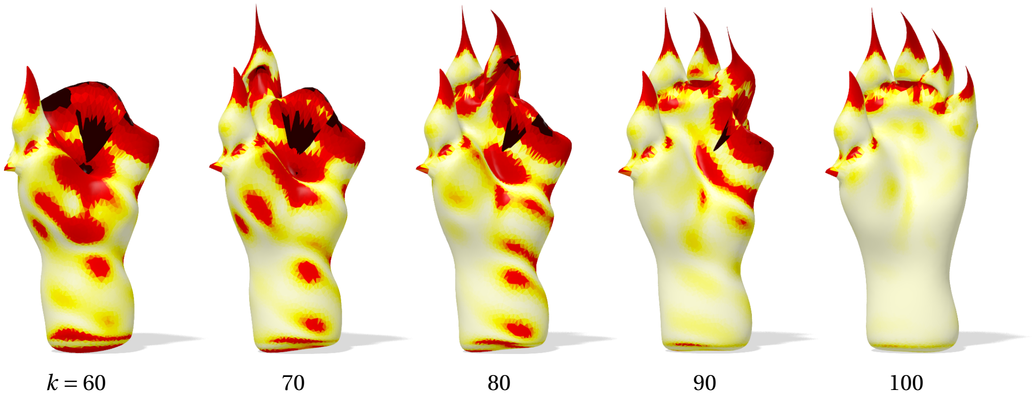


Fig. 5.35: Incremental hand reconstruction using EO basis functions [33] and the binary regions of Figure 5.31.

Comparison to other pipelines

Differently from our method, both CMM [110] and elliptic operator (EO) [33] do not allow to build upon and enrich a given set of basis functions. In particular, CMM relies on the assumption that the set of localized basis functions covers the entire surface, while EO employs a soft potential with global support. In Figure 5.33 we compare the reconstruction error of standard MH, EO using the potential defined in [33], our method using the latter potential as a soft region v_{EO}, our method using the binary regions of Figure 5.31, and CMM using a covering set of compressed modes. The better performance of LMH is motivated by two key properties. First, we allow to easily control the surface regions for localizing the refinement (unlike CMM); second, LMH gain benefit from the underlying global harmonics, enriching them where needed and maintaining good quality outside the support of the regions (unlike EO). For completeness, we also test the performance of EO when fed with a sequence of binary potentials (the finger regions in Figure 5.31). We emphasize that EO is not designed to operate in conjunction with an existing basis, hence there is no natural way to implement an incremental update as the one shown in Figure 5.31: the harmonics computed on each bi-

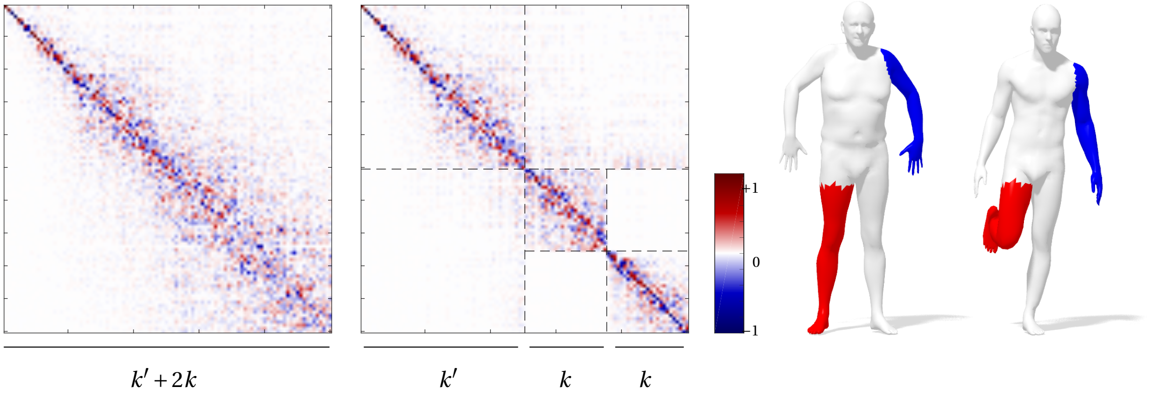


Fig. 5.36: Functional map matrices w.r.t. the standard Laplacian eigenbasis (left) and w.r.t a “mixed” basis composed of k' Laplacian eigenfunctions and $k+k$ localized harmonics (middle). The maps encode the ground-truth correspondence between the two shapes shown on the right; the regions used for the computation of LMH are highlighted in red and blue. Note the block-diagonal structure of the second matrix, a manifestation of the capability of LMH to encode local information compactly.

nary region would not have an underlying global structure to attach to. For this reason, we provide an extra region (the palm) where 50 EO basis functions are computed, and this basis is incrementally updated with 10 EO basis functions per finger. The result is shown in Figure 5.35.

Shape correspondence

As we saw in the Chapter 4 Ovsjanikov et al. [115] proposed to represent correspondences between shapes by a linear operator (called *functional map*).

As a linear operator, the functional map T admits a matrix representation $\mathbf{C} = (c_{ij})$ w.r.t. bases $\{\phi_i^{\mathcal{M}}\}$ and $\{\phi_j^{\mathcal{N}}\}$ on $L^2(\mathcal{M})$ and $L^2(\mathcal{N})$ respectively,

$$Tf = \sum_{ij\geq 1} \langle \phi_i^{\mathcal{M}}, f\rangle_{L^2(\mathcal{M})} \underbrace{\langle T\phi_i^{\mathcal{M}}, \phi_j^{\mathcal{N}}\rangle_{L^2(\mathcal{N})}}_{c_{ji}} \phi_j^{\mathcal{N}}, \tag{5.83}$$

for an arbitrary $f \in L^2(\mathcal{M})$. By choosing the Laplacian eigenfunctions on $\mathcal{M}$ and $\mathcal{N}$ as the bases $\{\phi_i^{\mathcal{M}}\}$ and $\{\phi_j^{\mathcal{N}}\}$, one can truncate the series

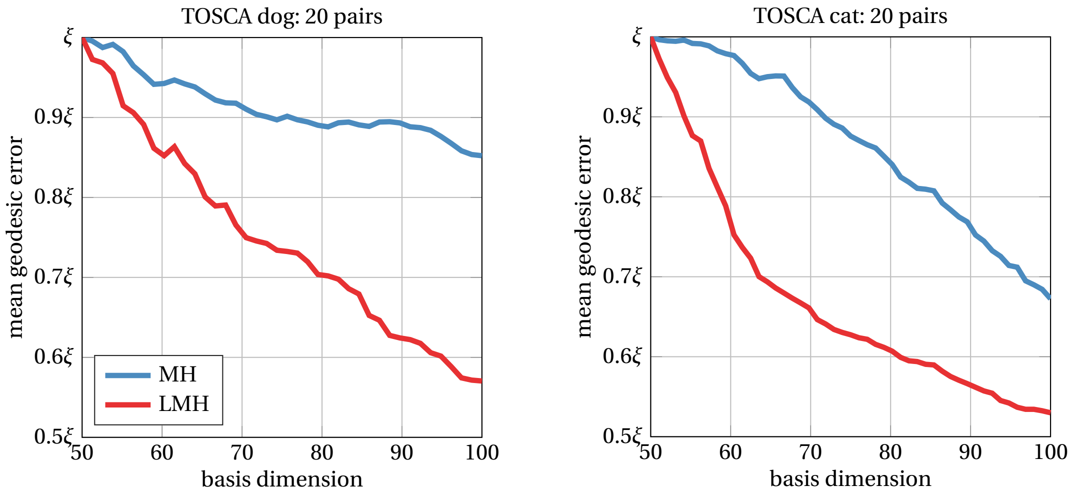


Fig. 5.37: Mean geodesic error vs. number of basis functions used in the functional map representation. Note that LMH lead to an increase in accuracy, resulting in turn in a more compact representation of the correspondence. Here ξ is defined as the mean geodesic error of MH at $k' = 50$.

(5.83) to the first k' terms – hence obtaining a compact representation which can be interpreted as a band-limited approximation of the full map. Correspondence problems can then be phrased as searching for a $k' \times k'$ matrix $\mathbf{C}$ minimizing simple data fidelity criteria [111, 115] or exhibiting a particular structure depending on the correspondence setting [80, 122, 128].

Similar to the previous experiments, the standard Laplacian eigenbasis may not be the best choice in the presence of fine details: the low-pass nature of the spectral representation of the map, embodied in matrix $\mathbf{C}$, negatively affects the quality of the representation at a point-wise level. Indeed, recovering a point-to-point map from a functional map is considered a difficult problem in itself [129, 149], and is at the heart of several applications dealing with maps.

LMH can be directly employed for representing functional correspondence *in conjunction* with the Laplacian eigenbasis:

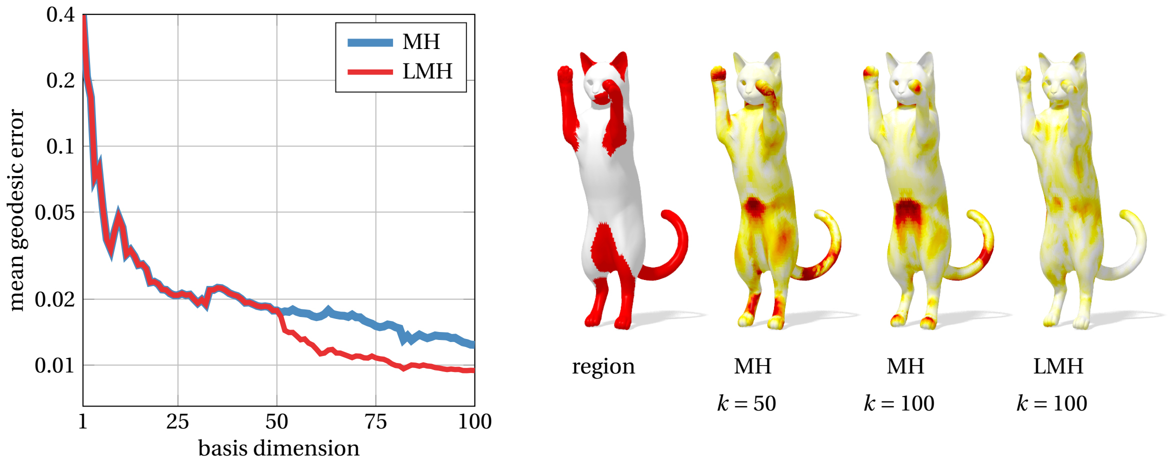


Fig. 5.38: Accuracy improvement in the functional map representation, obtained by introducing LMH after $k' = 50$ Laplacian eigenfunctions. On the right we show the regions used for LMH and the geodesic error (encoded as hot colors, growing from white to dark red) obtained for different configurations. Note the higher accuracy attained by LMH for the same amount of basis functions.

$$Tf = \sum_{\ell,m=1}^{k+k'} \langle \omega_\ell^{\mathcal{M}}, f \rangle_{L^2(\mathcal{M})} \langle T\omega_\ell^{\mathcal{M}}, \omega_m^{\mathcal{N}} \rangle_{L^2(\mathcal{N})} \omega_m^{\mathcal{N}} , \tag{5.84}$$

where $\omega \in \Omega$ and $\Omega = \{\phi_i\}_{i=1}^{k'} \cup \{\psi_j\}_{j=1}^{k}$ is the union of the standard and localized manifold harmonics. Note that the formula above allows for 'cross-talk' between the MH and LMH bases, and it can be seen as a localized *refinement* to some initial correspondence represented in the (global) Laplacian eigenbasis. An example of the resulting correspondence matrix **C** is shown in Figure 5.36.

In Figure 5.37 we show a quantitative comparison between the two representations (5.83) and (5.84). For this experiment we use near-isometric shapes from the TOSCA dataset [19]. For each pair of shapes, we use their ground-truth point-to-point correspondence to construct functional maps of increasing size in the Laplacian eigenbasis and in the LMH basis. For the latter, we use Eq. (5.84) with $k' = 50$ and k increasing from 1 to 50. The harmonics $\{\psi_\ell^{\mathcal{M}}, \psi_m^{\mathcal{N}}\}_{\ell,m=1}^{k}$ are localized to the regions having large reconstruction error, computed as in the previous experiments. Note that even though these regions can be arbitrarily disconnected and irregular (see, e.g., Figure 5.38), our framework can be applied without modifications. A point-to-point map is recovered from each functional map using the nearest neighbor approach [115]. We measure the correspondence quality via its *geodesic error*. Assume a point-to-point match $(x, y) \in \mathcal{M} \times \mathcal{N}$ is recovered, whereas the ground-truth correspondence is (x, y^*); we compute the quantity

$$\epsilon(x) = \frac{d_{\mathcal{N}}(y, y^*)}{\sqrt{\text{Area}(\mathcal{N})}}, \tag{5.85}$$

where $d_{\mathcal{N}}$ is the geodesic distance on $\mathcal{N}$. For the second set of experiments we consider a challenging setting of shape correspondence known as *deformable object-in-clutter* [37]. In this scenario, the task is to match a given model to a scene where the model appears in a different pose, and in the simultaneous presence of clutter (extra objects) and missing parts. The problem was recently tackled in [37] using the functional map representation; to our knowledge, this method represents the current state of the art for this class of problems. As data for these tests we use the entire dataset adopted for the comparisons in [37]. The dataset consists of 3 TOSCA models (*cat, centaur, dog*) and 150 synthetic scenes in which

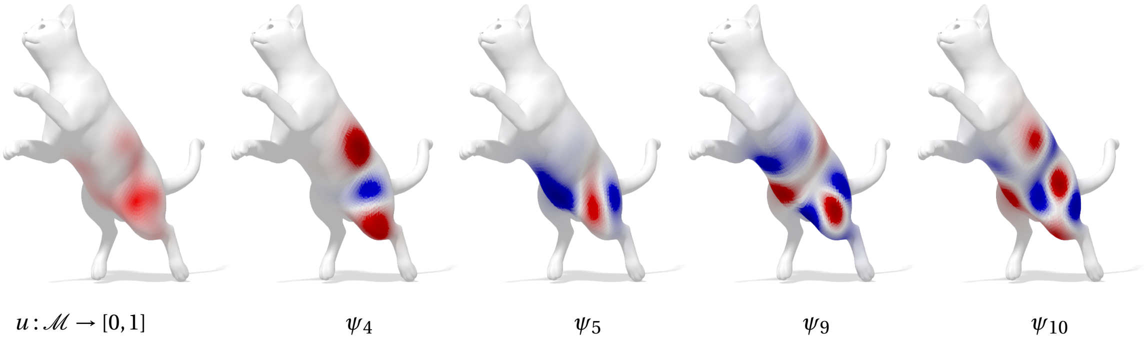


Fig. 5.39: Localized manifold harmonics on a *soft* region encoded by function u. No thresholding is required in order to obtain a valid set of functions minimizing the energy (5.33).

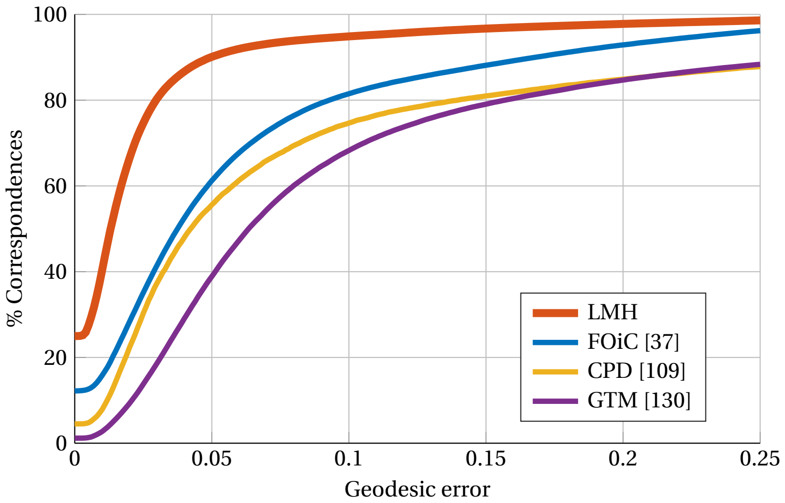


Fig. 5.40: Comparisons with the state of the art in deformable object-in-clutter. All methods use the same input data.

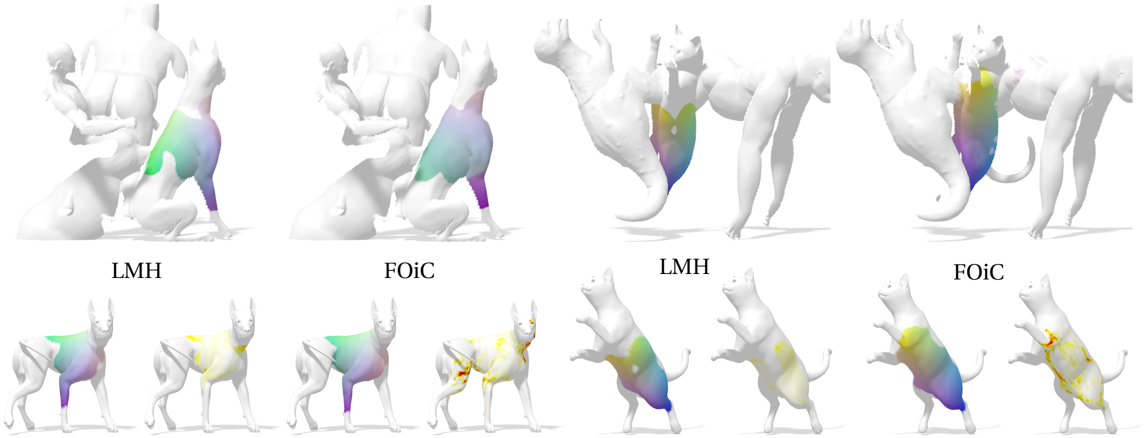


Fig. 5.41: Qualitative comparisons between our LMH-based approach for deformable shape correspondence in clutter and the state of the art [37]. For each experiment we show the dense correspondence (corresponding points have same color) and the geodesic error (hot colors growing from white to dark red).

the models appear. Sparse point-to-point matches (around 10) between models and scenes, obtained using the approach of [127], are also provided. Given m input matches, we construct a mixture of m Gaussians

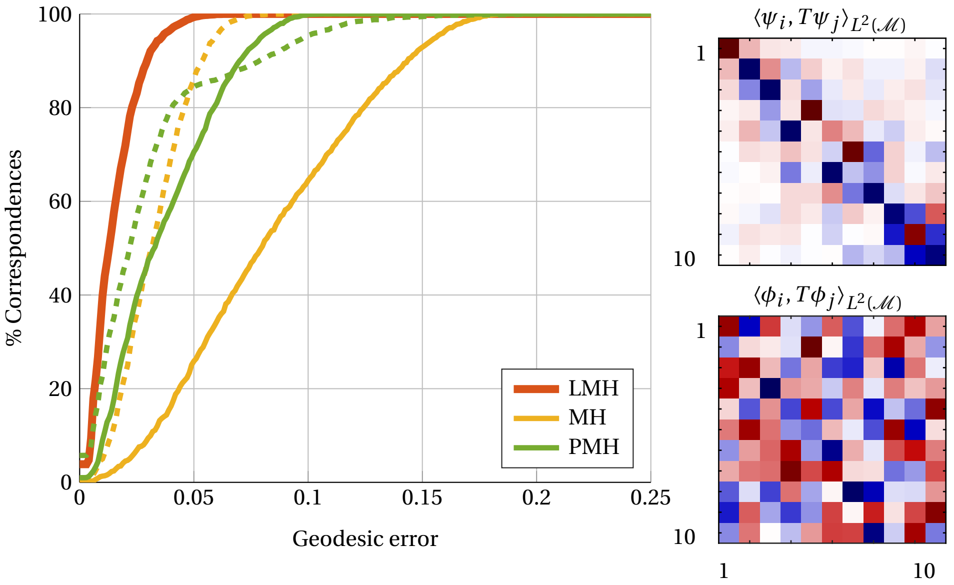


Fig. 5.42: *Left*: Correspondence accuracy in the object-in-clutter setting. Each curve corresponds to a functional map expressed in a different basis, using the same input data. We show the performance when using $k = 10$ (solid curves) and $k = 50$ (dashed curves) basis functions. In this example, MH could not reach the quality of LMH for any choice of k. *Right*: Functional map matrices in the LMH basis $\{\psi\}$ (top) and in the MH basis $\{\phi\}$ (bottom).

with equal variance (set to 1% of the shape diameter) to define a *soft* region u on both model and scene (see Figure 5.39). We then construct a functional map $\mathbf{C}$ upon the input sparse correspondence, and represent it w.r.t. $k = 15$ localized manifold harmonics computed on the soft regions (note that here we do *not* use any global eigenfunction, i.e., we set $k' = 0$ in (5.84)). Finally, we recover from $\mathbf{C}$ a dense point-to-point map localized on u using the intrinsic ICP approach of [115]. The results of this experiment are reported in Figure 5.40 quantitatively and in Figure 5.41 qualitatively. Despite the simple approach, our method gains a significant improvement in accuracy, of up to 25% upon the state of the art on this benchmark, highlighting the inherent robustness of LMH to missing parts and topological artifacts. Finally, in Figure 5.42 we compare (on a single pair of shapes) our pipeline with the counterparts obtained by replacing LMH with MH and PMH.

Limitations

Perhaps the biggest limitation of our approach lies in the availability of regions (or soft counterparts thereof) upon which to carry out the localized spectral analysis. Such information may not be available in certain unsupervised applications, where it is often difficult to define a meaningful segmentation – indeed, an inherently task-specific notion suggesting the use of data-driven approaches. Further, similarly to the classical setting, it is not obvious how to choose the number of harmonics to employ for a given task.

5.4.6 A non-optimal localized basis

In this Subsection we introduce an alternative method to obtain a new set of k basis functions for a manifold $\mathcal{M}$ that are *smooth*, *orthonormal*, and *localized* on a region $R \subseteq \mathcal{M}$, that could be added to a given orthonormal basis functions $\phi_1, \ldots, \phi_{k'}$, for example the first k' Laplacian eigenfunctions. We define the new basis solving an optimization problem for which it is not guaranteed an optimal solution. In addition, the proposed non optimal solution is not very efficient. For these reasons we prefer the LMH solution. This alternative solution generates the local basis by filtering the standard MH. It therefore remains in the spirit of this thesis.

We are looking for a set of functions $\{\psi_j\}_{j=1}^{k}$ that has the properties already explained.

We can formulate this problem as a constrained optimization problem in a optimization variable $\Psi = [\psi_1 \ldots \psi_k]$, a $N \times k$ matrix of which columns are the basis functions. The constrains and the cost functions of the optimization problem are properly chosen to meet the targets listed above. First of all, in order to reduce the dimensionality we use the eigenbasis to achieve a coefficients-based representation of our optimization variable. Given $\{\phi_l\}_{l=1}^{\tilde{N}}$, with $\tilde{N} \leq N$, a subset of the eigenfunctions instead of looking for the functions ψ_j we can consider its coefficients in the Fourier transform, in other words:

$$\psi_j = \sum_{l=1}^{\tilde{N}} \langle \psi_j, \phi_l \rangle_{L^2(\mathcal{M})} \phi_l = \sum_{l=1}^{\tilde{N}} x_{l,j} \phi_l.$$

We so obtain a $\tilde{N} \times k$ matrix $X = (x_{l,j})$ of Fourier coefficients. At this point we can use X as optimization variable. X can be view as the Fourier representation of Ψ. Once we have X we can return to the spatial representation $[\psi_1 \dots \psi_k] = [\phi_1 \dots \phi_{\tilde{N}}]\, X$, that we can abbreviate in a matrix form as $\Psi = \Phi X$. In this context we present below how we reach the proprieties listed above imposing constraints to the coefficients matrix X.

1. Orthonormality

The claim of orthonormality of the basis is equivalent to the following request:

$$\Psi^\top \Omega \Psi = I.$$

In terms of coefficients we have:

$$\Psi^\top \Omega \Psi = (\Phi X)^\top \Omega \Phi X = X^\top \Phi^\top \Omega \Phi X = X^\top I X = X^\top X.$$

If we are looking for a solution X that belongs to $Mat_{\mathbb{R}}(\tilde{N}, K)$ the manifold of real valued matrices with dimension $\tilde{N} \times K$, the constraint $X^\top X = I$ can be forced imposing to find the solution in the Stiefel manifold, the submanifold of $Mat_{\mathbb{R}}(\tilde{N}, K)$ defined as:

$$\left\{ A \in Mat_{\mathbb{R}}(\tilde{N}, K) | A^\top A = I \right\}.$$

We optimize the solution of our cost function imposing to the solution to be a point on the Stiefel manifold, this constraint ensure us to obtain orthonormal set of functions.

2. Smoothness

in order to ensure the smoothness of the function ψ_k we add the Dirichlet energy as a term of our cost function. As we have already seen the Dirichlet energy can be written in terms of the LBO as:

$$tr(\Psi^\top \Delta_{\mathcal{M}} \Psi),$$

where tr is the trace operator. The minimization of this energy guarantees the smoothness of the basis functions. Using X the Fourier representation of Ψ we rewrite this energy in the following equivalent formulation:

$$tr(X^\top \Lambda_{\mathcal{M}} X), \tag{5.86}$$

where $\Lambda_{\mathcal{M}}$ is the diagonal matrix with the element of position (l, l) equal to λ_l the l-th eigenvalue of $\Delta_{\mathcal{M}}$.

3. Locality

For this property we start from a subdivision of the mesh in Q semantic segments, eventually overlapped, $\mathcal{S}_1, \dots, \mathcal{S}_Q$ such that:

$$\mathcal{M} \subseteq \bigcup_{q=1}^{Q} \mathcal{S}_q.$$

This segmentation can be represented as a set of Q indicator functions $\{r_q\}_{q=1}^{Q}$ where:

$$r_q(i) = \begin{cases} 1, & \text{if } i \in \mathcal{S}_q \\ 0, & \text{otherwise} \end{cases}.$$

Now suppose that K the cardinality of our basis can be divided by Q without loss of generality, so there exists a value $p \in \mathbb{N}$ such that $K = pQ$. From these indicator functions we can get a mask that allows functions to be non-zero only in the segments that we select. Let e be the constant function equal to 1 on all the vertices. We can place side by side p copies

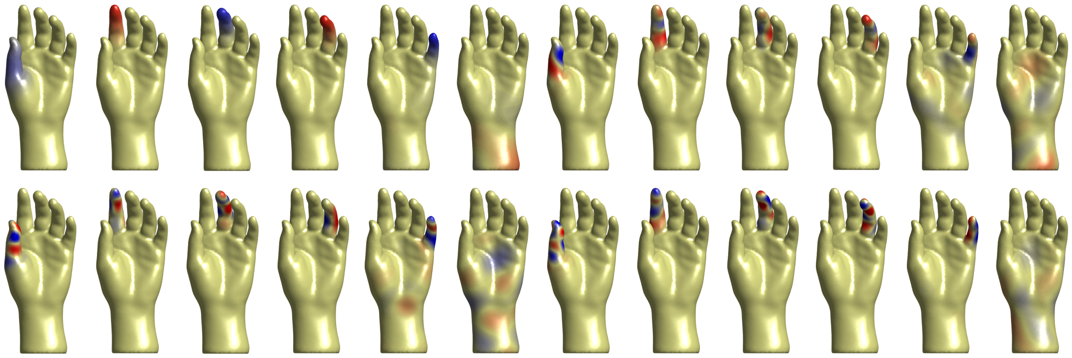


Fig. 5.43: Some ordered basis functions on the six segments of an hand computed imposing the $MASK$ filter and the "triangular filter". As can be seen the new basis are localized but in some case there is some energy outside the selected region ()

of the matrix $\left[e-r_1,\ldots e-r_Q\right]$, obtaining a matrix $N\times K$, namely $MASK$. In $MASK$ e minus the Q regions are alternate and repeating p-times. So $MASK$ is 0 on the vertices that belong to the segments, which are alternate in the columns, and 1 on the rest. Now we consider the spatial representation of the functions Ψ, so $\Psi=\Phi X$ and we can write the following constraint:

$$\|\Psi\odot MASK\|_F^2\,, \tag{5.87}$$

where $\|\|_F$ is the Frobenius norm and $\odot$ is the element wise product between the two matrices. In order to make explicit the variable X we can write the equation 5.87 as

$$\|\Phi X\odot MASK\|_F^2\,, \tag{5.88}$$

Minimize this norm is equivalent to require that the function ψ_k restricts its non-zero values in the segment that corresponds to the k-th column of the matrix $MASK$.

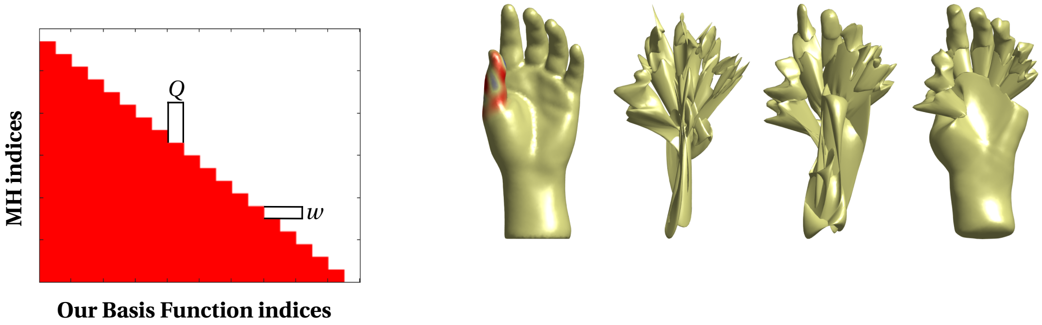


Fig. 5.44: On the left an example of a “triangular filter” used in our experiments. On the right the first basis function computed without the “triangular filter”, and reconstruction of the hand from left to right using 60, 120 and 180 basis functions computed without the “triangular filter”.

4. “Frequency value”

We known that the eigenbasis are sorted with respect to the absolute value of the corresponding eigenvalues that are intrinsically related to the frequencies of the Fourier basis that they form. This property is kindly desired for our basis Ψ in order to maintains a frequencies-based order, and allows meaningful as basis for the frequency-analysis. A natural way to get this property is to force the functions ψ_k to gradually be composed from an increasing number of eigenfunctions. Starting from the first eigenfunctions when k is small and adding some eigenfunctions as k increases. From the coefficients point o views this coincide to initially allows non-zero coefficients only in the first positions, and while k grows permit to more coefficients to be different from zero. To translate this facility in a bond we build a binary filter. We divide the number of eigenbasis $\tilde{N}$ by the number of frequencies that we want to obtain for every segments, namely u. Given u we can suppose that K the cardinality of the basis s equal to uQ, that is the number of frequencies for very regions multiplies for the number of regions. Less than approximations we can think that there is a number w such that $wu = \tilde{N}$. The filter T is a matrix of dimension $\tilde{N} \times K$ equal to 0 in the upper blocks with indices from 1 to $w(j-1)$ on the rows and from $1+(j-1)Q$ to jQ on the columns for every $j \in \{1,\dots,u\}$. All the others entries of T are equal to 1. Once we have T we can write the constraint explicitly as:

$$\|X \odot T\|_F^2\,, \tag{5.89}$$

An example of the obtained filter T can be visualized on the left of Figure 5.44. This filter is a blocks upper triangular matrix for this reason from now on we refer to it as “triangular filter”. Moreover this filter provides for all the segments a frequency-value functions for each of the new frequencies that we generate. As shown in Figure 5.44 without the constraint in equation 5.89 we can not guarantee to preserve a frequencies-based order. The fact of find high frequencies in the first positions is also very negative for many applications, as for example in reconstruction task, as shown in Figure 5.44 on the right.

Complexity

We use the optimization toolbox Manopt [16] to solve our optimization problems. The computation of around 300 functions of our localized basis ψ using 900 eigenfunctions takes on average 400 seconds for the single mesh version and 800 seconds for the shared version on two meshes with around 7000 vertices on a machine with 32GB of RAM using an Intel 3,6 GHz Core i7 processor.

Results and conclusion

We compare the results shown in Figure 5.31 with the ones in Figure 5.45. The heatmap used in Figure 5.45 allows to highlight more the error (yellow and orange). Also with the optimized basis we note the improvements in the accuracy of the reconstruction despite the use of the MH, highlighted in the blue boxes on the right. In the same time it is possible to see some not nice behaviour when we use the optimized local functions, as example in the bottom part of the wrist. Furthermore, it is also

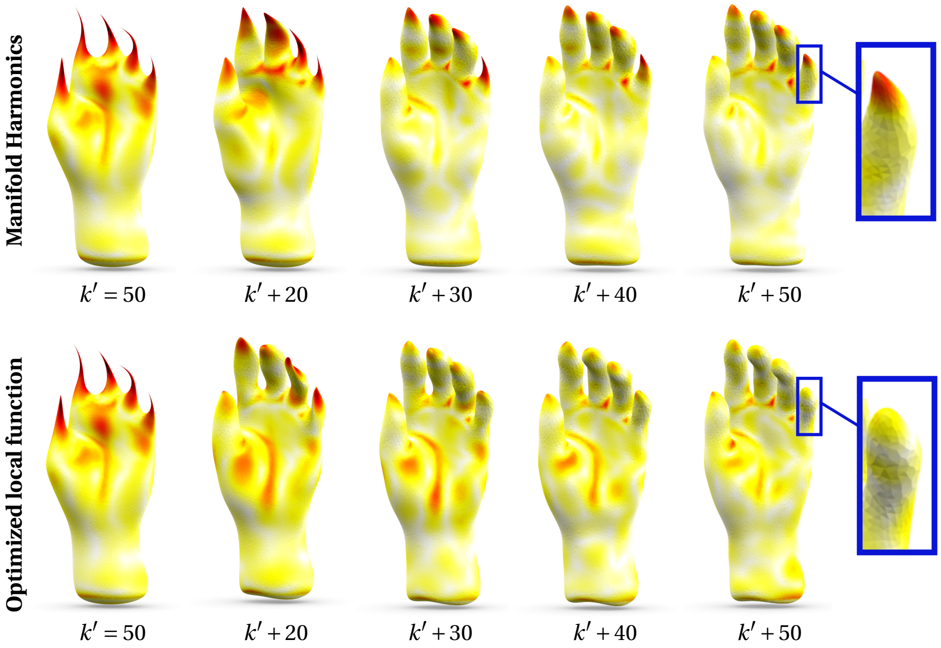


Fig. 5.45: *First row*: Surface reconstruction via (5.81) using the first k' to $k' + 50$ Laplacian eigenfunctions. *Second row*: Reconstruction via (5.82) with 10 optimized local functions per finger. Note the higher accuracy of the localized basis despite using the same number of MH. The heatmap encodes reconstruction error, growing from white to dark red.

clear as to each addition of 10 the improvement given by the optimized local functions is not so well localized, due to the non optimal solution that generates these functions. Indeed the results shown in Figure 5.45 are worse than those obtained with LMH and this alternative method is also not so efficient. For these reasons we prefer the *localized manifold harmonics*, which are also more look for a more theoretically sound.

Future directions

We introduced a new framework for spectral shape analysis and processing, allowing to perform operations which are localized to a given (possibly soft or disconnected) region of interest on the surface. Our framework is flexible, in that it can naturally enrich or fully replace the standard manifold harmonics in several tasks in graphics. We demonstrated its applicability in applications of geometry processing and shape correspondence, demonstrating a significant advantage if compared with the standard 'global' constructions based on the eigendecomposition of the Laplace operator. Despite the theoretical and empirical results provided within this paper, we feel that our study is still just 'scratching the surface' of a much broader area of research, with potentially extensive applications in geometry processing and graphics. Even though quite elusive at this stage, we foresee connections with results in *localization theory* (see, e.g., [47]), leading to a promising direction of research which we believe deserves a deeper exploration.

5.5 Conclusions

In this Chapter we propose two different methods to introduce localization in the spectral geometry processing. The first one, that can be summarized as WFT, is based on a classical signal processing tool, the windowed Fourier transform. We propose two versions of the WFT tool the isotropic and the anisotropic one. With WFT we introduce the use of a window which localizes the signal on the spatial domain, and then applies the standard global spectral analysis (the one with MH), to the localized signal. The localization with WFT is introduced on the spatial domain. The second method to introduce the localization in the spectral basis is LMH. Instead of analyze signals decomposing it through global basis, with LMH the decomposition is produced via a localized basis. In this case the localization is introduced in the spectral domain, with LMH indeed the signal remains the same but is the basis that defines the spectrum that is localized.

6

Localized non-spectral geometry processing

In this Chapter we move the attention from the spectral methods to some method related with the spatial geometry. Despite methods such as WFT and AWFT deal with the connection between geometric object in the $3D$ *space and their spectral representations, the methods presented in Chapter 5 mainly analyze points and surfaces from the spectral point of view. However, spectral analysis is not the unique kind of* local analysis *that can be adopted on surfaces. In this Chapter we introduce the* Discrete time evolution process *as an example of* $3D$ *spatial local analysis on surface.*

6.1 Need for spatial localization

The idea of localization is in some sense more intuitive in the non spectral case. If we define a spatial distance between points on a given surface $\mathscr{M}$ then for a point $x \in \mathscr{M}$ we can subdivide the surface in two regions, the neighborhood of x and the rest of the surface. This definition of a neighborhood of a point is something very standard in graph theory. The neighborhood of x can be seen as a localized context with respect to which the point x can be analyzed and described. Starting form an entire shape $\mathscr{M}$, is it clear how we could define different levels of localization around a point x that belongs to $\mathscr{M}$. An example of these levels of localization can be seen in Figure 6.1. In this figure for a pair of points highlighted with a small black spheres on $\mathscr{M}$ are proposed 5 different levels of localization depicted as red regions on the surface in an decreasing level of localization. On the first row analyzes a point on the top of a back thigh, while on the second row the point is taken from the neck of the Moose. As can be seen the first two levels on the left are very similar for both the points. In the level in the middle some difference starts to be represented. In the last two levels on the right the geometry of the neighborhoods is definitely different. In our work, *Discrete time evolution process*, we would like to describe and characterize the points on the surface encoding in an theoretically sound framework the information that is possible to see in Figure 6.1.

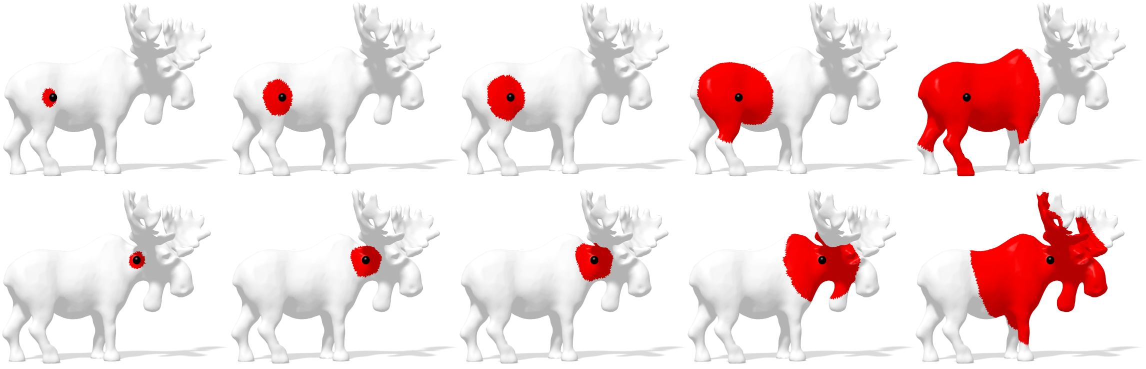


Fig. 6.1: Different levels of localization computed for two different points on the Moose shape. The points are highlighted with a small black spheres. In each row we show the neighborhoods of one of the two point in a decreasing level of localization from left to right.

6.2 Discrete time evolution process

The discrete time evolution process *is a framework for the shape analysis that starting from a generic pairwise relation between every pair of points on the surface, allows to obtain a more complete and informative* *description of each point with respect to different levels of localization of its neighborhood. In this spirit the* discrete time evolution process *can be view as a localized tool for geometry processing. For more details on the discrete time evolution process please refer to [103].*

6.2.1 Overview of the framework

We propose to encode the relation between points by exploiting an alternative evolution paradigm. Rather than considering a *continuous* time evolution we introduce a special operator that is applied iteratively on the surface. Our *process* operator is designed to explicitly integrate information across the shape by taking into account the relation of a given point to the rest of the surface. In particular, our operator is specified by a function that encodes the *direct* pairwise relations between surface points. Then, the iterative procedure allows our method to explore also *indirect* (or second-order) relations. This leads to a new *discrete-time* evolution scheme to represent the gradual change of the "context" of the each point. As an example when the relation function is defined by the geodesic distance, our process operator encodes the set of paths of gradually increasing lengths. Moreover, we effectively combine the contribution of each evolution state to obtain a final *score* that summarizes how a point is influenced by the rest of the shape after an infinite number of steps. Key to our approach is an observation that such multi-step computation can be done *exactly* and efficiently in practice by solving a single linear system of equations without requiring an approximation via a reduced spectral basis.

We show the benefits of this new framework by using it to derive a novel point signature. We build our process operator by using the geodesic distances as relation functions. Therefore, rather than considering only the shortest paths, which are known to be susceptible to noise, the iterative use of our operator captures the information about all paths of arbitrary lengths between each pair of points. Finally, we introduce a *multi-scale* strategy to capture information from both small and large neighborhoods, by controlling the length of the distance allowed in a single step. We demonstrate that our descriptor, which we call *Discrete time Evolution Process* descriptor (DEP), is highly discriminative and is more robust than other methods to several kinds of shape transformations such as non-isometric deformations and missing parts.

Our approach closely resembles methods based on diffusion geometry [5,18,35,50,140], especially in its use of an infinite number of paths to characterize points and their relations. More precisely our work is related to diffusion based methods in that it arises from an evolution process similarly to the HKS and WKS (already introduced in Chapter 3) which are respectively based on the evolution process of the heat and the motion of the particles on the surface. Our method exploits an alternative surface evolution paradigm and defines a new path-based multi-scale point descriptor, by capturing the paths of multiple lengths without the need to compute any spectral decomposition. Rather than focusing on the infinitesimal or differential characteristics of the shape (such as those defined by the Laplace-Beltrami operator), we argue for encoding the "integral" properties of points and their neighborhoods by considering the relations across all the other points on the shape, and by simulating a discrete-time evolution process. Nevertheless, by basing our descriptor directly on discrete time evolution and geodesic distances, rather than on the differential operator such as the Laplace-Beltrami operator, we are able to provide complementary information, with respect to existing diffusion-based signatures. Moreover, in contrast to these approaches, our descriptor can be computed *exactly*, without truncating an eigendecomposition. We demonstrate the effectiveness of our method in matching shapes across a wide range of challenging scenarios. For this, we provide results on benchmarks with increasing level of complexity and consider different matching strategies: i) based on a direct comparison of point descriptors, and ii) based on functional map [115] framework.

Our results outperform the state-of-the-art and show that the information contained in our DEP descriptor is alternative to the one captured by existing techniques. Our method is highly discriminative and it captures information that is alternative to the geometric attributes obtained by other traditional diffusion-based approaches. Even if our descriptor is built without employing any learning strategy our results are stable and robust even when the hypothesis of isometric transformation is violated.

6.2.2 Discrete time evolution process on manifolds

We define an iterative process that evolves on the manifold and that we observe at discrete and regular timestamps. Differently from the diffusion-based methods, such as HKS or WKS, our evolution process is not necessarily based on a differential equation that controls the process behaviour. Therefore, we do not need to know the evolution law that explains this process, but the process itself is fully derived from a generic pairwise relation function as described below.

Continuous shape

Let S be a smooth surface, and $\mathscr{F}(S,\mathbb{R})$ the set of real functions defined on S. We introduce a function d that represents a generic relation between each pair of surface points.

$$d : S \times S \longrightarrow \mathbb{R}$$

$$d : (x, y) \longmapsto d(x, y) \in \mathbb{R} \text{ s.t. } d(x, y) \geq 0 \ ,$$

Starting from the relation d we then define a process, that evolves in the discrete time setting and depends only on the underlying geometry of the surface. This process is governed by the relation d that represents how the each point x is "*influenced*" by every other point y on the surface.

We fix a finite time interval $\Delta t \in \mathbb{R}$, with $\Delta t > 0$, and divide the posi-

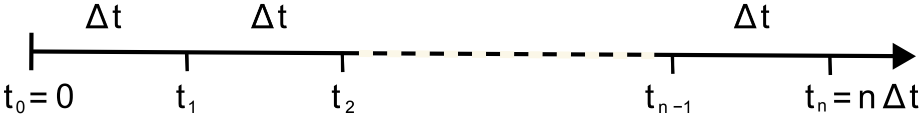


Fig. 6.2: Time discretization used for the discrete time evolution process.

tive real line in a discrete collection of instants $\{t_0 = 0, t_1, t_2, \ldots, t_{l-1}, t_l, \ldots\}$, where $t_l = l\Delta t$ as shown in Figure 6.2. Given an initial state represented by a real function $f_0 \in \mathscr{F}(S, \mathbb{R})$, we define the desired process as follows:

$$f_1(x) := \int_S d(x,y) f_0(y) \partial\mu(y), \tag{6.1}$$

where f_1 is the state after one time interval Δt, and $\partial\mu$ is the infinitesimal area element. We assume that the process is homogeneous and the evolution is the same at every time step. Thus, we can iterate this operation by obtaining the state after a generic number l of discrete intervals as:

$$f_l(x) := \int_S d(x,y) f_{l-1}(y) \partial\mu(y). \tag{6.2}$$

Using this relation, we introduce the *process operator* $\mathbf{P}$ that gives us the state of the process after the discrete interval of time. Therefore, we write:

$$\mathbf{P}(f_0) = f_1 = \int_S d(\cdot, y) f_0(y) \partial\mu(y) \tag{6.3}$$

and iterating on $l \in \mathbb{N}$ we obtain:

$$f_l = \mathbf{P}(f_{l-1}) = \mathbf{P}(\mathbf{P}(f_{l-2})) = \ldots = \underbrace{\mathbf{P}(\mathbf{P}(\cdots(\mathbf{P}(f_0))))}_{l-\text{times}} .$$

Finally, for every point x on S we define a scalar value that sums up the contributions of the process from every discrete time t_l with $l \in \{0, \ldots, +\infty\}$. Therefore, we introduce an *evolution process* score s at a point $x \in S$ as follows:

$$s(x) = \sum_{l=0}^{\infty} f_l(x) = f_0(x) + \sum_{l=1}^{\infty} \int_S d(x,y) f_{l-1}(y) \partial\mu(y). \tag{6.4}$$

Discrete shape

In the discrete setting we represent S by a triangular mesh $\mathscr{M}$ with N vertices $\mathcal{V} = \{v_i\}_{i=1}^{N}$. We divide the surface in barycells centered at every vertex v_i of the mesh, and denote by Ω_i their areas. Note that in this case the set $\mathscr{F}(S, \mathbb{R}) = \mathbb{R}^N$ and the function d corresponds to a matrix $\mathbf{D} \in \mathbb{R}^N \times \mathbb{R}^N$ where:

$$D_{i,j} = d(v_i, v_j) \in \mathbb{R} \text{ s.t. } d(v_i, v_j) \geq 0,\ \forall v_i,\ v_j \in \mathcal{V}\ .$$

Following the discussion above, we can redefine the same evolution process that we created in the continuous setting as follows. Given an initial state $f_0 \in \mathbb{R}^N$, our process is defined as

$$f_1(v_i) := \sum_{j=1}^{N} \Omega_j D_{i,j} f_0(v_j), \tag{6.5}$$

and f_1 is the state after one interval Δt. Here, the integral from equation 6.1 is replaced by the weighted sum according to the local areas. Then we can obtain the state after a generic number l of discrete time intervals as:

$$f_l(v_i) := \sum_{j=1}^{N} \Omega_j D_{i,j} f_{l-1}(v_j). \tag{6.6}$$

We build the diagonal matrix $\mathbf{A} = diag(\mathbf{A}_i)$, with the area $\mathbf{A}_i$ of the barycell centered in v_i. Then we can adopt a matrix notation to model the process. We denote by $\mathbf{P}$ the discrete *process operator* defined by the $N \times N$ real matrix $\mathbf{DA}$, more explicitly the element (i, j) of the matrix $\mathbf{P}$ is $\mathbf{P}(i, j) = \mathbf{A}_j d(v_i, v_j)$. Now we can write the discrete analogue of Eq. (6.3):

$$\mathbf{P} f_0 = f_1 = \mathbf{DA} f_0 = \sum_{j=1}^{N} \Omega_j D_{\cdot,j} f_0(v_j).$$

and iterating on $l \in \mathbb{N}$ we obtain:

$$f_l = \mathbf{P} f_{l-1} = \mathbf{P}(\mathbf{P}(f_{l-2})) = \ldots = \underbrace{\mathbf{P}(\mathbf{P}(\cdots(\mathbf{P} f_0)))}_{l-\text{times}} = \mathbf{P}^l f_0,$$

where $\mathbf{P}^l = (\mathbf{DA})^l$ is the process operator for l steps. Finally, we obtain our evolution process score via the discrete version of Eq. (6.4):

$$s(v_i) = \sum_{l=0}^{\infty} f_l(v_i) = f_0(v_i) + \sum_{l=1}^{\infty} \big(\sum_{j=1}^{N} \mathbf{P}^l(i,j) f_0(v_j) \big). \tag{6.7}$$

6.2.3 Analysis of Higher Order Relations

Here we discuss the meaning of the proposed evolution process in terms of higher order relations between points that belong to the surface. Note that while the process evolves our process operator $\mathbf{P}^l$ takes into account indirect links between the vertices. We consider these indirect links as higher order relations. More specifically, Equation (6.5) encodes the first-order relations of v_i and it can be rewritten as:

$$f_1(v_i) := \sum_{j=1}^{N} p_{i,j} f_0(v_j), \tag{6.8}$$

where $p_{i,j} := \mathbf{P}(i,j)$ represents how vertex v_i is "*influenced*" by v_j on the discretized shape. Now we evaluate the behaviour of the process in the higher order relations. Let σ denote a generic subset of $l+1$ vertices $\{v_{\sigma(0)}, \ldots, v_{\sigma(l)}\} \subseteq \mathcal{V}$, with possible repetitions. We can define the contribution w_σ of σ to the evolution process as:

$$\mathrm{w}_\sigma = \prod_{k=0}^{l-1} p_{\sigma(k),\sigma(k+1)} = \prod_{k=0}^{l-1} \Omega_{\sigma(k+1)} d(v_{\sigma(k)}, v_{\sigma(k+1)}), \tag{6.9}$$

Let $\mathbb{P}^l_{i,j}$ denote the collection of all the subsets $l+1$ vertices, starting with i ($\sigma(0) = i$) and ending with j ($\sigma(l) = j$). To account for all subsets in $\mathbb{P}^l_{i,j}$, and following standard linear algebra we compute:

$$\mathbf{P}^l(i,j) = \sum_{\sigma \in \mathbb{P}^l_{i,j}} \mathrm{w}_\sigma \tag{6.10}$$

Therefore, our process operator for l order $\mathbf{P}^l(i,j)$ represents how the v_i is "*influenced*" by all l-order relations between v_i and v_j.

Moreover, for every initial state f_0, we define the l-order evolution state at vertex v_i as:

$$f_l(v_i) = \sum_{j=1}^{N} \mathbf{P}^l(i,j) f_0(v_j), \tag{6.11}$$

Intuitively $f_l(v_i)$ encodes the quantity of the state f_0 "*absorbed*" in v_i from the l-order relations in the evolution process.

The *evolution process* score for each vertex of the mesh is defined summing over l as: $s(v_i) = \sum_{l=0}^{\infty} f_l(v_i)$, so it can be obtained as in Equation 6.7. Now, to generalize the computation of this score we introduce the score operator $\mathbf{S}$ as the geometric series of matrix $\mathbf{P}$:

$$\mathbf{S} = \sum_{l=0}^{\infty} \mathbf{P}^l. \tag{6.12}$$

Note, however, that since $\mathbf{S}$ increases exponentially with l, the infinite sum may diverge and so $\mathbf{S}$ may not be well-defined. To overcome this problem we employ a simple generating function strategy. Generating function regularization [56] is used to assign a consistent value for the sum of a possibly divergent series. There are different forms of generating functions [10] for some examples. To this end, we define the *regularized* score operator as:

$$\check{\mathbf{S}} = \sum_{l=0}^{\infty} r^l \mathbf{P}^l. \tag{6.13}$$

where r is a scalar regularization parameter. In order to ensure the convergence, we choose r so that $|r| < \frac{1}{\rho(\mathbf{P})}$, where $\rho(\mathbf{P})$ is the spectral radius of $\mathbf{P}$. In order to give a more formal argument, the value of r (used in the generating function) can be determined by relying on linear algebra [64].

Lets consider $\{\lambda_0, \dots, \lambda_{n-1}\}$ eigenvalues of the matrix $\mathbf{P}$, drawing from linear Algebra we can define the spectral radius $\rho(\mathbf{P})$ as:

$$\rho(\mathbf{P}) = \max_{\lambda_i \in \{\lambda_0, \dots, \lambda_{n-1}\}} \Big(|\lambda_i|\Big).$$

For the theory of convergence of the geometric series of matrices we have also that:

$$\lim_{l \to \infty} \mathbf{P}^l = 0 \iff \rho(\mathbf{P}) < 1 \iff \sum_{l=0}^{\infty} \mathbf{P}^l = (\mathbf{I} - \mathbf{P})^{-1}.$$

Furthermore another theoretical result indicated as Gelfand's formula states that for every matrix norm we have:

$$\rho(\mathbf{P}) = \lim_{k \longrightarrow \infty} ||\mathbf{P}^k||^{\frac{1}{k}}.$$

This formula leads directly to an upper bound for the spectral radius of the product of two matrices which commutes, given by the product of the spectral radii of the two matrices, that is for each pair of matrices $\mathbf{P}$ and $\mathbf{B}$:

$$\rho(\mathbf{PB}) \leq \rho(\mathbf{P})\rho(\mathbf{B}).$$

Starting from the definition of $\check{s}(i)$, and from the following trivial consideration

$$r^l \mathbf{P}^l = \left(r^l \mathbf{I}\right) \mathbf{P}^l = [(r\mathbf{I})\, \mathbf{P}]^l,$$

we can use Gelfand's formula on $r\mathbf{I}$ and $\mathbf{P}$ and thus obtain:

$$\rho\Big((r\mathbf{I})\, \mathbf{P}\Big) \leq \rho(r\mathbf{I})\rho(\mathbf{P}) = r\rho(\mathbf{P}), \tag{6.14}$$

For the property of the spectral radius: $\lim_{l \to \infty} (r\mathbf{P})^l = 0 \iff \rho(r\mathbf{P}) < 1$. Thus if we choose r such as $0 < r < \frac{1}{\rho(\mathbf{P})}$, then we have:

$$0 < \rho(r\mathbf{P}) = \rho\Big((r\mathbf{I})\, \mathbf{P}\Big) \leq \rho(r\mathbf{I})\rho(\mathbf{P}) = r\rho(\mathbf{P}) < \frac{1}{\rho(\mathbf{P})}\rho(\mathbf{P}) = 1\ ,$$

that implies $\rho(r\mathbf{P}) < 1$, and so that:

$$\check{\mathbf{S}} = \sum_{l=0}^{\infty} (r\mathbf{P})^l = (I - r\mathbf{P})^{-1}.$$

This choice of r allows us to have convergence in the sum that defines $\check{s}(i)$. From an algebraic view, $\check{\mathbf{S}}$ can be efficiently computed by using the convergence property of the geometric power series of a matrix [64]:

$$\check{\mathbf{S}} = (\mathbf{I} - r\mathbf{P})^{-1} \tag{6.15}$$

Matrix $\check{\mathbf{S}}$ encodes the information about the geometry between our set of vertices, and the chosen relation function d. Finally, we can obtain the evolution process scores for each vertex simply as:

$$\check{s}(v_i) = [(\check{\mathbf{S}} f_0)](i), \tag{6.16}$$

which can equivalently be computed by solving the linear system: $(\mathbf{I} - r\mathbf{P})(\check{s}) = f_0$. This interpretation makes it clear that, for every vertex v_i the computed score $\check{s}(i)$ is obtained by summing the contributions of all the relations starting at v_i, and evolving along the surface under the conditions imposed by the process operator $\mathbf{P}$, for all time scales, going to infinity. Note that from the pairwise relations encoded by $\mathbf{D}$ our evolution process allows to obtain and incorporate higher-order information. All the components involved in our framework are highlighted. In the next Subsection 6.2.5 we will give a deeper analysis of these components.

6.2.4 Connection to Continuous process

Our discrete time evolution process is defined by

$$f_{l+1} = \mathbf{P} f_l \, .$$

Let us now suppose that exists a matrix $\mathbf{B}$ such that:

$$\mathbf{P} = e^{\Delta t \mathbf{B}} \, .$$

This assumption is far from trivial, and it is possible only for a particular set of **P**. If we satisfy this condition, and focusing on the limit case of $\Delta t \rightarrow 0$, we can consider the continuous process associated to the following partial differential equation:

$$\frac{\partial f}{\partial t} = \mathbf{B} f \ .$$

It is well-known that for every initial state f_0 the state of this continuous process at time t is defined as:

$$f_t = e^{t\mathbf{B}} f_0 \ .$$

So we have that for every discrete time interval Δt and for every integer l:

$$f_{l\Delta t} = e^{l\Delta t\mathbf{B}} f_0 = \mathbf{P}^l f_0 \ ,$$

which corresponds to our discrete time evolution process. In the same spirit we can obtain a connection between the integration over all times of the continuous process and our evolution process score. In the continuous notation we can compute the following equation:

$$\int_0^\infty f_\tau d\tau = \int_0^\infty e^{\tau\mathbf{B}} f_0 d\tau = -\mathbf{B}^{-1} f_0 \ .$$

Now if we fix a time interval Δt as we did in the previous paragraph, we can rewrite this integration at discrete times as:

$$\sum_{l=0}^{\infty} \mathbf{P}^l f_0 \Delta t = (I - \mathbf{P})^{-1} f_0 \Delta t \ .$$

Assuming $\mathbf{P} = e^{\Delta t\mathbf{B}}$ as above, for $\Delta t \rightarrow 0$ we have:

$$(I - \mathbf{P})^{-1} f_0 \Delta t = -(\Delta t\mathbf{B})^{-1} f_0 \Delta t = -\mathbf{B}^{-1} f_0 \ .$$

Therefore, if we consider our evolution process score $\check{s} = (I - \mathbf{P})^{-1} f_0$ and forgetting Δt (as multiplicative constant value) $\check{s}$ corresponds to the integration over all times of the continuous process.

This highlights how the continuous process is related to our discrete time evolution process, and also how the score $\check{s}$ is related to the continuous process in this specific context.
Clearly this connection depends on a strong assumption, and holds only for very specific relation functions d. One of the main goals of our framework, however is to enable the use of a generic pair-wise relation without limiting its choice in order to meet some conditions. For this reason, we encode this relation via a discrete process, which allows us to obtain a score, even when the evolution is not governed by a continuous-time, diffusion-like procedure.

6.2.5 Proposed Descriptor

We investigate how the proposed evolution process described in previous paragraph can be exploited to define a new class of 3D point descriptors. The choice of the components of the evolution process is crucial to identify the encoded information. In particular, the main components are i) the relation function, ii) the regularization parameter, and iii) the starting state.

The relation function

The most important parameter is the relation function **D** that in principle can be defined by any positive two variables function. In practice, the characteristics of the chosen relation function determine the kind of information that is spread across the shape by the evolution process and therefore the effectiveness of the derived descriptor highly depends on this choice. In this context the most natural options are distance functions, kernels or generic (dis-)similarity measures. In our work, we build the process operator **P** using the geodesic distance $\mathcal{G}(v_i, v_j)$, i.e., the length of the shortest path on $\mathcal{M}$ between vertices v_i and v_j. We motivate this choice by the fact that geodesic distances are, by definition invariant under isometric transformations, and can be used to capture the geometry of the shape effectively. Moreover, as we demonstrate below, our discrete time evolution procedure that allows to incorporate information across an infinite set of paths helps to gain both informativeness and robustness against non-isometric shape changes. More specifically, we define the matrix of vertex relations $\mathbf{D}$ as:

$$\mathbf{D}(i,j) = 1 - \hat{\mathscr{G}}(v_i, v_j),$$

where $\hat{\mathscr{G}}(v_i, v_j) = \frac{\mathscr{G}(v_i,v_j)}{diam(\mathscr{M})}$ is the *normalized* geodesic distance and we denote with $diam(\mathscr{M})$ the diameter of $\mathscr{M}$, that is defined as the maximum of the geodesic distances between every pair of vertices on the surface. Therefore $\mathbf{D}(i,j) \in [0,1]$, $\mathbf{D}(i,i) = 1, \forall i$, and the process operator writes $\mathbf{P} = \mathbf{DA}$. This choice implies that each vertex absorbs more information from its neighborhood, decreasing gradually the influence of vertices that are further away from it. Moreover, the integration of the geodesic distance in our process operator leads to higher order relation $\mathbf{P}^l$ that is analogous to the $l-$order paths between pairs of points. Therefore, the interpretation of the evolution process is more intuitive as the encoding of paths at multiple lengths.

The regularization parameter

The second parameter that is important to fix is the regularization parameter. As mentioned in the previous section, r must be smaller than $\frac{1}{\rho(\mathbf{P})}$ to ensure the convergence of the regularized score operator defined in Equation 6.13. We keep this choice as a free value $r = \frac{c}{\rho(\mathbf{P})}$ parametrized by $c \in (0,1)$. It is worth noting that c determines the speed of convergence: values close to 0 means fast convergence and vice versa. In practice, when the convergence is fast (slow) the influence of the highest order relations is reduced (preserved). For instance if c is close to 1 the longest paths are just as relevant for the construction of the descriptor as the shorter ones.

Now we provide a discussion of the behaviour for the limit case of $r \to 0$.

In our method we compute our score v solving the following linear system:

$$(I - r\mathbf{P})v = e\ .$$

If we consider the exponential of the matrix $-r\mathbf{P}$ we have:

$$e^{-r\mathbf{P}} = \sum_{k=0}^{\infty} \frac{(-r)^k(\mathbf{P})^k}{k!} = I - r\mathbf{P} + \frac{r^2\mathbf{P}^2}{2} + \ldots$$

Now in the limit case of $r \to 0$ we can approximate $e^{-r\mathbf{P}}$ as

$$e^{-r\mathbf{P}} \approx I - r\mathbf{P}.$$

So we can write:

$$(I - r\mathbf{P})v \approx e^{-r\mathbf{P}}v = e$$

and multiplying left and right sides for $e^{r\mathbf{P}}$ we obtain:

$$v = e^{r\mathbf{P}}e\ .$$

From this point of view v is the state after a time r of a process governed by $e^{r\mathbf{P}}$. Generalizing this process for a initial state f at time 0 we obtain the state at time r as:

$$f_r = e^{r\mathbf{P}}\ .f$$

Using the previous approximation $(I - r\mathbf{P})v \approx e^{-r\mathbf{P}}v = e$ we can recover the partial differential equation that defines this process.

$$f_r = e^{r\mathbf{P}}f \approx (I + r\mathbf{P})f = f + r\mathbf{P}f$$

from which we have:

$$\frac{f_r - f}{r} = \mathbf{P}f, \text{ and for } r \to 0, \text{ we get } \frac{\partial f}{\partial t} = \mathbf{P}f\ .$$

So we can conclude that in the limit of $r \to 0$ our score can be computed as the first step of a different process for a time equal to r. This is not very useful for our framework because it is only a limit case and the connection with our method is given only for a infinitesimal time interval.

The starting state

A further important parameter to settle is the starting state f_0. The choice of this state also plays an important role in controlling the kind of information that is encoded. The options can be different and related to the specific application at hand. In our work we are interested in evaluating how the evolution process itself is able to encode the geometric information. Therefore we would like to keep the contribution of the starting state neutral. To this purpose, we choose a constant distribution on the surface as the initial state: $f_0 = \mathbf{e}$, i.e., the constant function, encoded with a vector with all the entries equal to 1.

Multi-scale Approach

Finally, in order to construct a multi-scale descriptor we can consider a *family* of weighted operators $\mathbf{P}_\delta$, parameterized by a scalar $\delta \in [0,1]$. We define the new matrix of vertex relations $\mathbf{D}_\delta$ as:

$$\mathbf{D}_\delta(i,j) = 1 - \hat{\mathscr{G}}_\delta(v_i, v_j),$$

where $\hat{\mathscr{G}}_\delta(v_i, v_j) = \hat{\mathscr{G}}(v_i, v_j)$ if $\hat{\mathscr{G}}(v_i, v_j) \leq \delta$ and 1 otherwise. Thus, the new process operator becomes $\mathbf{P}_\delta = \mathbf{D}_\delta \mathbf{A}$ by only considering geodesic balls of radius δ, which implies that in a single discrete time step, the relation is limited to points at distance δ. Intuitively, for small values of δ our matrix $\mathbf{P}_\delta$ makes a vertex dependent on a small neighborhood by capturing more local properties of the shape, while for larger values of δ, $\mathbf{P}_\delta$ exploits more global structures of the shape. This way, the parameter δ can be interpreted as *the speed* at which information is propagated across the shape in our discrete-time evolution process. Note that this is somewhat in contrast with diffusion-based methods, where all changes are completely global, since, e.g., the classical heat equation implies that heat propagates at infinite speed, which, in particular, is not compatible with the special theory of relativity [45].

ALGORITHM 2: Computation of DEP Descriptors

```
Input: $\mathscr{M}$ the mesh, $\{\delta_q\}_{q=1}^{Q}$ s.t $\delta_q \in [0,1]$ .
Output: $DEP$ the matrix of descriptors.
for $i = 1 : n$ do
    for $j = 1 : n$ do
        Compute $\mathbf{A}$, diagonal matrix, with $\mathbf{A}(i,i) = \mathbf{A}_i$ area of the barycell centered in $v_i$;
        Compute $\hat{\mathscr{G}}(v_i, v_j)$, the normalized geodesic distance between $v_i$ and $v_j$;
        $\hat{\mathscr{G}}(v_i, v_j) = \frac{\mathscr{G}(v_i,v_j)}{diam(\mathscr{M})}$;
    end
end
for $q = 1 : Q$ do
    if $\hat{\mathscr{G}}(v_i, v_j) \leq \delta_q$ then
        $\hat{\mathscr{G}}_{\delta_q}(v_i, v_j) = \hat{\mathscr{G}}(v_i, v_j)$;
    else
        $\hat{\mathscr{G}}_{\delta_q}(v_i, v_j) = 1$;
    end
    $\mathbf{D}_{\delta_q}(i,j) = 1 - \hat{\mathscr{G}}_{\delta_q}(v_i, v_j), \forall i, j \in \{1, \ldots, n\}$;
    Compute $\mathbf{P}_{\delta_q} = \mathbf{D}_{\delta_q}\mathbf{A}$;
    Compute $r_{\delta_q} = \frac{c}{\rho(\mathbf{P}_{\delta_q})}$, with $c \in (0,1)$;
    Solve the linear system $(\mathbf{I} - r_{\delta_q}\mathbf{P}_{\delta_q})\mathbf{v} = \mathbf{e}$;
    $\check{s}_q = \mathbf{v}$;
end
matrix $DEP = [\check{s}_1, \ldots, \check{s}_Q]$, s.t. the $i$th row, encodes the Discrete time Evolution Process descriptor for the
  vertex $i$.
```

Discrete time Evolution Process descriptor

Once the main components of the evolution process are fixed and the multi-scale paradigm is defined, we are ready to propose a new shape descriptor. We fix a set of Q scale values $\{\delta_1, \ldots, \delta_Q\}$ such that $\delta \in [0,1]$. For each choice of δ we construct an operator $\mathbf{P}_\delta$, and compute the score at scale δ. For this we solve the linear system:

$$(\mathbf{I} - r_\delta \mathbf{P}_\delta)\mathbf{v} = \mathbf{e}$$

and let the score vector at scale δ be $\check{s}_\delta = \mathbf{v}$. Here, as above, the score of vertex i equals $\check{s}_\delta(i)$. This way we create our Discrete time Evolution Process (DEP) descriptor, by assembling a vector of Q values to each vertex i :

$$DEP(i) = \left[\check{s}_1(i), \ldots, \check{s}_Q(i)\right].$$

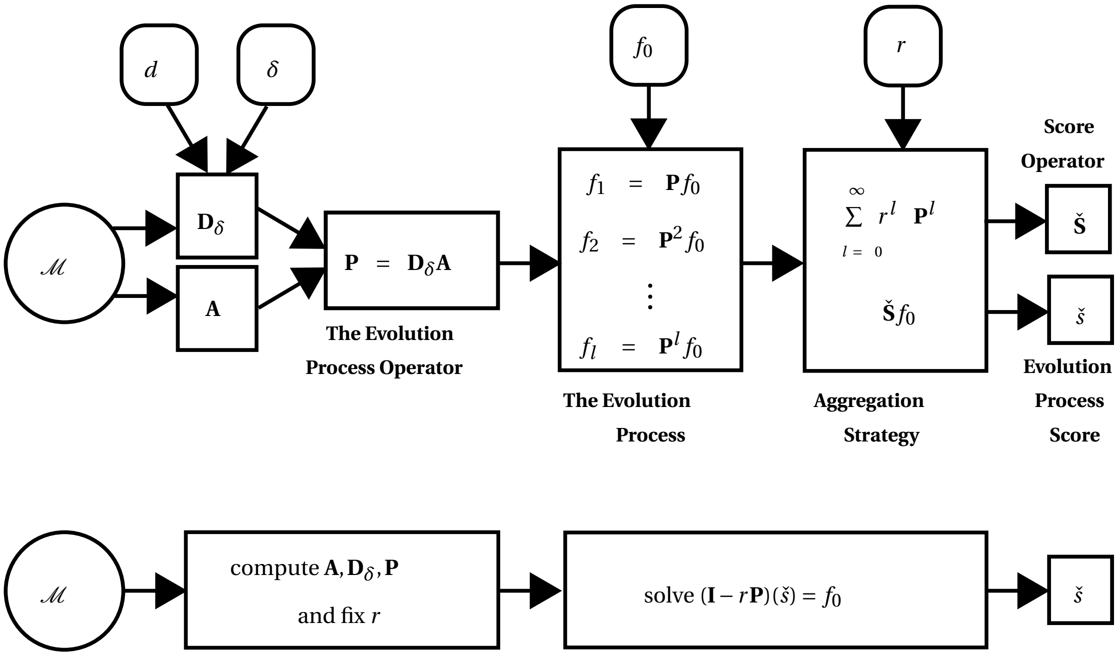


Fig. 6.3: A visualization of the entire discrete evolution process pipeline. The first row shows the pipeline that produces the discrete time evolution process. Starting with a triangle mesh $\mathscr{M}$ we introduce a relation function d and a scale parameter δ. The relation matrix $\mathbf{D}_\delta$ and the area elements matrix $\mathbf{A}$ are computed to obtain the evolution process operator $\mathbf{P}$. Then, fixing an initial state f_0 we run our discrete time evolution process by highlighting the involved discrete states. Finally, once a regularization parameter r is selected, we employ our aggregation strategy to obtain the score operator $\check{\mathbf{S}}$ and the evolution process score $\check{s}$. The second row shows how the score $\check{s}$ can be computed in practice, by solving a linear system.

In other words, for every vertex i we obtain a vector that represents in each of its dimensions the sum of the discrete time evolution process at vertex i, where the process performs steps of fixed maximum length. In Figure 6.3 we represent the entire pipeline to compute our evolution process score $\check{s}$ starting from the pairwise relations encoded by d and a set of parameters δ, f_0 and r. The first row explains the theoretical interpretation of our framework, while second row illustrates how it is computed efficiently in practice. For completeness, we summarize the construction of the Discrete time Evolution Process descriptor in Algorithm 2. In other

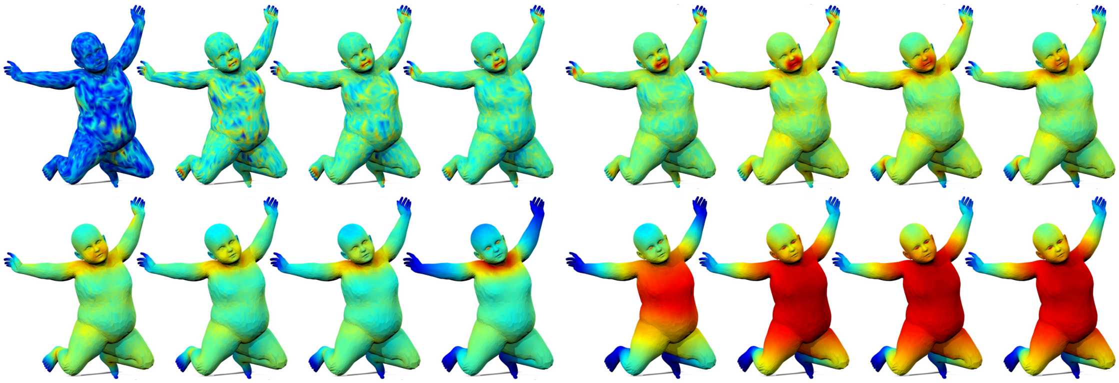


Fig. 6.4: Scores for multi-scale on a KIDS's shape. The scores at every vertex is plotted as a map on the shape, the values of δ are $\left[\frac{1}{100}, \frac{1}{50}, \frac{1}{40}, \frac{3}{100}, \frac{1}{25}, \frac{1}{20}, \frac{7}{100}, \frac{3}{40}, \frac{2}{25}, \frac{9}{100}, \frac{1}{10}, \frac{3}{20}, \frac{7}{20}, \frac{11}{20}, \frac{8}{10}, 1\right]$.

words, for every vertex i we obtain a vector that represents in each of its dimensions the sum of every time scales evolution process in that vertex given by a process that in a discrete time interval performs steps of fixed maximum length. Figure 6.4 shows the descriptors obtained using 16 different values of δ on a human shape (from the KIDS dataset [131]).

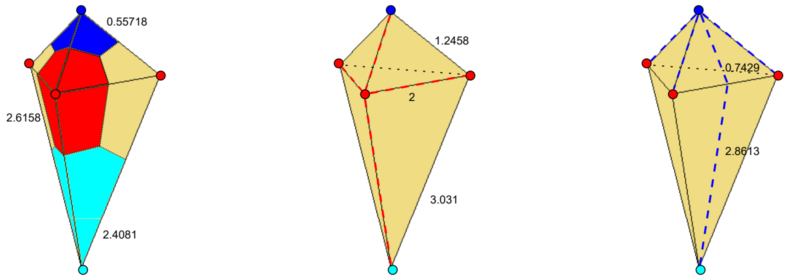
Contribution of higher order relations

In this section we analyze the importance of the higher order relations. Although the initial information (i.e., the geodesic distances) is already informative, it is not sufficient to compete with methods that are able to encode multiple paths between a pair of points such as spectral-based methods. Therefore, to disambiguate between points having accidentally the same geodesic distance we allow our evolution process to consider higher order paths. We compare the performance of using just a 1-step score with the infinite path descriptor on the toy-example in Figure 6.5. This shape is composed of the lateral surfaces of two tetrahedra having the same equilateral triangular bases. We can distinguish three types of points on this shape, the blue the red and the cyan. The three red points on the bases are considered the same because, thanks to the intrinsic symmetry of this shape, they only differ by a rigid rotation. The barycells associated to each vertex are shown with their value in the figure on the left. The geodesic distances are plotted with dashed lines in the other two figures. Table 6.5 shows the scores computed for the 1-step and for **Še** on all points in the surface showed in Figure 6.5. We can see that the 1-step score confuses the blue point with the three red points. Conversely, when we let the evolution process exploit paths with multiple steps, our **Še** scores is able to correctly distinguishing between different types of point. We can also obtain a theoretical bound on the robustness of our approach against shape perturbations. Indeed if we have a matrix of the linear system $M = (\mathbf{I} - r_\delta \mathbf{P}_\delta)$, and a perturbed matrix M' we can give the following upper bound [55]:

$$||\check{s}'_\delta - \check{s}_\delta|| \leq \mathcal{O}(\kappa(M)||M' - M||)||\check{s}_\delta||, \tag{6.17}$$

where $\check{s}_\delta$ and $\check{s}'_\delta$ are the solutions of the linear problem and the perturbed problem respectively, and $\kappa(M)$ is the condition number of the matrix. This bound depends on the norm of the perturbation $||M' - M||$. In practice we have observed that the condition number is well-behaved. We performed an experimental evaluation on different perturbed shapes from FAUST dataset. On average we obtained $\kappa(M) = 1.12$, which supports our claim that the solutions with our method are stable. Therefore, for small perturbations like in the case of near-isometric shapes our method can ensure a reliable solution.

In order to emphasize the robustness of our method we analyze the matching experiment reported in Figure 6.6. The same Duck shape is shown on two meshes with different density. It is clear that the geodesic distance information contained in the operator matrix **P** depends on the mesh structure that is used to discretize the mesh. But thanks to the integration at all the surface the DEP descriptors are more informative even in the presence of changes of the mesh structure. In Figure 6.6, we illustrate this behaviour. We show the 1-step descriptors and our descrip-



POINT COLOR	1-step	DEP
Blue	0.5197	1.0643
Red 1	0.5197	1.0483
Red 2	0.5197	1.0483
Red 3	0.5197	1.0483
Cyan	0.3373	0.5912

Fig. 6.5: Toy-example. On the left on a simple shape three type of points are highlighted: blue, red, and cyan. On the left the barycells associated to each vertex. Dashed lines are the geodesic paths: the four geodesic paths from one of the red points in the middle left, and the four geodesic paths from the blue point on the middle right. In the table on the right of the Figure are reported the values of descriptors at points in the toy example.

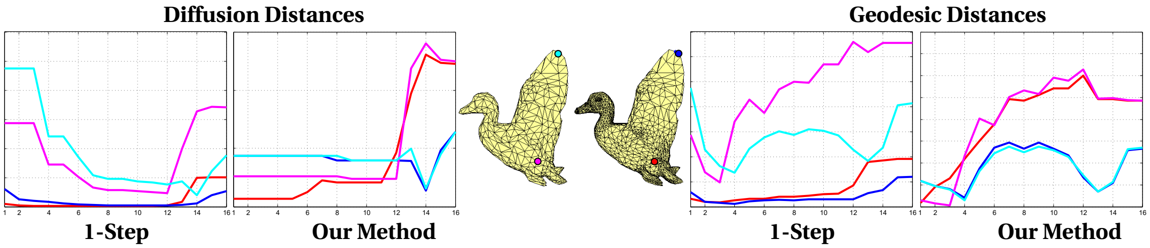


Fig. 6.6: Comparison between 1-step approach and our method using diffusion (left) and geodesic distances (right) for two points on the Duck shape with two different meshes with 752 and 2497 vertices respectively.

tors for some selected points on the two Duck shapes. To emphasize the contribution of the evolution process we evaluate two different relation functions. More precisely, other than using the geodesic distance we introduce also the diffusion distance [35] to build the process operator. Then a descriptor is defined as described in Section 6.2.5 (in practice the geodesic distance is simply substituted by the diffusion distance in the construction of the matrix **D**). Descriptors coming from diffusion distance are shown on the left, while those defined by the geodesic distance are shown on the right. In the 1-step case we can see how all the descriptors are very similar if they come from the same mesh independently from the kind of point that they represent, and also that the light blue and the red one are very close despite representing different points. Differently, when descriptors are defined with our new diffusion process the matching is correct. In particular, with our approach the performance is independent from the change of the mesh. In regions with similar density (ends of the wings) and in those with different density (belly) the proposed approach provided reliable results. Note that this behaviour is observed for both the chosen relation functions. This suggests that the contribution to obtain the correct matching is given by the proposed evolution process when higher order relations are considered.

6.2.6 Relation to Heat Kernel Signature

As mentioned in Subsection 6.2.1 our approach is related to existing techniques based on diffusion process and in particular with the Heat Kernel Signature. Moreover, in addition to the relation between our discrete time process and continuous diffusion described in Subsection 6.2.4, another informative connection can be obtained by considering the relation between the diffusion distance matrix [35] and the Heat Kernel Signature [140]. Namely, as pointed out by Sun et al. [140], for a fixed time parameter t:

$$\mathrm{HKS}_t(x) = \frac{1}{\mathbf{A}_{\mathcal{M}}} \int_{\mathcal{M}} d_t^2(x,y)\,dy - \frac{H_{\mathcal{M}}(t)}{\mathbf{A}_{\mathcal{M}}} + \frac{2}{\mathbf{A}_{\mathcal{M}}}.$$

Note that the last two terms do not depend on the vertex, and therefore, will not influence distances between descriptors, when comparing vertices on the same shape. In other words, the Heat Kernel Signature is closely related to the eccentricity of the squared *diffusion* distance. This means that for an appropriate discretization we can obtain:

$$\mathrm{HKS}_t = \mathbf{D}_f \mathbf{A}\mathbf{e},$$

using a matrix $\mathbf{D}_f$ properly constructed in the following way:

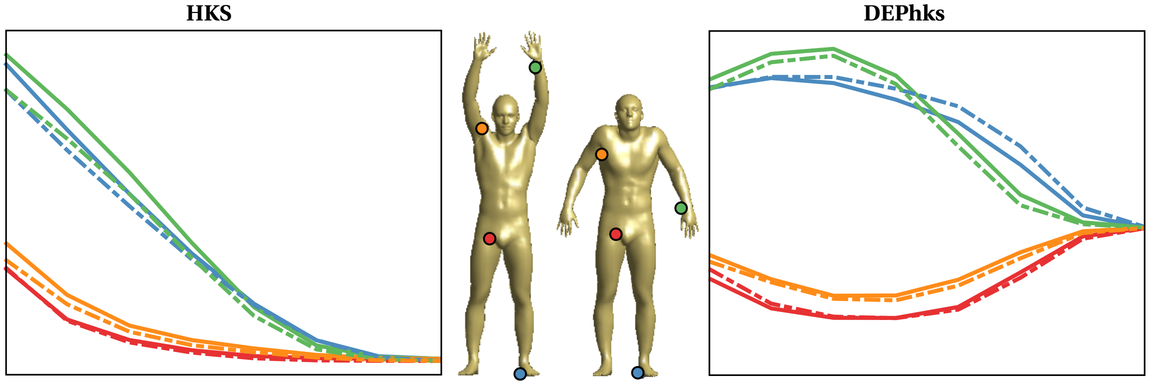


Fig. 6.7: Comparison between 1-step approach (that is equivalent to HKS) and our discrete time evolution version up to infinite (DEPhks). In the middle two human shapes from FAUST dataset with four vertices highlighted in coloured balls. On the sides the 16-dimensional descriptors: HKS descriptors (left) and DEPhks descriptors (right). Continuous lines refer to the shape on the left, while dashed lines represent the shape on the right.

$$\mathbf{D}_f(x,y) = \frac{1}{\mathbf{A}_{\mathcal{M}}} d_t^2(x,y) + \frac{1}{\mathbf{A}_y N}\left(\frac{2}{\mathbf{A}_{\mathcal{M}}} - \frac{H_{\mathcal{M}}(t)}{\mathbf{A}_{\mathcal{M}}}\right),$$

where N is the number of vertices in the surface discretization, and $\mathbf{A}_{\mathcal{M}} = \sum_{\forall j \in \mathcal{V}} \mathbf{A}_y$ is the total area of the surface. This shows how the HKS can be constructed marginalizing the rows of a matrix which represents specific relations between pairs of points, as proposed in our framework. The matrix $\mathbf{D}_f$ contains the pairwise relations between points on the surface encoded by the squared *diffusion* distance d_t^2. According to our framework, this can be considered as special relation function of a 1-step evolution process defined by $\mathbf{Pe}$, with $\mathbf{P} = \mathbf{D}_f\mathbf{A}$, which demonstrates a direct link between our approach and the HKS. Moreover, even though the diffusion distance is itself obtained by considering an infinite set of paths, by incorporating it into our framework and considering a discrete evolution process, we can obtain more reliable and stable connections between surface points, which further highlights the utility of our framework.

Figure 6.7 shows the improvement of our framework to the HKS descriptors. We select 16 time scales and compute the respective 16 matrices $\mathbf{D}_f$. A 16-dimensional HKS descriptor is computed for every selected vertex as the 1-step on the matrix $\mathbf{D}_f\mathbf{A}$ (Figure 6.7 left). Then, on the same matrices we compute the descriptors DEPhks related to our discrete time evolution process framework (controlled by the relation function $\mathbf{D}_f$ rather than the geodesic distances, Figure 6.7 right). As can be seen, the 1-step descriptors on different points are not very discriminative. In particular the blue point is confused with the green point. Conversely, with our method the behaviour of DEPhks descriptors is more coherent and the matching is correct.

6.3 Results

We have performed a wide range of experiments on several datasets to demonstrate the utility of our DEP descriptor in two application scenarios: i) point-to-point matching using nearest neighbor search in descriptor space, ii) incorporating our approach into the functional maps framework. In functional maps we also evaluate how our score operator could improve the standard performances. Furthermore from these results we analyze qualitatively the performance of our descriptor in comparison with the widely used HKS and WKS. Finally we explore the behaviour of our descriptor in different settings and varying the choice of parameters.

Point-to-Point Matching

In order to evaluate the ability of our DEP descriptor to associate corresponding points of different shapes we consider the following data sources (for more details please refer to Section 2.5 in Chapter 2): FAUST [13] a human shapes dataset with available ground truth point-wise correspondence. We use the whole FAUST dataset (100 shapes), along with additional shapes that have been edited by adding different types of noise. In addition we perform a test also using the more noisy real scans of humans. CAESAR another human shapes dataset [121]. We use a random selected subset of 21 shapes from the *CAESAR-fitted meshes* collection in which a template is fitted. For every shape we have around 6k vertices with 1:1 ground-truth correspondence. KIDS [131] a human dataset in which we find two different shape classes (*kid* and *fat kid*) in 16 different poses. In our test we uniformly down sample to approximately 6k vertices maintaining the 1:1 ground-truth correspondence and we can not guarantee to maintain the same connectivity for all the meshes. MISC dataset is composed of pairs of highly non-isometric shapes such as a horse and an elephant. Manually generated ground truth point-wise correspondences are available for a dense subset of points in this dataset. See e.g., Figure 6.8 for examples of shapes from this dataset.

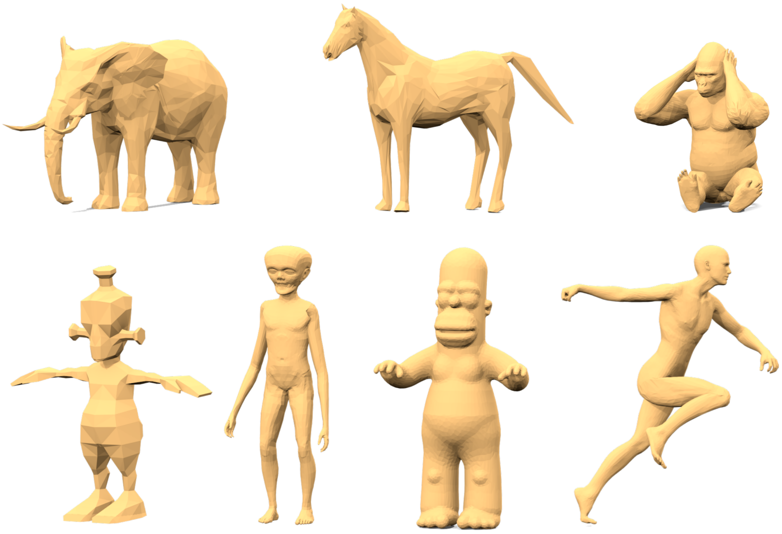


Fig. 6.8: Examples of shape from MISC dataset.

SHREC'11 *Partial* class benchmark [17]. The class includes one full human shape (i.e., the *null* shape) and 5 versions of its simulated transformations of pose deformation with strong partiality. These meshes were resampled to around 6K vertices. For every dataset we take all possible pairs of shapes. Then we randomly select 1K vertices on one shape and we compute the closest vertex in the descriptor space among all the vertices from the second shape. For the MISC dataset, on the first shape we use all the vertices for which the correct correspondence is given. We evaluated the performance of our descriptor using the *receiver operator characteristic* (ROC), already introduced in the previous Chapter. We compared our method with the following descriptors:

- HKS [140] with 100 dimension.
- WKS [5] with 100 dimension.
- SHOT [144] with 320 dimension.
- AWFT [104] with 100 dimension.

We use the code and settings available on-line. In order to be coherent with other methods our proposed descriptor is estimated at 100 scales (DEP100). Note that HKS and WKS are the closest methods to ours as described in Subsection 6.2.6. SHOT [144] is a local descriptor that encodes very different information with respect to our method. AWFT [104], to which is dedicated Section 5.3 of this thesis, also adopts a very different approach to encode local information and it represents the state of the art for point-to-point matching without the use of a learning procedure.
In Figure 6.9 we evaluate the performance of our DEP descriptor, in comparison with the other methods. We perform three different tests by varying the selected shapes: i) FAUST with all pairs from the entire dataset of 100 shapes (Figure 6.9 left), ii) FAUST-intra with only the pairs belonging to the same subject (Figure 6.9 middle), and iii)FAUST-extra with only pairs from two different subjects (Figure 6.9 right). Our method outperforms the competitors in all these settings. The greater improvement is achieved in the FAUST extra where strong non-isometric deformations are observed. Therefore, we claim that our descriptor is particularly effective in the case of change of subjects for which the isometric relation is

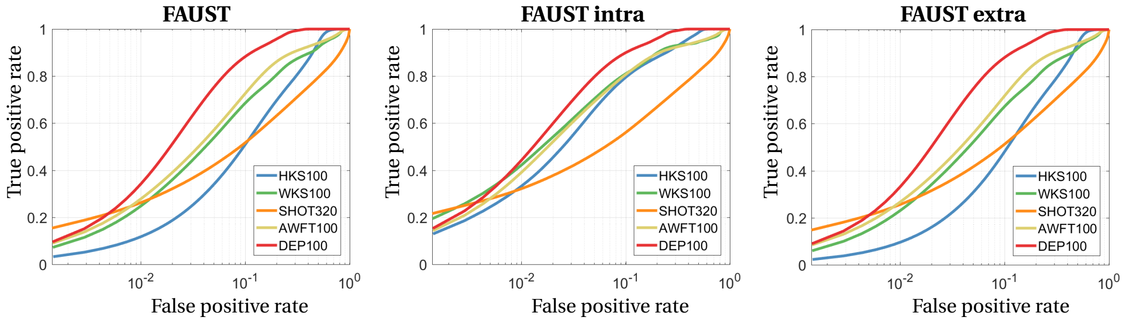


Fig. 6.9: Performance evaluation on the pairs FAUST dataset, using all 100 shapes (left), allowing only matching between shapes of the same subject (middle) and only between shapes from different subject (right). Comparison with 4 different descriptors. Next to the descriptor name, we show its dimensionality.

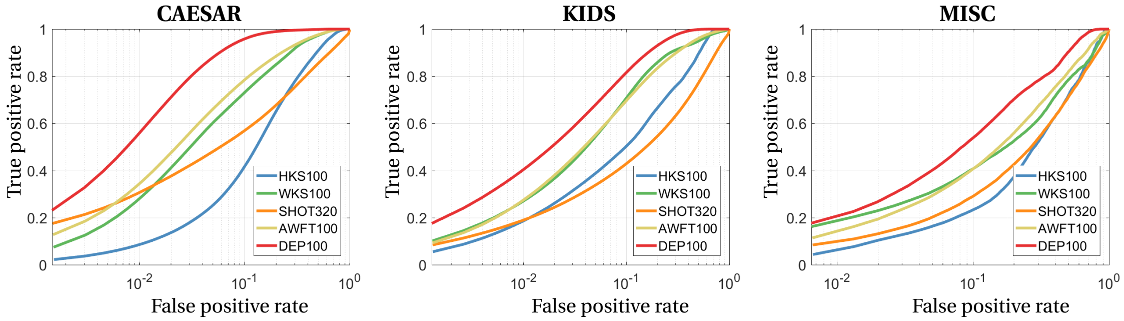


Fig. 6.10: Performance evaluation on CAESAR dataset (left), KIDS dataset (middle) and on the pairs *Elephant and Elephant subsampled, Elephant and Horse subsampled, Alien and Robot, Homer and Alien, Boy and Baby* and *Man* (7 shapes), *Gorilla* (5 shapes), *and Woman* (12 shapes) from MISC dataset (right). Comparison with 4 different descriptors. Next to the descriptor name we shown its dimensionality.

Table 6.1: AUC of ROC curves

dataset	**HKS100**	**WKS100**	**SHOT320**	**AWFT100**	**DEP100**
CAESAR	0.8069	0.9083	0.8034	0.9228	**0.9762**
KIDS	0.8210	0.8955	0.7279	0.8951	**0.9449**
MISC	0.6568	0.7227	0.6584	0.7513	**0.8317**
FAUST	0.8387	0.8736	0.7569	0.8946	**0.9577**
FAUST intra	0.9258	0.9176	0.7830	0.9163	**0.9635**
FAUST extra	0.8300	0.8692	0.7543	0.8924	**0.9572**
MEAN	0.8132	0.8645	0.7473	0.8788	**0.9385**

clearly violated. Figure 6.10 shows the pointwise matching evaluation on other benchmarks: i) CAESAR (Figure 6.10 left), KIDS (Figure 6.10 middle), and iii) MISC (Figure 6.10 right). In the human datasets (i.e., CESAR and KIDS) the shapes are geometrically near-isometric although the natural articulated motion of humans and the change of the subject can lead to possibly significant geodesic distortions. These kinds of distortion are not as strong as in FAUST. On these datasets DEP clearly outperforms all the competitors. In particular we exceed the performance of AWFT, the most recent competitor in our analysis. The curves on the right are obtained on the MISC dataset. On this dataset the point-to-point matching is very difficult since the shapes are not related by an isometric transformation, and therefore methods that are very sensitive to non-isometric changes are likely to fail. Moreover, in general the meshes have different resolution and different connectivity. We evaluate the average performance between all the pairs of shapes on the vertices for which the correspondences are known. In this setting our method also clearly outperforms all the other descriptors. As in the FAUST-extra in the presence of more non-isometric deformations, that are characteristic to the MISC dataset, the DEP descriptor increases its positive gap with all the competitors. In order to give a numerical comparison of the performance, we report the AUC (area under the curve) for every dataset in Table 6.1. Our DEP descriptor showed superior performance over all the other methods across all the datasets with improvements of at least 5% and on average around 6%. In particular, our method is able to improve standard methods based on a diffusion geometry like HKS and WKS by confirming the benefit of our alternative evolution process. Moreover, our DEP descriptor is preferable to AWFT that to the best of our knowledge represents the state of the art of descriptors without learning.

Functional Maps

The experiments above suggest that our descriptor can identify related points across different shapes. Below we show how our general approach, which includes the discrete evolution process and the derived DEP descriptor can be used to obtain entire *maps* across shapes. For this, we use the *functional map* framework to which is dedicated Chapter 4 of this thesis. As we have already seen to estimate this we can exploit i) functional and ii) commutativity constraints. A functional constraint (i) is defined from a set of corresponding descriptors. The commutativity constraint (ii) is introduced by some commutative operator. In the original setting i) is introduced by standard descriptors like HKS and WKS and ii) is imposed by the Laplace-Beltrami Operator (LBO). A detailed description can be found in Chapter 4.

The discrete time evolution process provides a new set of continuous functions $\check{s}$, the score of our process for different selections of parameters and a new shape operator $\check{\mathbf{S}}$. In the following experiments we propose to inject our evolution process scheme into the functional map framework by introducing our DEP descriptor for the functional constraint, and by exploiting our regularized score operator $\check{\mathbf{S}}$ as a commutative operator.

Dataset and Evaluation

We choose to perform our tests on FAUST because it is the largest datasets and it contains more isometric and non-isometric variations. Moreover as we have done in the point to point matching evaluation, we perform three different tests on FAUST considering all the dataset, only the intra subject pairs and finally restricting to pairs belonging to different subjects.

All the following evaluations are obtained by randomly selecting 10 pairs of shapes from the 100 available in the dataset and we plot the average performances. The test on the entire dataset is obtained with 10% of intra subject shapes and 90% of extra subject as happens in the whole dataset. As in the original version we use a post processing step to the obtained functional map based on a high-dimensional ICP, which also results in a point-to-point map between shapes (see [115] for details).

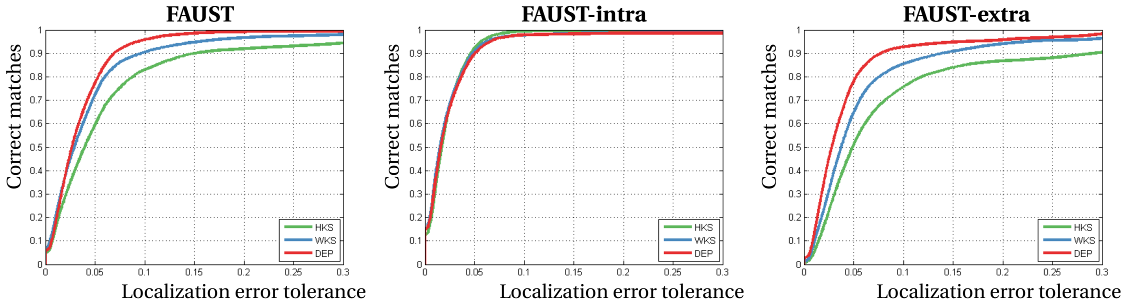


Fig. 6.11: Test varying the functional constraints functional maps framework. Visualization of the error rates given an unnormalized radius r, the percentage y of the points that are mapped by the correspondence at a distance at most r from their ground-truth image.

We evaluate the performance using the *correspondence quality characteristic* [76], previously introduced in Chapter 5, that is the standard evaluation used for functional maps. These curves show the percentage of nearest-neighbor matches that are at most r-geodesically distant from the ground truth correspondence. Here, we accept the symmetric images of ground truth correspondences, and use the minimum between the distance from the matched point to the ground truth and its symmetric image, as done in prior work [115].

Functional constraints

Starting from [115] many works have tried to explore the quality and the best method for the selection of the descriptor constraints to be used in this framework (See [116] Chapter 2 for an overview). In [36], for example, the authors compute optimal descriptor weights by learning the contribution of each descriptor for the estimation of the overall matching. In this experiment we adopt a new approach proposed recently in [111] that is based on the enhancement of the descriptor preservation constraints. In [111] the authors noticed that the original functional constraints [115] do not capture all of the information contained on a given descriptor. Rather than only preserving the descriptor *values*, as done before, they

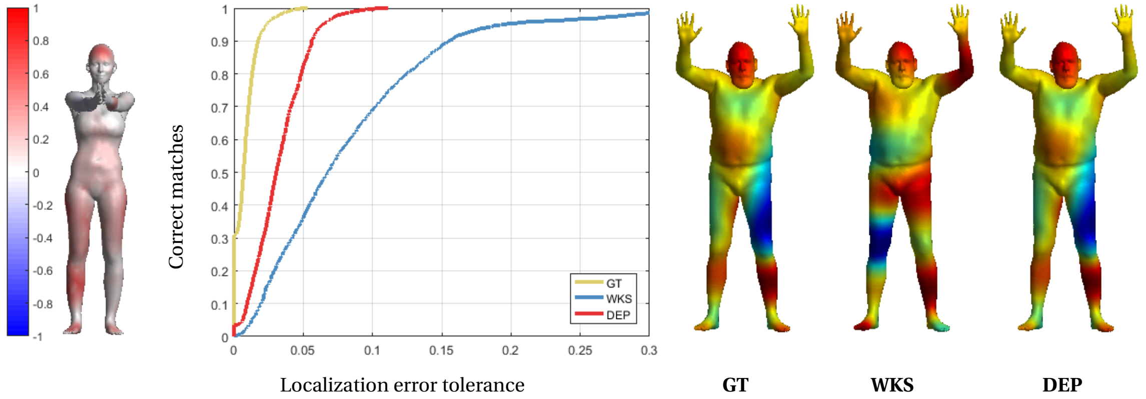


Fig. 6.12: Evaluation of a functional map on a pair of shapes from two different subjects in the FAUST dataset. In the middle the evaluation curves for ground truth (GT) and the comparison between WKS and DEP are shown. The colors on the left show points that are matched better (red) or worse (blue) with DEP compared to WKS. On the right we show on the second shape the transportations of a smooth function defined on the the first shape, using GT, WKS or DEP.

have demonstrated that by preserving *function products* with descriptors leads to a significant improvement in map quality, even in the presence of a few descriptor functions. Moreover, the authors show how the resulting new constraints can be efficiently encoded via commutativity of the unknown map with linear operators defined by the descriptors, which retains the overall efficiency of the framework. Note that in addition to the descriptors, in [115] and [111] authors proposed to add some consistent segmentation of the shapes as functions to be preserved. Here we prefer to use only descriptors in order to evaluate strictly their contribution to the framework. We consider HKS100, WKS100 and our DEP100, and from each of these we select 6 equally spaced scales as input descriptors. Explicitly we use the first, the last and all the scales that are multiples of 20, in order to represent all the features contained in the whole collection. In Figure 6.11 we show the performance of the functional maps framework using the three different descriptors in the three different tests. As can be seen the best performance on FAUST is obtained with DEP descriptors.

In particular, on shapes for undergoing strong non-isometric deformations, our DEP descriptor (red line) clearly outperforms the competitors. In the FAUST intra test, where the isometry is preserved we note that the three descriptors achieve the same results. This experiment confirms the utility of our descriptor for the computation of shape matching within the functional map framework. In Figure 6.12 we show some details for particular pairs of non isometric shapes. We compute functional maps from the female shape on the left to the male shape on the right of this figure. In the middle we plot the evaluation curves. DEP clearly outperforms WKS and is also very close to the ground truth (GT) performance. On the shape on the left we plot the differences between the error of the functional maps correspondences computed with WKS or DEP in the following way. If F_{WKS} and F_{DEP} are the functional maps computed using WKS and DEP respectively, and GT is the ground truth map, then we can define the function g at every vertex v_i as:

$$g(v_i) = \|(GT - F_{\text{WKS}})\delta_{v_i}\| - \|(GT - F_{\text{DEP}})\delta_{v_i}\| \,,$$

where δ_{v_i} is the function equal 0 everywhere but equal to 1 in v_i. The norm $\|(GT - F)\delta_{v_i}\|$ is a measure of the local error of transportation for the functional map F in v_i. Plotting this function g on the shape we have that the Red area are the ones for which F_{DEP} is more precise while in Blue we can see where F_{WKS} is better. More intense color corresponds to greatest difference in performances. On the right we can see on the second shape the transportations of a function defined on the the first shape, using GT, WKS or DEP. Here, GT corresponds to the original function defined on the first shape. In comparison, the transported function using DEP is similar to GT, unlike the one obtained using WKS.

Commutativity constraint

In Figure 6.13 we evaluate different commutativity constraints. In order to give more emphasis to the contribution of the operators, here

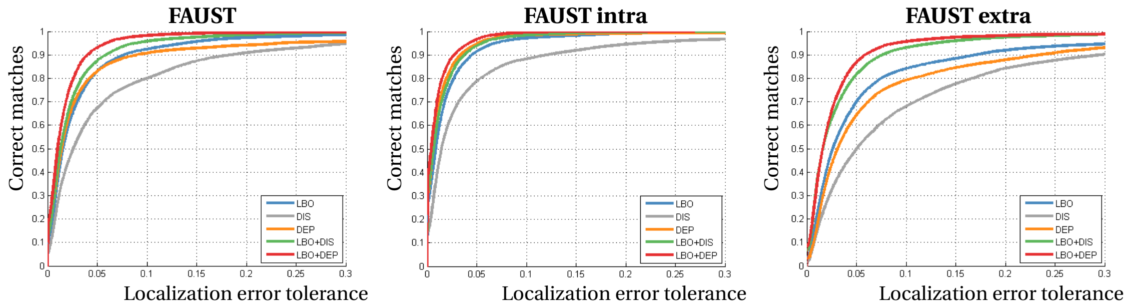


Fig. 6.13: Test varying the commutativity constraints functional maps framework. Visualization of the error rates given an unnormalized radius r, the percentage y of the points that are mapped by the correspondence at a distance at most r from their ground-truth image.

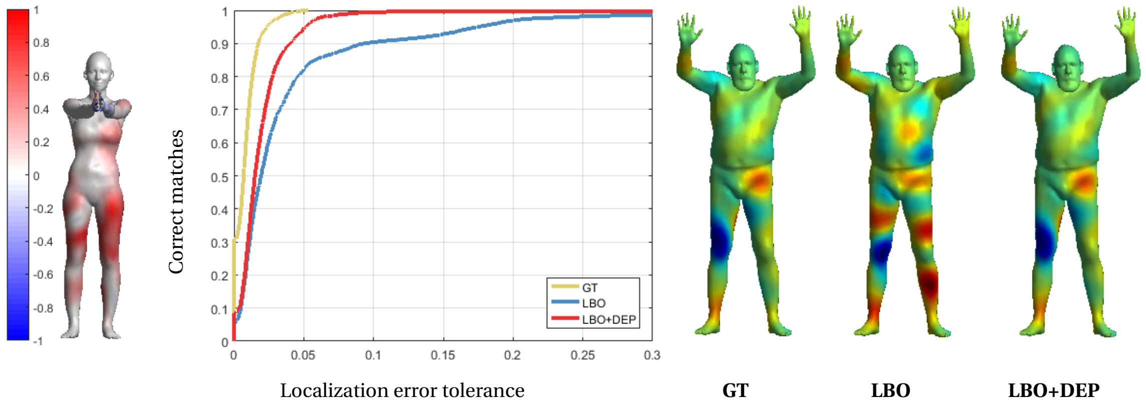


Fig. 6.14: Evaluation of a functional map on a pair of shapes from two different subjects in the FAUST dataset. In the middle the evaluation curves for ground truth (GT) and the comparison between LBO and LBO+DEP are shown. The colors on the left show points that are matched better (red) or worse (blue) with LBO+DEP. On the right we show on the second shape the transportations of a smooth function defined on the the first shape, using GT, LBO or LBO+DEP.

we use the standard functional constraint and the standard set of preserved functions (HKS and WKS), [115], without any pre-computed segment correspondences. We use i) our regularized score operator $\check{\mathbf{S}}$ (DEP) defined in Equation 6.13 ii) the Laplace Beltrami Operator (LBO) as in the original version of functional map, and iii) the matrix of geodesic distances (DIS), as basic approach to introduce the geodesic information in this framework, as proposed in [2]. We also compare the performance of each operator with the pairs (LBO+DIS) and (LBO+DEP). This way we can see which operator based on geodesic distances adds more information to the original LBO. As can be seen while in the FAUST-intra test (Figure 6.13 middle) the improvement given by adding DEP to LBO is not significant, the gap between LBO and LBO+DEP clearly grows in the FAUST-extra test (Figure 6.13 right). This result confirms that in the discrete time evolution process framework also the operator $\check{\mathbf{S}}$ is more stable with respect to non-isometric deformation.

As can be seen in Figure 6.13 the performance using DEP alone is very close to the standard performance using only LBO. When DEP and LBO are integrated the performance is definitely improved. Note also that the quality of the maps obtained with (LBO+DEP) significantly exceeds (LBO+DIS), confirming that our evolution process provides an effective approach to further improve the contribution of geodesic distances in this framework. In Figure 6.14 we show some details for a particular pair of non isometric shapes. This figure contains the same analysis of Figure 6.12, but we compare LBO and LBO+DEP in the Commutativity constraint. In the middle we plot the evaluation curves. DEP clearly improves the results of LBO and LBO+DEP is closer to the ground truth (GT) performance. On the right we can see on the second shape, the transportations of a function defined on the the first shape, using GT, LBO or LBO+DEP respectively. These results show that adding DEP to LBO results in maps that are significantly better than those obtained using LBO alone and that are very close to GT.

Evaluation on Real Scans dataset

We evaluate the proposed method on the full real scans dataset [13] following the settings reported on [30] . For this experiment to give a direct comparison with [30] we evaluate our results using the correspondence quality characteristic [76] with error measured in centimeters.

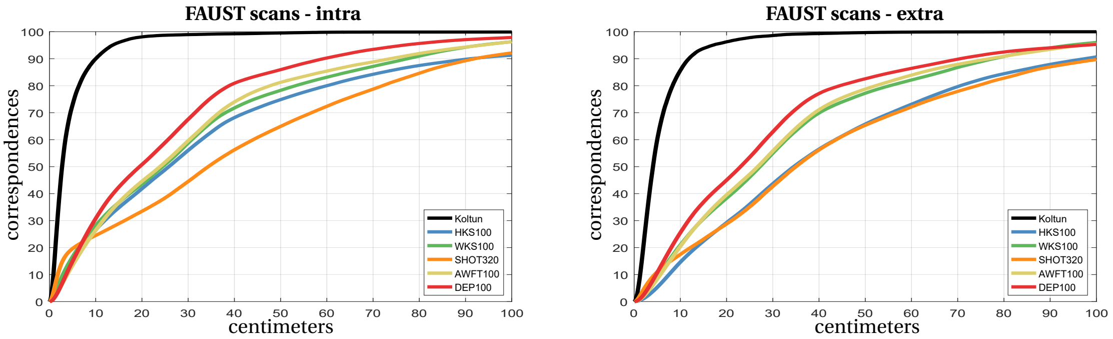


Fig. 6.15: Cumulative error distribution on the FAUST real scan training dataset. The results are an average on 50 pairs provided by [30], 50 for intra-subject (left) and 50 for extra-subject (right). The matching is computed via distances in the descriptor space. In black we show the performance of [30] for reference.

Figure 6.15 shows point to point matching results on 50 intra-subject and 50 extra-subject pairs. We compare our DEP100 descriptor with [30] and the 4 descriptors used in the point-to-point matching section, i.e., i) HKS100, ii) WKS100, iii) SHOT320 and iv) AWFT100. Note that as expected the method proposed in [30] shows the best performance. These excellent results were indeed obtained by using an extrinsic alignment of the shapes, which is not used by the other methods. On the other hand, among the pure descriptor-based methods, our DEP descriptor outperforms all other alternatives and is comparable with most of the methods evaluated in [30].

Qualitative Evaluation

In order to obtain a visual evaluation of the proposed approach we show the dissimilarity maps of some pairs of shapes for some fixed points, as done in Chapter 5 for the evaluation of others descriptors. For instance in Figure 6.16 we fix a point on the *Male* shape in a first pose (the red ball), and show its dissimilarity (i.e., the Euclidean distance on the descriptor space) with all the points on the second pose of the *Male* shape. The minimum of the dissimilarity for every descriptor is highlighted with a white ball. The distances grow under varying from cold colors (similarity) to warm colors (no similarity). It is clear that both HKS and WKS

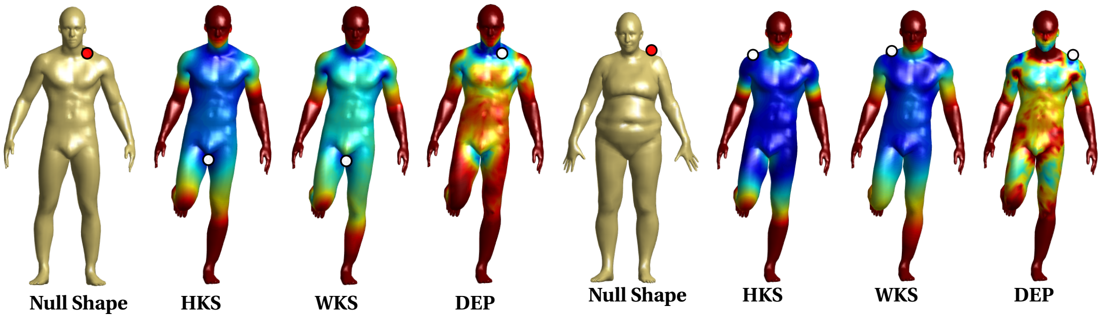


Fig. 6.16: Visualization of dissimilarity maps. On the left a point is selected on the Man shape in the first pose (left), and the dissimilarity maps on the Man shape in a different pose for HKS, WKS and our method respectively from left to right. On the right a point is selected on the Woman shape in the first pose (left), and the dissimilarity maps on the Man shape in a different pose for HKS, WKS and our method respectively from left to right.

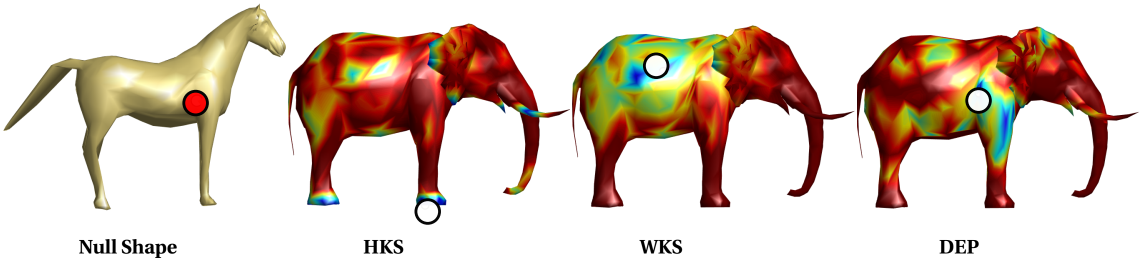


Fig. 6.17: Visualization of dissimilarity maps. The selected point on the Horse shape (left), and the dissimilarity maps on the Elephant shape for HKS, WKS and our method respectively from left to right.

methods are not able to localize the selected point, and the white ball is far from the target point. Conversely, our method gives more local result and identifies the correct corresponding point. Gradually losing the isometry property between the shapes the robustness of our method is highlighted. This is shown on the right in Figure 6.16 for Female and Male subjects from FAUST dataset, in Figure 6.17 for Horse and Elephant from MISC and on the left in Figure 6.18 for Robot and Alien from MISC. In particular in the last two pairs from MISC the two shapes are clearly non-isometric and their meshes are totally different in the number of vertices

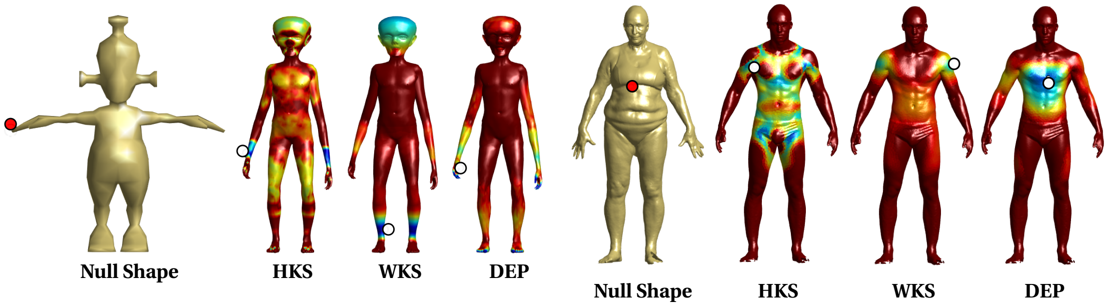


Fig. 6.18: Visualization of dissimilarity maps. On the left the selected point on the Robot shape (left), and the dissimilarity maps on the Alien shape in a different pose for HKS, WKS and our method respectively from left to right. On the right the selected point on the scanned Woman shape (left), and the dissimilarity maps on the scanned Man shape for HKS, WKS and our method respectively from left to right.

and connections. We perform the same test using a pair of real scans from FAUST dataset to evaluate the robustness of the proposed method against noise and missing parts. Real scans have more than 160K vertices and the number of vertices is different for different scans. These high-resolution, triangulated, non-watertight meshes present a lot of challenging features. Also in this case our descriptor correctly identifies a local region of points on the shape without spurious areas as in the HKS and WKS results as we show on the right of Figure 6.18. A similar evaluation analysis is shown in Figure 6.19 for testing the robustness of proposed descriptor against strong subsampling and topological noise (i.e., glued fingers and missing parts). This experiment is carried out on a small collection of shapes from the FAUST dataset with different poses of the same subject and different subjects with several kinds of noise (see the caption of Figure 6.19 for an exhaustive list). We observe that our DEP descriptor is robust to smoothing, subsampling and topological noise and shows better localization and accuracy than other methods. We also evaluate the robustness of our DEP for partial and broken parts from the SHREC’11 benchmark. Note that spectral methods such as HKS and WKS are known to be sensitive to this kind of failure, since the

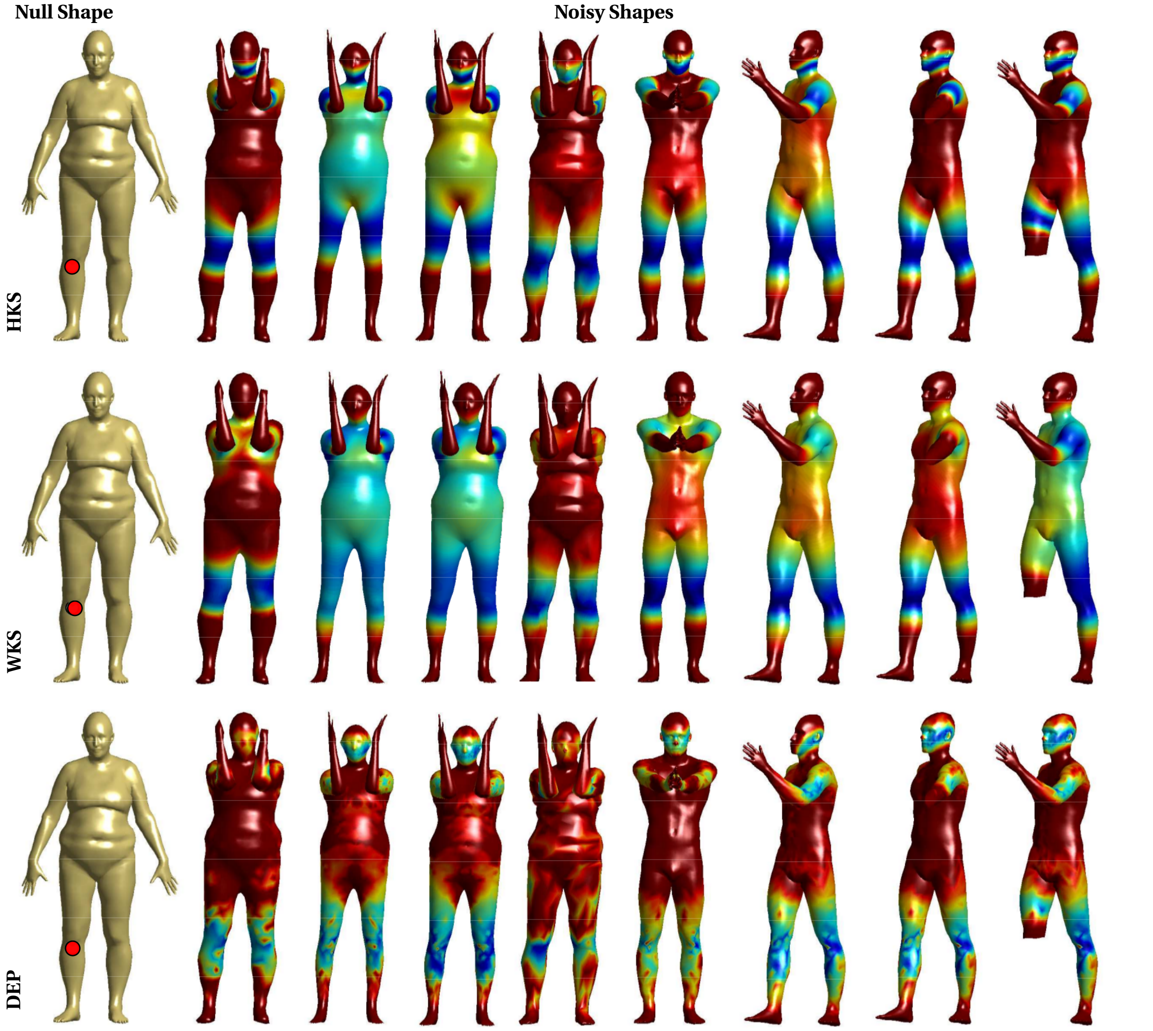


Fig. 6.19: Visualization of the dissimilarity maps for shapes with different kinds of noise. The vertex on the right knee of the clean shape on the left is selected, and the dissimilarity maps on shapes with different kind of noise are shown. In the first row for HKS, in the second row for WKS and finally for our method. Respectively from left to right the original shape, an isometric remeshing, two smoothed versions of the surface, a subsampled mesh with 1000 vertices, topological noise (glued fingers) and the last three are partial views or surfaces with missing parts.

Laplace operator changes its spectral representation. Figure 6.20 shows the dissimilarity maps for some selected points for different partial mod-

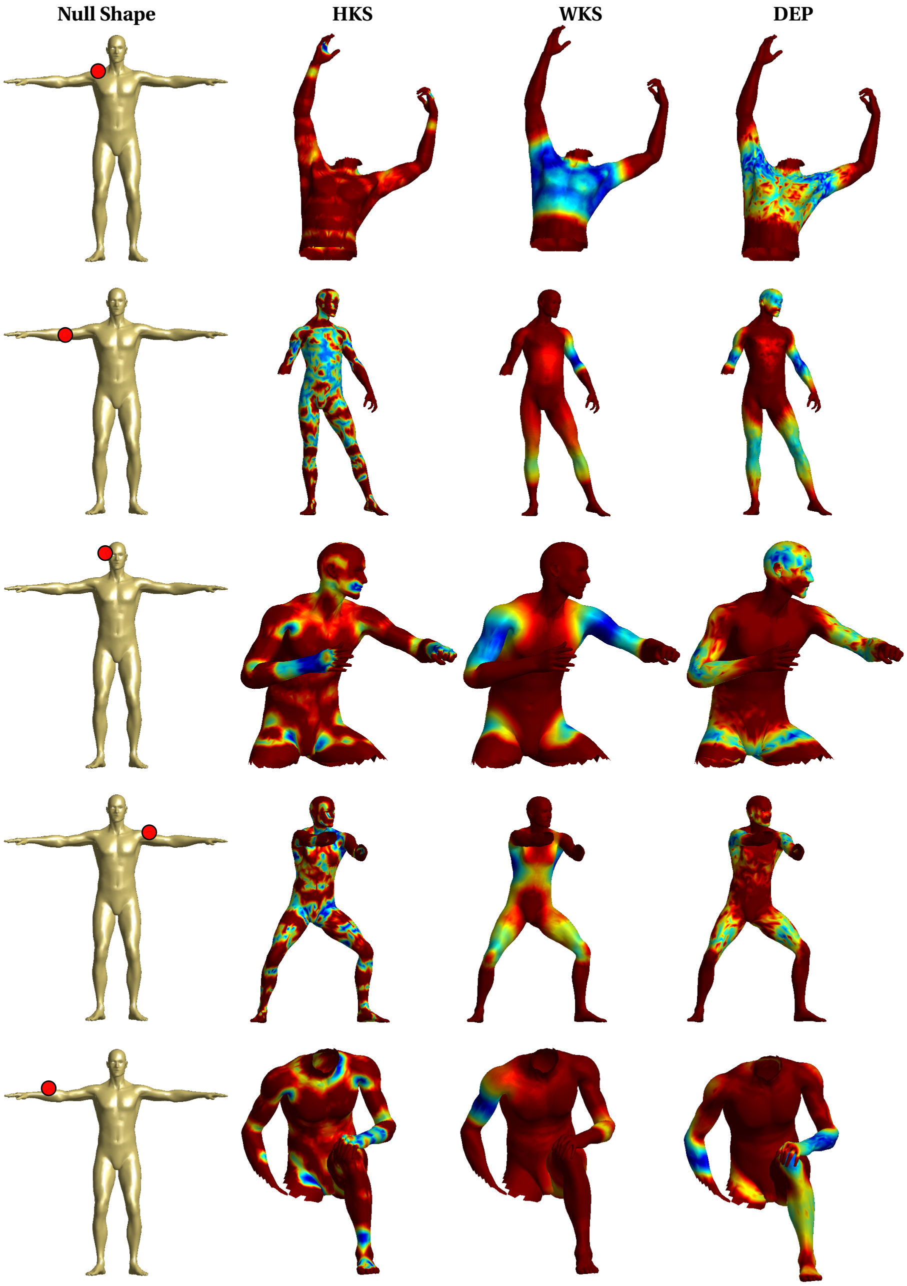


Fig. 6.20: Visualization of the dissimilarity maps for partial matching. The selected points on the null shape, and the dissimilarity maps on different partial models for HKS, WKS and our method respectively from left to right.

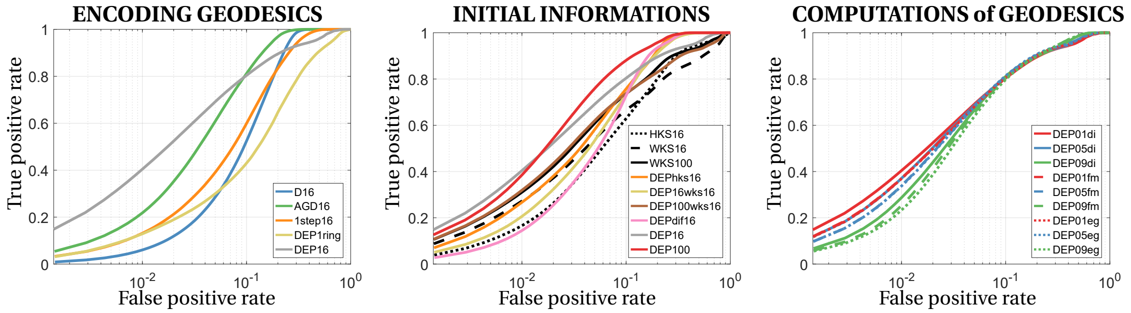


Fig. 6.21: Performance evaluation on FAUST dataset (test for 10 different poses of 2 different subjects) in different settings. From left to right: comparison with different methods to encode geodesic distances (left), comparison starting from different initial information matrices (middle), comparison starting from different computations of geodesic distances and regularizations (right).

els. We note that HKS is very sensitive to this kind of shape alteration and it highlighted in general wrong areas. WKS performed clearly better but it resulted in several ambiguous parts. Differently, our method is robust in all the selected experiments.

Comparison of parameters choices in different settings

Finally, to highlight the flexibility of the proposed framework, we evaluate the performance of point to point matching across a range of parameter choices: i) different approaches to encode the geodesic distance, ii) different relation functions, iii) different choice of regularization values. In Figure 6.21 we show a set of such evaluations. In order to access the role of geodesic distance and the importance of higher order relations we consider five methods to encode the information contained in the matrix **D**:

– Basic geodesic distance (D16). We use the matrix **D** introduced in Section 6.2.5 as process operator and perform 1-step of our evolution process with $f_0 = \mathbf{e}$. Note that in this case the contribution of the areas **A** is not considered.
– 1-step (1-step16). As described in Section 6.2.5 we consider only the first order relation (i.e., 1-step of the evolution process with $f_0 = \mathbf{e}$).

- Average Geodesic Distance (AGD16). According to the method introduced in [60], we compute the average geodesic distances for every point at each scale as descriptor. Note that in AGD the areas are considered and differently from 1-step16 a more effective normalization method is introduced (see [60] for more details).
- The 1−ring version of our method (DEP1ring). For every vertex we compute the geodesic distances to each point in the 1−ring (i.e., the length of the edge). Different scales of this descriptor are then obtained using 16 different values of the regularization parameter for $r \in (0,\ 1)$.
- Our descriptor (DEP16). We evaluate our descriptor as described in 6.2.5 with $f_0 = \mathbf{e}$. In this case the higher order relations are considered by evaluating the evolution process up to infinite.

We fix 16 different scales $\delta \in [0,1]$ in order to obtain for every method a 16−dimensional descriptor as defined in Section 6.2.5, with exception of the DEP1ring method. Figure 6.21 on the left shows the performance of the considered approaches. As expected when the geodesic distance is not combined with the areas (D16) the performance are the worst. In the same way starting only from the 1 ring information the obtained descriptor does not perform well. This construction is indeed strongly related to the mesh, and is not stable under near-isometric or non-isometric deformations. Moreover, we confirm with a more exhaustive evaluation the benefit of higher order relations discussed in Section 6.2.5. In particular, our method (DEP16) clearly outperforms both 1-step16 and AGD16.
As second setting we evaluate the evolution process by defining different relation functions:

- Diffusion distance (DEPdiff16). The relation function is defined as in Section 6.2.5 where the normalized diffusion distance is used instead of the geodesic distance.

- HKS from diffusion distance (DEPhks16). The relation function is defined by the matrix $\mathbf{D}_f$ as proposed in Section 6.2.6 where the HKS is derived from the diffusion distance.
- Relation from other descriptor 1 (DEP16wks16). In this case a different approach is exploited. We compute a generic descriptor for each vertex, namely the WKS. We define as relation function the Euclidean distance on the descriptor space.
- Relation from other descriptor 2 (DEP100wks16). The same as the previous one but the WKS descriptor is $100-$dimensional rather than $16-$dimensional.
- Our descriptor (DEP16). The relation function is defined starting from the geodesic distance as described in Section 6.2.5.

We fix 16 different scales $\delta \in [0,1]$ in order to obtain for every method a multi-scale descriptor with the same dimensionality. In more details, in DEPdiff16 we fix a time value for each scale and compute the descriptor using the normalization procedure described in Section 6.2.5. For DEPhks16 we use the 16 matrices $\mathbf{D}_f$ (using 16 different time scales). Finally DEP16wks16 and DEP100wks16 are obtained as described in Section 6.2.5 where the normalized Euclidean distance among descriptors is used instead of $\hat{\mathscr{G}}$. For all the considered relation functions we compute the evolution process and the evolution score by obtaining different versions of the DEP descriptor. To complete the evaluation we add in the comparison DEP100, HKS16, WKS16, and WKS100. Figure 6.21 middle, shows the results. As above, this test also highlights the importance of the higher order relations captured using our discrete evolution process. In particular, it is interesting to observe how DEPhks16 (which employs our evolution process on the same relation function that generates the HKS at 1-step as explained in Section 6.2.6) clearly outperforms HKS16. Moreover, even if WKS16 performs better than DEP16wks16 we observe that DEP100wks16 is comparable with WKS100 showing that our DEP approach is able to obtain the same performance but with a much lower dimensional descriptor (from 100- to 16-dimensional). Overall, our DEP descriptor based on geodesic distance (DEP16) achieves the best performance. Conversely, the use of diffusion distance (DEPdiff16) seems not convincing. From this test we conclude that the best choice for the relation function is derived from the geodesic distance as in our DEP descriptor, even if the other choices have confirmed the effectiveness of our evolution process scheme. Finally, we evaluate the dependence of the proposed method on the regularization parameter. In Figure 6.21 on the right we compute DEP descriptors with three different values of $c \in (0, 1)$ in the computation of the parameter r as described in Section 6.2.5. The choice of c gives different weights to the paths with a large number of steps (and vice versa). We evaluate $c = 0.1$, $c = 0.5$ and $c = 0.9$. Note that the performances are very similar suggesting that the best choice of c can be estimated with respect to the task at hand. In general the choice $c = 0.1$ consistently gives good and stable performance and therefore we fix this value throughout our experiments. Finally, in Figure 6.21 on the right we also evaluate the performance of our descriptor by computing the geodesic distance on surface using different methods. Namely, we compare Dijkstra algorithm (*di*) [105], fast marching (*fm*) [77] and exact geodesic (*eg*) [141]. As can be seen on the right of Figure 6.21, the performance is similar across different choices, and for computational efficiency we use Dijkstra's algorithm in the following experiments.

Complexity

The complexity of our method is dominated by computing and storing the pairwise geodesic distance matrix $\mathscr{G}$. In practice, we use Dijkstra's algorithm to approximate the geodesic distances on a triangle mesh. By using a straightforward non-optimized implementation, our method required, on average, on a triangle mesh with around 7000 vertices, just over 10 minutes to compute the descriptors of all points across a range of 100 scales and just over 1 minute across a range of 16 scales, on a machine with 8GB of RAM using an Intel 2,6 GHz Core i7 processor. HKS and WKS take a few seconds on the same machine.

6.4 Locality in discrete time evolution process and take home message

In this last Section we want briefly explore the use of discrete time evolution process for representing a single point on a surface. This Section could be seen as an extension of the paper [103], or as an introduction to possible future work related to the same paper.
Instead of considering as starting state a constant distribution on the surface $f_0 = \mathbf{e}$ as done in the previous Section, here we fix the starting state as a delta in a point on the surface; in this case we can write $f_0 = \delta_x$, for a given $x \in \mathcal{M}$. With this choice $f_0(x) = 1$ and f_0 is equal to 0 everywhere else. Now if we apply the DEP framework to this starting state we obtain the discrete time evolution process of the point x represented as a delta function. Also this starting state has a clear connection with heat diffusion. Indeed, as we have seen in Chapter 3, for every $y \in \mathcal{M}$ and for every time $t \in \mathbb{R}$, the heat kernel $h_t(x, y)$ represents the amount of heat which is transferred from x to y in the time t, starting with the initial heat distribution δ_x. This connection allows us to compare how the heat kernel and our discrete time evolution process represent a point x in different scales of time for the heat diffusion, and of localization for DEP. In Figure 6.22 we compare these two different process on two human shapes. Each row of the Figure 6.22 is dedicated to a different shape. The shape on the second row has glued hands as a topology error. For both process the initial state is equal to a delta function in the point x on the left shoulder of the shapes that is highlighted with a small white sphere. The five shapes on the left represent the heat diffusion process for five different increasing scales of time. While the five shapes on the right represent the discrete time evolution process for five different decreasing scales of localization.

We rescale all the diffusion states shown in the interval $[0,1]$ for visualization. Comparing the DEP with the heat kernel we notice two aspects for which DEP is better. First the localization. As can be seen the heat kernel is initially well localized around the point at least the first shape on the left. As time scale goes on the localization is lost, and the heat is globally diffused on the shape. Looking at DEP, as the localization scale decreases the size of the considered neighborhood increases but the localization remains centered at the fixed point. Second the stability with respect to the topology error. For the heat kernel comparing the first and the second rows it is clear that the third and fourth shapes are different, in the diffusion on the arm opposite to the point. DEP seems more stable with respect to the variation in the topology, although in the fourth and fifth shapes there are some small changes. This motivates the analysis of the use of DEP to solve problems related to topological variations. This could be the idea for a future work based on DEP. In this Chapter we introduced the discrete time evolution process, an alternative evolution paradigm defined on surfaces. Our process is based on a pairwise relation defined o the surface. We start describing a point as the set of its neighborhoods as depicted in Figure 6.1. These initial informations can be seen as a local representation of the point in different scales of locality. The discrete time evolution process and its iterative procedure allows us to explore the relation between a point and the entire shape starting from an initial local relation. In this spirit the discrete time evolution process encode a global representation of the points starting from a local characterization. The discrete time evolution process can be thus view as a tool for a from local to global description of the surface.

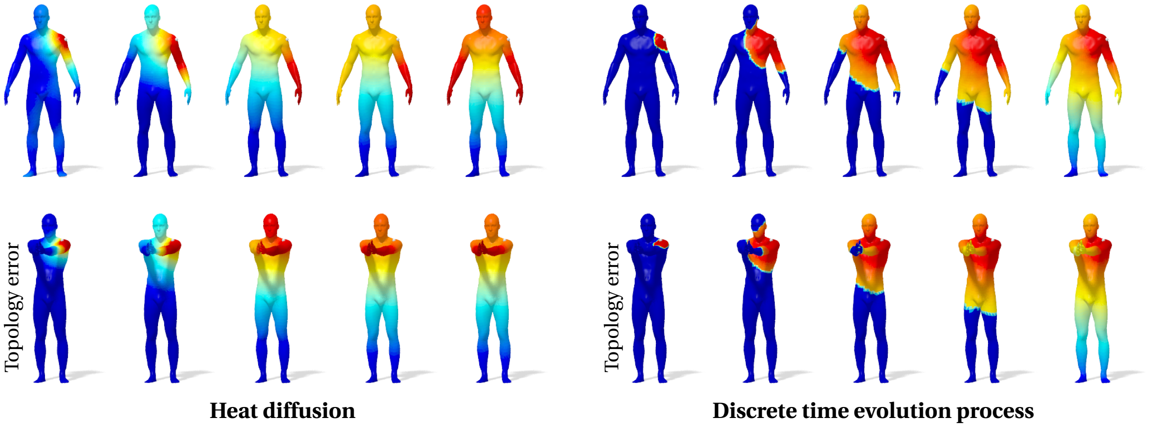


Fig. 6.22: Comparison between heat diffusion and discrete time evolution process. The comparison is performed on two surfaces, one for each row. These shapes are different poses of the same subject, with topology error (glued hands) on the shape on the second row. A point on the shapes is selected, the same on both shapes, on the left shoulder of the shapes and it is highlighted with a small white sphere. On the left we show the heat diffusion from a heat source concentrated in the point for different time scales. On the right, starting from the indicator function of the point as initial state, we show the discrete time evolution process For different localization scales.

7

Conclusion

In this thesis we observed how the localized approaches could improve many applications in the geometry processing field. Our main focus is on localized analysis for point based-description of surfaces, looking at how a pointwise characterization and description of shapes can help in many applications and problems.

We describe how to extend local signal processing tools such as the windowed Fourier transform to 2-dimensional surfaces embedded in $\mathbb{R}^3$. We propose a new family of localized basis for spectral geometry processing that allows us to obtain a good spectral localization together with a good spatial localization and increase the representation of signals defined on manifolds. We also introduce the discrete time evolution process as a non spectral alternative to localized analysis of surfaces and to standard diffusion process.

The proposed methods could be analyzed under different points of view.

- *Spectral and non-spectral method.* We can distinguish the methods according to whether the spectrum of the shapes is involved or not. This is the difference between the localizations proposed in Chapter 5 and the one proposed in Chapter 6.
- *The domain of the localization.* This domain could be spectral as in the case of the WFT and AWFT methods, where a spectral window g is defined through its Fourier coefficients. The localization domain could

be spatial as it is for LMH where the localization is obtained fixing a region on the surface.

– *The direction of localization.* We can move from global to local as done in WFT, AWFT and LMH methods, imposing locality to the global spectral analysis. Conversely we can start from local information and obtain a global or a less local description of the shape, as done in DEP descriptor. From local pairwise relation to a final score that encodes properties of the entire shape. In the same way in the brain classification work we start from local descriptors and we obtain a global classification of the shapes.

Our practical contributions are many and derive from the application of the proposed methods to problems such as point-to-point matching, salient point detection, shape segmentation, shape classification and shape correspondence in different settings.

The obtained results give arise to many future directions. First we could try to modify the functional kernel proposed in Chapter 4 in order to obtain a kernel that is less sensitive to local extrinsic variations improving the detection of more substantial morphological abnormalities associated with the analyzed disease. This can be done adopting other types of descriptors, for example extrinsic descriptors, in the optimization of the map. Another possible extension of the functional map for brain classification method could be to analyze more in details the deformations that identify the abnormalities. Adding to the classification already obtained a localization of the main deformation. The functional kernel could be also profitably exploited in other applications such as shape retrieval.

With the WFT and the AWFT it is possible to investigate new applications such as 2D or 3D texture classification, analyzing with these tools functions as curvatures, gradients and others that could characterize these textures. Some investments could be made in the direction of defining more Gabor like instruments, of which AWFT is only a first attempt.

The general LMH basis can be improved in different applications with a task specific and unsupervised definition of the regions on which localize the basis functions. A possible application of this work could be the transfer of regions for which the standard MH are not optimal. New interesting directions can be also to consider the localization of the eigenfunctions of other operators. LMH could be also seen as a possible new solution for the localized spectral analysis that we face with WFT and AWFT. We can fix a definition for the neighbour of each point, then localize in this neighbour the LMH and use the computed function in order to obtain the localized spectral representation of the selected point.

Finally as we shown DEP analysis can be extended to more general functions and not only to the constant functions as done in this thesis. Similarly, other pairwise relations can be considered instead of geodesic distances. Probably some spectral pairwise relations can be adopted instead of the geodesic distances, speeding up computationally the method and improving the intrinsic properties encoded by DEP. The DEP descriptor does not exhaust the analysis of the paths on surfaces. Other kinds of analysis widely used for graphs could be adopted on surfaces with possible practical results.

We hope that the tools proposed in this thesis could be used, reinforced and developed by the geometry processing community, and that localization could generate new solutions and promote much innovative research in this field.

References

1. Y. Aflalo, H. Brezis, and R. Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160, 2015.
2. Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. Spectral generalized multi-dimensional scaling. *Int. J. Comput. Vision*, 118(3):380–392, 2016.
3. Mathieu Andreux, Emanuele Rodolà, Mathieu Aubry, and Daniel Cremers. Anisotropic laplace-beltrami operators for shape analysis. In *Proc. NORDIA*, pages 299–312. Springer, 2014.
4. Dragomir Anguelov et al. SCAPE: Shape completion and animation of people. *TOG*, 24(3):408–416, 2005.
5. M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV*, 2011.
6. Kayhan N Batmanghelich, Dong Hye Ye, Kilian M Pohl, Ben Taskar, and Christos Davatzikos. Disease classification and prediction via semi-supervised dimensionality reduction. In *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1086–1090, 2011.
7. Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Proc. NIPS*, 2000.
8. Mirela Ben-Chen and Craig Gotsman. Characterizing shape using conformal factors. In *Proc. 3DOR*, 2008.
9. Pierre Bérard, Gérard Besson, and Sylvain Gallot. Embedding riemannian manifolds by their heat kernel. *Geometric & Functional Analysis*, 4(4):373–398, 1994.
10. E Bergshoeff. Ten physical applications of spectral zeta functions. *Classical and Quantum Gravity (CQG)*, 13(7), 1996.
11. James Bergstra et al. Theano: a CPU and GPU math expression compiler. In *Proc. SciPy*, June 2010.
12. Francesco Bianconi and Antonio Fernández. Evaluation of the effects of gabor filter parameters on texture classification. *Pattern Recognition*, 40(12):3325–3335, 2007.
13. Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, 2014.
14. Davide Boscaini, Jonathan Masci, Simone Melzi, Michael M Bronstein, Umberto Castellani, and Pierre Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. In *Computer Graphics Forum*, volume 34, pages 13–23. Wiley Online Library, 2015.
15. Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M. Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. *Computer Graphics Forum*, 35(2), 2016.
16. N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research*, 15:1455–1459, 2014.
17. E. Boyer, A. M. Bronstein, M. M. Bronstein, B. Bustos, et al. SHREC 2011: robust feature detection and description benchmark. In *Proc. 3DOR*, pages 71–78. Eurographics Association, 2011. `http://tosca.cs.technion.ac.il/book/shrec_correspondence.html`.
18. A. M. Bronstein and M. M. Bronstein. Shape recognition with spectral distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):1065–1071, 2011.

19. A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes.* Springer, 2008.
20. Alex Bronstein, Yoni Choukroun, Ron Kimmel, and Matan Sela. Consistent discretization and minimization of the l1 norm on manifolds. In *Proc. 3DV*, pages 435–440, Stanford, CA, 2016. IEEE.
21. Alexander M Bronstein, Michael M Bronstein, Leonidas J Guibas, and Maks Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)*, 30(1):1, 2011.
22. Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proc. CVPR*, 2010.
23. J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. In *Proc. ICLR*, 2014.
24. Kerstin Bunte, Eemeli Leppäaho, Inka Saarinen, and Samuel Kaski. Sparse group factor analysis for biclustering of multiple data sources. *Bioinformatics*, 32(16):2457–2463, 2016.
25. U. Castellani, E. Rossato, V. Murino, M. Bellani, G. Rambaldelli, C. Perlini, L. Tomelleri, M. Tansella, and P. Brambilla. Classification of schizophrenia using feature-based morphometry. *Journal of Neural Transmission*, 119:395–404, 2012.
26. Umberto Castellani, Pasquale Mirtuono, Vittorio Murino, Marcella Bellani, Gianluca Rambaldelli, Michele Tansella, and Paolo Brambilla. A new shape diffusion descriptor for brain classification. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 426–433. 2011.
27. Umberto Castellani, Alessandro Perina, Vittorio Murino, Gianluca Rambaldelli Marcella Bellani, Michele Tansella, and Paolo Brambilla. Brain morphometry by probabilistic latent semantic analysis. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 177–184, 2010.
28. Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2:27:1–27:27, 2011.
29. Isaac Chavel. *Eigenvalues in Riemannian geometry*, volume 115. Academic press, Orlando, FL, 1984.
30. Qifeng Chen and Vladlen Koltun. Robust nonrigid registration by convex optimization. In *International Conference on Computer Vision (ICCV)*, 2015.
31. Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transaction on Graphics (TOG)*, 28(3):73:1–73:12, 2009.
32. Dong S. Cheng, Vittorio Murino, and Mário Figueiredo. Clustering under prior knowledge with application to image segmentation. In P. B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 401–408. MIT Press, 2007.
33. Y. Choukroun, A. Shtern, A. Bronstein, and R. Kimmel. Hamiltonian operator for spectral shape analysis. *arXiv:1611.01990*, 2017.
34. David Cohen-Steiner and Jean-Marie Morvan. Restricted delaunay triangulations and normal cycle. In *Proceedings of symposium on Computational geometry*, pages 312–321. ACM, 2003.
35. Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006.
36. Étienne Corman, Maks Ovsjanikov, and Antonin Chambolle. Supervised descriptor learning for non-rigid shape matching. In *European Conference on Computer Vision (ECCV) Workshops (NORDIA)*, pages 283–298. Springer, 2014.
37. L. Cosmo, E. Rodolà, J. Masci, A. Torsello, and M. M. Bronstein. Matching deformable objects in clutter. In *Proc. 3DV*, pages 1–10, Stanford, CA, 2016. IEEE.
38. Anders M Dale, Bruce Fischl, and Martin I Sereno. Cortical surface-based analysis: I. segmentation and surface reconstruction. *Neuroimage*, 9(2):179–194, 1999.
39. Pablo de Castro, Fabrício de França, Hamilton Ferreira, and Fernando Von Zuben. Applying biclustering to text mining: an immune-inspired approach. *Artificial Immune Systems*, pages 83–94, 2007.
40. Matteo Denitto, Simone Melzi, Manuele Bicego, Umberto Castellani, Alessandro Farinelli, Mario A. T. Figueiredo, Yanir Kleiman, and Maks Ovsjanikov. Region-based correspondence between 3d shapes via spatially smooth biclustering. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.
41. Manfredo Perdigão do Carmo. Riemannian geometry. 1992. *Bikhausen, Boston.*
42. Sara Dolnicar, Sebastian Kaiser, Katie Lazarevski, and Friedrich Leisch. Biclustering overcoming data dimensionality problems in market segmentation. *Journal of Travel Research*, 51(1):41–49, 2012.

43. Helin Dutagaci, Chun Cheung, and Afzal Godil. Evaluation of 3d interest point detection techniques via human-generated ground truth. *The Visual Computer*, 28:901–917, 2012.
44. N. Dym, H. Maron, and Y. Lipman. DS++: A flexible, scalable and provably tight relaxation for matching problems. *arXiv:1705.06148*, 2017.
45. Ernst Rudolf Georg Eckert and Robert M Drake. Analysis of heat and mass transfer. 1987.
46. Asi Elad and Ron Kimmel. On bending invariant signatures for surfaces. *PAMI*, 25(10):1285–1295, 2003.
47. M. Filoche, S. Mayboroda, and B. Patterson. Localization of eigenfunctions of a one-dimensional elliptic operator. *Contemporary Mathematics*, 581:99–116, 2012.
48. J Fourier. Mémoire sur la propagation de la chaleur dans les corps solides [memoir on the propagation of heat in solid bodies]. *Nouveau Bulletin des sciences par la Société philomatique de Paris*, 6:215–221, 1807.
49. Vignesh Ganapathi-Subramanian, Boris Thibert, Maks Ovsjanikov, and Leonidas Guibas. Stable region correspondences between non-isometric shapes. In *Computer Graphics Forum*, volume 35, pages 121–133. Wiley Online Library, 2016.
50. K. Gebal, J. A. Bærentzen, H. Anæs, and R. Larsen. Shape analysis using the auto diffusion function. *CGF*, 28(5):1405–1413, 2009.
51. Guido Gerig, Martin Styner, D Jones, Daniel Weinberger, and Jeffrey Lieberman. Shape analysis of brain ventricles using spharm. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA)*, pages 171–178. IEEE, 2001.
52. Guido Gerig, Martin Styner, Martha E Shenton, and Jeffrey A Lieberman. Shape versus size: Improved understanding of the morphology of brain structures. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 24–32, 2001.
53. Polina Golland, W Eric L Grimson, and Ron Kikinis. Statistical shape analysis using fixed topology skeletons: Corpus callosum study. In *Information Processing in Medical Imaging*, pages 382–387, 1999.
54. Polina Golland, W Eric L Grimson, Martha E Shenton, and Ron Kikinis. Detection and analysis of statistical differences in anatomical shape. *Medical image analysis*, 9(1):69–86, 2005.
55. Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.
56. Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1994.
57. Boris Gutman, Yalin Wang, Jonathan Morra, Arthur W Toga, and Paul M Thompson. Disease classification with hippocampal shape invariants. *Hippocampus*, 19(6):572, 2009.
58. Bernard Haasdonk and Claus Bahlmann. Learning with distance substitution kernels. In *Pattern Recognition*, pages 220–227. 2004.
59. A Ben Hamza and Hamid Krim. Geodesic object representation and recognition. In *Proc. DGCI*, 2003.
60. Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proc. SIGGRAPH*, pages 203–212, 2001.
61. Sepp Hochreiter, Ulrich Bodenhofer, Martin Heusel, Andreas Mayr, Andreas Mitterecker, Adetayo Kasim, Tatsiana Khamiakova, Suzy Van Sanden, Dan Lin, Willem Talloen, et al. Fabia: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527, 2010.
62. Roger A Horn and Charles R Johnson. *Matrix analysis.* Cambridge university press, 2012.
63. Q. Huang, H. Su, and L. Guibas. Fine-grained semi-supervised labeling of large shape collections. *TOG*, 2013.
64. J. H. Hubbard and B. B. Hubbard, editors. *Vector Calculus, Linear Algebra, and Differential Forms: A Unified Approach.* Pearson, 2nd edition, 2001.
65. Hoppe Hugues. Progressive meshes. In *Computer Graphics (SIGGRAPH 96 Proceedings)*, pages 99–108, 1996.
66. J. Hwang, I. K. Lyoo, S. R. Dager, S. D. Friedman, J. S. Oh, J. Y. Lee, S. J. Kim, D. L. Dunner, and P. F. Renshaw. Basal ganglia shape alterations in bipolar disorder. *Am J Psychiatry*, 163(2):276–285, 2006.
67. Athirai A Irissappane, Siwei Jiang, and Jie Zhang. A biclustering-based approach to filter dishonest advisors in multi-criteria e-marketplaces. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 1385–1386. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
68. Hemant Ishwaran and J Sunil Rao. Spike and slab variable selection: frequentist and bayesian strategies. *Annals of Statistics*, pages 730–773, 2005.

69. Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.
70. Andrew E. Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *PAMI*, 21(5):433–449, 1999.
71. Sarang C Joshi, Michael I Miller, and Ulf Grenander. On the geometry and shape of brain sub-manifolds. *International Journal of Pattern Recognition and Artificial Intelligence*, 11(08):1317–1343, 1997.
72. Mark Kac. Can one hear the shape of a drum? *The American Mathematical Monthly*, 73(4):1–23, 1966.
73. E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D mesh segmentation and labeling. *TOG*, 29(3), 2010.
74. Zachi Karni and Craig Gotsman. Spectral compression of mesh geometry. In *Proc. CGIT*, pages 279–286, New York, NY, 2000. ACM Press/Addison-Wesley Publishing Co.
75. Mehdi Kaytoue, Victor Codocedo, Aleksey Buzmakov, Jaume Baixeries, Sergei O Kuznetsov, and Amedeo Napoli. Pattern structures and concept lattices for data mining and knowledge processing. In *Machine Learning and Knowledge Discovery in Databases*, pages 227–231. Springer, 2015.
76. V. G. Kim, Yaron Lipman, and T. Funkhouser. Blended intrinsic maps. *ACM Transaction on Graphics (TOG)*, 30(4):79:1–79:12, 2011.
77. Ron Kimmel and James A Sethian. Computing geodesic paths on manifolds. *PNAS*, 95(15):8431–8435, 1998.
78. J. Koenderink and A. van Doorn. Surface shape and curvature scales. *Image and Visual Computing*, 10:557–565, 1992.
79. Iasonas Kokkinos, Michael M. Bronstein, Roee Litman, and Alexander M. Bronstein. Intrinsic shape context descriptors for deformable shapes. In *Proc. CVPR*, 2012.
80. A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst. Functional correspondence by matrix completion. In *Proc. CVPR*, pages 905–914, Boston, MA, 2015. IEEE.
81. A. Kovnatsky, K. Glashoff, and M. M. Bronstein. MADMM: a generic algorithm for non-smooth optimization on manifolds. In *Proc. ECCV*, pages 680–696, Cham, Switzerland, 2016. Springer International Publishing.
82. Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.
83. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.
84. R. Lehoucq, D. Sorensen, and C. Yang. Arpack users' guide: Solution of large scale eigenvalue problems with implicitly restarted Arnoldi methods, 1997.
85. R. B. Lehoucq and D. C. Sorensen. Deflation techniques for an implicitly restarted Arnoldi iteration. *SIAM Journal on Matrix Analysis and Applications*, 17(4):789–821, 1996.
86. Bruno Lévy. Laplace-Beltrami eigenfunctions towards an algorithm that understands geometry. In *Proc. SMI*, pages 13–25, Washington, DC, 2006. IEEE.
87. Bruno Lévy and Hao Richard Zhang. Spectral mesh processing. In *SIGGRAPH 2010 Courses*, pages 8:1–8:312, New York, NY, 2010. ACM.
88. Zhouhui Lian et al. A comparison of methods for non-rigid 3D shape retrieval. *Pattern Recognition*, 46(1):449–461, 2013.
89. Roee Litman, Alex Bronstein, Michael Bronstein, and Umberto Castellani. Supervised learning of bag-of-features shape descriptors using sparse coding. *CGF*, 33(5):127–136, 2014.
90. Roee Litman and Alexander M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *PAMI*, 36(1):170–180, 2014.
91. Derek Liu, Alec Jacobson, and Keenan Crane. A Dirac operator for extrinsic shape analysis. *Computer Graphics Forum*, 36(5), 2017.
92. Herve Lombaert, Michael Arcaro, and Nicholas Ayache. Brain transfer: Spectral analysis of cortical surfaces and functional maps. In *International Conference on Information Processing in Medical Imaging*, pages 474–487, 2015.
93. David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
94. Richard H MacNeal. *The solution of partial differential equations by means of electrical networks.* PhD thesis, California Institute of Technology, 1949.

95. S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE TCBB*, 1:24–44, 2004.
96. G Maicas, AI Muñoz, G Galiano, A Ben Hamza, E Schiavi, Alzheimer's Disease Neuroimaging Initiative, et al. Spectral shape analysis of the hippocampal structure for alzheimer's disease diagnosis. In *Trends in Differential Equations and Applications*, pages 17–32. 2016.
97. Stéphane Mallat. *A wavelet tour of signal processing.* Academic press, 1999.
98. Siddharth Manay, Daniel Cremers, Byung-Woo Hong, Anthony J Yezzi, and Stefano Soatto. Integral invariants for shape matching. *PAMI*, 28(10):1602–1618, 2006.
99. J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. ShapeNet: Convolutional neural networks on non-Euclidean manifolds. *arXiv:1501.06297*, 2015.
100. Diana Mateus, Radu Horaud, David Knossow, Fabio Cuzzolin, and Edmond Boyer. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *Proc. CVPR*, 2008.
101. S. Melzi, E. Rodolà, U. Castellani, and M. M. Bronstein. Localized manifold harmonics for spectral shape analysis. *Computer Graphics Forum*.
102. Simone Melzi, Alessandro Mella, Letizia Squarcina, Marcella Bellani, Cinzia Perlini, Mirella Ruggeri, Carlo Alfredo Altamura, Paolo Brambilla, and Umberto Castellani. *Functional Maps for Brain Classification on Spectral Domain*, pages 25–36. Springer International Publishing, 2016.
103. Simone Melzi, Maks Ovsjanikov, Giorgio Roffo, Marco Cristani, and Umberto Castellani. Discrete time evolution process descriptor for shape analysis and matching. *Transaction on Graphics (TOG)*, 2017.
104. Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael Bronstein. Shape analysis with anisotropic windowed fourier transform. In *Proc. International Conference on 3D Vision (3DV)*, pages 470–478, Stanford, CA, 2016. IEEE.
105. Joseph SB Mitchell, David M Mount, and Christos H Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.
106. Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.
107. Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos A Coello Coello. Survey of multiobjective evolutionary algorithms for data mining: Part ii. *Evolutionary Computation, IEEE Transactions on*, 18(1):20–35, 2014.
108. Patrick Mullen, Yiying Tong, Pierre Alliez, and Mathieu Desbrun. Spectral conformal parameterization. 27(5):1487–1494, 2008.
109. A. Myronenko and X. Song. Point set registration: Coherent point drift. *TPAMI*, 32(12):2262–2275, Dec 2010.
110. Thomas Neumann, Kiran Varanasi, Christian Theobalt, Marcus Magnor, and Markus Wacker. Compressed manifold modes for mesh processing. *Computer Graphics Forum*, 33(5):35–44, 2014.
111. Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017.
112. Ali Oghabian, Sami Kilpinen, Sampsa Hautaniemi, and Elena Czeizler. Biclustering methods: Biological relevance and application in gene expression analysis. *PloS one*, 9(3), 2014.
113. Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing.* Prentice Hall Press, Upper Saddle River, NJ, USA, 2009.
114. Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Shape distributions. *TOG*, 21(4):807–832, 2002.
115. Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012.
116. Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, pages 9:1–9:60, 2016.
117. Maks Ovsjanikov, Quentin Mérigot, Facundo Mémoli, and Leonidas Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564, 2010.
118. Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

119. Mark Pauly, Richard Keiser, and Markus Gross. Multi-scale feature extraction on point-sampled surfaces. *CGF*, 22(3):281–289, 2003.
120. Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2(1):15–36, 1993.
121. Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. In *ArXiv*, March 2015.
122. Jonathan Pokrass, Alexander M Bronstein, Michael M Bronstein, Pablo Sprechmann, and Guillermo Sapiro. Sparse modeling of intrinsic correspondences. In *Computer Graphics Forum*, volume 32, pages 459–468. Wiley Online Library, 2013.
123. Roland Priemer. *Introductory Signal Processing*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1991.
124. Hamed Rabiei, Frédéric Richard, Muriel Roth, Jean-Luc Anton, Olivier Coulon, and Julien Lefèvre. The graph windowed fourier transform: a tool to quantify the gyrification of the cerebral cortex. In *Workshop on Spectral Analysis in Medical Imaging (SAMI)*, 2015.
125. Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace–beltrami spectra as "shape-dna" of surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.
126. Kathleen M Robinette, Hans Daanen, and Eric Paquet. The caesar project: a 3-d surface anthropometry survey. In *Second International Conference on 3-D Digital Imaging and Modeling (3DIM)*, pages 380–386. IEEE, 1999.
127. E. Rodolà, A.M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *Proc. CVPR*, pages 182–189, Washington, DC, June 2012. IEEE.
128. E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017.
129. E. Rodolà, M. Moeller, and D. Cremers. Point-wise map recovery and refinement from functional correspondence. In *Proc. VMV*, pages 25–32, Aachen, Germany, 2015. Eurographics Association.
130. Emanuele Rodolà, Andrea Albarelli, Filippo Bergamasco, and Andrea Torsello. A scale independent selection process for 3D object recognition in cluttered scenes. *International Journal of Computer Vision*, 102(1-3):129–145, 2013.
131. Emanuele Rodolà, S Rota Bulò, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4177–4184, 2014.
132. Emanuele Rodolà, Samuel Rota Bulò, and Daniel Cremers. Robust region detection via consensus segmentation of deformable shapes. In *Computer Graphics Forum*, volume 33, pages 97–106. Wiley Online Library, 2014.
133. Raif M Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP*, 2007.
134. Raif M Rustamov. Multiscale biharmonic kernels. *Computer Graphics Forum*, 30(5):1521–1531, 2011.
135. Ahmed H Sameh and John A Wisniewski. A trace minimization algorithm for the generalized eigenvalue problem. *SIAM Journal on Numerical Analysis*, 19(6):1243–1259, 1982.
136. Eerika Savia, Kai Puolamäki, and Samuel Kaski. Latent grouping models for user preference prediction. *Machine Learning*, 74(1):75–109, 2009.
137. David I Shuman, Benjamin Ricaud, and Pierre Vandergheynst. A windowed graph fourier transform. In *IEEE Statistical Signal Processing Workshop (SSP)*, pages 133–136, 2012.
138. David I Shuman, Benjamin Ricaud, and Pierre Vandergheynst. Vertex-frequency analysis on graphs. *arXiv:1307.5708*, 2013.
139. Ivan Sipiran and Benjamin Bustos. Harris 3D: a robust extension of the harris operator for interest point detection on 3D meshes. *Visual Computer*, 27(11):963–976, 2011.
140. Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *CGF*, 28(5):1383–1392, 2009.
141. Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J Gortler, and Hugues Hoppe. Fast exact and approximate geodesics on meshes. *ACM Transaction on Graphics (TOG)*, 24(3):553–560, 2005.
142. G. Taubin. Geometric Signal Processing on Polygonal Meshes. In *Eurographics 2000 - STARs*. Eurographics Association, 2000.

143. Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, New York, NY, 1995. ACM.
144. Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *International Conference on Computer Vision (ICCV)*, pages 356–369, 2010.
145. Aydin Ulas, R P W Duin, Umberto Castellani, M Loog, P Mirtuono, Manuele Bicego, Vittorio Murino, Marcella Bellani, Silvio Cerruti, Michele Tansella, and Paolo Brambilla. Dissimilarity-based detection of schizophrenia. *International Journal of Imaging Systems and Technology*, 21(2):179–92, 2011.
146. Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library, 2008.
147. V.N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.
148. E. Veronese, U. Castellani, D. Peruzzo, M. Bellani, and P. Brambilla. Machine learning approaches: from theory to application in schizophrenia. *Computational and Mathematical Methods in Medicine*, 2013:1–12, 2013.
149. M. Vestner, R. Litman, E. Rodolà, A. M. Bronstein, and D. Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *Proc. CVPR*, Honolulu, HI, 2017. IEEE.
150. Gang Wang and Yalin Wang. Multi-scale heat kernel based volumetric morphology signature. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 751–759. 2015.
151. Thomas Windheuser, Matthias Vestner, Emanuele Rodolà, Rudolph Triebel, and Daniel Cremers. Optimal intrinsic descriptors for non-rigid shape analysis. In *Proc. BMVC*. BMVA Press, 2014.
152. Max A. Woodbury. *Inverting modified matrices*. Number 42 in Statistical Research Group Memorandum Reports. Princeton University, Princeton, NJ, 1950.
153. Matthew D. Zeiler. ADADELTA: An adaptive learning rate method. *arXiv:1212.5701*, 2012.