Spectral Signatures for Non-rigid 3D Shape Retrieval

Frederico Artur Limberger

Doctor of Philosophy

University of York

Computer Science

May 2017

Abstract

This thesis addresses problems associated with computing spectral shape signatures for non-rigid 3D object retrieval. More specifically, we use spectral shape analysis tools to describe the characteristics of different 3D object representations. This thesis tries to answer whether spectral shape analysis tools can enhance classical shape signatures to improve the performance of the non-rigid shape retrieval problem. Furthermore, it describes the stages of the framework for composing non-rigid shape signatures, built from the shape Laplacian.

This thesis presents four methods to improve each part of the framework for computing spectral shape signatures. The first stage comprises computing the right shape spectrum to describe 3D objects. We introduce the Kinetic Laplace-Beltrami operator which computes enhanced spectral components from 3D meshes specific to non-rigid shape retrieval and we also introduce the Mesh-Free Laplace Operator which computes more precise and robust spectral components from 3D point clouds. After computing the shape spectrum, we propose the Improved Wave Kernel Signature, a more discriminative local descriptor built from the Laplacian eigenfunctions. This descriptor is used throughout this thesis and it achieves, in most cases, state-of-the-art performances. Then, we define a new framework for encoding sparse local descriptors into shape signatures that can be compared to each other. Here, we show how to use the Fisher Vector and Super Vector to encode spectral descriptors and also how to compute dissimilarities between shape signatures using the Efficient Manifold Ranking. Furthermore, we describe the construction of the Point-Cloud Shape Retrieval of Non-Rigid Toys dataset, aimed in testing non-rigid shape signatures on point clouds, after we evidenced a lack of point-cloud benchmarks in the literature. With these ingredients, we are able to construct shape signatures which are specially built for non-rigid shape retrieval.

Contents

List of Tables

List of Figures

Acknowledgements

My deepest and sincere gratitude goes first and foremost to my supervisor Professor Richard Wilson for their huge support and encouragement over the past three years. I have been very fortunate to work with him and learn from him. I will always remain indebted to him for his guidance, regular meetings and meaningful suggestions that helped me complete this dissertation in a timely manner.

My gratitude also goes to my assessor Dr. Edwin Hancock for his precise assessment of my work. My sincere thanks to all my colleagues and friends at the CVPR group for the help they provided when needed.

Many thanks are due to the Science Without Borders program from Brazil's government for providing me the financial support. Many thanks also to all my professors at UFRGS and UFSM that shaped my knowledge and understanding of computer science.

I am grateful to all my friends in the UK for their help and support during my stay at York and for making my life easier and more enjoyable while I was far away from my home. I would also like to thank all my Brazilian friends that shared this journey with me, for their motivation and encouragement.

Lastly and most importantly, I wish to thank all my family members for their constant support. My very special thanks go to my mother Marta Miriam Bracht Limberger, my father Décio José Limberger and my sister Mabel Thais Limberger to whom I owe everything I am today. I am grateful to them for educating me and preparing me for my future. These years have been tough without them but they have supported me unconditionally. This work would have been impossible without their love and support throughout my entire life.

Declaration

I declare that this thesis is a presentation of original work, which I undertook at the University of York during 2014 - 2017, and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References.

Some parts of this thesis have been published in conference proceedings; where items were published jointly with collaborators, the author of this thesis is responsible for the material presented here.

Conference Papers

- **Frederico A. Limberger** and Richard C. Wilson. **Feature Encoding of Spectral Signatures for 3D Non-Rigid Shape Retrieval**. In Proceedings of the British Machine Vision Conference (BMVC), pages 56.1-56.13. BMVA Press, September 2015.

- Lian, Z.; Zhang, J.; Choi, S.; ElNaghy, H.; El-Sana, J.; Furuya, T.; Giachetti, A.; Guler, R. A.; Lai, L.; Li, C.; Li, H.; **Frederico A. Limberger**; Martin, R.; Nakanishi, R. U.; Neto, A. P.; Nonato, L. G.; Ohbuchi, R.; Pevzner, K.; Pickup, D.; Rosin, P.; Sharf, A.; Sun, L.; Sun, X.; Tari, S.; Unal, G.; Wilson, R. C. **SHREC15 Track: Non-rigid 3D Shape Retrieval**. Eurographics Workshop on 3D Object Retrieval, 2015.

- **Frederico A. Limberger**, Richard C. Wilson, M. Aono, N. Audebert, A. Boulch, B. Bustos, A. Giachetti, A. Godil, B. Le Saux, B. Li, Y. Lu, H.-D. Nguyen, V.-T. Nguyen, V.-K. Pham, I. Sipiran, A. Tatsuma, M.-T. Tran, S. Velasco-Forero. **SHREC'17**

Track: Point-Cloud Shape Retrieval of Non-Rigid Toys. Eurographics Workshop on 3D Object Retrieval, 2017.

- S. Biasotti, E. Moscoso Thompson, M. Aono, A. Ben Hamza, B. Bustos, S. Dong, B. Du, A. Fehri, H. Li, **Frederico A. Limberger**, M. Masoumi, M. Rezaei, I. Sipiran, L. Sun, A. Tatsuma, S. Velasco Forero, R. C. Wilson, Y. Wu, J. Zhang, T. Zhao, F. Fornasa and A. Giachetti. **SHREC17 Track: Retrieval of surfaces with similar relief patterns**. Eurographics Workshop on 3D Object Retrieval, 2017.

Chapter 1

Introduction

Shape analysis has attracted the interest of many researchers in the computer vision and pattern recognition communities in the past few years. This is because three-dimensional (3D) models are becoming more practical and they are appearing more often in many types of applications. These models can be acquired in many different ways using scanners or cameras, and can be captured even from non-expert users in this field. They can also be created or edited by 3D modeling software with a low learning curve. This explains the gigantic number of models available in the Internet for free in databases like Google 3D Warehouse [Google, 2015] containing millions of models. Usually in shape analysis, models are represented by their boundaries or shell, not by their volume.

Shape analysis comprises examining and describing the form of geometric shapes so computers can detect similarities between objects in a database. Principal applications in this case are: search for deformations of the same object [Lian et al., 2011, Lian et al., 2015, Pickup et al., 2014b], search for objects from same categories [Biasotti et al., 2014], match parts of the shapes or find correspondences [Bronstein et al., 2010a], mesh segmentation [Chen et al., 2009], understand shape changes for medical purposes [Castellani et al., 2011] and face recognition [Samir et al., 2006, Maalej et al., 2011]. The problem is that shapes are represented in the three-dimensional space without any standardization on size, orientation or discretization. Therefore, analyzing shapes in order to find similarities between 3D models is not a matter of comparing vertices but finding more robust representations that can be compared properly and efficiently.

1.1 The problems

The usual way to compare digital objects is by using a signature, also called fingerprint (the name was used for the first time in shape analysis by Reuter [Reuter et al., 2005]), that carries most important information about the shape. It is desirable that this representation is stable against non-rigid deformations of the same shape and typical rigid transformations that 3D models undergo, like rotation, mirroring, etc. Shape signatures usually represent local characteristics of shapes, which are associated with their vertices. They usually capture information about the neighborhood of a vertex and so they can be directly applied to some important tasks like point correspondence and shape segmentation. There are many ways to address the problem of representing the local properties of surfaces. The most well known methods use different variations of the scale-invariant feature transform (SIFT) [Lowe, 1999] or the Laplace-Beltrami operator (LBO).

Another constraint, that comes from how the data is registered, is the way shapes are stored in memory. The two most common ways are using point clouds or meshes. The problem is that methods designed to compute descriptors depend on the way shapes are described. This is a problem since point clouds are less informative than meshes when both are equally dense. For computing the Laplace-Beltrami Operator, there exist methods to deal with both representations, although existing methods are only approximations for the true LBO. The most well know are the *cotangent scheme* [Meyer et al., 2003] for meshes and the *PCD Laplace (Point-Cloud Data Laplace)* [Belkin et al., 2009] for point clouds.

On the other hand, local descriptors cannot be immediately applied to the problem of shape retrieval, because this task is not addressed by comparing descriptors, but by comparing global signatures. In this thesis, we refer to *signatures* as the global shape representation and to *descriptors* as the local shape representation (coupled to vertices). Creating signatures is not a simple task since shapes can have arbitrary number of vertices, edges and faces. To create an ideal generic representation of a shape model all important characteristics shall be preserved during the encoding process. Even so, the global representation must compress local characteristics using the same representation in order to facilitate comparisons.

There are many different approaches to encode descriptors into signatures. They succeed

in representing local features in a same-basis space, however, there is not one encoding process that is always better than others. They depend on the characteristics of the descriptors.

Regarding the problem being faced (dataset characteristics), it is necessary to compute different local descriptors which handle important deformations in a specific manner. For instance, in the non-rigid shape retrieval problem, descriptors have to be invariant where joint motions occur even though their neighborhood is different after the motion. To address, for instance, a benchmark of solid objects, like chairs, wardrobes and beds, it is important to have a signatures that can learn the common transformations of same-class objects.

1.2 Our goals

In our study, we are interested in coding different aspects of shapes to improve the efficiency of spectral descriptors in shape analysis. More specifically, we are interested in a general framework for spectral signatures, which computes a designed LBO for the problem in hand and generates descriptors which carry all important shape characteristics. After computing descriptors, it is necessary to compute a fast and stable encoding scheme which does not reduce the descriptor potential. Finally, it is necessary to compute signature discrepancies in a fair way that reflect the real differences between objects. With all these components, the questions that we want to answer are: Can we use spectral analysis to construct enhanced spectral descriptors which can perform at state-of-the-art performance? Is it possible to achieve efficient shape retrieval using only spectral components? If not, what are the tools that we need to increase spectral shape retrieval performance? Our efforts are aimed to give the best answers to these questions throughout this thesis.

1.3 Contributions

To achieve the goal of creating enhanced spectral signatures for 3D shape retrieval we make the following contributions in this thesis.

1.3.1 Improved Wave Kernel Signature

In spectral shape analysis, the shape spectra is represented by the eigenvalues and eigenfunctions of the eigendecomposition of the Laplace-Beltrami operator. These statistics are used to describe localized characteristics of the shape. By using the different frequencies of the eigenfunctions it is possible to characterize one particular position of the shape in many different ways by varying the vibration modes. One of the problems when creating a spectral shape signature is how to model the weighting of the shape spectra, which will define how much information about local and global aspects shall be taken into account.

The weighting is described by a filter which controls the significance of different frequencies. Whether the majority of the weightings are concentrated in the low frequencies, only the major topological structure of the object will be captured by the signature. On the other hand, if we account all the shape spectra equally, the high frequencies will aggregate noise to the descriptor, since these relate to information from different parts of the shape.

After analyzing the effects of different weightings to the WKS, we proposed to create filters which are more stable to the gross structure of the shape but also aggregate more high frequency information about the shape. By creating more stable filters along the frequencies of the shape spectrum it is possible to create a better signature for non-rigid shape retrieval which captures both the major structure of the object and the fine details at the same time.

1.3.2 Robust encoding framework

Once local descriptors are computed from 3D models, it is necessary to write these descriptors on a commensurable basis so they can be compared to each other. There exist many different encoding schemes that have this goal, however, some are more robust than others. In this thesis, we propose to use state-of-the-art encoding schemes in conjunction with spectral signatures. We show how to encode local spectral descriptors with Fisher Vector [Perronnin and Dance, 2007] and Super Vector [Zhou et al., 2010]. This combination had never been done before. Furthermore, we show experimentally that is possible to use these encoding methods with spectral descriptors once they can be approximated by Gaussian Mixture Models. We present experiments which prove that this aggregation increase the performance of spectral

signatures.

For this, we exhaustively evaluate the performance of our framework by a series of experiments computed over many different spectral descriptors, datasets and settings. In the end, we compare our results with state-of-the-art signatures.

1.3.3 Kinetic Laplace-Beltrami operator

Given that the shape spectra should be more informative and discriminative than the results were showing us, we decided to undertake the more fundamental strategy of designing an LBO that is specialized in non-rigid shape retrieval. The Kinetic Laplace-Beltrami operator (KLBO) takes a more general approach to modify the dynamics that occur on the shape's surface.

Therewith, we introduce a new framework that can be modified to deal with the problem at hand. We propose a number of different weightings to bias most important features on the model, leading to a formulation which is specially built for the problem. We explain how it is possible to deal with the non-rigid shape retrieval problem by weighting the kinetic energy by a map that is small in areas on the shape where articulations are likely to occur and stable to non-rigid motions.

We present many experiments that show that the KLBO is very good at retrieving non-rigid shapes, comparing to the classic LBO. We also show that KLBO descriptors can also be used with state-of-the-art encoding schemes, since descriptors can be approximated by GMMs. In the end, we show a full comparison with other signatures designed for shape retrieval.

1.3.4 Non-rigid point-cloud dataset

On the other hand, meshes are not the only discrete shape representation. 3D point-cloud objects are the immediate result of 3D scans of real 3D objects. Given the necessity of comparing 3D non-rigid shapes based directly on a rough 3D scan of the object we have created a point-cloud dataset which aims in evaluating how methods perform on the non-rigid point-cloud retrieval task. In this way, our dataset is the first of its kind. In this task, a shape retrieval method has to worry about common scanning problems, mainly caused by

object self-occlusions, however, this task cannot be classified as a part-based shape retrieval problem because the gross structure of the shape is always presented (only fine details are missing). Therefore, we created this dataset to promote the development of non-rigid 3D shape retrieval of point clouds.

This dataset has the objective of evaluating shape retrieval methods that can be computed directly and efficiently from point clouds. For this, we present a dataset named *PRoNTo: Point-Cloud Shape Retrieval of Non-Rigid Toys*, which has been scanned by us and contains 100 3D non-rigid point clouds, divided in 10 classes, with 10 model per class. Every point cloud was resampled to two sizes: one with 4K points and another with 10K points to demonstrate the effect of different sampling ratios on the descriptors. All models were sampled to the same number of points in either datasets so that variations in model sizes does not influence the descriptors.

The dataset is available in the SHREC'17 PRoNTo website [Limberger and Wilson, 2017]. We show in the end a summary of the methods submitted to the benchmark and evaluation statistics resulted from shape retrievals on the dataset. From there, we draw our conclusions based on the results.

1.3.5 Mesh-Free Laplace-Beltrami operator

The Mesh-Free Laplace operator (MFLO) is an optimized operator for computing an isotropic discrete Laplace operator for point clouds. The MFLO does not need to estimate a local tangent plane nor compute a Voronoi diagram for each sample like other approaches [Belkin et al., 2009, Liu et al., 2012], speeding up Laplacian-matrix computation. Our operator approximates the Laplace-Beltrami operator on the unknown manifold which the point cloud was sampled, under certain sampling conditions.

The standard Graph Laplacian (GL) can be used to compute the Laplacian of point clouds. However, there are two problems with constructing this operator for a point cloud. Firstly, the point cloud does not have a mesh and therefore we need to produce a connection scheme for the points to construct a graph. Secondly, some arbitrary connection scheme will not faithfully reproduce the LBO of the underlying surface. To replicate the Laplace-Beltrami operator on the underlying manifold a weighting scheme must account the distance

between points on the manifold and not on the Euclidean space.

With this in mind, we approximate manifold distances using Taylor series of the function on the manifold in normal co-ordinates at every point of the point cloud. This scheme approximates the underlying manifold if points are isotropically sampled from the manifold and sufficiently dense.

We show experiments of the MFLO for computing the LBO and constructing spectral signatures for respective point clouds. Then, we perform shape retrieval in the PRoNTo dataset, and also in the sampled versions of SHREC'10, SHREC'11 and SHREC'15 non-rigid datasets. We also compare the MFLO with other methods that compute the LBO for point clouds: the PCDLaplace [Belkin et al., 2009], Point-Based Manifold Harmonics [Liu et al., 2012] and the Graph Laplacian.

1.4 Thesis structure

The remainder of the thesis is organized as follows: Chapter 2 reviews the research literature related to the work presented in this thesis. Chapter 3 shows how we improve the WKS to non-rigid shape retrieval and Chapter 4 shows how to create global signatures from local descriptors using state-of-the-art encoding methods. Then, Chapter 5 presents improvements to the computation of the LBO for non-rigid shape retrieval by weighting the kinetic energy on the manifold. Furthermore, Chapter 6 describes the conception of the non-rigid point-cloud dataset, created to test non-rigid shape retrieval of point-cloud shapes. Chapter 7 shows how to compute a point-cloud version of the discrete LBO using an isotropic edge-weighting scheme to characterize features on the unknown underlying manifold. Chapter 8 concludes this thesis and points to several problems that deserve further research.

Chapter 2

Literature Review

In this section, we review methods related to different formulations for the Laplace-Beltrami operator, non-rigid shape descriptors proposed in the literature and also methods to encode local descriptors into shape signatures. First, we define the LBO and then comment on related methods. Second, we group non-rigid shape descriptors into three categories: spectral-based, geometry-based, and learning-based descriptors, and discuss their innovations. Then, we review feature encoding methods and, finally, we detail non-rigid shape retrieval datasets proposed in the literature.

2.1 Laplace-Beltrami operator

The Laplace operator generalized to operate on functions defined on a Riemannian manifold $\mathcal{M}$ (2D in our case) is known as the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$. It is a linear operator defined as the divergence of the gradient taking functions into functions

$$\Delta_{\mathcal{M}} f = - \bigtriangledown \cdot \bigtriangledown_{\mathcal{M}} f \tag{2.1}$$

given that f is a twice-differentiable real-valued function. The negative sign is simply to respect the standard convention for graph Laplacians. The eigendecomposition of the LBO, represented by its eigenvalues and eigenfunctions, is known as the spectrum of the shape.

In the discrete case ($\Delta_M f$), there are several methods concerned in computing the LBO for meshes [Taubin, 1995, Desbrun et al., 1999, Pinkall et al., 1993, Meyer et al., 2003, Reuter

et al., 2006, Xu, 2006, Levy, 2006, Belkin et al., 2008]. Meyer *et al.* proposed the cotangent weight scheme [Meyer et al., 2003] which we utilize in our construction. We consider $f : \mathcal{V} \rightarrow \mathbb{R}$, where the ith component $f(i)$ is the function value at the vertex i in $\mathcal{V}$. Using Meyer et al.'s discretization, the LBO is written as

$$\Delta_M f = Lf = A^{-1}Wf \tag{2.2}$$

where L is the discrete Laplace operator, A is a positive definite diagonal matrix and the elements of W are given by

$$W(i,j) = \begin{cases} \frac{(\cot \alpha_{ij} + \cot \beta_{ij})}{2} & \text{if } (i,j) \in E, \\ -\sum_{k \neq i} W(i,k) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

where α_{ij} and β_{ij} are the angles $\angle(\mathbf{v}_i\mathbf{v}_a\mathbf{v}_j)$ and $\angle(\mathbf{v}_i\mathbf{v}_b\mathbf{v}_j)$ as shown in Fig. 2.1. The diagonal elements A_{ii} are the Voronoi areas (green polygon) associated to the vertex $\mathbf{v}_i$.

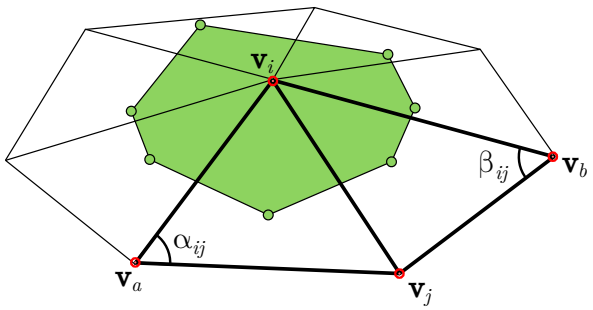


Figure 2.1: Illustration of the angles α_{ij} and β_{ij}, and the Voronoi area (green polygon) associated to the vertex $\mathbf{v}_i$ of the cotangent weight scheme [Meyer et al., 2003].

In order to find the spectrum of the shape, the eigendecomposition of the LBO is computed. The eigenvectors and eigenfunctions are solutions of the generalized eigenproblem

$$L\phi = \lambda\phi \tag{2.4}$$

$$W\phi = \lambda A\phi. \tag{2.5}$$

Xu [Xu, 2004] showed that the cotangent weight scheme does not converge in general, and Belkin *et al.* [Belkin et al., 2008] proposed a new discretization method that converges even when meshes present imperfections.

Recently, Andreux *et al.* [Andreux et al., 2014] proposed an anisotropic LBO which keeps key properties of the isotropic LBO and benefits from a more semantically meaningful diffusion process. Later, Boscaini *et al.* [Boscaini et al., 2016] constructed shape descriptors from anisotropic oriented diffusion kernels which were learned using a deep learning architecture.

There has been also an increasing focus on computing the discrete LBO for point sets. Belkin *et al.* [Belkin et al., 2009] introduced a sort of meshing procedure where a local tangent space is estimated for each point. From there, they compute an operator, named PCDLaplace, that converges to the LBO as the point cloud become denser. Luo *et al.* [Luo et al., 2009] compute a Voronoi weighting scheme to estimate integrals over the manifold using a geodesic metric. Later, Liu *et al.* [Liu et al., 2012] built a symmetrizable discrete LBO over point-sampled manifold surfaces, named *Point-Based Manifold Harmonic Transform* (PB-MHT). Differently from Luo et al.'s work, they construct Voronoi diagrams based on the Euclidean distance, and show that their solution converges better than previous techniques.

After computing the shape spectra (eigenvalues and eigenfunctions), it can be used in many ways to represent the intrinsic shape. The Laplacian matrix L has many properties when computed over an undirected graph. For example, the multiplicity of the eigenvalue zero of the Laplacian matrix is equal to the number of connected components of the graph. For computing appropriate descriptors, it is necessary to take this into consideration. Also, good methods for computing L create a positive-semidefinite matrix, which generates only positive eigenvalues. In the next section, we will focus in showing methods that use the shape spectra to create shape representations.

2.2 Spectral-based descriptors

In the past years, spectral-based descriptors have been earned attention by the community since they satisfy most of the desirable properties of local descriptors. Spectral-based techniques take advantages of intrinsic properties of spectral graph theory to present an elegant

solution to non-rigid shape retrieval. This solution relies on the analysis of the eigensystem (eigenvalues and eigenfunctions) of the Laplace-Beltrami operator.

To create a spectral descriptor one needs to organize the shape spectra in a way that it is representative for the respective application. The shape spectra is the spectrum of the Laplace-Beltrami operator of the shape. The LBO of the shape can be computed from meshes and point clouds using discrete versions of the LBO, as shown in the previous section. Then, spectrum of the LBO is given by the eigendecomposition of the LBO matrix, which results in its eigenvalues and eigenfunctions (λ, ϕ).

These spectral components were first used to represent shapes by Reuter *et al.* [Reuter et al., 2005] in 2005. They used the eigenvalues of the LBO sorted in ascending order as a shape fingerprint for shape identification and comparison. Soon after, Rustamov [Rustamov, 2007] incorporated the eigenfunctions of the LBO to describe the global shape using GPS embedding. Since then, descriptors normally use both the eigenvalues and eigenfunctions to represent object's shapes. In the following section, we describe in more details the principal shape signatures created from the shape's spectral components.

2.2.1 Shape DNA

Creating signatures using the spectra of the shape was first addressed by Reuter *et al.* [Reuter et al., 2005, Reuter et al., 2006]. In this seminal paper, the authors use a collection of eigenvalues of the LBO to represent local properties

$$\text{ShapeDNA} = 0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \lambda_4 \leq \ldots \leq \lambda_n. \tag{2.6}$$

Although Reuter's signature ensures that it can recognize shapes in different poses, there exist compact non-isometric shapes that have the same spectra, therefore, they cannot be fully distinguished only by the eigenvalues of the LBO.

2.2.2 Global point signature

To overcome Shape DNA's drawback, Rustamov [Rustamov, 2007] uses all the spectra (eigenvalues and eigenvectors) of a shape to create the Global Point Signature (GPS). The GPS

of a point $\mathbf{p}$ is giving by

$$\mathrm{GPS}(\mathbf{p}) = \left(\frac{1}{\sqrt{\lambda_1}}\phi_1(\mathbf{p}), \frac{1}{\sqrt{\lambda_2}}\phi_2(\mathbf{p}), \frac{1}{\sqrt{\lambda_3}}\phi_3(\mathbf{p}), \ldots, \frac{1}{\sqrt{\lambda_n}}\phi_n(\mathbf{p}) \right) \tag{2.7}$$

He also proposes an isometric-invariant shape embedding, where the global signature is computed by a histogram of pairwise distances [Osada et al., 2002] between points in the embedding space. This approach solves the previous problem of distinguishing non-isometric shapes with the same spectra but introduces the sign correction problem to the eigenfunctions.

2.2.3 Heat kernel signature

Sun *et al.* [Sun et al., 2009] introduced a deformation-invariant signature based on the heat kernel, solving the sign correction problem of the GPS. Named the Heat Kernel Signature (HKS), the descriptor is based on the diffusion of heat over the surface of the model, governed by the heat equation

$$\Delta_{\mathcal{M}} u(x,t) = \frac{\partial u}{\partial t}(x,t), \tag{2.8}$$

where u is a function in respect to space and time that requires to satisfy the Dirichlet boundary condition $u(x,t) = 0$ for all $x \in \partial M$ during all t. Given a starting distribution of heat at time t, the purpose is to measure how the heat is diffused across the shape, and, from this distribution, compute a heat-based descriptor. The HKS can be seen as the remaining heat at x between an interval of time $[t_0, t_1]$ ($t_0 > 0$ and $t_1 > t_0$). This is computed by the heat kernel k_t:

$$\mathrm{HKS}(x,t) = k_t(x,x) = \sum_{k=1}^{\infty} \phi_k(x)^2 e^{-\lambda_k t}, \tag{2.9}$$

where λ are the eigenvalues and ϕ the eigenfunctions of the Laplace-Beltrami operator $\Delta_{\mathcal{M}}$.

This formulation with the heat kernel has a number of good properties:

- **Intrinsic**: The invariance of the LBO leads to an invariance of the heat kernel under isometric transformations.
- **Informative**: The heat kernel contains all the information about geometric properties of the shape.

- **Multi-scale**: It contains information about different scales (time measurements) leading to local and global knowledge.

- **Stable**: The heat kernel is stable under small perturbations of the underlying manifold.

2.2.4 Scale-invariant heat kernel signature

Although the HKS can describe a shape at multiple scales (at different time samples) it is not scale-invariant. This means that if there are two identical shapes but they appear in different sizes, their HKS will be different. Bronstein and Kokkinos proposed a normalization to the HKS that makes it totally scale invariant (SI-HKS) [Bronstein and Kokkinos, 2010].

This way, sampling the HKS logarithmically in time ($t = \alpha^\tau$) at each point x results in:

$$h_\tau = HKS(x, \alpha^\tau) \tag{2.10}$$

From now, it is possible to normalize the signature since the scaling of the shape h_τ to h'_τ, results in a shift in time by $s = 2\log_\alpha \beta$ and in a scaling in amplitude by β^2:

$$h'_\tau = \beta^2 h_{\tau+s} \tag{2.11}$$

According to Bronstein and Kokkinos [Bronstein and Kokkinos, 2010] it is possible to remove the constant β^2 by taking the logarithm of h, and then the discrete derivative with respect to τ:

$$\dot{h}' = \dot{h}_{\tau+s} \tag{2.12}$$

Then, by taking the discrete Fourier transform it is possible to remove the time shift

$$H'(\omega) = H(\omega)e^{2\pi\omega s} \tag{2.13}$$

where H and H' are the Fourier transform of h and h' respectively, and $\omega \in [0, 2\pi]$. Finally, by taking the Fourier transform modulus it is possible to eliminate the phase and have a

signature that is invariant to its scale

$$|H'(\omega)| = |H(\omega)| \tag{2.14}$$

2.2.5 Wave kernel signature

Aubry *et al.* [Aubry et al., 2011b] introduced the Wave Kernel Signature (WKS), which evaluates the probability of measuring a quantum mechanical particle at a specific location $x \in X$ by varying its energy E. Aubry *et al.* use the Schrodinger equation governed by the wave function $\psi(x,t)$ (Equation 2.15) to describe the quantum mechanical behavior of particles over an object surface.

$$i\Delta_{\mathcal{M}}\psi(x,t) = \frac{\partial\psi}{\partial t}(x,t), \tag{2.15}$$

The Wave Kernel Signature is very similar to the heat equation but it has different induced dynamics. Rather than use different time intervals, Aubry *et al.* compute the descriptor at different energy scales using band-pass filters. The WKS is given by

$$\mathrm{WKS}(E,x) = \sum_{k=1}^{\infty} \phi_k(x)^2 f_E(\lambda_k)^2 \tag{2.16}$$

where f_E (Equation 2.17) is a distribution that can properly characterize shape properties at different scales.

$$f_E(\Lambda_k)^2 = \exp\left[\frac{-(e-\log(\Lambda_k))^2}{2\sigma^2}\right] \qquad e \in [\log\Lambda_1, \log\Lambda_{max}] \tag{2.17}$$

By choosing a log-normal distribution function to account for spectrum variances, the WKS present some undesirable properties which will be further discussed in Chapter 3. These properties have been grounds of investigation in this thesis in order to create a more robust signature. Nonetheless, the WKS can also be made scale invariant following the same framework of Bronstein and Kokkinos [Bronstein and Kokkinos, 2010].

2.2.6 Other spectral methods

Many researchers have made modifications to the computation of previous spectral approaches to increase the retrieval performance of spectral signatures for certain problems. In this section, we detail these methods and comment on their novelty.

Raviv *et al.* [Raviv et al., 2010] extended the heat kernel signature, normally computed from meshes, to a volumetric descriptor computed from the volume of an object. In this case, the LBO is computed from a voxelized version of the 3D model to generate the Volumetric Heat Kernel Signature (VHKS). Another modification to the HKS was made by Castellani *et al.* [Castellani et al., 2011], where they introduce the Global Heat Kernel Signature (GHKS), which is a concatenation of histograms for a fixed number of HKS scales. They used the GHKS for brain classification to distinguish patients affected by schizophrenia.

Li and Hamza [Li and BenHamza, 2013] introduced a spectral graph wavelet framework to retrieve shapes in non-rigid databases, where local descriptors are extracted using the spectral graph wavelet transform. Masoumi *et al.* [Masoumi et al., 2016] improved previous work [Li and BenHamza, 2013] by incorporating the vertex area in the computation of the descriptor. Recently, Ye and Yu [Ye and Yu, 2015] took advantage of functional operators to design a framework specifically for encoding non-rigid geometries by using a context-aware integral kernel operator on a manifold. Li *et al.* [Li et al., 2016] computed a descriptor for non-rigid shape retrieval based on the HKS which is only computed on assigned keypoints to reduce computational complexity. Mohamed and Hamza [Mohamed and Hamza, 2016] created a descriptor based on the spectral shape skeleton computed from the second eigenfunction of the LBO, and compared these skeletons using a graph matching framework.

2.3 Geometry-based descriptors

Geometry-based descriptors use statistics computed on primitive geometric attributes, for instance, distance between any two points and/or histograms of primitive geometric attributes, to characterize 3D models. In the following subsections, we will discuss the main geometry-based descriptors methods proposed in the literature to describe 3D shapes. First, we discuss the two first shape descriptors created to describe objects: Spin Images [Johnson,

1997] and Shape Context [Belongie and Malik, 2000]. Then, we focus our attention to the category of descriptors which use the Scale-Invariant Feature Transform (SIFT) to represent local features. Finally, we describe algorithms which perform Multidimensional Scaling to transform the shape into its canonical form and remove the influence of shape articulations.

2.3.1 Spin images

One of the first works that addresses shape signatures is Spin Images [Johnson, 1997] . Johnson created a parameterization in a local space, around a point and a tangent plane, where intuitively a plane is rotated around the normal vector of each point accumulating the remaining points of the shape as the plane sweeps space. This accumulation of points creates a spin image, which can be used to compare shapes.

Accumulation of points depend on point density of the shape, since a mesh with T points will generate T Spin Images. Johnson uses a map function to map points from the three dimensional space to two dimensional space. Because spin images are generated around a local base space, they are invariant to rigid transformation of the shape. For each point in the model $\mathbf{x}_i$ one needs to compute the parameters α and β (Eq. 2.18) which represent the radial distance and the axial distance, respectively, from the origin $\mathbf{p}_i$ of the normal vector $\mathbf{n}_i$. This two distances create a spin map, which is accumulated into discrete bins to create the Spin Image.

$$(\alpha, \beta) = (\sqrt{\|\mathbf{x} - \mathbf{p}\|^2 - (\mathbf{n} \cdot (\mathbf{x} - \mathbf{p}))^2}, \mathbf{n} \cdot (\mathbf{x} - \mathbf{p})) \tag{2.18}$$

In order to match shapes using spin images, it is necessary to calculate spin images from all points of all models in the database. This way, two spin images P and Q with N bins each are compared using a similarity measure

$$C(P, Q) = \operatorname{arctanh}(R(P, Q))^2 - \lambda \left(\frac{1}{N - 3} \right), \tag{2.19}$$

where R is the correlation coefficient

$$R(P,Q) = \frac{N\sum \mathbf{p}_i\mathbf{q}_i - \sum \mathbf{p}_i \sum \mathbf{q}_i}{\sqrt{(N\sum \mathbf{p}_i^2 - (\sum \mathbf{p}_i)^2)(N\sum \mathbf{q}_i^2 - (\sum \mathbf{q}_i)^2)}}. \tag{2.20}$$

This function returns a large value if two spin images have a large number of overlapping bins. Also, shapes can be compared using compressed spin images, reducing the dimension of the data from M spin images to S Eigen spin images computed by PCA. Then with a projection of these images it is possible to search for a shape using a closest point search procedure and this allows the use of efficient techniques for matching spin images.

2.3.2 Shape context

The idea of Shape Context (SC) is similar with the one of spin images. Belongie and Malik [Belongie and Malik, 2000] create local bases in the shape and represent the shape from each local base. But instead of using an accumulator of points, they use a log-polar histogram of relative coordinates. For this, the boundaries of the objects must be detected and be uniform sampled, but the last it is not mandatory. Initially, they applied SC for measuring shape similarity and recovering point correspondences. Belongie [Belongie et al., 2000] also applied Shape Context for correspondence recovery and 2D shape-based object recognition from pictures taken from objects in distinct views. Later, Kokkinos *et al.* [Kokkinos et al., 2012] developed an intrinsic version of the shape context for 3D models, that can be seen as a meta descriptor which can be applied to any photometric or geometric property field.

2.3.3 Scale-invariant feature transform

Scale-Invariant Feature Transform (SIFT) is a feature descriptor that have been first applied to describe features in images [Lowe, 1999]. Due to its great success it was also extended to describe features in 3D. As the name suggests, it is invariant to scale but also invariant to rotation and translation and partially invariant to light changes and affine 3D projection. SIFTs generally are assigned to high-contrast regions, like on the edges of images. Once they are localized, the descriptor is created by computing an orientation histogram around the keypoint. These keypoints can be matched in different images by identifying their nearest

neighbours.

Many variants of this algorithm have been proposed in the literature: 3D SIFTs were applied to action recognition in video sequences where the third dimension represents the temporal space [Flitton et al., 2010, Laptev et al., 2007, Laptev and Lindeberg, 2006, Scovanner et al., 2007]. Other variants tried to reduce the computation time of SIFT: DSIFT [Vedaldi and Fulkerson, 2008] and SURF [Bay et al., 2008]. When it comes to 3D object descriptors, SIFTs are usually computed from multiple viewpoints of the same object [Ohbuchi et al., 2008, Furuya and Ohbuchi, 2009, Furuya and Ohbuchi, 2014, Ohbuchi and Furuya, 2010] or applied directly to 3D objects by using 3D SURF [Knopp et al., 2010].

In the multi-view scenario, depth images are taken from the object in many different viewpoints. Then, 2D SIFTs are computed from these 2D range images. From there, bag-of-features are computed and compared using the calculated SIFTs as features. In [Ohbuchi et al., 2008], authors compute salient local visual features (SSIFT) from range images. In [Furuya and Ohbuchi, 2009], they chose to change SSIFT for DSIFT, to improve retrieval accuracy of highly articulated models. In [Ohbuchi and Furuya, 2010], they compute different types of SIFTs on the range images (SSIFT, DSIFT and 1SIFT) to aggregate local and global information at the same time and combine them into a unique descriptor.

In [Knopp et al., 2010], 3D SIFTs are extracted directly from the 3D shape. First, the 3D shape is voxelized using the intersection of faces with the 3D grid. Then, features points are chosen as the local extremes of the Hessian filter responses, where S is the absolute value of the determinant of the Hessian matrix

$$S(\vec{x},\sigma) = |H(\vec{x},\sigma)| = \left| \begin{pmatrix} L_{xx}(\vec{x},\sigma) & L_{xy}(\vec{x},\sigma) & L_{xz}(\vec{x},\sigma) \\ L_{yx}(\vec{x},\sigma) & L_{yy}(\vec{x},\sigma) & L_{yz}(\vec{x},\sigma) \\ L_{zx}(\vec{x},\sigma) & L_{zy}(\vec{x},\sigma) & L_{zz}(\vec{x},\sigma) \end{pmatrix} \right|. \tag{2.21}$$

Thus, 3D SURF descriptors are computed around each feature points, and finally, bag-of-features is computed likewise.

Furuya and Ohbuchi [Furuya and Ohbuchi, 2014] fused local features to create a more powerful descriptor. They computed SIFTs and merged them using an anchor manifold

graph. As a more extensive comparison, Furuya and Ohbuchi [Takahiko Furuya, 2015] developed a feature aggregation algorithm named *Diffusion-On-Manifold* to encode local features into a global descriptor. They also proposed a new local feature called *Position and Orientation Distribution*, that describes the oriented-points distribution using a *Sphere-Of-Interest*.

2.3.4 Multidimensional scaling

Many techniques have also been proposed to handle non-rigid shape retrieval by first applying multidimensional scaling (MDS). MDS transforms models into their canonical form, therefore, removing the influence of shape motions and simplifying the computation of 3D descriptors, which do not need to worry about shape articulations.

Elad and Kimmel [Elad and Kimmel, 2003] computed bending invariant signatures by applying MDS on the intrinsic geodesic distances between surface points, computed from the fast marching on triangulated domains algorithm. Lian *et al.* [Lian et al., 2013] propose to apply an algorithm called Clock Matching to depth-buffer images, captured around the 3D objects in their canonical forms. After this, Bag-of-Features is applied to create a quantization of features to every view (CM-BOF). Finally, a multi-view shape matching is applied to compute the dissimilarities between models. Also based on MDS, Li *et al.* [Li et al., 2014a] proposed a hybrid descriptor (MDS-ZFDR) by combining distance-based and curvature-based features. Pickup *et al.* [Pickup et al., 2015] computed canonical forms based on Euclidean distances. By substituting MDS with their canonical-form method in the work of Lian *et al.* [Lian et al., 2013], they created shape descriptors that perform at a comparable accuracy but present lower computation times. In a similar way to canonical forms, Pickup *et al.* [Pickup et al., 2016a] created a descriptor by unbending the skeleton of the mesh and used it with Lian et al.'s work [Lian et al., 2013] to perform non-rigid shape retrieval.

2.4 Learning-based descriptors

Recently, there have been works concerned with applying machine learning methods to geometric data. Note that these are intrinsically different from the previous mentioned ap-

proaches, since previous ones do not use any offline training data. This is a recent and important field which deserves attention. Their drawback is that they require some prior knowledge such as training data so they can learn class attributes.

Litman *et al.* [Litman et al., 2014] worked with supervised learning to replaced the unsupervised construction of the dictionary in BoF by a supervised learning scheme via bi-level optimization. Later, Litman and Bronstein [Litman and Bronstein, 2014] defined a generic family of parametric spectral descriptors, from where HKS and WKS are particular cases. They show how to learn an optimized descriptor by taking into account the statistics of shapes. The new spectral descriptor is learned by modifying the filters (kernel) that are computed from the shape spectra. Boscaini *et al.* [Boscaini et al., 2015] proposed a localized spectral convolutional neural network using the windowed Fourier transform to represent local shape structures. Using this learning procedure, they were able to create class-specific descriptors for deformable shapes. Giachetti *et al.* [Giachetti et al., 2016] used learned subspace projections to differentiate human body subjects using classical signatures (ShapeDNA and Histogram of area projection transform).

2.5 Feature encoding methods

The idea of creating global shape signatures from local descriptors is neither new nor straightforward. Aubry *et al.* [Aubry et al., 2011b] states that comparing 3D shapes is a difficult computational challenge. The most common way of encoding local descriptors is by using histograms of the local feature vectors. This strategy is widely used because it removes the local dependence of each descriptor by writing them as an association to a vocabulary. This vocabulary can be computed by segmenting a training set into semantically coherent clusters (K-means) or by finding the probability of each sample belonging to each cluster (Gaussian Mixture Model). In the next two sections we describe the most common methods to compute a visual vocabulary (K-means and GMM) and in the following sections we describe different encoding schemes.

2.5.1 *K*-means clustering

K-means is the most popular clustering algorithm to construct a geometric vocabulary. In this method, each sample is assigned to the cluster which has the nearest mean. Given a set of observations $\{\mathbf{x}_1, ..., \mathbf{x}_T\}$, k-means can be formulated as a minimization of the squared sums of each cluster:

$$\arg\min_{\mathbf{S}} \sum_{k=1}^{K} \sum_{\mathbf{x} \in S_k} ||\mathbf{x} - \mu_k||^2, \tag{2.22}$$

where μ_k is the mean of points in S_k. K-means is used, in this context, to create a dictionary for either soft or hard encodings, *e.g.* for Histogram encoding or Kernel codebook encoding.

2.5.2 Gaussian mixture model clustering

A Gaussian Mixture Model (GMM) is a probabilistic distribution function u_λ (Eq. 2.23 and 2.24) composed by a set of K Gaussian distributions. Each Gaussian is represented by three parameters w_k, μ_k and Σ_k which represents respectively the weight, the means and the covariance matrix. GMM clustering can be seen as a soft clustering method where to each sample is assigned a probability of belonging to each cluster. Usually an iterative algorithm is used to converge to a local optimum solution in order to find the parameters of each Gaussian. A GMM is used to create dictionaries for the Fisher Vector and the Super Vector.

$$u_\lambda(x) = \sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \Sigma_k) : \sum_{k=1}^{K} w_k = 1 \tag{2.23}$$

$$\mathcal{N}(x|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^D |\Sigma_k|}} \exp[-\frac{1}{2}(x - \mu_k)^\top \Sigma_k^{-1}(x - \mu_k)] \tag{2.24}$$

2.5.3 Histogram encoding

Histogram encoding or Vector Quantization (VQ) is the precursor of all other encodings and it is represented by a histogram of the quantized local descriptors assigned in a hard manner. The vocabulary is built by learning a codebook of size K using k-means. Then, each local descriptor is associated to one codeword. This histogram of all associations is the Histogram encoding vector and it has the same size of the codebook (K). Sivic and Zisserman [Sivic

and Zisserman, 2003] use VQ to search for user outlined objects in videos.

2.5.4 Kernel codebook

Kernel codebook encoding is the standard Bag-of-Feature (BoF) encoding. It encodes local descriptors by assigning them to visual words. This removes the local dependence of each local descriptor by describing the probability of occurrence (soft assignment version) of each local descriptor in a geometric vocabulary. Firstly, a vocabulary of words $P = \{\mathbf{p}_1, ..., \mathbf{p}_K\}$ of size K is generated by an unsupervised learning algorithm like vector quantization through k-means. This algorithm clusters similar features from a training set thus generating the "visual words". After this, local descriptors $X = \{\mathbf{x}_1, ..., \mathbf{x}_T\}$ are assigned to the vocabulary. Each association can be seen as the probability of each descriptor $\mathbf{x}_i$ to be assigned to each visual word $\mathbf{p}_i$. This is computed by:

$$\theta(\mathbf{x}_i) = c(\mathbf{x}_i) \exp\left[-\frac{||\mathbf{x}_i - \mathbf{p}_i||_2^2}{2\sigma^2}\right], \tag{2.25}$$

where $c(x)$ is selected to normalize $\theta(\mathbf{x}_i)$ ($|\theta(\mathbf{x}_i)|_1 = 1$). Integrating this over the entire shape

$$\Gamma(X)_{BoF} = \int_X \theta(\mathbf{x}_i) d\Omega(\mathbf{x}_i), \tag{2.26}$$

leads to the BoF representation of the shape ($\Gamma(X)_{BoF}$), where $\Omega(\mathbf{x}_i)$ is the standard area measure on X, and so the descriptor has the same size as the vocabulary (K).

Ovsjanikov *et al.* [Ovsjanikov et al., 2009] and Bronstein *et al.* [Bronstein et al., 2011] have used BoF to combine spectral signatures (HKS and SI-HKS, respectively) to represent shapes, removing the local dependence of each local descriptor. They calculate the distance between two shapes using a Hamming metric between bag-of-features of each shape in the Hamming space. They also propose the Spatially Sensitive Bags of Features (SSBoF) to avoid losing the spatial relation between geometric words during the encoding process. More recently, Litman *et al.* [Litman et al., 2014] developed a supervised learning approach to construct the dictionary of BoF model, showing significant improvements in performance over the compared methods.

There are other signatures that also make use of the BoF framework but do not use

spectral local descriptors. In [Furuya and Ohbuchi, 2009], Furuya *et al.* use BoF to encode SIFTs generated by depth images rendered in different views from a 3D model. In a recent benchmark of 3D shape retrieval [Li et al., 2014c], Tatsuma (*Depth Buffered Super-Vector Coding*) and Furuya [Ohbuchi and Furuya, 2010] have also applied BoF to their local features. In a benchmark of retrieval of non-rigid 3D human models [Pickup et al., 2014a], Tatsuma (*Bag-of-Features approach with Augmented Point Feature Histograms*), Bu (*High-level Feature Learning for 3D Shapes*) and Li [Li et al., 2014a], besides Litman *et al.* [Litman et al., 2014], have encoded local features using BoF. In both most recent benchmarks, the best results were obtained by techniques that have used the Bag-of-Features framework.

2.5.5 Locality-constrained linear coding

Similar to other encoding methods, Locality-constrained linear coding (LLC) needs to compute a dictionary of "visual words" using k-means. Thus, it projects each descriptor $\boldsymbol{x}_i$ into a set of $M < K$ closest visual words closest to $\boldsymbol{x}_i$. Let $\mu_1, \ldots, \mu_K$ be the visual vocabulary generated by k-means and $\pi_1, \ldots, \pi_M$ the indices of the M closest visual words μ_k closer to $\boldsymbol{x}_i$. The M visual words can be denoted as $B = [\mu_{\pi_1}, \ldots, \mu_{\pi_M}]$. The LLC is given by the coefficients $\alpha \in R^M$ of the approximation $\boldsymbol{x}_i \approx B\alpha$

$$[\Gamma_{LLC}(\boldsymbol{x}_i)]_{\pi_m} = \alpha_m, m = 1, \ldots, M. \tag{2.27}$$

The LLC encoding can be obtained by max-pooling

$$[\Gamma_{LLC}]_j = \max_{i=1,\ldots,N} [\Gamma_{LLC}(\boldsymbol{x}_i)]_j \tag{2.28}$$

Wang *et al.* [Wang et al., 2010] proposed and use LLC to classify images in three benchmarks (Caltech-101, Caltech-256 and Pascal VOC 2007) achieving state-of-the-art performance at the time but being overtaken by [Chatfield et al., 2011] later.

2.5.6 Fisher vector

The Fisher Vector encoding characterizes a large set of vectors by their deviation from a vocabulary, generating a high-dimensional gradient vector representation. Let $X = \{\mathbf{x}_t, \mathbf{x}_t \in \mathbb{R}^D, t = 1...T\}$ be a large set of vectors and $\lambda = \{w_k, \boldsymbol{\mu}_k, \Sigma_k, k = 1...K\}$ a set of parameters of a probability density function (pdf) u_λ, where w_k, $\boldsymbol{\mu}_k$ and Σ_k are respectively the weight, mean vector and covariance matrix of a Gaussian k of the GMM. Let us assume that the generation process of X can be modelled by the parameters of u_λ. The gradient of the log-likelihood, also called *Fisher score*, describes the contribution of each parameter to the generation process and is given by [Jaakkola and Haussler, 1998]:

$$G_\lambda^X = \nabla_\lambda \log u_\lambda(X|\lambda) \tag{2.29}$$

The dimensionality of the gradient vector does not depend on the sample size T, but depends on the dimension D of each sample vector and the number of parameters K of u_λ, since each dimension is evaluated with each parameter of the pdf (Eqs. 2.23 and 2.24). To be able to compare G_λ^X vectors they must be properly scaled. According to [Jaakkola and Haussler, 1998] a measure of pairwise similarity between samples is given by the Fisher Kernel:

$$K(X, Y) = G_\lambda^X F_\lambda^{-1} G_\lambda^Y, \tag{2.30}$$

where F_λ is the *Fisher information matrix*,

$$F_\lambda = E_{x \sim u_\lambda}[G_\lambda^X G_\lambda'^X]. \tag{2.31}$$

Scaling G_λ^X by $F_\lambda^{-1/2}$ give us the FV representation (Γ_λ^X) of the sample X with respect to the parameters λ:

$$\Gamma_\lambda^X = F_\lambda^{-1/2} G_\lambda^X. \tag{2.32}$$

To encode local descriptor using FV, the parameters of a Gaussian Mixture Model (2.23) are estimated using the Expectation Minimization (EM) algorithm in order to optimize a Maximum Likelihood criterion. Then, local descriptors are written wrt. the probabilistic

model, which means express X by its gradient in respect to u_λ. This is done by associating each vector x_t to a mode k in the GMM. First, we compute the association strength (soft assignment) that is given by the posterior probability [Perronnin and Dance, 2007, Perronnin et al., 2010a]:

$$q_{tk} = \frac{\exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_k)]}{\Sigma_{i=1}^{K} \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_i)]}. \tag{2.33}$$

Second, for each mode k and each descriptor dimension $j = 1..D$, we can compute the deviation vectors (gradient) with respect to the mean and covariance, respectively,

$$u_{jk} = \frac{1}{T\sqrt{w_k}} \sum_{i=1}^{T} q_{ik} \frac{x_{ji} - \mu_{jk}}{\sigma_{jk}}, \tag{2.34}$$

$$v_{jk} = \frac{1}{T\sqrt{2w_k}} \sum_{i=1}^{T} q_{ik} \left[\left(\frac{x_{ji} - \mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right]. \tag{2.35}$$

Finally, the FV representation of a shape S is the concatenation of the vectorization of the matrices u and v. It generates a high-dimensional vector with $2KD$ dimensions, where D is the size of each local descriptor:

$$\Gamma_{FV} = \Gamma_\lambda^X = [\mathbf{u}_1^\top \dots \mathbf{u}_K^\top, \mathbf{v}_1^\top \dots \mathbf{v}_K^\top]^\top \tag{2.36}$$

Recently, image retrieval and classification tasks have been improved by the use of FV [Csurka and Perronnin, 2011, Schneider and Tuytelaars, 2014], which has shown many advantages over BoF and being this a particular case of BoF [Sanchez et al., 2013]. Chatfield *et al.* used FV to encode SIFT descriptors achieving state-of-the-art accuracy on Pascal VOC 2007 and Caltech-101 dataset [Chatfield et al., 2011].

In 3D shape analysis FV has been used by Furuya and Onbuchi [Furuya and Ohbuchi, 2014] to encode statistical descriptors (DSIFT and MO1SIFT). They fuse both signatures using a technique called Multi-Feature Anchor Manifold, however, in this case SV and LLC performs better than FV and shows to be more accurate in *SHREC 2014 Large-scale Comprehensive 3D Shape Retrieval* dataset [Li et al., 2014c].

2.5.7 Super vector

Super Vector encoding (SV) [Zhou et al., 2010] is similar to the Fisher Vector encoding. In their framework Zhou *et al.* perform a nonlinear map to create a high-dimensional sparse vector. Differently from FV, SV only considers two-order differences between descriptors and clusters means. Instead of the third-order differences they add another component related to the mass of each cluster. Thus, the magnitude of their signature is $K(D+1)$.

The Super Vector encoding can be calculated by the following expressions:

$$p_k = \frac{1}{N}\sum_{i=1}^{N} q_{ik} \tag{2.37}$$

$$s_k = s\sqrt{p_k} \tag{2.38}$$

$$\mathbf{u}_k = \frac{1}{\sqrt{p_k}}\sum_{i=1}^{N} q_{ik}(\mathbf{x}_t - \boldsymbol{\mu}_k) \tag{2.39}$$

As we can see in Equation 2.39, the Super Vector encoding normalizes each element by the square root of the posterior probability ($\sqrt{p_k}$) in contrast to the prior probability ($\sqrt{w_k}$). The final descriptor (Γ_{SV}) is given by the following concatenation:

$$\Gamma_{SV} = [s_1, \mathbf{u}_1^\top, ..., s_K, \mathbf{u}_K^\top]^\top \tag{2.40}$$

2.6 Non-rigid shape retrieval datasets

Many datasets were proposed in the literature to test shape descriptors robustness against shape articulations. In the following subsections, we present the most important datasets created to test non-rigid shape retrieval performances.

2.6.1 McGill 3D shape benchmark

While Bronstein *et al.* [Bronstein et al., 2006] non-rigid dataset was worried about non-rigid shape correspondences, the *McGill 3D Shape Benchmark* (MSB) [Siddiqi et al., 2008] was the first shape retrieval benchmark focused on testing descriptors performance against shape articulations. The McGill dataset, published in 2008, had some of their models adapted

from the *Princeton Shape Benchmark* (PSB) [Shilane et al., 2004], some downloaded from different repositories and others created by them using CAD modeling tools. The dataset is available in "mesh" form (triangulated) and "voxelized" form, a representation that we rarely see nowadays.

The majority of recent descriptors do not explicitly use the McGill benchmark to test their methods since there have been other benchmarks proposed that use McGill models in their collections. These are the cases of [Lian et al., 2010, Lian et al., 2011, Lian et al., 2015] where they provide the classification ground truth of the data so that descriptors can test their performance. On the other hand, only part of this dataset is focused in non-rigid shape retrieval, which led these models to be adapted to an entire non-rigid dataset later [Lian et al., 2010].

2.6.2 Shape retrieval contest of non-rigid 3D models

The *Shape Retrieval Contest of Non-rigid 3D Models* dataset was proposed as a track of the Shape Retrieval Contest (SHREC) in 2010 by Lian and Godil [Lian et al., 2010]. It is a non-rigid shape retrieval benchmark where its models were taken solely from McGill dataset. 200 models were extracted from it to ensure every class had the same number of models. Since a classification file was given with the dataset, it was then possible to compute accurate statistics about descriptors performance and compare to other algorithms.

2.6.3 Shape retrieval contest of non-rigid 3D watertight meshes

Considering that [Lian et al., 2010] had only 200 models, Lian and Godil [Lian et al., 2011] created a new dataset with 600 non-rigid models in the SHREC in 2011. This time, the mesh models were extracted from a number of databases [TOSCA, 2009, Lian et al., 2010, Shilane et al., 2004, Siddiqi et al., 2008]. All meshes were made watertight, *i.e.*, there are no holes, cracks or missing features on the mesh. This facilitates the formulation of shape descriptors that require a manifold structure.

2.6.4 Shape retrieval of non-rigid 3D human models

There is also another much more challenging non-rigid benchmark, proposed by Pickup *et al.* [Pickup et al., 2014b], which features exclusively human models in two different datasets: one synthetic (SHREC'14S) and another real (SHREC'14R). The synthetic dataset features "made up" data created using DAZ Studio, and the real dataset was created by scanning real human participants. In the Synthetic dataset, models have in average 60K vertices while in the real datasets models have 15K vertices.

The main difference from these datasets is that the disparities between classes are much more smaller than compared to other datasets. This makes harder the detection of features that can distinguish classes, making it a very challenging benchmark.

2.6.5 Non-rigid 3D shape retrieval

To create an even more complex non-rigid dataset, Lian and Zhang [Lian et al., 2015] took different models from [Lian et al., 2011] and more models from 3D Warehouse [Google, 2015]. With all these models in hand, they created 23 deformed version of each model by articulating them around their joints resulting in 1200 models organized in 50 categories. This dataset was published in the SHREC in 2015 and it is the most complete non-rigid dataset already created to test descriptors performance against shape articulations.

Following in Table 2.1, we show a summary of the non-rigid datasets proposed in the literature for non-rigid shape retrieval and their numbers.

2.7 Summary

In this chapter we have examined the principal components related to 3D shape analysis, focusing in shape signatures developed to address the problem of shape retrieval. We have seen that in order to create a shape signature it is commonly first necessary to compute a local descriptor, then create a vocabulary using a set of these descriptors and write them in respect to the vocabulary by using an encoding method.

The review then proceeded to look at the main local descriptors proposed in the literature and how they operate. These have been divided in spectral-based methods, geometry-based

Table 2.1: Number of models, classes, models within the same classes and models sources of each benchmark. The numbers of models within the same classes are the same except in the MSB, which vary between 20 and 30, depending of the class.

Benchmark	# Models	# Classes	# Models within classes	Sources
MSB	457	19	20-30	[Shilane et al., 2004]
SHREC'10	200	10	20	[Siddiqi et al., 2008]
SHREC'11	600	30	20	[TOSCA, 2009, Lian et al., 2010, Shilane et al., 2004, Siddiqi et al., 2008]
SHREC'14S	300	15	20	created by authors
SHREC'14R	400	10	40	created by authors
SHREC'15	1200	50	24	[TOSCA, 2009, Google, 2015, Lian et al., 2011, Shilane et al., 2004, Siddiqi et al., 2008]

methods and learning-based methods. Geometry methods include Spin Images, which was the first 3D shape descriptor created to represent 3D models, Shape Context and many methods based on SIFT. On the other hand, spectral methods concern those which use the Laplace-Beltrami operator to compute their signature. Among these, it is important to mention Shape DNA, GPS, HKS, SIHKS and WKS. Nonetheless, learning-based methods try to learn relevant class dissimilarities to create descriptors with are easily distinguishable.

The review then considers techniques to combine local descriptors, called encoding schemes. The main methods commonly used are VQ, BoF, FV and SV. Furthermore, it also reviews two methods to create visual vocabularies: K-means and GMM clustering.

Finally, last section of the literature review details recent non-rigid shape retrieval benchmarks created to test shape signatures performances. The three more important here are Shape retrieval contest of non-rigid 3D models (SHREC'10), Shape retrieval contest of non-rigid 3D watertight meshes (SHREC'11) and Non-rigid 3D shape retrieval (SHREC'15). The next chapters report efforts being done to improve shape retrieval by using improved spectral signatures on the principal non-rigid shape retrieval benchmarks.

Chapter 3

Improved Wave Kernel Signature

In this chapter, we detail the investigation made on the Wave Kernel Signature, more specifically, on the WKS filters that weight the shape spectra to create shape descriptors. As said previously, the log-normal distribution function used to handle the spectrum variance within same class shapes presents some problems. First, it blurs the high frequencies of the spectrum and, second, it concentrates the majority of weightings in the low frequencies of the spectrum. Therefore, in this chapter we propose improvements to the WKS in these two directions. We also propose improvements in the direction of a more robust shape signature to the non-rigid shape retrieval problem by aggregating shape curvatures into the descriptor formulation.

The chapter was published in [Limberger and Wilson, 2015] and it is organized in the following way. We introduce the topic and motivate the reader in Section 3.1. Then, we explain how we deal with the eigenvalue weighting problems in Section 3.2. We show how to create a signature which is more robust to non-rigid shape deformations in Section 3.3. Finally, in Section 3.4, we show experiments where we are able to observe improvements due to the two proposed changes. Retrieval performances are shown only in Chapter 4, where a global signature for the shape is computed from the new local descriptors.

3.1 Introduction

Spectral methods have gained increased attention recently for their advantageous properties. They are intrinsic by construction, invariant to isometric deformations, relatively stable against articulations and can be efficiently computed for 3D shapes (they do not require vertex correspondences between models). Applications of spectral methods can be seen in several areas: shape analysis [Levy, 2006], shape retrieval [Reuter et al., 2005], correspondence [Bronstein et al., 2010a] and segmentation [Chen et al., 2009]. The main idea of spectral methods is to use the eigenvalues and eigenfunctions of the (discrete) Laplace-Beltrami operator (LBO) to compute spectral signatures, which capture local and global information of the shape.

The traditional Wave Kernel Signature (WKS) [Aubry et al., 2011b] has proved to be a useful tool for finding non-rigid shape correspondences. In their paper, Aubry *et al.* used the solutions of the Schrödinger's equation to create a wave-based descriptor for shape analysis. The WKS is a computed by a collection of band-pass filters applied over the shape spectrum as shown in Figure 3.2. Despite its correspondence functionality, as initially proposed, it can also be applied to non-rigid shape retrieval.

The Heat Kernel Signature [Sun et al., 2009] was also applied to retrieve shapes in many other occasions [Bronstein et al., 2011, Bronstein and Kokkinos, 2010, Litman et al., 2014, Ovsjanikov et al., 2009]. The HKS applies low-pass filters to the shape spectrum, which can be seen in Figure 3.1, acquiring mainly global shape information.

We performed an analysis of the local descriptors cited above, along with the SIHKS, to understand the reasons why these signatures can satisfactorily retrieve models from non-rigid databases. Nonetheless, we then investigate ways to improve retrieval performances of these descriptors. By analyzing the HKS and WKS filters (Figure 3.1 and 3.2) we can see that the WKS aggregates more information about the shape spectrum in the descriptors. It also usually achieves better retrieval scores when compared to the HKS and SIHKS. For these reasons, we investigate ways to improve the WKS to better describe features for 3D non-rigid shape retrieval.

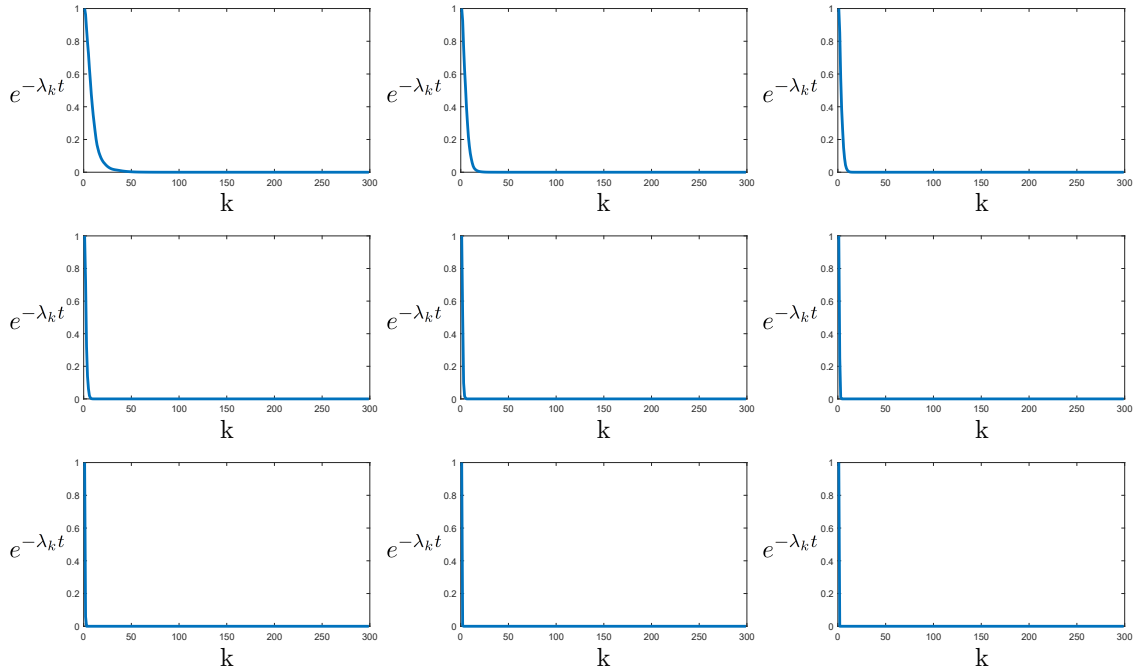


Figure 3.1: Illustration of the HKS's low-pass filters ($e^{-\lambda_k t}$) applied to the shape spectrum. There are shown the frequencies 20, 30, 40, 50, 60, 70, 80, 90 and 100, respectively.

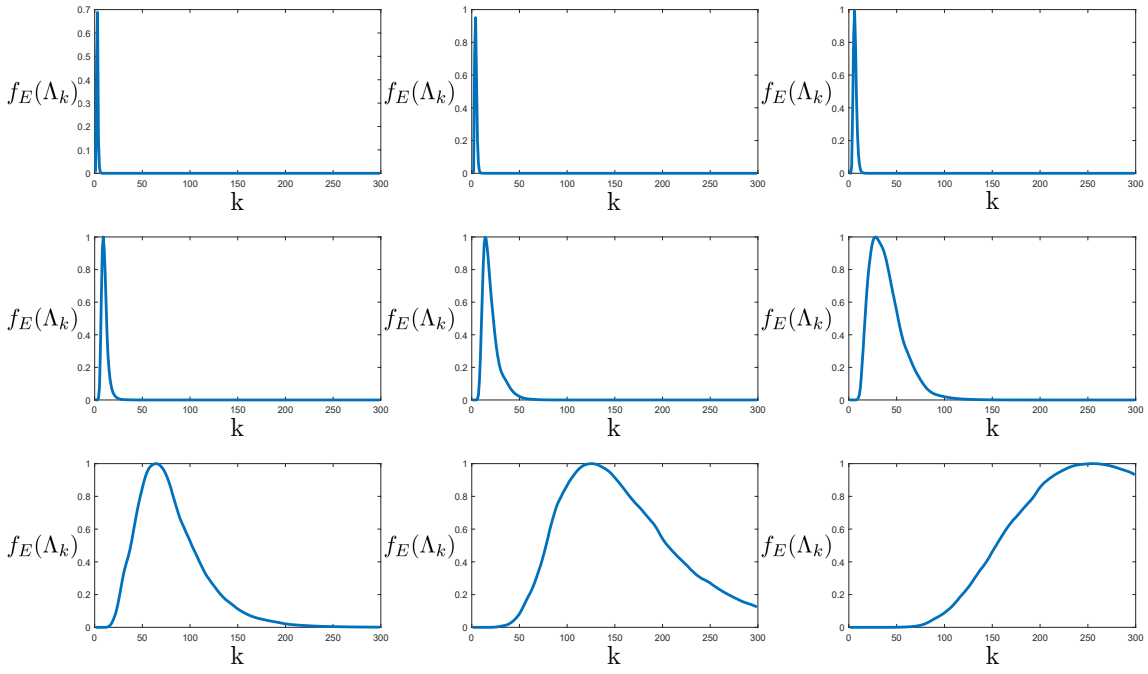


Figure 3.2: Illustration of the WKS's band-pass filters ($f_E(\lambda_k)$) applied to the shape spectrum. There are shown the frequencies 20, 30, 40, 50, 60, 70, 80, 90 and 100, respectively.

3.2 Adaptive eigenvalue scaling

In order to construct a spectral signature, it is necessary to apply a filter to the shape spectra (eigenvalues of the LBO) in a way that we can create a feature vector for each point of the shape. Each element of the feature vector is a combination of different frequencies of the spectral eigenfunctions. These filters must be optimized to deal with noise and deformations of the shape. We show in Figures 3.1 and 3.2 the filters used by the HKS and the WKS to compose their descriptors.

By looking into the filters of spectral signatures, we can see that the low frequencies of the band-pass filters (WKS) (Eq. 3.1) are very narrow (Figure 3.2), *i.e.*, they only account for a very specific part of the spectrum. On the other hand, the high frequencies are very large, aggregating a huge part of the spectrum in only one descriptor frequency. While the low frequencies of the descriptor can suffer from noise since they are a very small part of the spectrum, the high frequencies do not add much information considering it fuses many details in one descriptor frequency.

$$f_E(\Lambda_k)^2 = \exp\left[\frac{-(e-\log(\Lambda_k))^2}{2\sigma^2}\right] \qquad e \in [\log\Lambda_1, \log\Lambda_{max}] \tag{3.1}$$

The WKS filter (Eq. 3.1) aims in accounting for the difference that the shape spectrum (eigenvalues) can have within the same class. Although the shape spectra should be the same, articulating shape joints cause a small change in the eigenvalues that needs to be accounted for. The WKS has a probability density function given by a log-normal distribution to account for this variance. This means the WKS assumes the eigenvalues of same-class shapes vary according to a log-normal distribution, *i.e.*, being the majority of the absolute differences close to zero and very few larger than zero.

Following this argument, we looked into the differences of eigenvalues of same-class shapes. In Figure 3.3, we plot eight histograms that represent the difference of eigenvalues of 20 shapes of ants from SHREC'10 with their respective normalized errors. We did the same plots for other shapes as well from SHREC'10 and most of them have shown similar outcomes.

In the first plot, we show the histogram of the eigenvalue differences without any scaling;

in the next five plots, we show different power scalings with our proposed cube root scaling (which we are going to talk about soon); finally, we show the logarithmic scaling used by the WKS and also the logarithmic scale with base 10. Then, on each histogram we draw a normal distribution (red curve) fitted to the histogram that accounts for the eigenvalue variance. We show the respective normalized errors

$$E_2 = \frac{||H - F||_2}{||H||_2} \tag{3.2}$$

of this fitting at the top of each histogram, where H are the respective heights of the histogram bins and F is the normal distribution. Clearly, the power scaling functions have a smaller error compared to the logarithmic functions. The smallest fitting error in the histograms is given by the cubic root scaling. With this new scaling, we have a signature which accounts better for the difference of eigenvalues of same-class shapes.

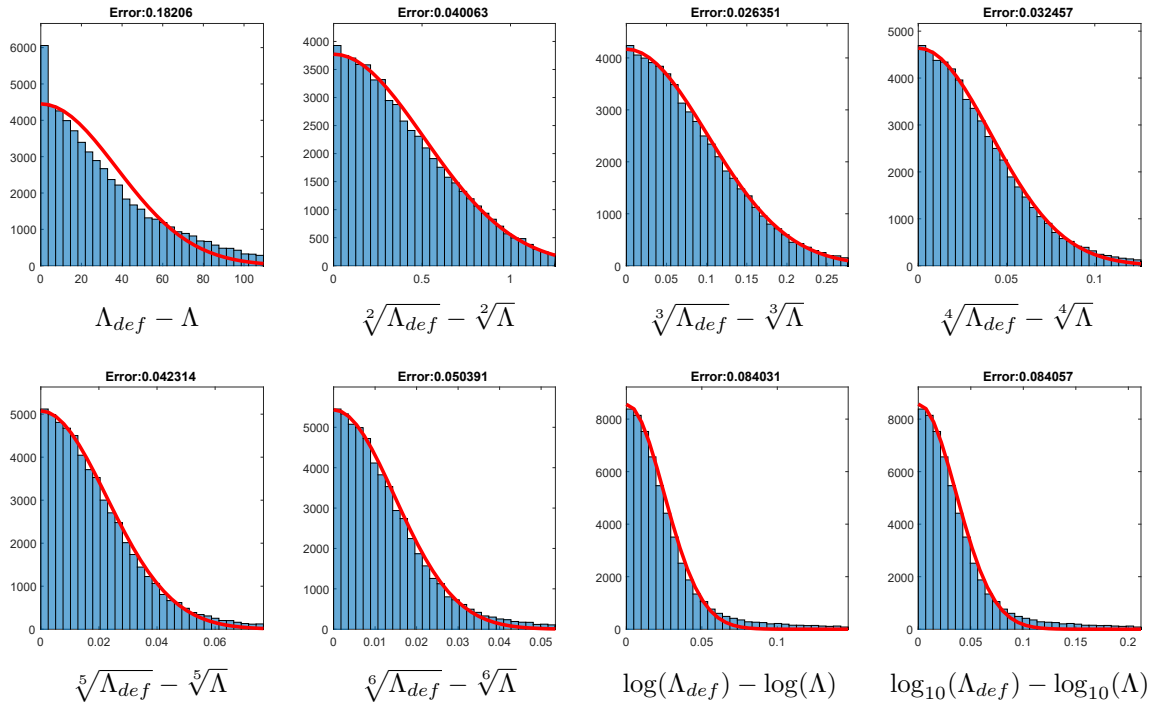


Figure 3.3: Analysis of the differences of eigenvalues in different scales. As you can see, the cubic root scaling has the smallest fitting error. This points to a signature which acknowledges better the difference of eigenvalues of shapes from the same class since the majority of the histogram bars are now "inside" the normal distribution (below the red curve). This leads to signatures that are more similar to shapes that belong to the same class.

The point here is that the real data (the difference of eigenvalues of the same-class shapes)

are not log-normally distributed, therefore, the WKS do not account accurately for changes in the shape spectra using a log-normal distribution. On the other hand, the cubic root scaling is better in this way. It precisely accounts the difference of the eigenvalues can have when the shape is articulated. Retrieval results in the entire benchmark confirm the success of this strategy and these are shown in the following chapters. We do not show evaluation performances here because we still need to define the other parts of the framework for shape retrieval.

The difference in the scaling can also be seen in the band-pass filters. Figure 3.4 shows the new filters when using a cube root scaling instead of logarithmic scaling. Note how the cube root scale does not suffer from the same problems of the logarithmic scale that we detailed before. The first frequencies are not as sharp as in the WKS, reducing the influence of noise in the filter, and the last frequencies do not blur the entire spectrum, aggregating more information to the descriptor. This leads to a signature the is informative along all its frequencies, because the filters are more equally distributed across the spectrum.

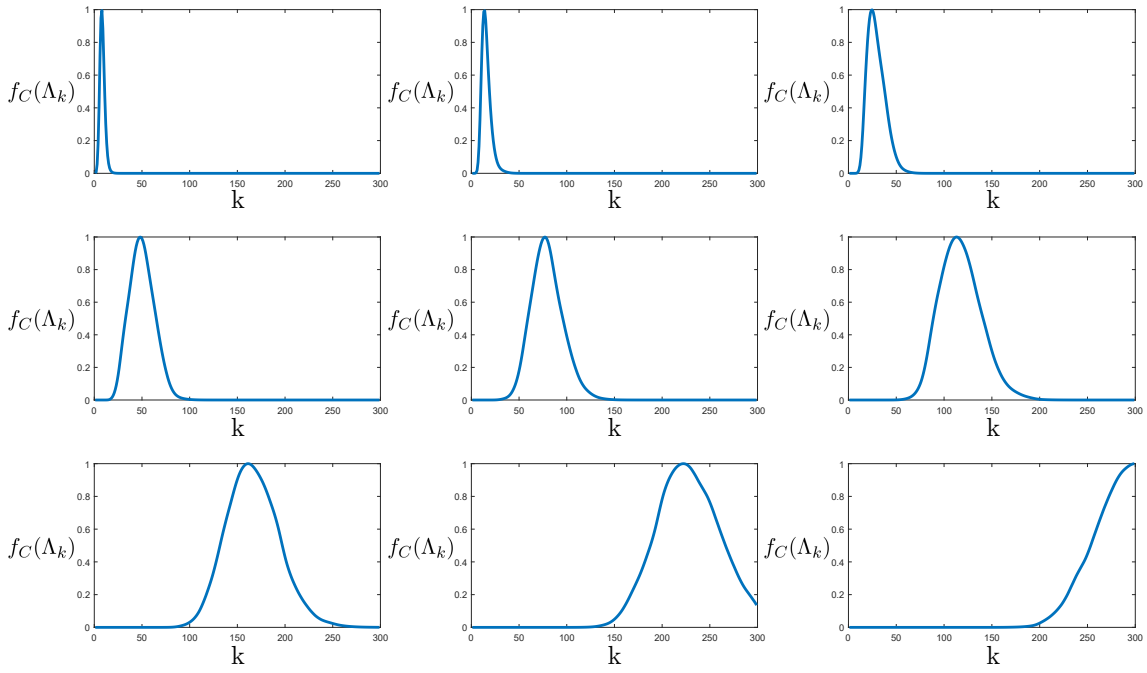


Figure 3.4: Illustration of the IWKS's band-pass filters ($f_C(\Lambda_k)$) which will be applied to the shape spectrum. There are shown the frequencies 20, 30, 40, 50, 60, 70, 80, 90 and 100, respectively. In this graphs, it is possible to see that the filters are more distributed along the frequencies of the signature. Furthermore, the last frequencies do not blur the spectrum completely as it happens in the HKS and in the WKS.

To decrease the difference of the eigenvalues from shapes that belong to the same class we also proposed a second improvement. By reducing the influence of shape joints in the descriptor we were able to increase the retrieval performance of the IWKS even more. More details and the final IWKS equation are given in the following section.

3.3 Curvature aggregation

To start addressing the effect of articulation, we analyzed the outcomes of joint motions in the spectrum of the shape, and we noted that descriptors were consistently different after deforming the respective 3D shapes. With this in mind, we believe that removing the influence of shape joints in the shape descriptor will give a better representation of the intrinsic shape. To reduce this influence we need to downweight joint regions so that they do not influence the final descriptor.

Curvatures are a good and automatic way of producing maps over the surface of objects, and they represent how the local patch is bent and also its bending direction. We start looking at different sorts of curvatures that could represent what we need. To compute curvatures over the surface we use the work described in [Rusinkiewicz, 2004]. Rusinkiewicz computes face curvatures by computing the second fundamental tensor ($\mathbf{II}$) of triangle faces

$$\mathbf{II}(X,Y) = \begin{bmatrix} \mathbf{II}(X_1,Y_1) & \mathbf{II}(X_1,Y_2) \\ \mathbf{II}(X_2,Y_1) & \mathbf{II}(X_2,Y_2) \end{bmatrix}. \tag{3.3}$$

Vertex curvatures are estimated by averaging adjacent triangle curvatures weighted by the respective Voronoi area of the face. This is done by transforming the coordinates system of the $\mathbf{II}$ matrix from the face to the vertex coordinate system. Finally, principal curvatures and directions are extracted by computing the eigenvalues and eigenvectors of $\mathbf{II}$ in the vertex coordinate system.

By computing the eigendecomposition of $\mathbf{II}(X,Y)$ two eigenvalues are found k_1 and k_2 which are called the principal curvatures (P), and two eigenvectors v_1 and v_2, which are called the principal directions which we observe the principal curvatures. These curvatures can be combined to define other curvature attributes, which can be seen in Table 3.1. Figure

3.5 shows some curvatures plotted over the shape of a dinosaur.

Table 3.1: Different statistics that can be computed from the principal curvatures.

Curvature	Formula	Characteristic
Maximum Principal Curvature	k_1	Maximum folding at a certain point
Minimum Principal Curvature	k_2	Minimum folding at a certain point
Gaussian	$k_1 k_2$	Indicates whether a surface has been warped or not
Mean	$H = \frac{k_1+k_2}{2}$	The average curvature of the local patch
Curvedness	$(k_1^2+k_2^2)^{1/2}$	Measure the intensity of folding

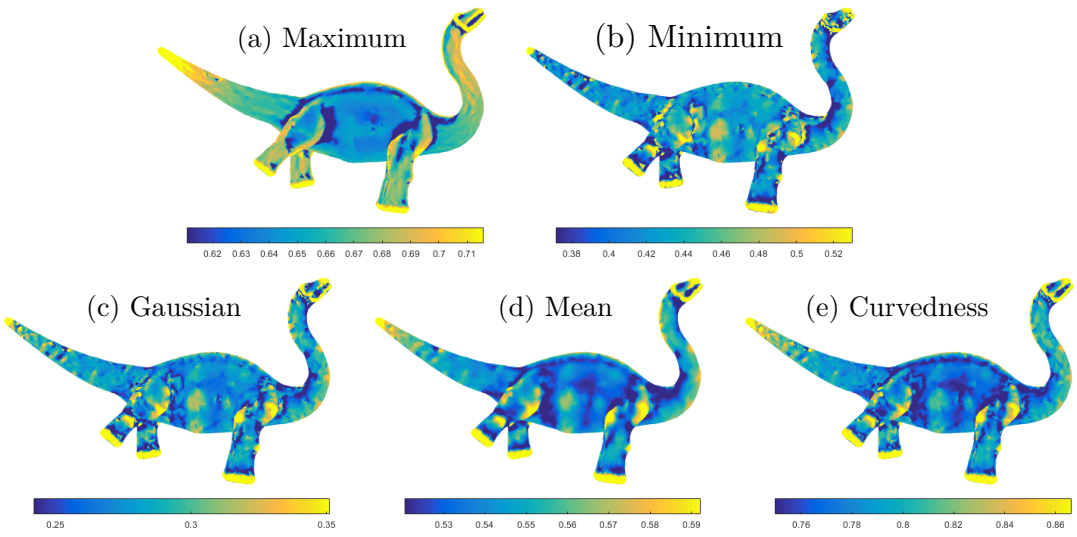


Figure 3.5: Plot of different curvatures over the shape of a dinosaur.

Combining these maps with spectral signatures can generate even more discriminative and meaningful maps which are crucial for object recognition. The maximum principal curvature has very small or negative values on joint regions, therefore, combining the maximum principal curvature with spectral signatures result in signatures which are less variant to shape motions. The maximum principal curvature was chosen based on experiments on the signatures to retrieve shapes in non-rigid databases. Thus, we aggregate shape curvatures

by shifting spectral descriptors by the maximum principal curvature, thus computing:

$$\mathrm{IWKS}(\mathbf{x},\mathbf{e}) = \sum_{k=1}^{\infty} \phi_k(\mathbf{x})^2 f_C(\Lambda_k,\mathbf{e})^2 + \alpha C(\mathbf{x}) \qquad f_C(\Lambda_k,\mathbf{e})^2 = \exp\left[\frac{-(\mathbf{e}-\sqrt[3]{\Lambda_k})^2}{2\sigma^2}\right] \quad (3.4)$$

where $\mathbf{e} \in [\sqrt[3]{\lambda_f}, \sqrt[3]{\lambda_{max}}]$, λ_f is the first non-zero eigenvalue, $C(\mathbf{x}) = max(k_{1\mathbf{x}}, 0)$, α normalizes C accordingly to the signature values, and (Λ, ϕ) are eigenvalues and eigenfunctions of the LBO.

The IWKS is designed to reduce the influence of articulation in non-rigid shapes since it downweights joint regions of the shape. However, it also does something more. The map also weights different structures of the shape, increasing the separability of these regions. Both characteristics can be seen in Figure 3.6, where joint regions are less weighted (in blue) and the two primitives presented on the object (spheres and tube) can be easily distinguished.

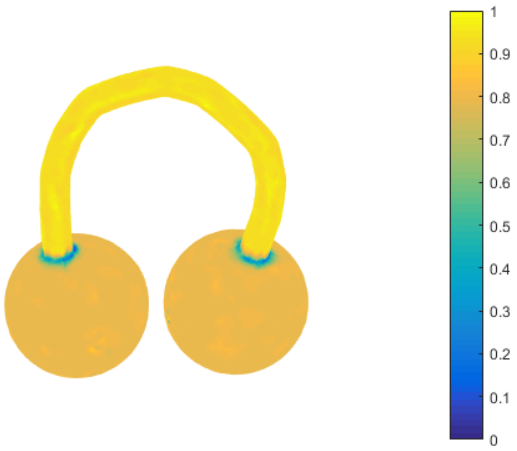


Figure 3.6: Plot of the maximum curvature over the shape of *twoballs*. As can be seen, the curvature can "label" parts which have the same meaning. Furthermore, it can reduce the influence of shape motion by downweighting joint regions.

Therefore, the final IWKS is given by Equation 3.4. It can be used instead of the HKS, SIHKS or WKS to describe and retrieve shapes in non-rigid databases. In the next section, we perform experiments with the IWKS together with our encoding framework for non-rigid shape retrieval.

3.4 Experiments

In this section, we show an analysis of the Improved Wave Kernel signature applied to different models and compare against other spectral descriptors. We focus in showing here that our signature is much more informative than other signatures, once it benefits from a better filtering of the shape spectra.

The IWKS aims in creating a signature which is invariant to shape motions. This way, to compute distances between pairs of shapes it is first necessary to create a global representation of the shape. The creation of global signatures from local descriptors is going to be addressed in the next chapter (4). There, it is possible to find quantitative experiments on non-rigid 3D shape retrieval, their performance statistics and comparisons against state-of-the-art signatures. Here, we focus in comparing only qualitative experiments of the local descriptors generated by different spectral signatures (HKS, SIHKS, WKS and of course IWKS).

By comparing first the WKS with the IWKS, we can easily see from Figures 3.2 and 3.4 that the IWKS filters are more equally distributed along the spectrum frequencies. It is also possible to see that, for the first frequencies, the IWKS filters will go through the frequencies faster, considering the mean of the filter, while the WKS will start slowly. This happens because of the difference in the weighting (logarithmic vs cubic root). However, in the last frequencies, the WKS filters will have to be wide to cover the final spectral frequencies causing many frequency descriptors to be the same. In Figure 3.7, we can see that for the same descriptor frequency showed on the dinosaur models, the IWKS presents more high frequency information than the WKS because, as previously said, the IWKS filter goes through the first spectrum frequencies quicker but never blurs any information, regardless of the filter frequency.

Figure 3.8 shows a comparison among HKS, SIHKS, WKS and IWKS signatures. On the left, we show the four descriptors plotted over the shape of a dinosaur. On the right, we show the respective signature plotted over all frequencies. If the reader pays attention to the last frequencies of the plots, he will see that the last frequencies of the HKS, SIHKS and WKS are blurred (flat horizontal lines) because of the distribution of filters along the

spectrum. The filters that represent the last frequencies are too wide and aggregate the same information. This leads to the generation of the horizontal lines which do not add any information to the descriptor. On the other hand, the last frequencies of the IWKS still have useful information which can be used to distinguish one shape from another.

3.5 Timing analysis

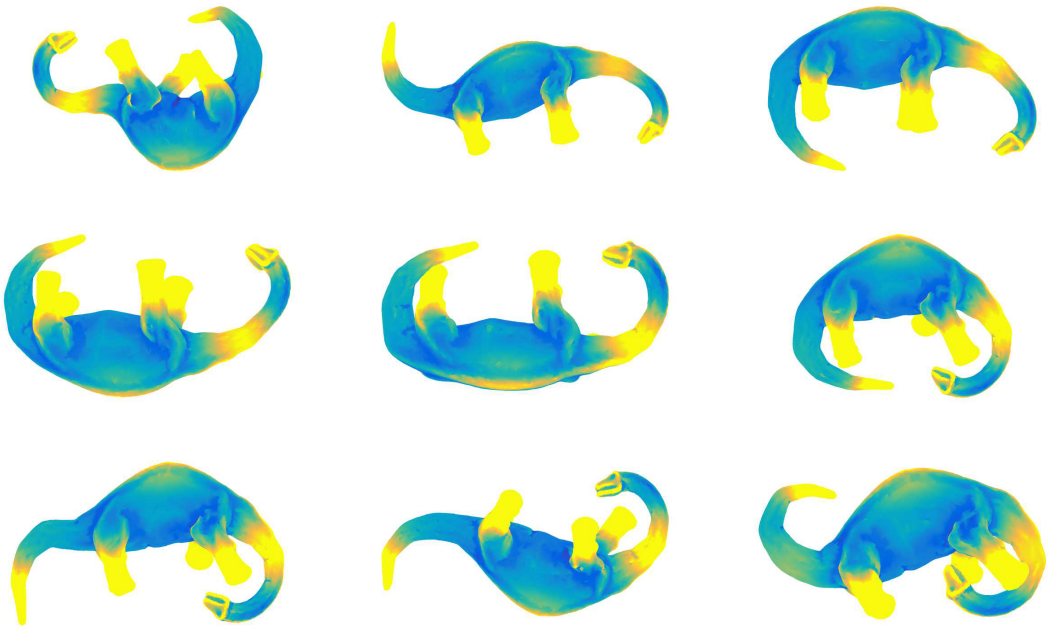
The IWKS has the same computational complexity of the WKS. After computing the eigendecomposition of the LBO, the computational complexity for computing the IWKS is $O(|\Lambda|F)$, where F is the number of times we sample the descriptor at different energies. In our case these parameters are fixed and are $|\lambda| = 300$ and $F = 100$. The average time for computing the IWKS is 0.01 seconds. This time does not account for computing the shape spectrum.

3.6 Summary

The Improved Wave Kernel signature is a result of an empirical analysis of the differences among eigenvalues of same-class shapes. In the original paper of the Wave Kernel Signature, Aubry *et al.* [Aubry et al., 2011b] have chosen the function to handle differences between eigenvalues using a theoretical analysis. This resulted in the logarithm scale function of the original WKS. However, when applied over distinct datasets this function behaves differently. By using an empirical analysis of the differences of eigenvalues we found a scale function that behaves accordingly the data. In other words, the scaling function depends on the data and, in order to maximize retrieval performance, the scaling function should be a result of an empirical analysis of the training dataset.

The *eigenvalue scaling* depends on the data so it should adapt according the data. In the *SHREC'15 Non-Rigid 3D Shape Retrieval*, we analyzed the difference between eigenvalues from the articulated shape of an ant and we discover that the distribution function is better fitted using a power scale ($\text{eigs}^{1/3}$), which is the cubic root. This modification in the scaling gives a better Gaussian fit to the data and improves the retrieval performance of the benchmark.

Besides the eigenvalue scaling function, we also improve spectral signatures by adding an additional ingredient to their formulation, the *curvature aggregation*. This feature is designed to increase the separation of shape features. It does that in two ways. First, it reduces the influence of articulations, since it downweights joint regions of the shape. Second, it labels different parts of the shape by equally weighting regions which belong to the same category. To choose what is the weight that will be applied to the descriptor we have computed and analyzed many shape curvatures. We chose to use here the maximum curvature given it gave the best retrieval performances and it has shown to be stabler and smoother than other extrinsic properties.



(a)

(b)

Figure 3.7: Comparison plot between the WKS (a) and the IWKS (b) at same color scale from 0 (blue) to 2 (yellow). We show 9 dinosaur models (from SHREC'15) of each signature all at the 30th frequency of the respective descriptor. We can see that the IWKS is more informative along the entire shape (higher frequency information), not only giving separation between legs and body trunk. This is because the IWKS filters are sharper from the 20th frequency onwards and do not blur completely the spectrum. Thus, for the same descriptor frequency, the IWKS carries more information about the shape, especially in the last descriptor frequencies.

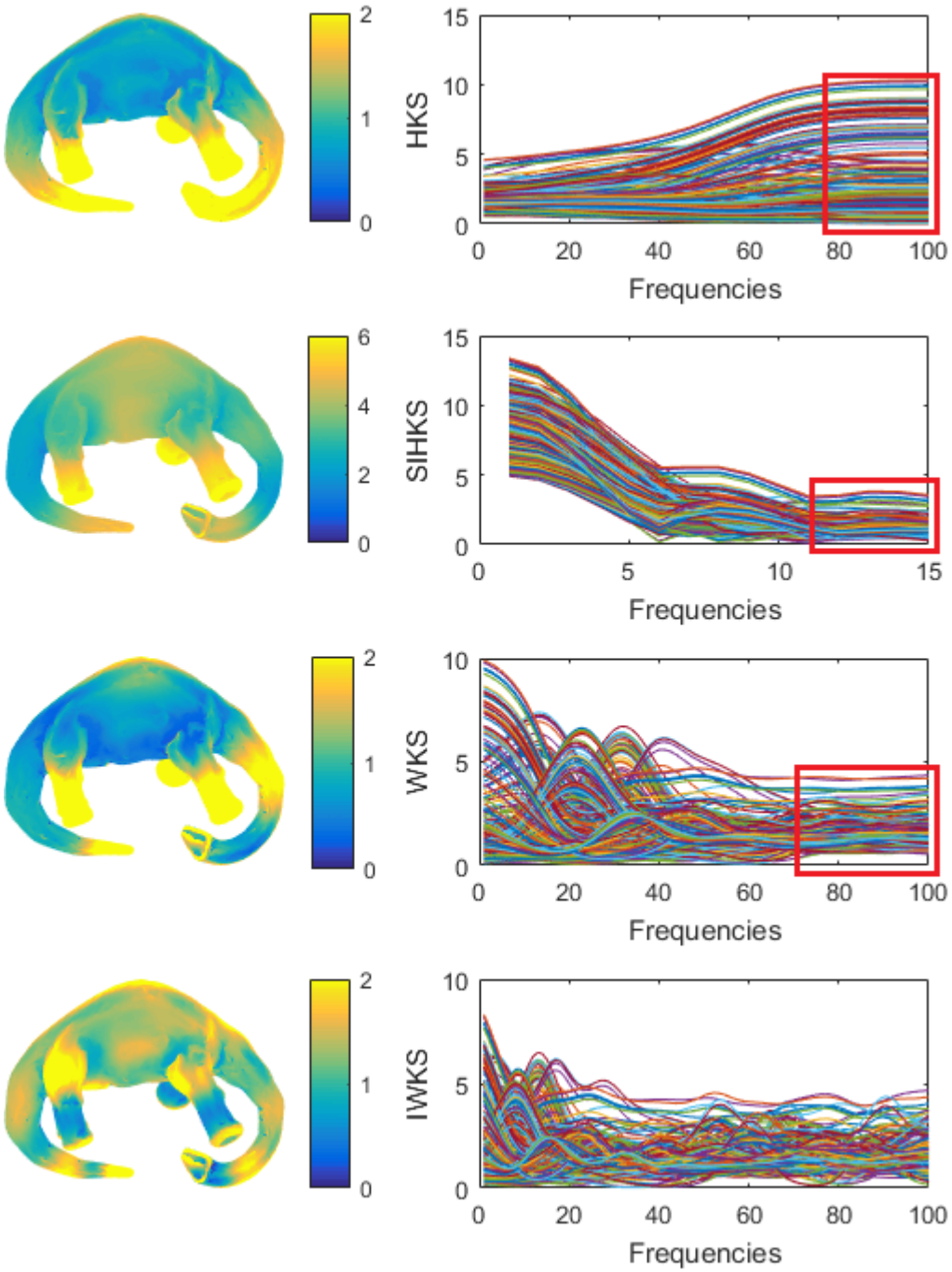


Figure 3.8: Comparison plot between (in order from top to bottom) the HKS, SIHKS, WKS and IWKS respectively. On the left, it is shown dinosaur models coloured by the 30th frequency (5th for SIHKS). On the right, are plotted 5% of the descriptors for all frequencies (x-axis) for the respective model. It is possible to see that the IWKS is more informative, both in the dinosaur model (because it contains more details) and in the signature plot, since the last frequencies are not blurred and do not appear as horizontal lines in the plot like the other descriptors. Horizontal lines do not add any information to the descriptor once those descriptor frequencies are the same.

Chapter 4

Using Robust Encodings With Spectral Signatures

In this chapter, we propose to use state-of-the-art encoding methods combined with spectral signatures to represent 3D shapes. Although spectral signatures have many desirable properties to describe 3D shapes, for instance being invariant under rigid transformations and stable against non-rigid transformations, they do not perform so well in recent benchmarks.

A large number of local descriptors have been created to represent local characteristics of geometric shapes for the purpose of many computer vision, geometry processing and shape analysis tasks. A local descriptor is a compact representation that characterizes a small region of a shape. They usually capture information about the neighbourhood of a vertex and so they can be directly applied to some important tasks like point correspondence and shape segmentation. For this purpose, it is desirable to compute signatures that are invariant under rigid, non-rigid and isometric deformations, the typical deformations that 3D models undergo. However, local descriptors cannot be immediately applied to the problem of shape retrieval, because this task is not addressed by comparing local features but by comparing global signatures (signatures that represent the shape as a whole). Creating a global signature is not a simple task since shapes can have arbitrary number of vertices, edges and faces. To create a generic representation of a shape all important characteristics should be preserved during the encoding process. Even so, the global representation must compress

local characteristics using a commensurable basis in order to facilitate comparisons.

The chapter was published in [Limberger and Wilson, 2015], but before that, we participated in the SHREC'15 contest [Lian et al., 2015]. The remaining of this chapter is organized in the following way. We discuss related work in Section 4.1. Then, we show how our encoding framework is structured in Section 4.2. We show how to compute signature distances in Section 4.3. Finally, the proposed framework is evaluated and compared in Section 4.4.

4.1 Related methods

In this section, before introducing the encoding framework, we provide a brief background about related methods that use the Bag-of-Features paradigm (BoF) applied to shape retrieval. We also mention where the Fisher Vector and the Super Vector were used in the literature to improve retrieval.

The idea of creating shape signatures from local signatures is neither new nor straightforward. The usual way to describe a set of local descriptors into a shape-level signature for retrieval and classification purposes is building a Bag-of-Features model to remove the local dependency of each descriptor by writing local properties as a histogram of their occurrences. Although recent approaches mainly use the classical BoF, the use of other encoding methods can bring many advantages over the traditional method. For instance, the Fisher Vector combines the strengths of generative and discriminative models [Perronnin and Dance, 2007]. While the BoF characterizes a sample by the number of occurrences of visual words, FV is characterized by the deviation from a probabilistic vocabulary. Chatfield *et al.* [Chatfield et al., 2011] states that Fisher Vector and Super Vector encodings are better than other encoding methods, since they carry extra information about the displacement between descriptors and visual words.

With the popularization of the BoF paradigm, Ovsjanikov *et al.* [Ovsjanikov et al., 2009] and later Bronstein *et al.* [Bronstein et al., 2011] proposed to encode local descriptors to a global representation of the shape, removing the local dependence of each descriptor. They used Bag-of-Features to obtain a visual word-based representation of the shape that can be compared efficiently. They formulate their signature as the normalized probability of

associating each geometric word from the vocabulary with all vertices of the shape. The distance between two shapes is given by a Hamming metric between the bag-of-features of each shape in the Hamming space.

There are other signatures that also make use of the BoF framework but do not use spectral local descriptors. In [Furuya and Ohbuchi, 2009], Furuya *et al.* use BoF to encode SIFTs generated by depth images rendered in different views from a 3D model. In a recent benchmark of 3D shape retrieval [Li et al., 2014b], Furuya and Ohbuchi [Furuya and Ohbuchi, 2013] have also applied BoF to their local features. In a non-rigid 3D shape retrieval benchmark of human models [Pickup et al., 2014a], Tatsuma (Bag-of-Features approach with Augmented Point Feature Histograms), Bu (High-level Feature Learning for 3D Shapes) and Li [Li et al., 2014a], besides Litman *et al.* [Litman et al., 2014], have encoded local features using BoF. In other recent benchmarks [Li et al., 2014c, Li et al., 2012], the best results were obtained by techniques that have used BoF or similar frameworks.

The use of Fisher Vectors to classify and retrieve images has been recently addressed by a number of researchers. Perronnin and Dance [Perronnin and Dance, 2007] applied FV to the problem of image categorization, Perronnin *et al.* [Perronnin et al., 2010a] proposed a compressed form of FV to retrieve images in a large database, Sanchez *et al.* [Sanchez et al., 2013] and Csurka and Perronnin [Csurka and Perronnin, 2011] show that the FV framework is the state-of-the-art approach for classification and retrieval purposes since it has a more efficient representation of an image. Takeyoshi and Kikinis [Vohra et al., 2002] use FV to classify patients with epilepsy, Schneider and Tuytelaars [Schneider and Tuytelaars, 2014] applied FV in sketch classification and Simonyan *et al.* [Simonyan et al., 2013] create a face descriptor achieving state-of-the-art performance on a challenge benchmark. The use of Gaussian Mixture Models (basic concept of FV) to characterize 3D shapes was first experimentally addressed by [Aubry et al., 2011a].

On the other hand, Super Vector (SV) [Zhou et al., 2010] has shown to be a good encoding to represent local features. Super Vector is a nonlinear mapping from the descriptor space to a high-dimensional sparse vector. Algorithmically, it can be seen as a simple extension of Vector Quantization. In a recent benchmark [Lian et al., 2015], SV was used to aggregate local 3D shape features achieving the best performance among other participants. In [Li

et al., 2014c], Tatsuma used Super Vector to encode features extracted from rendered depth buffer images, achieving the best performance in the benchmark.

4.2 Encoding framework

In this section, we propose an efficient and discriminative encoding framework to address the problem of creating global signatures for 3D models from local descriptors based on the spectrum of the shape, for the purpose of shape retrieval and classification. In conjunction with creating the global representation, we also propose to use Manifold Ranking to compute the dissimilarities between encodings.

We propose the use of Fisher Vector or Super Vector to encode spectral signatures thus describing the entire representation of a shape. Until now, the Fisher Vector had never been used to represent 3D models before. Differently from [Bronstein et al., 2011, Ovsjanikov et al., 2009], our approach uses a Gaussian Mixture Model as a dictionary of probabilistic visual words and encodes the global signature using three orders statistics (0-th, 1-st, 2-nd) rather than using only the first order. Further, while BoF generates a K-dimensional histogram, where K is the vocabulary size, Fisher Vector encoding generates a high-dimensional vector with $2KD$ dimensions, where D is the size of each local descriptor, being more discriminative but still simple to compare, since all shapes are encoded in the same basis.

This encoding framework is used during many parts of this thesis for the purpose of creating a global shape signatures from local shape descriptors.

4.2.1 Spectral Signatures

In this section, we explain how to compute spectral descriptors from 3D meshes and the parameters that we used for each signature. All the settings shown here are used throughout this thesis when we compute the spectral signatures HKS, SIHKS, WKS and IWKS.

First, we start by the discretization of the LBO. We use the algorithm of Meyer *et al.* [Meyer et al., 2003] even though it has shown to not converge in general [Xu, 2004]. By using watertight meshes, we did not experience any convergence problems. In Section 2.1, we show a description of this discretization in more details. When performing the eigenvalue decompo-

sition, we compute the first 300 eigenvalues and eigenfunctions of the LBO to characterize shapes using the generalized eigenproblem (Equation 2.5).

Regarding spectral descriptors, we used the respective available author implementations [Aubry et al., 2011b, Bronstein and Kokkinos, 2010, Limberger and Wilson, 2015, Sun et al., 2009]. We have chosen the parameters that maximize retrieval performance in the general case and used the same throughout all experiments, regardless of what benchmark we are testing. In the following, λ_i represent the eigenvalue at position i in ascending order and λ_f is the first nonzero eigenvalue.

The **Heat Kernel Signature** and the **Scale-Invariant Heat Kernel Signature** are sampled 100 times in the time interval $[log(4\ln(10)/\lambda_{300)}, log(4\ln(10)/\lambda_2)]$, according to author's implementation. Then, we sample the first 15 frequencies of the SIHKS after computing scale normalization using Fast Fourier transform. The **Wave Kernel Signature** is sampled 100 times in the energy interval $[\log(\lambda_2), \log(\lambda_{300})]$. The WKS variance used in all datasets was $wksvar = 6$. The **Improved Wave Kernel Signature** is sampled 100 times in the energy interval $[\sqrt[3]{\lambda_f}, \sqrt[3]{\lambda_{300}}]$. The IWKS variances are $iwksvar = 5$, $iwksvar = 2.5$ and $iwksvar = 3.75$ for SHREC'10, '11 and '15, respectively. The variances of WKS and IWKS have a different impact on the respective signature because the weighting filters are different, *i.e.*, if the same IWKS variances would be used in the WKS they would generate different outcomes.

After analyzing the IWKS shift [Limberger and Wilson, 2015] carefully, described in Chapter 3, we have found better results using a normalized curvature distribution

$$S(\mathbf{x}) = \frac{m \cdot C(\mathbf{x})}{\text{mean}(C(\mathbf{x}))} \tag{4.1}$$

where m is a parameter representing the desired mean after scaling the curvatures $C(\mathbf{x})$. This makes curvatures more stable even when shapes are different in size or deformed. This shift is applied to all signatures (HKS, SIHKS, WKS and IWKS). Finally, we use $S(\mathbf{x})$ in signature shift, differently from [Limberger and Wilson, 2015], which uses $C(\mathbf{x})$. Fig. 4.1 shows a plot of $S(\mathbf{x})$ over different deformed shapes. It is easy to see that $S(\mathbf{x})$ is less weighted on joint regions (mainly parts near leg joints) and it remains stable across deformations of

the shape (see the tail and the neck).

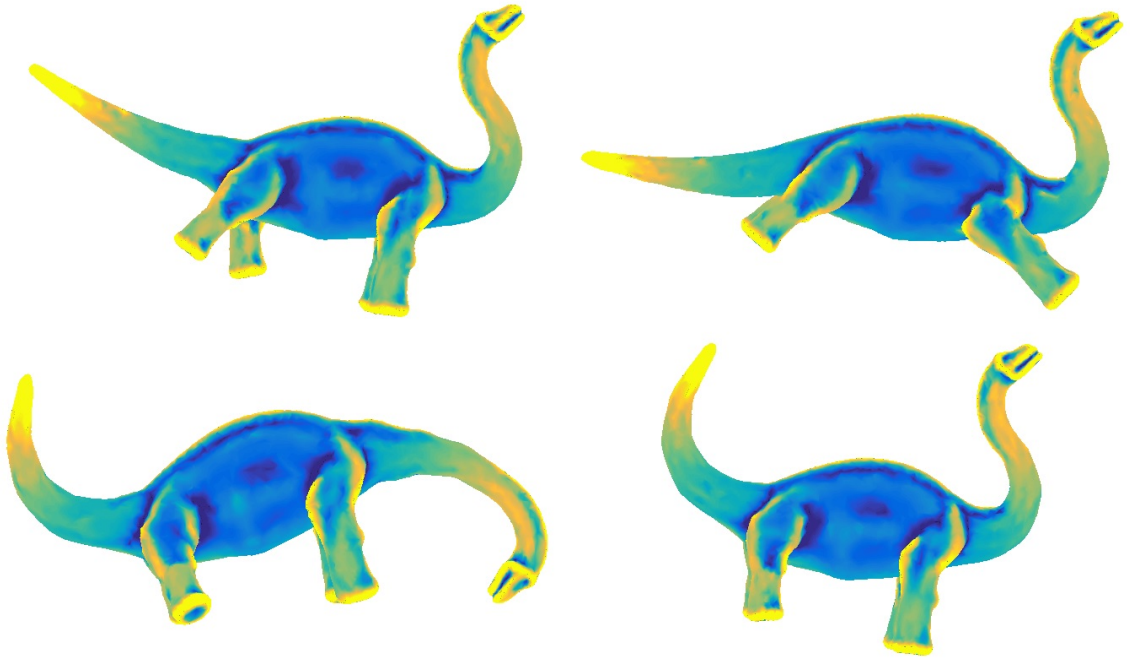


Figure 4.1: Plot of $S(\mathbf{x})$ on different dinosaur models from SHREC'15 benchmark using the same colormap. Blue stands for low values and yellow stands for high values. As can be seen, the positive curvatures remain stable along non-rigid deformations of the shape. Models are respectively 69, 171, 323 and 393.

All the four signatures are shifted by curvature to increase separation of features as described above, where $m = 0.3$, $m = 0.5$ and $m = 0.5$ for each respective dataset: SHREC'10, '11 and '15.

4.2.2 Encoding Spectral Signatures

Similarly to images, encoding methods can be applied to shapes. Although shapes have a complex structure in a 3-dimensional space, encodings can be applied effectively after calculating proper local descriptors, which must respect the following properties:

1. Isometry-invariant signature: It is essential that a shape can be described by isometry-invariant descriptors to avoid prior alignment (placing the model in the same orientation).

2. Multi-level signature: It is necessary to describe a shape in multiple scales. For this, all descriptors must have the same size although the number of descriptors still can

depend on the shape (number of vertices).

The spectral signatures fulfill all these requirements. Each frequency of the descriptor is seen as a layer that describes the entire shape. Encoding methods are applied to these layers to encode all the information in a high dimensional vector. Moreover, there are other good properties that descriptors should hold to properly represent shapes, for instance being stable against non-rigid motions and different sampling rates. Although these are very important for the success of shape retrieval, the encoding process do not depend on them.

In order to encode local descriptors we need to characterize them by their deviation from a generative model. Let

$$S = \{\mathbf{x}_t, \mathbf{x}_t \in \mathbb{R}^D, t = 1...T\} \tag{4.2}$$

be a set of local descriptors of a shape S, where T is the number of vertices and D the descriptor dimension, and

$$\lambda = \{w_k, \mu_k, \mathbf{\Sigma}_k, k = 1...K\} \tag{4.3}$$

a set of parameters of a Gaussian Mixture Model p_λ, where w_k, μ_k and $\mathbf{\Sigma}_k$ are respectively the weight, mean vector and covariance vector of the k-th Gaussian of the GMM. We assume that covariances matrices are diagonal thus writing them as vectors. The distribution of descriptors $p_\lambda(x)$ is given by

$$p_\lambda(x) = \sum_{k=1}^{K} w_k \mathcal{N}(x|\mu_k, \Sigma_k) : \sum_{k=1}^{K} w_k = 1. \tag{4.4}$$

The Gaussian Mixture Model parameters are estimated using the Expectation-Maximization (EM) algorithm [Sanchez et al., 2013] in order to optimize a Maximum Likelihood criterion for the data X. The EM is an iterative algorithm that estimates parameters of a statistical model (X), when data is missing from the observation $\mathbf{x} = (x_1, x_2, \dots)$. In our case, we would like to learn the parameters of a Gaussian Mixture Model, which are the means and covariances of the probabilistic distributions. Thus, EM algorithm is what allows us to infer those parameters values. In order to create a dictionary of GMMs for shape retrieval, the EM algorithm selects the K more distinct features, using k-means, from a subset of models to compose the dictionary of mixture models.

To compute the EM-algorithm we first initialize it with k-means clustering, giving a label to each data point. Then EM computes Gaussian mixture models with this respective information (means and variances from the clusters) and assigns a posterior probability to each data point belonging to each cluster (Expectation part). With these probabilities, it re-estimate new means and variances to fit the new data points (Maximization part). By iterating these last two steps, the EM-algorithm converges to the model parameters.

Thus, differently from K-means algorithm that assigns each sample to a cluster, EM-algorithm finds a Gaussian Mixture model which gives the probability that each data point belongs to each cluster. EM algorithm is computed from a set of descriptors from the whole dataset. All vertex descriptors from this set are concatenated in one single vector of features so the algorithm can assemble similar feature kernels regardless of their location on the shapes. We chose to use the first 29 models of each dataset to compute the respective dictionary. The number of models was chosen as a middle point between an undersampled and oversampled dictionary (to contain at least one feature of each type and to not contain many duplicates). Occam's razor intuition says that if we have two different hypothesis to explain what is happening in the data we should pick the simplest possible explanation, which means, the solution that has fewer clusters K but still converge to a similar solution. Therefore, we chose to compute GMMs with $K = 38$ components.

The choices of the number of components and number of models depend on the data, however, they do not fluctuate much at a certain interval. The chosen parameters are selected because they shown to produce good results in the retrieval problem. It is rather difficult to estimate the best value for K, since one wants to maximize the likelihood (L), but one does not want a high number of parameters (P). Minimizing P and maximizing L it is not simple. Usually people see how well the GMMs work on the output result (retrieval performance in our case) over different values of K and then people pick the best K. We chose these parameters because they have shown to be robust to all sorts of models in the available non-rigid benchmarks.

Thereby, we are now able to compute the FV (Γ_{FV}) and the SV (Γ_{SV}). The FV produces three-order-deviation vectors ($\mathbf{q}, \mathbf{u}, \mathbf{v}$) from the vocabulary to characterize the set of local descriptors. The first order is the association strength (soft assignment), which is computed

by the posterior probability

$$q_{tk} = \frac{\exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_k)^\top \Sigma_k^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_k)]}{\Sigma_{i=1}^{K} \exp[-\frac{1}{2}(\mathbf{x}_t - \boldsymbol{\mu}_i)^\top \Sigma_i^{-1}(\mathbf{x}_t - \boldsymbol{\mu}_i)]}. \tag{4.5}$$

Then, the second and third orders are computed w.r.t. the mean and covariance. For each mode k and each descriptor dimension $j = 1..D$, deviation vectors are computed

$$u_{jk} = \frac{1}{T\sqrt{w_k}} \sum_{i=1}^{T} q_{ik} \frac{x_{ji} - \mu_{jk}}{\sigma_{jk}}, \tag{4.6}$$

$$v_{jk} = \frac{1}{T\sqrt{2w_k}} \sum_{i=1}^{T} q_{ik} \left[\left(\frac{x_{ji} - \mu_{jk}}{\sigma_{jk}} \right)^2 - 1 \right] \tag{4.7}$$

where σ_{jk} are the square roots of the covariances Σ_k. In the end, FV is given by the vectorization and concatenation of the matrices u_{jk} and v_{jk}.

$$\Gamma_{FV} = \Gamma_\lambda^X = [\mathbf{u}_1^\top \dots \mathbf{u}_K^\top, \mathbf{v}_1^\top \dots \mathbf{v}_K^\top]^\top \tag{4.8}$$

Differently from the FV, the SV only considers two-order-deviation vectors $(\mathbf{q}, \mathbf{u})$ but it adds a component related to the mass of each cluster $(\mathbf{s})$

$$\begin{aligned} p_k &= \frac{1}{N} \sum_{t=1}^{N} q_{tk} \qquad s_k = s\sqrt{p_k} \\ \mathbf{u}_k &= \frac{1}{\sqrt{p_k}} \sum_{t=1}^{N} q_{tk}(\mathbf{x}_t - \boldsymbol{\mu}_k) \end{aligned} \tag{4.9}$$

where s is a weight to balance s_k and $\mathbf{u}_k$ numerically. Finally, SV is given by

$$\Gamma_{SV} = [s_1, \mathbf{u}_1^\top, ..., s_K, \mathbf{u}_K^\top]^\top \tag{4.10}$$

Following [Perronnin et al., 2010b], we also apply Power Normalization (4.11) and L2 Normalization (4.12) to FV and SV signatures to properly compare shapes

$$PN(z) = sign(z)|z|^{1/2} \tag{4.11}$$

$$L2N(z) = \frac{z}{\|z\|}. \tag{4.12}$$

Figure 4.2 shows the pipeline of our encoding framework. Thus, we create a dissimilarity matrix by computing distances between every model in the database using Manifold Ranking, which is the subject of the next section.

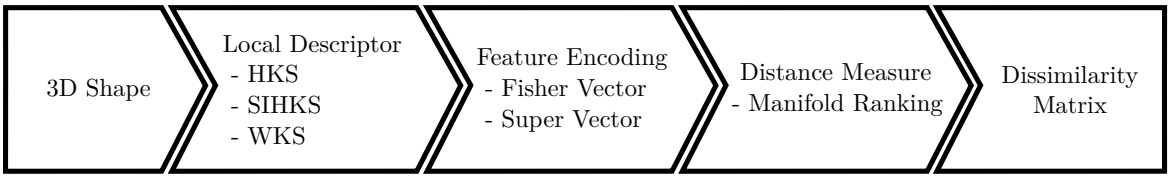


Figure 4.2: Encoding framework. First, local descriptors are computed from 3D models. Then, one of the state-of-the-art statistical encodings schemes is applied to the descriptors. Finally, Manifold Ranking is computed to gauge the differences between descriptors.

4.3 Manifold Ranking

After encoding the descriptors in the same basis, they now need to be compared. One way to do that is by computing Euclidean distances between feature vectors. However, this is not an efficient distance measure as signatures are represented by high-dimensional vectors (usually more than 6K values) which can be strongly affected by outliers. Another way is using Principal Component Analysis (PCA) to reduce features dimensionality. This is a better idea that generally lead to superior results, however, some information will be lost in the process, even by removing only less-important dimensions. An even better method is Manifold Ranking (MR) [Zhou et al., 2004], which is a graph-based ranking algorithm that captures information about the entire descriptor but it does not suffer from outliers.

The MR algorithm leads to a better separation of features than using a pairwise euclidean distance by exploiting the global structure of the intrinsic manifold, created from the feature vectors. First, it computes the intrinsic manifold based on the descriptors, generated by either FV or SV. It then computes the similarity between descriptors by navigating manifold graph edges, similar to a diffusion process. Therewith, a relative ranking score is assigned to each feature vector, differently from a pairwise similarity, as usually employed by dissimilarity measures. To compute the similarities of another model in respect to the database, the same process is repeated, this time starting at the respective descriptor's node. MR is becoming

a standard way to compute dissimilarities of feature vectors in large datasets as shown by many researchers [Li et al., 2012, Li et al., 2014c, Lian et al., 2010, Lian et al., 2011, Pickup et al., 2014b, Takahiko Furuya, 2015].

However, Manifold Ranking is computationally expensive on large datasets, which significantly limits its applicability to a variety of data types. This way, we use the Efficient Manifold Raking (EMR) [Xu et al., 2011], which is a faster version of the MR. EMR addresses differently the graph construction and ranking computation stages, which are the MR algorithm's bottleneck. Therefore, experimental results show that EMR has comparable recall performance to the MR, but it has a much better response time as the size of the graph becomes larger. By computing an efficient ranking function, EMR also allows out-of-sample retrieval, crucial to real-world retrieval systems.

4.4 Experiments

In this section, we demonstrate that local spectral descriptors can be used with Gaussian Mixture Model dictionaries. Furthermore, we also perform experiments on the encoding framework using three different datasets and report the results. We commence by fitting local descriptors by Gaussian Mixture Models using the EM algorithm.

4.4.1 GMM dictionaries and spectral descriptors

In this section, we perform an empirical analysis on spectral shape descriptors to determine whether they can fit a GMM. We plotted many descriptor frequencies to analyse their shapes. Each descriptor frequency is used as a feature at a certain scale. We show in Figure 4.3 five different randomly-selected descriptor frequencies for each spectral signature (HKS, SIHKS, WKS, IWKS) computed from the shape LBO.

In every example we fit a GMM with five components using the iterative EM-algorithm to show that is possible to approximate every shape feature histogram, even with a small number of components. This enables the use of GMM dictionaries with spectral signatures since features can be described by Gaussian Mixture Models and suggests, by extension, that it is possible to use FV and SV encodings schemes with spectral descriptors.

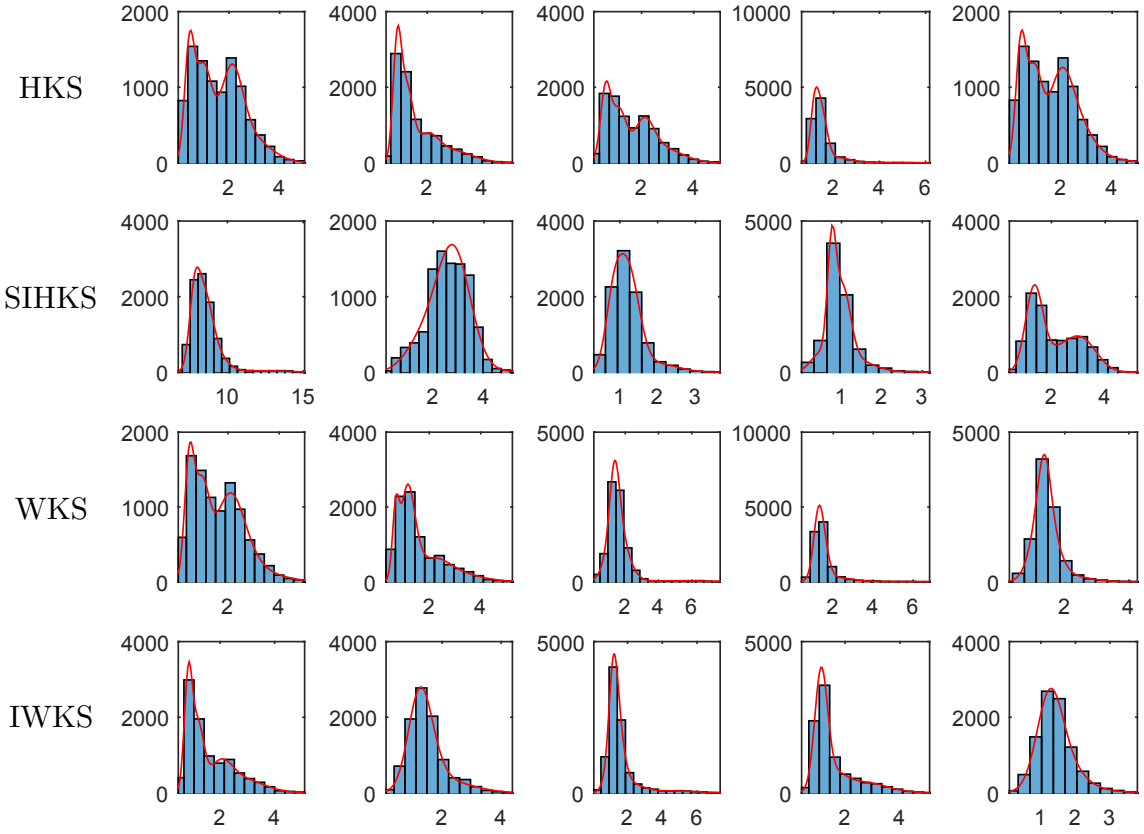


Figure 4.3: Histograms of randomly-chosen shape features plotted with the respective learned GMM with 5 components for different LBO signatures. Each row represents features from HKS, SIHKS, WKS and IWKS, respectively, from the first model of the database. The better it is possible to approximate these features the more precise will be the shape encoding.

The residuals R from the approximation are computed by summing all errors for each bin and normalizing by the signal size

$$R = \sum_{i=1}^{h} \frac{|\delta_i|}{v}, \tag{4.13}$$

where h is the number of histogram bins, δ_i is the difference from the histogram i-th bin value to the GMM sampled in the x-axis at bin's midpoint and v is the number of vertices in the model. Therefore, we analyze and plot the residuals by fitting feature histograms with different number of components for the LBO in Figure 4.4. The residual plots show that the error decreases as we increase the number of GMM components, stabilizing the error at a very small value, approximately 0.002.

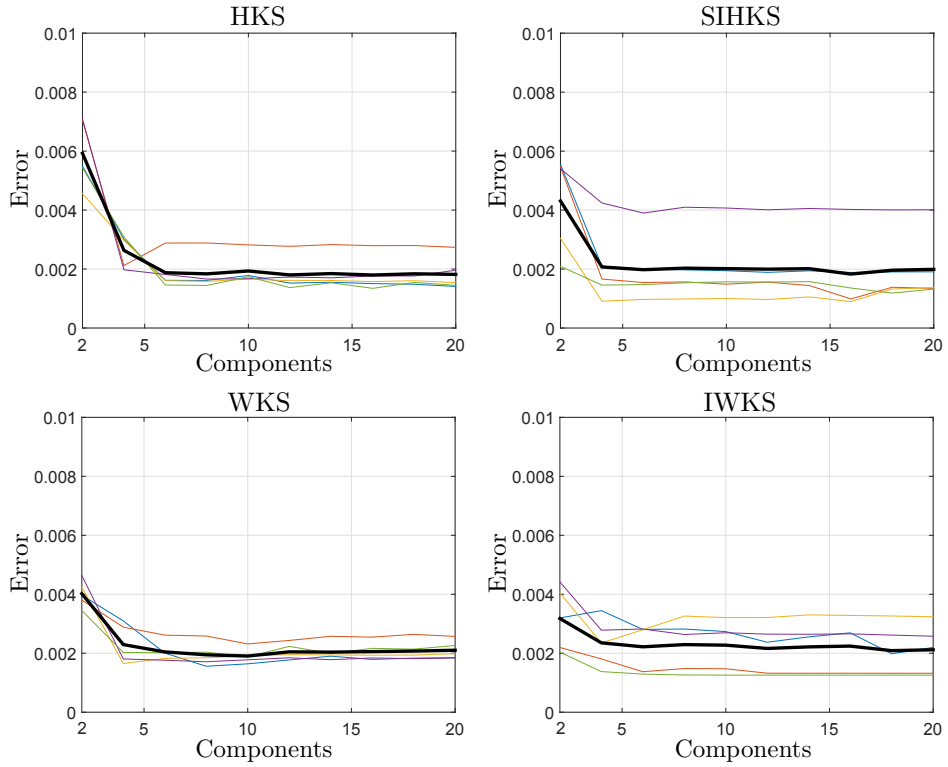


Figure 4.4: Residuals of the GMM fitting on the five feature histograms of Figure 4.3 for each **LBO signature**. The black line represents the average loss of the five histograms and coloured curves represent the individual errors of the five randomly chosen features. As can be seen, the error stabilizes in most of cases when are used in average 5 components or more. As the error converges to a very small value, it is possible to use GMMs to present shape features, once the error approximation of features will be very small.

4.4.2 Benchmark experiments

In this section, we perform an exhaustive evaluation of the encoding framework on three different datasets (SHREC'10, SHREC'11 and SHREC'15). We compare the use of different spectral descriptors (the Heat Kernel Signature (HKS) [Sun et al., 2009], the Scale-invariant Heat Kernel Signature (SI-HKS) [Bronstein and Kokkinos, 2010], the Wave Kernel Signature (WKS) [Aubry et al., 2011b]) and the Improved Wave Kernel Signature (IWKS) [Limberger and Wilson, 2015] in recent benchmarks by encoding them using the Fisher Vector and Super Vector paradigms.

In the following, we also perform an evaluation to compare the performance of *Euclidean distance* (EUC) and *Manifold Ranking* (MR) to compute descriptor dissimilarities. For this, we show six different statistics computed for each descriptor. These statistics are Nearest

Neighbour (NN), First Tier (FT), Second Tier (ST), E-measure (E), Discounted Cumulative Gain (DCG) and mean Average Precision (mAP). These are default statistics proposed by Shilane *et al.* [Shilane et al., 2004] to evaluate shape retrieval problems.

The NN statistic refers to the percentage of closest matches that belongs to the same class as the query. The highest score is 1.0 when every model in the database retrieves at first a model of the same class. The FT and ST give the percentage of models that belongs to query's class that appear within the top K matches. For FT, $K = |C| - 1$ and for ST $K = 2 * (|C| - 1)$, where $|C|$ is the number of members of the query's class. Thus, FT has only the highest score of 1.0 whether, for every query, it retrieves all the remaining models of the same class at first. On the other hand, ST looks at twice the number of matches, therefore, it usually gives a higher value than FT, since it has more space to find correct matches. The e-Measure statistic is given by the precision and recall of the first 32 retrieved models (first page). If classes contain 32 or more models, the maximum value to e-Measure is 1.0, otherwise, it will have a lower upper bound. In the datasets used in this thesis the number of models within classes are smaller than 32 so the maximum score for e-Measure is lower than 1.0. DCG is a statistic which weights correct matches logarithmically based on their position on the retrieval list. The more in the front of the list the more it will score, thus giving more importance to the first results. The maximum score for DCG is 1.0. Finally, mAP is given by the area below the precision and recall curve. Considering the plot as a square 1×1, the maximum value is 1.0 when the result produces a horizontal line across the top of the plot.

In Tables 4.1, 4.3 and 4.5 are shown the retrieval performances of spectral descriptors in the three benchmarks. We use the Laplace-Beltrami operator computed using Meyer's approach [Meyer et al., 2003]. Then, we compute local spectral descriptors and use FV and SV to encode them to shape signatures. In these tables, dissimilarities are computed using Euclidean distance after reducing the dimensionality of the feature vector to 50 using Principal Component Analysis. Analogously, in Tables 4.2, 4.4 and 4.6 are shown retrieval statistics of the same descriptors, however, when dissimilarities are now computed using EMR. EMR is a powerful tool to describe the differences of feature vector because it takes into consideration all objects when computing the difference between two objects. It uses a

graph, where the nodes are the objects. To compute the dissimilarity between two models, it walks over the edges of this graph to determine the distance between the two nodes.

Alongside every table, we show Precision and Recall curves of all descriptors on the three datasets using Euclidean distance in Tables 4.5, 4.7 and 4.9 and using Efficient Manifold Ranking distances in Tables 4.6, 4.8 and 4.10, to compute signature dissimilarities. On this graphs, the bigger is the volume of the curve the better is the performance of the method. In all three datasets, we clearly see an overall improvement of the descriptors when using EMR instead of EUC by looking at the curve of the graphs.

Tables 4.1 and 4.2 and Figures 4.5 and 4.6 show the performance of the spectral framework in SHREC'10. The method that performs the best in this benchmark is the WKS, achieving always the best scores. The IWKS does not perform quite well here because of the differences of same-class shapes. More discussion about this can be find in the next Chapter.

On the other hand, Tables 4.3 and 4.4 and Figures 4.7 and 4.8 show the performance of our spectral framework in SHREC'11. We can see that by using EMR we have significantly improved the performance of the best approach (FV-IWKS) which now performs almost perfectly (mAP 0.997). In the same way, in Tables 4.5 and 4.6 and Figures 4.9 and 4.10, FV-IWKS also performs the best, also improving its performance when using EMR.

In general, by analyzing the performance of all methods, it is possible to state that wave-based methods have a clear dominance over heat-based methods for the computation of spectral signatures for non-rigid shape retrieval when using Fisher Vector or Super Vector encoding schemes. Comparing encoding schemes, SV and FV seem to perform similarly when computing spectral signatures, alternating between the best method.

A full comparison with the state-of-the-art methods will be given in the next Chapter when we introduce the Kinetic Laplace-Beltrami operator, which is a new way for computing the shape spectra for non-rigid shape retrieval.

To summarize, we show in Table 4.7 the improvements of the MR over Euclidean distance when applied to the three datasets using DCG as retrieval score. We can see that, in the majority of the cases (except one), EMR improves the retrieval performance of the benchmark over Euclidean distance, by computing more reliable distances between the shape descriptors.

Table 4.1: Retrieval performance comparison of the different spectral signatures computed with the **LBO** combined with FV and SV applied to the **SHREC'10 benchmark**. Dissimilarities are computed using **Euclidean distance**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9863	0.8029	0.9043	0.6570	0.9530	0.868
FV-SIHKS	0.9863	0.7157	0.8512	0.6173	0.9280	0.812
FV-WKS	**0.9925**	**0.8194**	**0.9279**	**0.6738**	**0.9631**	**0.886**
FV-IWKS	0.9888	0.6686	0.8112	0.5793	0.9161	0.775
SV-HKS	0.9425	0.7294	0.8278	0.6017	0.9147	0.815
SV-SIHKS	0.9750	0.7341	0.8747	0.6313	0.9337	0.818
SV-WKS	**0.9925**	0.7819	0.8889	0.6455	0.9497	0.848
SV-IWKS	0.9813	0.6960	0.8326	0.5957	0.9213	0.800

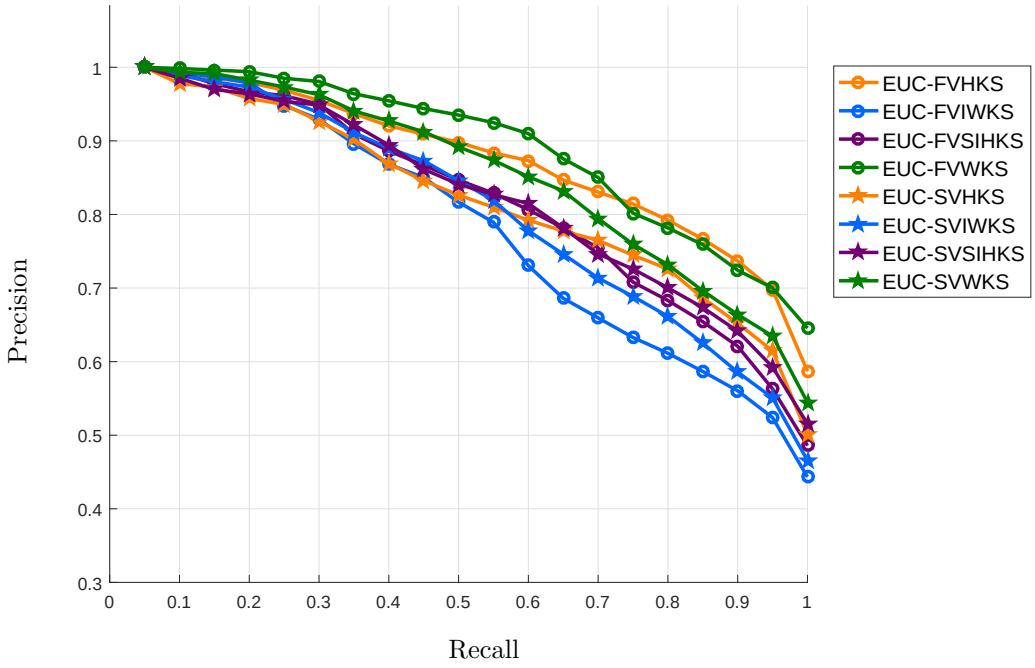


Figure 4.5: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'10** benchmark. The shape spectra is computed by the **LBO**. Distances are computed using **Euclidean distance**. Equal colours represent the same local descriptor. As shown above, the best performance in this benchmark is EUC-FVWKS.

Table 4.2: Retrieval performance comparison of the different spectral signatures computed with the **LBO** combined with FV and SV applied to the **SHREC'10 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9638	0.8462	0.9350	0.6796	0.9570	0.885
FV-SIHKS	0.9400	0.7975	0.9152	0.6594	0.9403	0.841
FV-WKS	0.9813	0.8802	0.9369	0.6912	0.9679	0.894
FV-IWKS	0.9525	0.7676	0.8866	0.6439	0.9359	0.856
SV-HKS	0.9525	0.8291	0.8988	0.6583	0.9438	0.896
SV-SIHKS	0.9638	0.8157	0.9220	0.6714	0.9479	0.846
SV-WKS	**0.9900**	**0.8924**	**0.9629**	**0.7053**	**0.9773**	**0.920**
SV-IWKS	0.9638	0.8137	0.9010	0.6558	0.9489	0.901

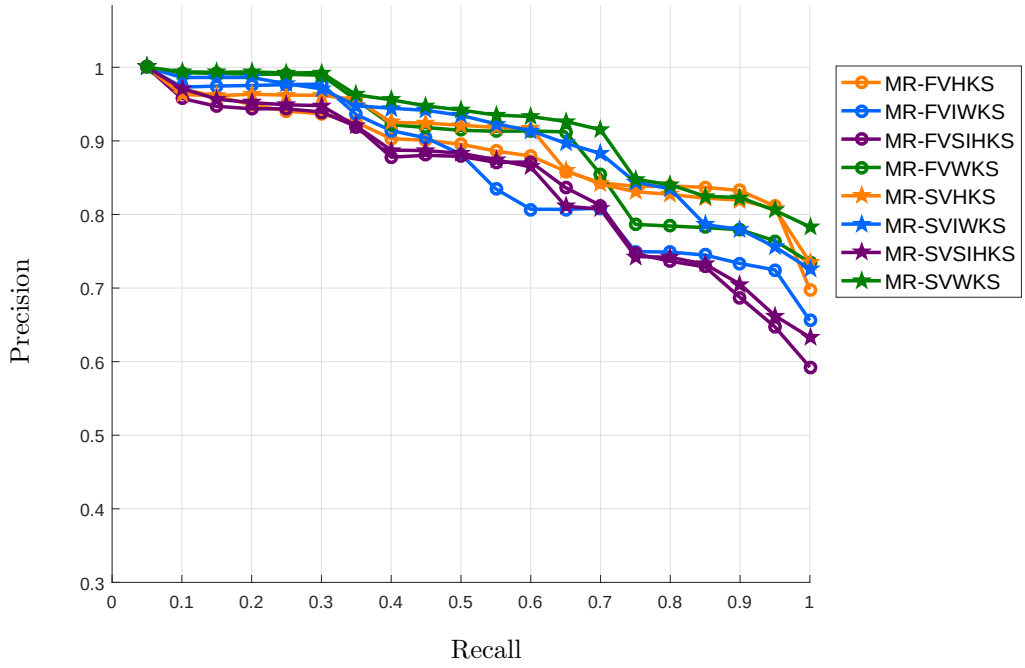


Figure 4.6: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'10** benchmark. The shape spectra is computed by the **LBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the best performance in this benchmark is MR-SVWKS.

Table 4.3: Retrieval performance comparison of the different spectral signatures computed with the **LBO** combined with FV and SV applied to the **SHREC'11 benchmark**. Dissimilarities are computed using **Euclidean distance**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9867	0.8743	0.9530	0.6996	0.9697	0.923
FV-SIHKS	0.9967	0.9314	0.9800	0.7229	0.9870	0.963
FV-WKS	0.9967	0.9398	0.9749	0.7210	0.9870	0.964
FV-IWKS	0.9996	**0.9672**	**0.9842**	**0.7318**	**0.9937**	**0.980**
SV-HKS	0.9717	0.7949	0.8936	0.6523	0.9419	0.858
SV-SIHKS	**1.0000**	0.9239	0.9747	0.7189	0.9840	0.955
SV-WKS	0.9967	0.9189	0.9632	0.7129	0.9822	0.951
SV-IWKS	0.9988	0.9536	0.9803	0.7283	0.9899	0.975

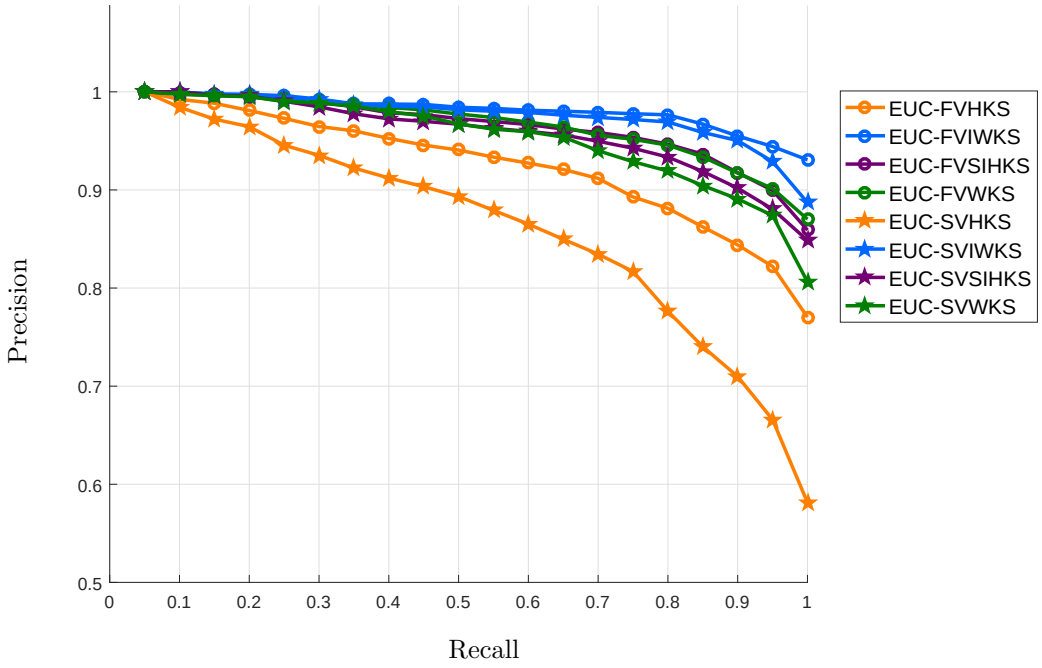


Figure 4.7: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'11** benchmark. The shape spectra is computed by the **LBO**. Distances are computed using **Euclidean distance**. Equal colours represent the same local descriptor. As shown above, the two best performances in this benchmark (EUC-FVIWKS and EUC-SVIWKS) use our proposed descriptor.

Table 4.4: Retrieval performance comparison of the different spectral signatures computed with the **LBO** combined with FV and SV applied to the **SHREC'11 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9683	0.9118	0.9839	0.7175	0.9764	0.949
FV-SIHKS	0.9900	0.9762	0.9982	0.7377	0.9943	0.989
FV-WKS	**0.9983**	0.9710	0.9982	0.7413	0.9967	0.990
FV-IWKS	**0.9983**	**0.9852**	**0.9999**	**0.7446**	**0.9976**	**0.997**
SV-HKS	0.9250	0.8661	0.9567	0.6963	0.9498	0.906
SV-SIHKS	0.9967	0.9545	0.9953	0.7327	0.9889	0.973
SV-WKS	0.9967	0.9777	0.9944	0.7367	0.9949	0.988
SV-IWKS	0.9600	0.9554	0.9926	0.7303	0.9798	0.971

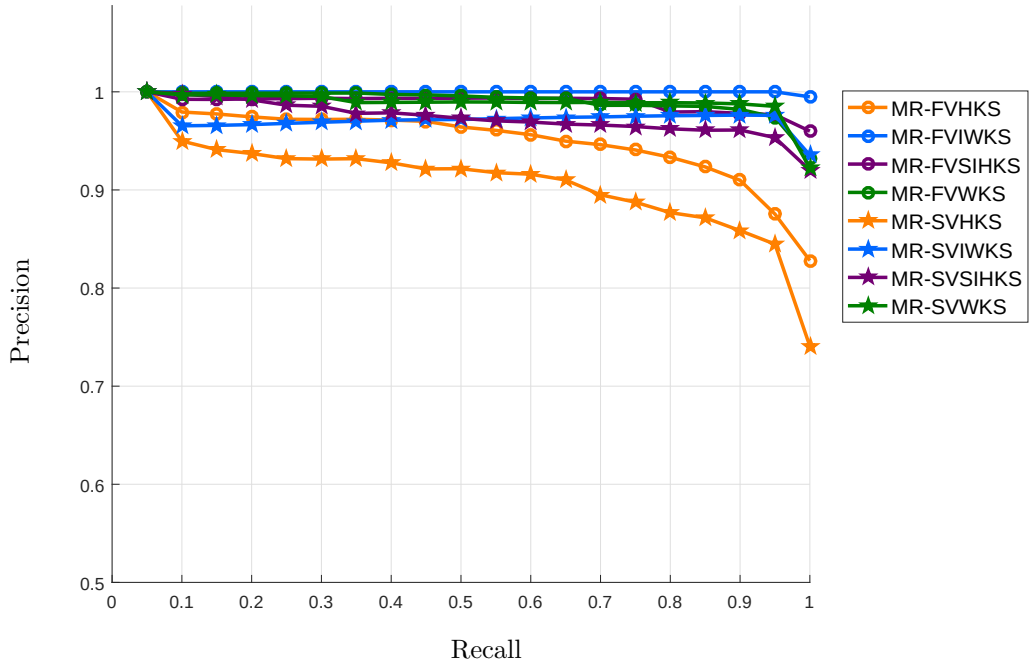


Figure 4.8: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'11** benchmark. The shape specta is computed by the **LBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the best performance in this benchmark (MR-FVIWKS) use our proposed descriptor.

Table 4.5: Retrieval performance comparison of the different spectral signatures computed with the **LBO** combined with FV and SV applied to the **SHREC'15 benchmark**. Dissimilarities are computed using **Euclidean distance**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9567	0.7489	0.8292	0.6661	0.9134	0.807
FV-SIHKS	0.9658	0.8104	0.8770	0.7102	0.9382	0.919
FV-WKS	0.9725	0.8628	0.9183	0.7511	0.9553	0.924
FV-IWKS	**0.9975**	**0.9463**	**0.9801**	**0.8102**	**0.9884**	**0.969**
SV-HKS	0.9217	0.6168	0.7061	0.5564	0.8539	0.724
SV-SIHKS	0.9642	0.7559	0.8371	0.6698	0.9222	0.906
SV-WKS	0.9600	0.7685	0.8520	0.6842	0.9243	0.872
SV-IWKS	0.9867	0.8748	0.9387	0.7649	0.9683	0.924

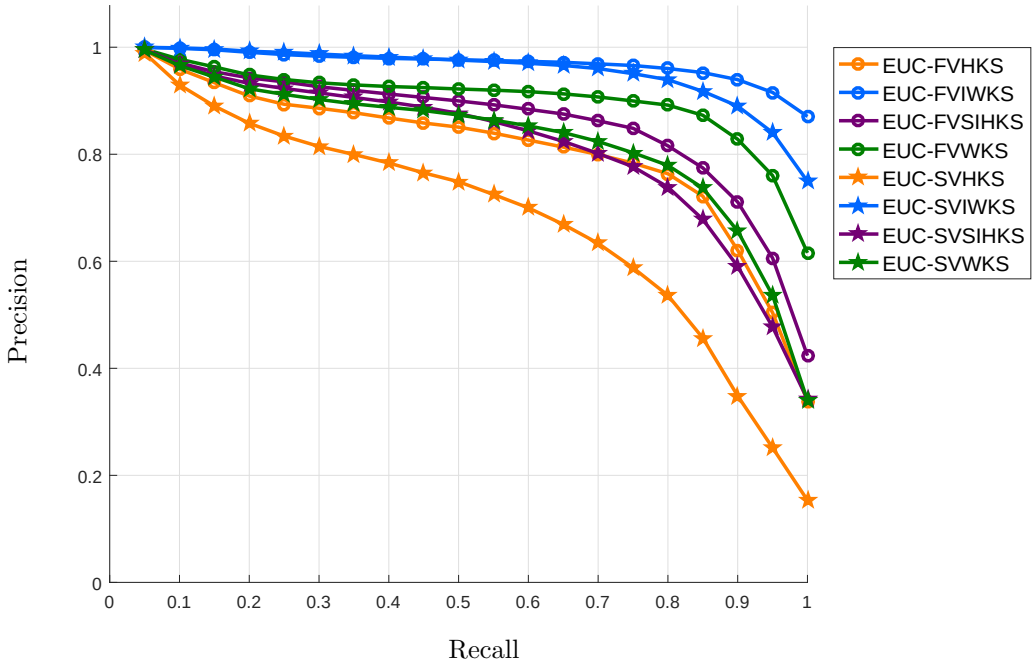


Figure 4.9: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'15** benchmark. The shape spectra is computed by the **LBO**. Distances are computed using **Euclidean distance**. Equal colours represent the same local descriptor. As shown above, the two best performance in this benchmark (MR-FVIWKS and MR-SVIWKS) use our proposed descriptor.

Table 4.6: Retrieval performance comparison of the different spectral signatures computed with the **LBO** combined with FV and SV applied to the **SHREC'15 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9394	0.8356	0.8858	0.7210	0.9373	0.876
FV-SIHKS	0.9850	0.9371	0.9622	0.7944	0.9795	0.957
FV-WKS	0.9771	0.9241	0.9487	0.7820	0.9725	0.937
FV-IWKS	**0.9936**	**0.9736**	**0.9922**	**0.8211**	**0.9933**	**0.979**
SV-HKS	0.9107	0.7579	0.8268	0.6645	0.9039	0.805
SV-SIHKS	0.9779	0.9188	0.9553	0.7823	0.9734	0.940
SV-WKS	0.9640	0.8960	0.9275	0.7620	0.9611	0.912
SV-IWKS	0.9821	0.9475	0.9722	0.8029	0.9820	0.957

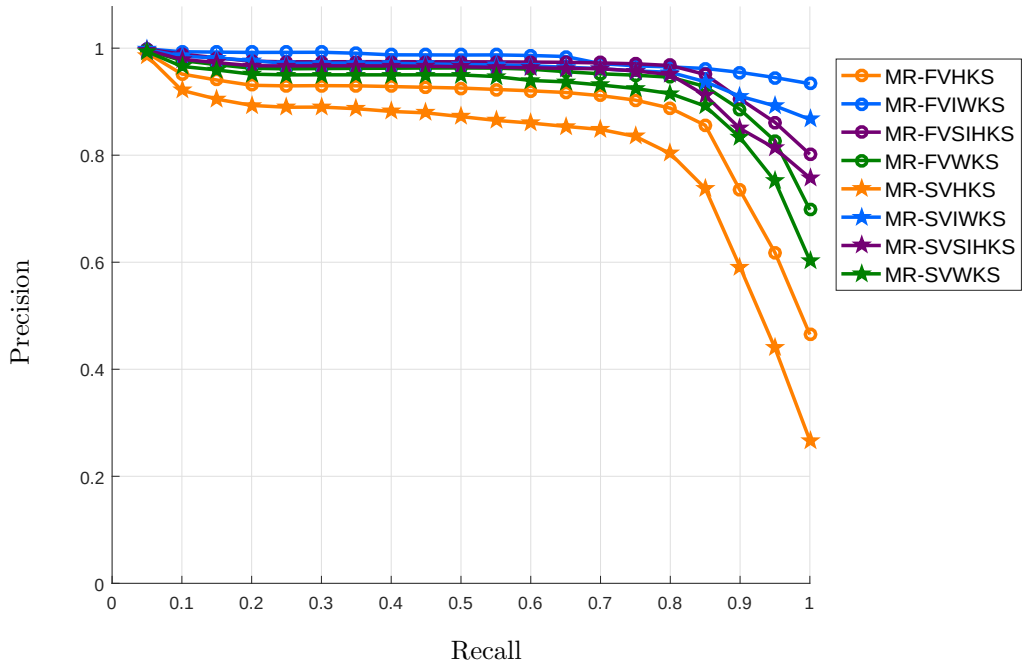


Figure 4.10: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'15** benchmark. The shape spectra is computed by the **LBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the two best performance in this benchmark (MR-FVIWKS and MR-SVIWKS) use our proposed descriptor.

Table 4.7: Performance improvements of EMR over EUC using DCG as retrieval score.

Method	SHREC'10	SHREC'11	SHREC'15
FV-HKS	0.42%	0.69%	2.62%
FV-SIHKS	1.33%	0.74%	4.40%
FV-WKS	0.49%	0.98%	1.80%
FV-IWKS	2.16%	0.39%	0.50%
SV-HKS	3.19%	0.84%	5.85%
SV-SIHKS	1.52%	0.50%	5.55%
SV-WKS	2.91%	1.29%	3.98%
SV-IWKS	3.00%	-1.02%	1.41%

4.5 Summary

In this chapter, we have shown how to encode local spectral descriptors using the Fisher Vector or the Super Vector. These, compute discriminative measurements of similarity between a statistical model, which is represented by a dictionary of features, and the respective shape descriptors. These measurements of correlation give the shape a unique signature which is based on its local characteristics. Furthermore, by computing enhanced distances with the Efficient Manifold Ranking algorithm, we improve shape retrieval performance of classical spectral features, *e.g.* the HKS, SIHKS, WKS and IWKS.

We can also examine the performance of our new proposed descriptor. As can be seen in the previous experiments, the IWKS achieves state-of-the-art performance comparing to other spectral signatures on SHREC'11 and SHREC'15. This is due to the fact that the IWKS aggregates more information than other descriptors thus being able to better differentiate from other classes of objects.

Chapter 5

Kinetic Laplace-Beltrami Operator

To fully benefit from the power of the shape spectra it is necessary to learn the class-specific features of the spectrum that are more important to the problem at hand since the shape spectra has a lot of information and it is hard to code all the important information into a signature. On the other hand, we decided to take a step back and modify the shape spectra itself so it can be more robust to non-rigid deformations. This way, the potential of shape spectra can be fully exploited for the desirable application.

In this chapter, we deal with the problem of creating robust spectral signatures for meshes. Point-cloud models will be addressed later in this thesis. Thus, we present a new method for non-rigid 3D shape retrieval by computing enhanced spectral signatures from meshes. Therefore, we introduce the Kinetic Laplace-Beltrami Operator (KLBO), a method for computing spectral components which is intrinsically less variant to non-rigid deformations. The KLBO downweights joint regions by a curvature-based kinetic term, removing the influence of shape's articulations on shape descriptors. The weighting is small in areas on the shape where articulations are likely to occur and also stable to rigid and non-rigid motions. This weighting changes the diffusion speed on the surface where articulations are likely to occur decreasing their influence in the final descriptor.

The KLBO has participated in a shape retrieval contest in the *SHREC'17 track: Retrieval of surfaces with similar relief patterns* [Biasotti et al., 2017], where it achieves the best performance. This chapter is organized in the following way: we discuss and propose our method in Section 5.1. Then, we show how it performs on recent non-rigid benchmarks

in Section 5.2. We analyse its time complexity in Section 5.3. Finally, we summarize the benefits of our method in Section 5.4.

5.1 Method

The family of spectral methods, exemplified by Sun et al. [Sun et al., 2009] and Aubry et al. [Aubry et al., 2011b], are very attractive for 3D shape representation because they are isometrically invariant, easy to make scale invariant, partly resistant to shape deformations, and easy to calculate even for large meshes. They are also resistant to some types of noise, which appears in the high-frequency part of the shape spectrum and can be downweighted. The essence of these methods is to define a dynamic equation on the surface of the shape (for example, the heat equation or the wave equation) and use the solution to extract information about the shape.

To be clear, differently from a weighted manifold [Grigor'yan, 2006], we start by defining the Lagrangian of the dynamics using *classical field theory*. Thus, we weight the physical field using a smooth positive kinetic density. Both methods share some functional similarities, however, our formulation for the problem is completely different. We begin by defining the Lagrangian density of the system

$$\mathcal{L}(\phi, \bigtriangledown\phi, \dot{\phi}, \mathbf{x}, t) = T - V \tag{5.1}$$

where T is the kinetic energy (K.E.) and V is the potential energy, and ϕ represents a field defined over the space (i.e. over the surface of the object). The *action* of the system is given by the integral of the Lagrangian density:

$$S(\mathcal{L}) = \int \mathcal{L} d\mathbf{x} dt \tag{5.2}$$

The dynamics of the system can be recovered from Hamilton's principle, which states that the action should be stationary for the true dynamic evolution of the system. This leads to

the Euler-Lagrange equation for the dynamics:

$$\frac{\partial \mathcal{L}}{\partial \phi} - \bigtriangledown \cdot \frac{\partial \mathcal{L}}{\partial \bigtriangledown \phi} - \frac{\partial}{\partial t}\left(\frac{\partial \mathcal{L}}{\partial \dot{\phi}}\right) = 0 \tag{5.3}$$

By defining an appropriate Lagrangian and solving the resulting Euler-Lagrange equations, we can find a shape signature that weights kinetic energy differently across the field. The kinetic energy is generated by different forms of motions. The movement of joint regions can be physically interpreted as translational (when one part is moved from one place to another), rotational (when the joint is rotated) and/or vibrational (when part of the shape is also deformed by the motion), depending on the type of articulation and deformation. Thus, we weight the kinetic energy to remove joint-articulation's effect on shape signatures.

In the general scaled Lagrangian:

$$\mathcal{L} = \frac{1}{2}\dot{\phi}\dot{\phi^*} + \frac{1}{2}(\phi\dot{\phi^*} - \phi^*\dot{\phi}) - \frac{1}{2}\nabla\phi \cdot \nabla\phi^* - \phi^* V \phi \tag{5.4}$$

ϕ is a (possibly) complex field, so there are in fact two "fields" corresponding to the real part and the imaginary part. In practice it is easier to consider the field ϕ and its complex-conjugate ϕ^* which are linear combinations of the real and imaginary parts and so do not affect the calculations. The first two terms are kinetic energy terms, the first corresponding to a standard K.E. proportional to the square of the velocity. The second is a K.E. term where the energy increases with the size of the field. The second two terms are field potentials, the first related to the gradient and the second to some external potential field V.

From this point, we can define the shape descriptor by choosing appropriate terms from (5.4). To define the heat equation we choose second and third terms

$$\mathcal{L} = \frac{1}{2}(\phi\dot{\phi^*} - \phi^*\dot{\phi}) - \frac{1}{2}\nabla\phi \cdot \nabla\phi^* \tag{5.5}$$

which applied to (5.3) gives dynamics

$$\nabla^2\phi = \dot{\phi}. \tag{5.6}$$

This is the heat equation and, ultimately, leads to the definition of the family of Heat Kernel Signatures. In the same way, to define the wave equation we choose first and third terms

$$\mathcal{L} = \frac{1}{2}\dot{\phi}\dot{\phi}^* - \frac{1}{2}\nabla\phi \cdot \nabla\phi^* \tag{5.7}$$

which gives dynamics

$$\nabla^2\phi = \ddot{\phi}. \tag{5.8}$$

which is the wave equation and ultimately leads to the Wave Kernel Signature of Aubry *et al.* [Aubry et al., 2011b].

To reduce the effect of object articulations, we introduce a kinetic energy weighting term (first to the wave equation) into the Lagrangian

$$\mathcal{L} = \frac{k}{2}C(\mathbf{x})\dot{\phi}\dot{\phi}^* - \nabla\phi \cdot \nabla\phi^* \tag{5.9}$$

By applying this time (5.9) to (5.3) we get

$$\bigtriangledown^2\phi = kC(\mathbf{x})\ddot{\phi} \tag{5.10}$$

where k is a normalization term and $C(\mathbf{x})$ is a spatially varying weighting function which is small in areas on the shape where articulations are likely to occur and also stable to non-rigid motions. Standard separation of variables and discretization gives

$$L\phi_x = \lambda K\phi_x \tag{5.11}$$

where K is a diagonal matrix where its diagonal elements are $C(\mathbf{x})$. Putting these elements together, following a standard discretization procedure from (2.2), the eigenvectors associated with the signature are solutions of the generalized eigenproblem

$$W\boldsymbol{\phi} = \lambda AK\boldsymbol{\phi}. \tag{5.12}$$

The kinetic term can be modified in the heat Lagrangian in the same way:

$$\mathcal{L} = \frac{k}{2} C(\mathbf{x})(\phi\dot{\phi}^* - \phi^*\dot{\phi}) - \nabla\phi \cdot \nabla\phi^* \tag{5.13}$$

and this leads to exactly the same spatial eigenvectors of (5.12), although the solution is different due to difference in the time derivatives. To derive a descriptor which is less variant to non-rigid motions we merely need to choose an appropriate function $C(\mathbf{x})$ which is smaller in articulated points than in rigid areas of the shape.

The weighting function $C(\mathbf{x})$ needs to reduce the effect of areas which are most different when comparing non-rigid shapes. When a human model moves its arm, what happens is that the arm joint region changes its curvature. At one side it becomes smaller (more negative) and at the other side it becomes bigger (more positive). After analyzing the structure of shapes we found a relation between the positive volume of the local surface patches and their joints. By using this analysis, joint regions are consistently less weighted than other regions. We compute the volume of a surface patch by integrating a quadric patch, which is a representation of this surface in the local coordinate system

$$\iint\limits_{x^2+y^2<R^2} (k_1 x^2 + k_2 y^2 - z)dxdy \tag{5.14}$$

Here k_1 and k_2 are the principal curvatures of the surface patch since they are the eigenvalues of the symmetric matrix in the second fundamental form

$$II = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \tag{5.15}$$

After integrations of (5.14), the volume inside the circle with radius R centered at the respective vertex $\mathbf{v}_i$ is defined as

$$\frac{\pi R^4}{2}\frac{k_1 + k_2}{2} - \pi z R^2 \tag{5.16}$$

where $(k_1 + k_2)/2$ is the **mean curvature** (H), $\pi R^4/2$ is a scaling factor and z dictates

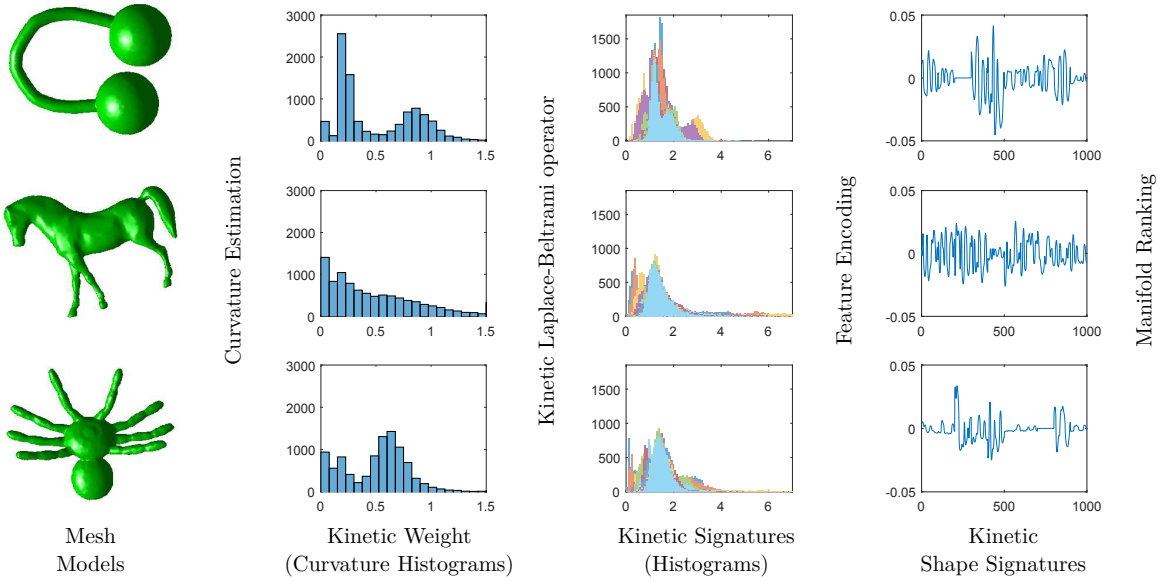


Figure 5.1: Pipeline proposed for the non-rigid shape retrieval problem for triangle meshes. By weighting the kinetic energy on the Euler-Lagrangian equation we reduce the effect of shape articulations, causing same-class shapes' signatures to be closer to each other. Then, by encoding the kinetic signatures using either Fisher Vector (Section 2.5.6) or Super Vector (Section 2.5.7) we are able to compare shapes efficiently using Manifold Ranking technique.

the position of the reference plane defined by the surface patch's normal. By removing the scaling factor and taking $z = 0$ we find that the volume is proportional to the mean curvature thus we define the weighting function $C(\mathbf{x})$ as

$$C(\mathbf{x}) = \max(\epsilon, H(\mathbf{x})), \tag{5.17}$$

where $H(\mathbf{x})$ is the mean curvature of the surface patch at position $\mathbf{x}$ and ϵ a very small number (e.g. 10^{-8}). The scaling factor is not significant because a scaling normalization is performed after this stage. To facilitate computations, we extract directly mean curvatures from shapes by computing a curvature tensor at each vertex, according to [Rusinkiewicz, 2004].

With these ingredients, Equation (5.11) can be solved as a generalized eigenvalue problem. The resulting eigensystem is then used to construct a shape signature following the appropriate method for the particular Lagrangian, i.e. using Equation (5.9) for either the WKS or IWKS or (5.13) for either the HKS or SIHKS.

Therefore, we call these methods, using modified kinetic energy terms in the Lagrangian, *Kinetic Laplace-Beltrami operator* or KLBO methods. Fig. 5.1 summarizes the main steps of the KLBO pipeline.

This method has similarities with the *curvature aggregation* (Section 3.3) in the way both reduce the influence of shape motions, however, each method does that in a different way and therefore they can be used with each other. The KLBO reduces the influence of articulations by changing the diffusion speed on the surface where articulations are likely to occur, *i.e.*, changing the shape spectrum, while the *curvature aggregation* only downweights joint regions after the signatures have been computed.

5.2 Experiments

In this section, we present detailed experiments on the shape retrieval performance and speed of our algorithm. For that, we use three non-rigid databases: SHREC'10 [Lian et al., 2010], SHREC'11 [Lian et al., 2011] and SHREC'15 [Lian et al., 2015]. However, we start by showing that the KLBO features can be approximated by Gaussian Mixture models and therefore can be used with Fisher Vector and Super Vector encoding schemes.

Figure 5.2 shows 5 randomly chosen shape features computed with the KLBO and their respective GMMs approximated using the Expectation-maximization algorithm. Figure 5.3 shows errors generated by these respective approximations. We can see that errors stabilize around 0.002 for all signatures (HKS, SIHKS, WKS and IWKS). The low error generated by the approximation tells us that GMM is also a good way to represent KLBO features.

From now, we present retrieval performance of our KLBO descriptor. First, we show detailed statistics of our framework for spectral signatures when using the KLBO instead of the LBO. Thus, we use the shape spectra computed from the KLBO to construct spectral descriptors (HKS, SIHKS, WKS and IWKS). Then, we encode these descriptors either with FV or SV. Finally, we perform shape retrieval on benchmarks by computing Efficient Manifold Ranking distances between signatures.

The detailed results of these experiments can be seen in Tables 5.1, 5.2 and 5.3, applied respectively in SHREC'10, SHREC'11 and SHREC'15. See that, IWKS is ranked first for

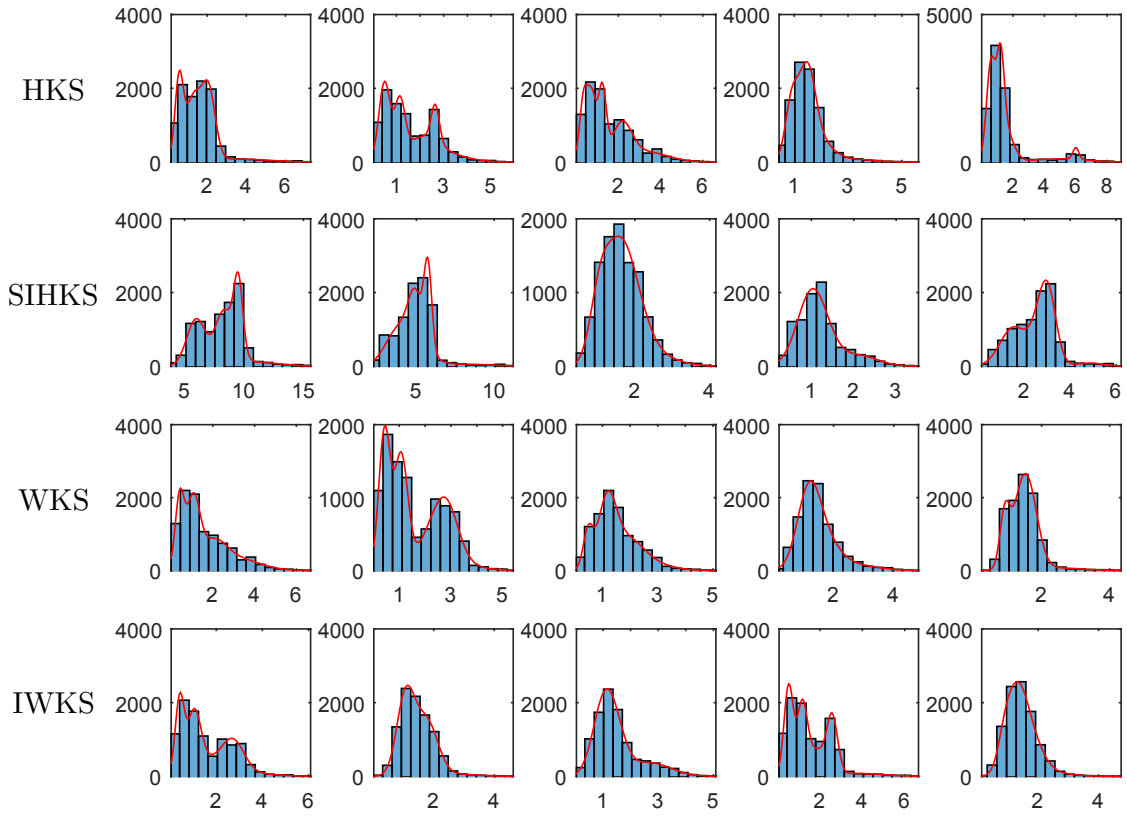


Figure 5.2: Histograms of randomly-chosen shape features plotted with the respective learned GMM with 5 components for different **KLBO signatures**. Each row represents features from HKS, SIHKS, WKS and IWKS, respectively, from the first model of the database.

SHREC'11 and SHREC'15, while WKS is ranked first for SHREC'10 and SHREC'11.

The reason for the IWKS be ranked low in SHREC'10 is because models are sometimes quite different, *i.e.*, the models are not the same. This aspect can be seen in figure 5.4. While other spectral methods rely most of their features on gross topological structures, the IWKS is more sensitive to changes on the shape of the model.

For comparing other mesh descriptors with our KLBO, we took state-of-the-art descriptors from the literature and the algorithms that performed better on each dataset. All these results were taken from the respective author's papers.

Table 5.5 shows a comparison of our best run on SHREC'10 (KLBO-SVWKS, see Table 5.1) against best descriptors taken from SHREC'10 benchmark (MR-BF-DSIFT-E, DMEVD_run1, CF) and other state-of-the-art techniques, where referenced. Our descriptor exhibit the best retrieval score when compared to all other methods. It is worth noting that the SHREC'10 models are substantially different in nature to the others and also very challenging because

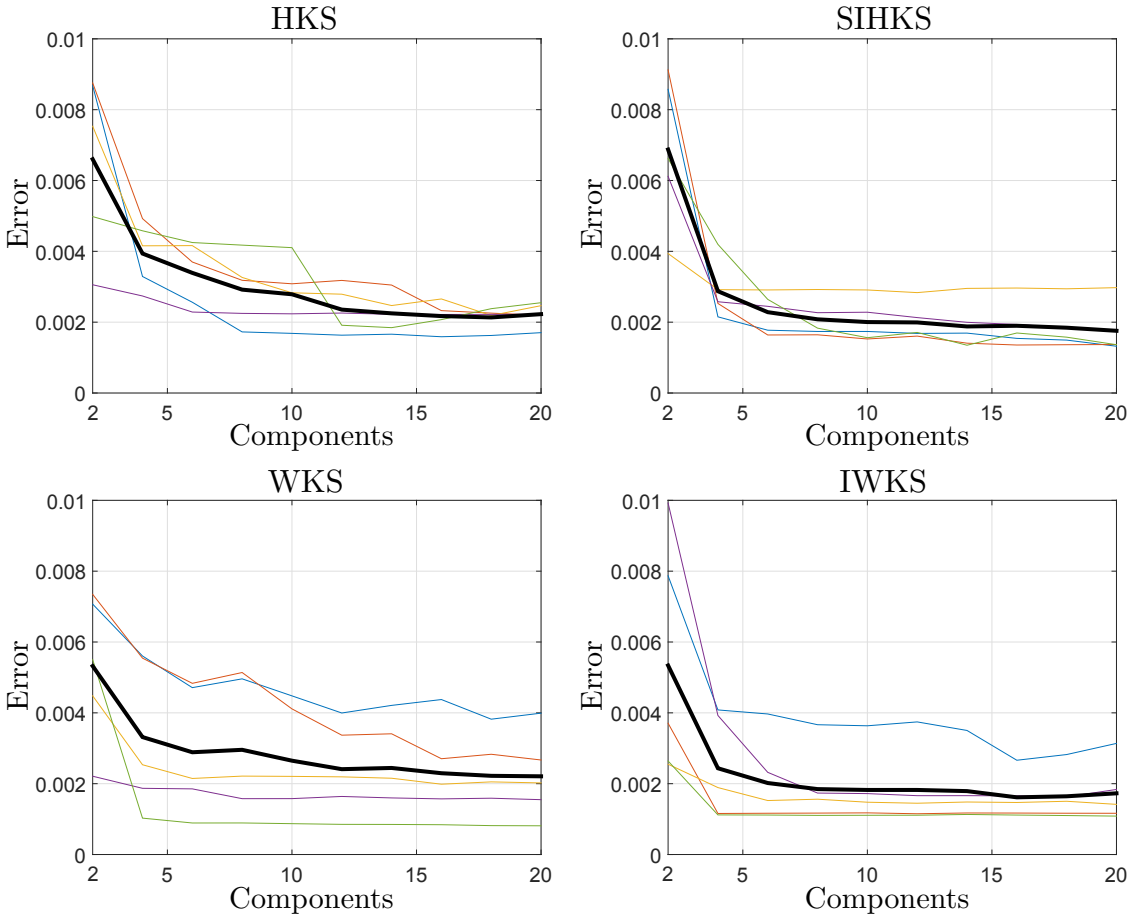


Figure 5.3: Residuals of the GMM fitting on the five feature histograms of Figure 5.2 for each KLBO signature. The black line represents the average loss of the five histograms and coloured curves represent the individual errors of the five randomly chosen features. As can be seen, the error stabilizes in most of cases when are used in average 5 components or more.

sometimes classes are very similar (octopuses, crabs and spiders have similar shapes and other times shapes can be deformed instead of only articulated, as shown in Figure 5.4). This dataset also contains some noise which is naturally most handled by spectral signatures.

Table 5.6 compares the performance of KLBO-FVWKS, KLBO-SVIWKS and KLBO-SVSIHKS against the best methods on SHREC'11 benchmark (SD-GDM-meshSIFT, MDS-CM-BOF, OrigM-n12-nrmA, FOG+MRR, BOGH, LSF) and other state-of-the-art techniques, where referenced. As shown, our methods clearly outperforms all others, achieving a perfect retrieval score (mAP 100.0%). It is important to note that this result is achieved without any training or learning method during the local descriptor computation, which

means that the algorithm inspects models only once for computing local descriptors. Finally, our methods are capable of retrieving all 19 correct matches for the 30 classes, i.e. if any object from these classes is taken as query it will retrieve all remaining shapes from the same class at first.

On SHREC'15 datasets, our method is the second best (see Table 5.7), performing only marginally below the best. Although it does not achieve the first position it is interesting to note the stability of the results on the three non-rigid benchmarks. Other methods, for example SV-LSF_kpaca50 achieves an excellent retrieval score on SHREC'15, however, its performance on SHREC'11 is not among the top tier.

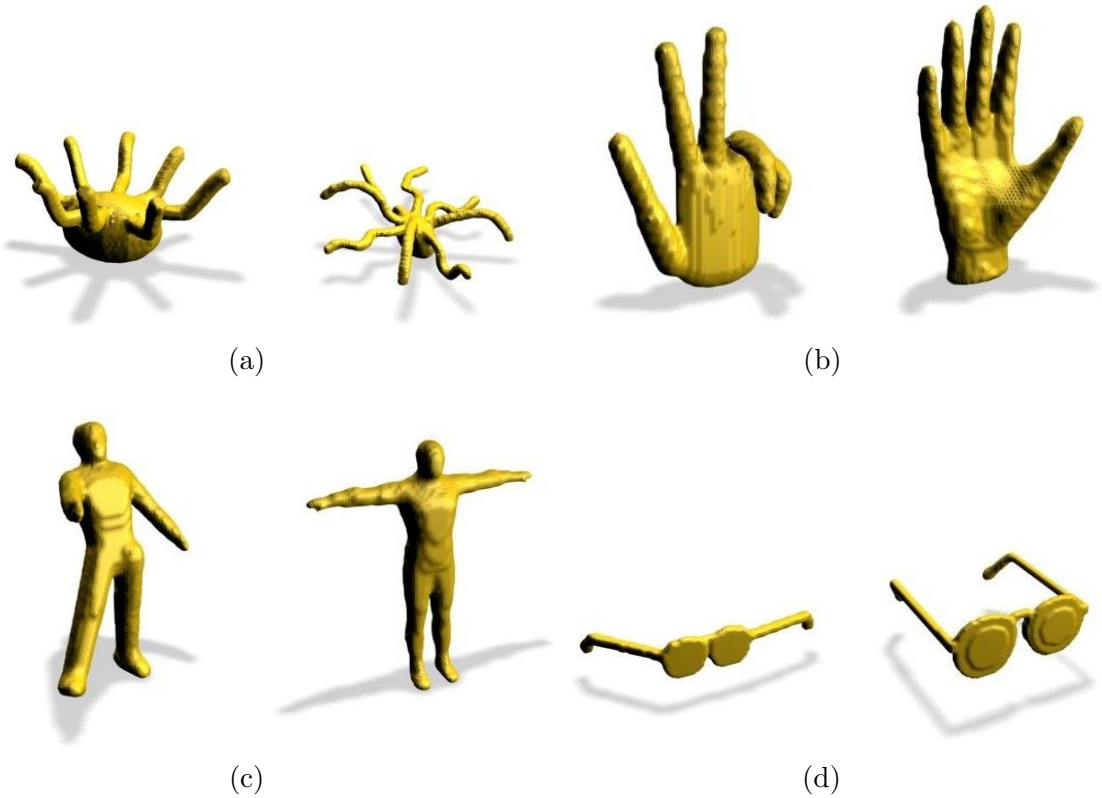


Figure 5.4: Models from SHREC'10 showing that they are substantially distinct even belonging to the same class. (a) shows *Octopuses* that differ in thickness. (b), (c) and (d) show respectively *Hands*, *Humans* and *Spectacles* that differ in shape. Thumbnail images taken from the track website [Lian et al., 2010].

We also tested the KLBO on the SHREC'14 dataset [Pickup et al., 2014b]. Tables 5.8 and 5.9 show performance statistics on the Synthetic and Real dataset, respectively, and Figures 5.8 and 5.9 show the respective Precision and Recall curves.

Table 5.1: Retrieval performance comparison of the different spectral signatures computed with the **KLBO** combined with FV and SV applied to the **SHREC'10 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9850	0.9088	0.9567	0.7043	0.9774	0.945
FV-SIHKS	0.9788	0.8764	0.9630	0.7045	0.9706	0.909
FV-WKS	0.9888	0.9266	0.9787	0.7148	0.9811	0.963
FV-IWKS	0.9425	0.8047	0.9361	0.6767	0.9444	0.846
SV-HKS	0.9650	0.8761	0.9433	0.6878	0.9645	0.919
SV-SIHKS	0.9888	0.8665	0.9601	0.7052	0.9722	0.922
SV-WKS	**0.9950**	**0.9656**	**0.9952**	**0.7328**	**0.9938**	**0.991**
SV-IWKS	0.9650	0.8682	0.9755	0.7129	0.9676	0.907

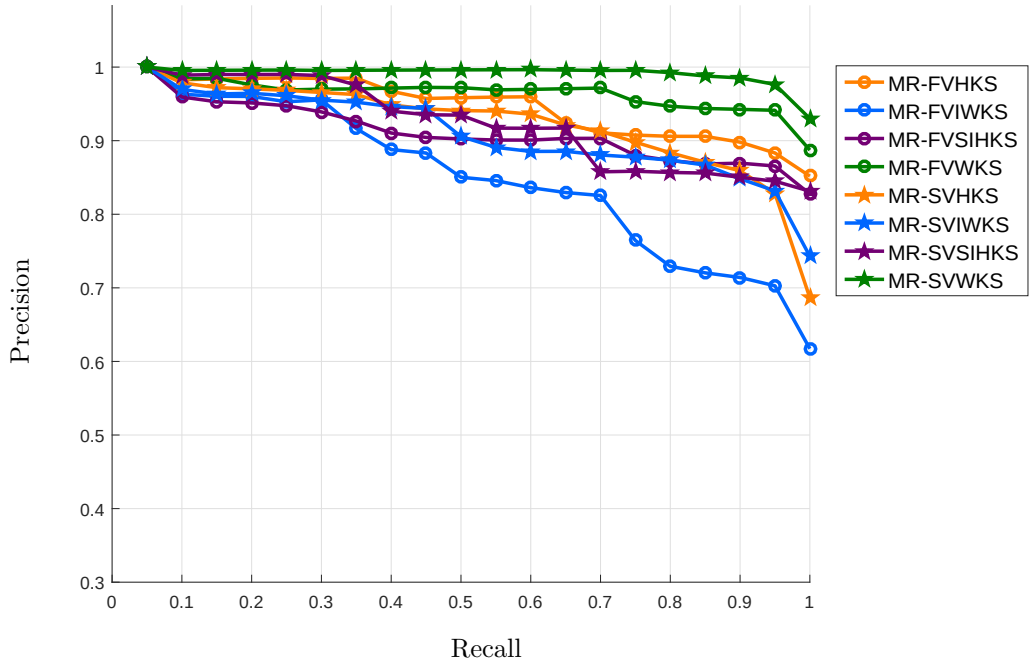


Figure 5.5: Precision and Recall plot of different spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'10 benchmark**. The shape spectra is computed by the **KLBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. The best performances are MR-FVWKS and MR-SVWKS.

Table 5.2: Retrieval performance comparison of the different spectral signatures computed with the **KLBO** combined with FV and SV applied to the **SHREC'11 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9983	0.9967	0.9967	0.7426	0.9985	0.997
FV-SIHKS	**1.0000**	0.9864	**1.0000**	0.7425	0.9986	0.995
FV-WKS	**1.0000**	**1.0000**	**1.0000**	**0.7451**	**1.0000**	**1.000**
FV-IWKS	**1.0000**	0.9966	0.9999	0.7441	0.9996	0.999
SV-HKS	0.9933	0.9792	0.9932	0.7361	0.9946	0.989
SV-SIHKS	**1.0000**	**1.0000**	**1.0000**	**0.7451**	**1.0000**	**1.000**
SV-WKS	**1.0000**	0.9983	0.9983	0.7439	0.9998	0.999
SV-IWKS	**1.0000**	**1.0000**	**1.0000**	**0.7451**	**1.0000**	**1.000**

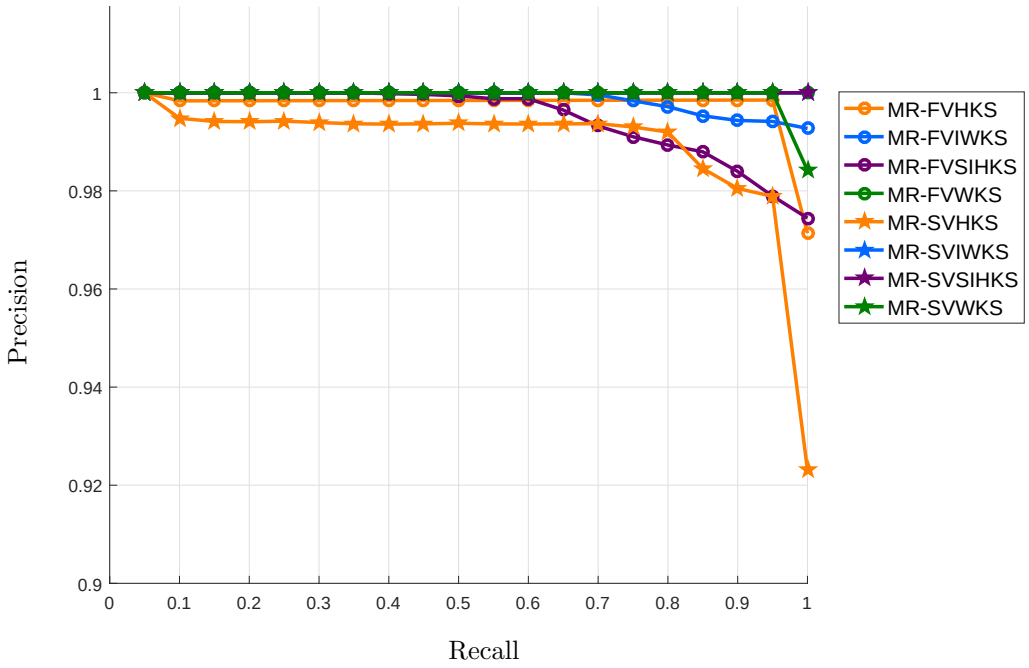


Figure 5.6: Precision and Recall plot of different spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'11 benchmark**. The shape spectra is computed by the **KLBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. The best performances are MR-FVWKS, MR-SVSIHKS and MR-SVIWKS with 100% of retrieval performance.

Table 5.3: Retrieval performance comparison of the different spectral signatures computed with the **KLBO** combined with FV and SV applied to the **SHREC'15 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9465	0.8492	0.8792	0.7225	0.9391	0.877
FV-SIHKS	0.9852	0.9473	0.9604	0.7988	0.9790	0.956
FV-WKS	0.9823	0.9382	0.9559	0.7925	0.9770	0.957
FV-IWKS	**0.9992**	**0.9859**	**0.9933**	**0.8269**	**0.9959**	**0.992**
SV-HKS	0.9367	0.8050	0.8487	0.6918	0.9221	0.852
SV-SIHKS	0.9867	0.9527	0.9621	0.8019	0.9813	0.963
SV-WKS	0.9800	0.9297	0.9464	0.7858	0.9744	0.945
SV-IWKS	0.9969	0.9792	0.9886	0.8232	0.9943	0.986

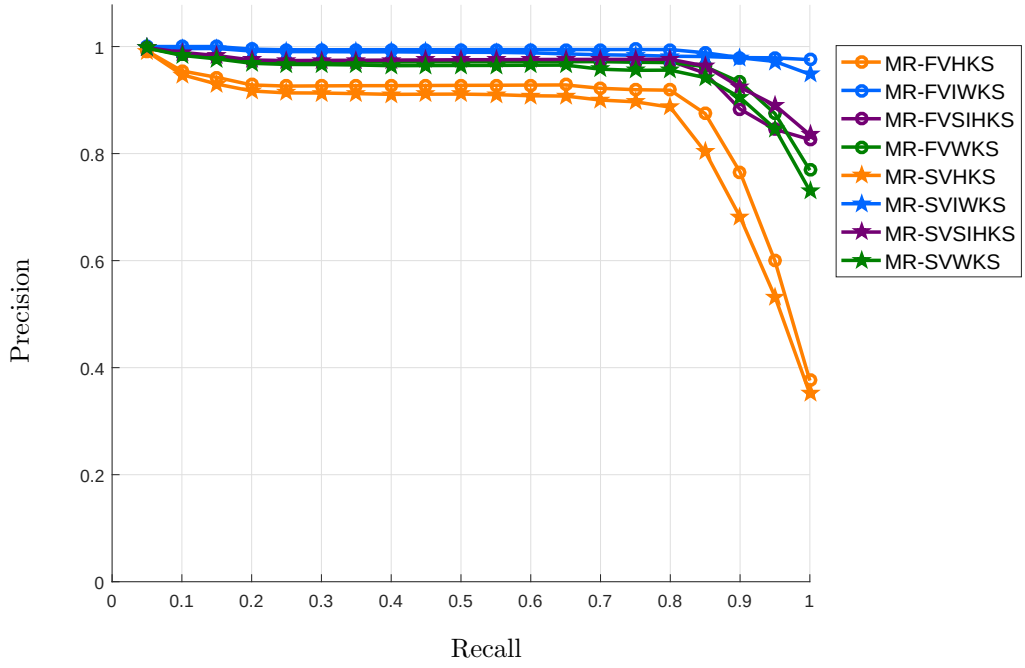


Figure 5.7: Precision and Recall plot of different spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'15 benchmark**. The shape spectra is computed by the **KLBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the two best performance in this benchmark are MR-FVIWKS and MR-SVIWKS, which use our proposed descriptor.

Table 5.4: Performance improvements of the KLBO over the LBO using DCG as retrieval score. Dissimilarities are computed using Efficient Manifold Ranking.

Method	SHREC'10	SHREC'11	SHREC'15
FV-HKS	2.14%	2.26%	0.19%
FV-SIHKS	3.22%	0.43%	-0.05%
FV-WKS	1.37%	0.33%	0.47%
FV-IWKS	0.91%	0.20%	0.26%
SV-HKS	2.19%	4.72%	2.01%
SV-SIHKS	2.56%	1.12%	0.82%
SV-WKS	1.68%	0.49%	1.38%
SV-IWKS	1.97%	2.06%	1.26%

Table 5.5: Retrieval performance comparison with state-of-the-art signatures on SHREC'10. We show best runs from the three groups that performed better in SHREC'10 (MR-BF-DSIFT-E, DMEVD_run1, CF) and other recent descriptors that outperformed those, against our descriptor (KLBO-SVWKS). In bold are highlighted the best performances for each retrieval measure.

Descriptor	e-Measure	mAP [%]
KLBO-SVWKS	**0.7328**	**99.1**
ConTopo++ [Sfikas et al., 2011]	0.7140	97.6
MR-BF-DSIFT-E	0.7055	95.4
DMEVD_run1	0.7012	94.1
MDS-ZFDR [Li et al., 2014a]	-	94.1
MR-SVWKS	0.7053	92.0
EUC-SVIWKS [Limberger and Wilson, 2015]	0.5957	80.0
CF	0.5527	75.2

KLBO does not perform very well in these datasets because they are substantially different from the others in terms of class similarity. Both the Synthetic and Real dataset from SHREC'14 feature only human models, where the difference between classes are sometime minimal. Unfortunately, the KLBO fails in describing these small differences. It is interesting to note that the SIHKS performs the best considering the spectral descriptors. Further research is still needed to identify the reasons this is happening.

Table 5.6: Retrieval performance comparison with state-of-the-art signatures on SHREC'11. We show best runs from the six groups that performed better in SHREC'11 (SD_GDM-meshSIFT, MDS-CM-BOF, OrigM-n12-normA, FOG+MRR, BOGH, LSF) and other recent descriptors that outperformed those, against our descriptor (KLBO-FVWKS). In bold are highlighted the best performances for each retrieval measure.

Descriptor	e-Measure	mAP [%]
KLBO-FVWKS/KLBO-SVIWKS/KLBO-SVSIHKS	**0.7451**	**100.0**
MR-FVIWKS	0.7446	99.7
3DVFF [Furuya and Ohbuchi, 2014]	-	99.1
SD-GDM-meshSIFT	0.7358	98.5
EUC-FVIWKS [Limberger and Wilson, 2015]	0.7318	98.0
MDS-ZFDR [Li et al., 2014a]	-	97.5
SV-DSIFT [Takahiko Furuya, 2015]	-	97.2
R-BiHDM-L23 [Ye and Yu, 2015]	0.7300	-
SGWC-BoF [Masoumi et al., 2016]	0.7290	-
SV-LSF_kpaca50 [Takahiko Furuya, 2015]	-	96.2
Geodesic Distances (LS) [Pickup et al., 2015]	0.7170	-
MDS-CM-BOF	0.7166	95.0
OrigM-n12-normA	0.7047	94.4
FOG+MRR	0.6958	91.8
ConTopo++ [Sfikas et al., 2011]	0.6950	94.7
BOGH	0.6469	86.7
LSF	0.6327	85.1

5.3 Timing analysis

The KLBO has the same computational complexity of the LBO. It is comprises the computation of the matrices A, K and W, which have complexities $O(|V|)$, $O(|V|)$ and $O(|V|^2)$, respectively, and the eigendecomposition of the generalized eigenproblem. However, the computation of the matrices are dominated by the eigendecomposition of the LBO. Using SVD, the worst case scenario for computing the eigenfunctions is $O(|V|^3)$. On the other hand, using **eigs** function on Matlab can accelerate this process. This is because the LBO is sparse and there are specific methods to solve eigendecomposition when we do not need to compute all the $|V|$ eigenfunctions. For example, in our case we only need to compute the first 300 eigenvalues and eigenfunctions.

We show in Table 5.13 detailed running times to compute KLBO. In the table, we show

Table 5.7: Retrieval performance comparison with state-of-the-art signatures on SHREC'15. We show best runs from the six groups that performed better in SHREC'15 (SV-LSF_kpaca50, HAPT_run1, SPH_SparseCoding_1024, CompactBoHHKS10D, FV-WKS, ED-BCF_NW) and other recent descriptors in the literature against our descriptor (MR-KLBO-FVIWKS). In bold are highlighted the best performances for each retrieval measure.

Descriptor	e-Measure	mAP [%]
SV-LSF_kpaca50 [Takahiko Furuya, 2015]	**0.8357**	**99.8**
KLBO-FVIWKS	0.8269	99.2
MR-FVIWKS	0.8211	97.9
HAPT_run1	0.8150	97.7
EUC-FVIWKS [Limberger and Wilson, 2015]	0.8102	96.9
SPH_SparseCoding_1024	0.8047	96.8
SGWC-BoF [Masoumi et al., 2016]	0.7470	-
CompactBoHHKS10D	0.7465	90.1
SRG [Mohamed and Hamza, 2016]	0.7390	-
FV-WKS	0.7242	87.5
EDBCF_NW	0.7076	85.0

Table 5.8: Retrieval performance comparison of the different spectral signatures computed with the **KLBO** combined with FV and SV applied to the **SHREC'14 Synthetic benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.7333	0.4677	0.6250	0.4332	0.7761	0.5506
FV-SIHKS	**0.8883**	**0.8158**	**0.9512**	**0.6887**	**0.9403**	**0.8622**
FV-WKS	0.6342	0.4656	0.6708	0.4622	0.7609	0.5504
FV-IWKS	0.6475	0.5176	0.7638	0.5214	0.7977	0.6009
SV-HKS	0.7267	0.4296	0.5865	0.4081	0.7436	0.4943
SV-SIHKS	0.8667	0.7232	0.8402	0.6034	0.8975	0.6972
SV-WKS	0.6742	0.4636	0.6462	0.4474	0.7598	0.6344
SV-IWKS	0.6892	0.5023	0.7501	0.5056	0.7900	0.5347

average times to compute one model from each database. At the last two columns it is shown the total time to compute each of the benchmarks, using either FV or SV. The average computation times of Fisher Vector approach is considerably lower because we use VLfeat implementation [Vedaldi and Fulkerson, 2008], while Super Vector is completely implemented on Matlab. Complexities of FV and SV are similar, thus SV would have similar computation

Table 5.9: Retrieval performance comparison of the different spectral signatures computed with the **KLBO** combined with FV and SV applied to the **SHREC'14 Real benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent the best retrieval performance for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.0381	0.0359	0.0682	0.0461	0.2888	0.136
FV-SIHKS	0.1019	0.1130	**0.2280**	**0.1543**	0.3806	**0.222**
FV-WKS	0.0681	0.0616	0.1289	0.0916	0.3284	0.165
FV-IWKS	0.0738	0.0784	0.1586	0.1068	0.3430	0.192
SV-HKS	0.0625	0.0572	0.0961	0.0605	0.3060	0.149
SV-SIHKS	**0.1125**	**0.1163**	0.2242	0.1509	**0.3830**	0.221
SV-WKS	0.0725	0.0781	0.1596	0.1010	0.3410	0.173
SV-IWKS	0.0888	0.0894	0.1765	0.1154	0.3567	0.195

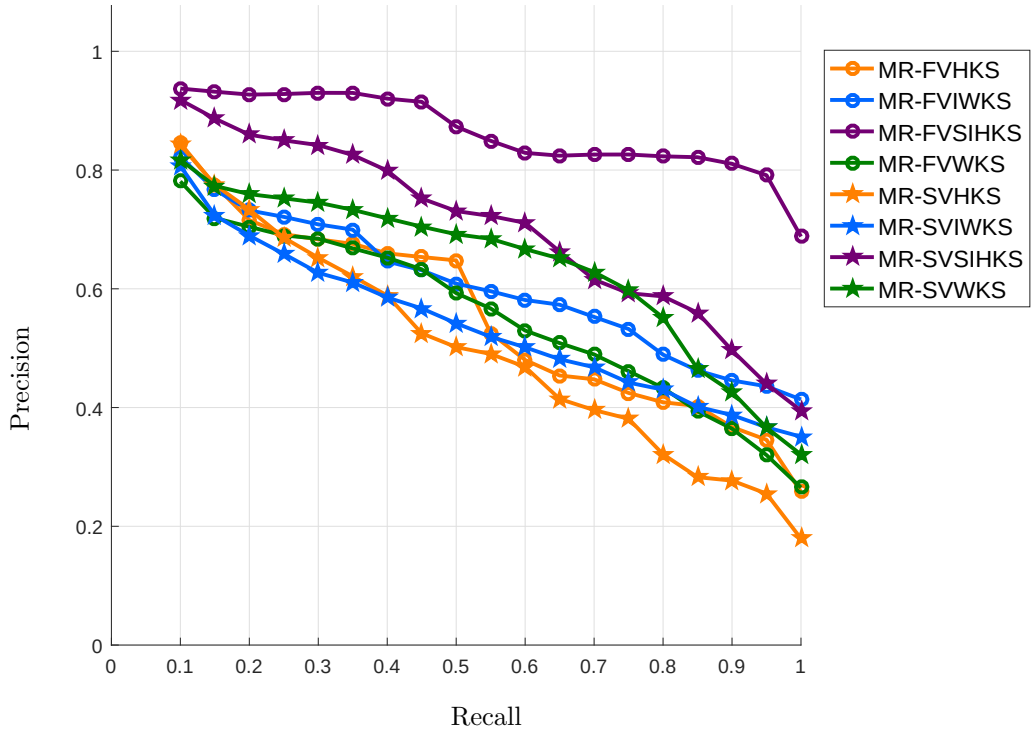


Figure 5.8: Precision and Recall plot of different spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'14 Synthetic benchmark**. The shape spectra is computed by the **KLBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor.

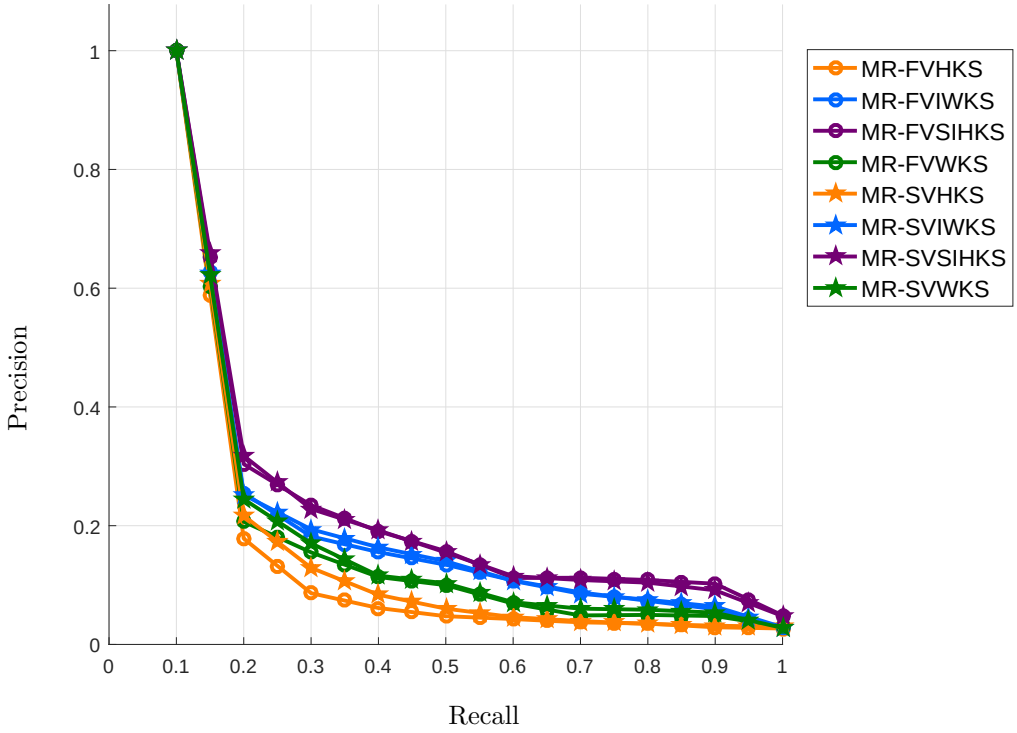


Figure 5.9: Precision and Recall plot of different spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **SHREC'14 Real benchmark**. The shape spectra is computed by the **KLBO**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor.

Table 5.10: Retrieval performance comparison of state-of-the-art descriptors for the **SHREC'14 Synthetic benchmark** [Pickup et al., 2014b]. Results were taken from the respective papers.

Method	E	DCG	mAP
supDLtrainS [Litman et al., 2014]	**0.721**	**0.975**	**0.954**
NPSR [Gasparetto et al., 2015]	-	-	0.950
Spectral Geometry [Li, 2013]	0.706	0.971	-
KLBO-FVSIHKS	0.689	0.940	0.862
Surface Area [Pickup et al., 2014b]	0.691	0.901	-
HAPT [Giachetti and Lovato, 2012]	0.655	0.936	0.817

time if it was implemented in like manner.

Table 5.11: Retrieval performance comparison of state-of-the-art descriptors for the **SHREC'14 Real benchmark** [Pickup et al., 2014b]. Results were taken from the respective papers.

Method	E	DCG	mAP
supDLtrainR [Litman et al., 2014]	**0.432**	**0.891**	**0.791**
NPSR [Gasparetto et al., 2015]	-	-	0.790
R-BiHDM-s [Ye et al., 2013]	0.387	0.781	0.640
HAPT [Giachetti and Lovato, 2012]	0.355	0.795	0.637
Surface Area [Pickup et al., 2014b]	0.326	0.571	-
KLBO-FVSIHKS	0.119	0.353	0.203

Table 5.12: Retrieval performance comparison of state-of-the-art descriptors for the **SHREC'17 Relief Patterns benchmark** [Biasotti et al., 2017]. Results were taken from the respective paper. There can be found more details about the other methods.

Method	E	DCG	mAP
KLBO-FVIWKS (Limberger)	**0.332**	**0.759**	**0.339**
LBPI (Tatsuma)	0.232	0.697	0.283
CMC-2 (Velasco-Forero)	0.261	0.686	0.271
IDAH-1 (Sun)	0.145	0.578	0.174

Table 5.13: Average computation times (in seconds) for computing one mesh signature for an average-sized model from each dataset. KLBO stands for computation of curvatures, eigenvectors and eigenvalues. HKS, SIHKS, WKS and IWKS stand for time to compute respective signatures. SHIFT stands for the shifting described in [Limberger and Wilson, 2015]. FV and SV stand for computation times of Fisher Vector and Super Vector. EMR represents the time to perform retrieval of one model. Total times to compute signatures and retrieve all models using either FV or SV are shown in Total-FV and Total-SV columns.

Benchmark	KBLO	HKS	SIHKS	WKS	IWKS	SHIFT	FV	SV	EMR	**Total-FV**	**Total-SV**
SHREC'10	24.74	0.10	5.39	0.05	0.06	0.04	6.03	28.89	0.07	**6,390**	**7,533**
SHREC'11	12.73	0.06	3.62	0.04	0.06	0.03	3.93	16.97	0.27	**10,668**	**12,625**
SHREC'15	15.24	0.07	3.77	0.04	0.05	0.03	5.02	34.91	0.58	**25,239**	**34,205**

5.4 Summary

In this chapter, we have presented a more fundamental approach to deal with the problem of non-rigid shape retrieval for meshes. We begun by defining 3D models in physical terms using classical field theory. With this approach, we are able to weight the physical field by a curvature term that reduces the effect of object articulation on the shape descriptors.

Therefore, in this case the shape spectra is naturally less variant to non-rigid deformations. We have shown through several experiments that the KLBO improves the performance over the classic LBO of spectral signatures in 3D non-rigid shape retrieval. We also showed that the KLBO can be used with the Fisher Vector and the Super Vector, state-of-the-art encoding schemes, which also improve retrieval performance. Furthermore, we verified that the KLBO and the LBO present similar computation time.

Chapter 6

Non-Rigid Point-Cloud Shape Retrieval Benchmark

In this chapter, we detail the creation of a point-cloud benchmark for non-rigid shape retrieval. The non-rigid shape retrieval area is concerned with retrieving models that are somehow deformed by its joints. The importance of shape retrieval is evidenced by the 11 years of the SHREC. In this thesis, we have been extensively addressing this problem for 3D meshes, however, 3D shapes can be also represented by three-dimensional point clouds, which are the immediate result of 3D scans of real 3D objects. Although some methods available in the literature use the point set data to create their shape signatures, we have not seen these methods being used directly to address the non-rigid point-cloud shape-retrieval problem since there are no specific point-cloud datasets available for this purpose. We have included our own method in this evaluation (the Mesh-Free Laplace-Beltami Operator), however, it will be fully explained only in the next chapter.

This chapter was published in [Limberger et al., 2017] as a SHREC'17 benchmark track, therefore, there were other researchers that submitted their algorithm results to be compared on this data. The rest of this chapter is organized in the following way. We introduce the subject of point-cloud shape retrieval and motivate the reader in Section 6.1. Then, we discuss peculiarities about the dataset and how it was constructed in Section 6.2. In Section 6.3, we discuss our evaluation methodology and in Section 6.4 we list the participants of the

SHREC'17 track: PRoNTo benchmark. In Section 6.5, we show many experiments on the dataset and analyze carefully poses and classes. Finally, Section 6.6 summarizes the chapter and points possible research directions.

6.1 Introduction

With the rapid development of virtual reality (VR) and augmented reality (AR), especially in gaming, 3D data has become part of our everyday lives. Since the creation of 3D models is essential to these applications, we have been experiencing a large growth in the number of 3D models available on the Internet in the past years. The problem now has been organizing and retrieving these models from databases. Researchers from all over the world are trying to create shape descriptors in a way to organize this huge amount of models, making use of many mathematical tools to create discriminative and efficient signatures to describe 3D shapes.

There are two distinct areas which concern shape retrieval: The first, non-rigid shape retrieval, which deals with the problem of articulations of the same shape [Lian et al., 2010, Lian et al., 2011, Lian et al., 2015], and second, comprehensive shape retrieval [Bronstein et al., 2010b, Li et al., 2014c, Savva et al., 2016], which deals with any type of deformation, for example, scaling, stretching and even differences in topology. While comprehensive shape retrieval is more general, non-rigid shape retrieval is as important when it is necessary to carefully classify similar objects that are in distinct classes [Pickup et al., 2016b].

Three-dimensional point clouds are the immediate result of scans of 3D objects. Although there are efficient methods to create meshes from point clouds, sometimes this task can be complex, particularly when point-cloud data present missing parts or noisy surfaces, for example, fur or hair. In this chapter, we are once more interested in the non-rigid shape retrieval task. This way, we created a non-rigid point-cloud shape retrieval benchmark (PRoNTo: Point-Cloud Shape Retrieval of Non-Rigid Toys), which was produced given the necessity of testing non-rigid shape signatures computed directly from unorganized point clouds, *i.e.*, without any connectivity information. This is the first benchmark ever created to test, specifically, the performance of non-rigid point-cloud models.

This benchmark is important given the need to compare 3D non-rigid shapes based directly on a rough 3D scan of the object, which is a more difficult task than comparing signatures computed from well-formed 3D meshes. Also because real-time interaction with 3D scanners and 3D objects are expected to be part of our everyday lives in the near future as suggested by the growth of VR and AR and the new 3D platform from Microsoft, which will include many 3D features in its products, and, consequentially, will take the third dimension to everyone's houses.

Scanning problems can occur when the laser hits first one part of the model leaving another part of the model unseen (in the same direction pointed by the scan head). Other problems can also be caused by object's specular materials and inner-reflections of the object and can lead to the insertion of outliers in the model. Objects with missing parts are intended to test signatures robustness against scanning problems since precise 3D-object scanning is a very time-consuming task. In the future, people will not spend hours capturing 3D objects. Object recognition and description methods need to be robust to an uncomplicated scan, and be able to classify objects created from a rough model examination.

6.2 Dataset

In this section we explain the dataset characteristics, information about models and the reasons we chose each one of the poses. Our dataset consists of 100 models that are derived from 10 different real objects. Each real object was scanned in 10 distinct poses by articulating them around their joints. The real objects can be seen in Figure 6.1 and the different poses can be seen in Figure 6.3, using the object *Monster* as example. The point clouds acquired by these scans suffer from common scanning problems like holes and missing parts resulted from self-occlusions of the shapes. Objects were scanned using the Head & Face Color 3D Scanner of Cyberware. This scanner makes a 360 degrees scan around the object estimating x, y, and z coordinates of a vertical patch. The scanning process captures an array of digitized points and also the respective RGB colours.

In the end, after all models being scanned, the point clouds were randomly rotated across the three axis X, Y and Z to test descriptors robustness against rigid transformations. Each

Figure 6.1: Different toys used to create the PRoNTo dataset.

point cloud was also resampled to, in average, 4K points each using Poisson-disk sampling (PDS) algorithm [Corsini et al., 2012]. The resampling step is important because it sets a standard size to the models and it makes the dataset less susceptible to be classified by trivial methods.

Although 3D points have being scanned with colours, this is not the benchmark purpose, thus colours were removed and only the point positions are made available. The file format was chosen as the ASCII Object File Format (.off), which, in this case, contains only vertex information. Figure 6.2 shows the entire framework to create the point-cloud models. First, we scan the model and get the rough object's scan (a). Then, we manually clean the supports and some outliers left from the scanning process (b). We perform an undersampling of the model using PDS algorithm (c). Finally, we remove colours (d). The orientation of the model is not relevant since we perform an arbitrary rotation of the model at last.

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

Figure 6.2: Framework for creating point-cloud models using model *Dog* as an example. In (a) we show the rough object's scan. In (b), it is shown the model cleaned from outliers and supporting wires. In (c), we show the undersampled version of (b) and, finally, in (d) we show the uncoloured model used in the benchmark with approximately 4K points.

6.2.1 Poses

Different poses have been captured to test different characteristics of descriptor's potential. They are: *Celebrating, Dance move, Default, Hands front, Open, Seated, Seated hands front, Straight, Walking* and *Tilted*. These can be seen in Figure 6.3 . In order to scan these poses, we implanted wires inside the objects. To position each object in the scanning area we also hang the object by a wire, which can be seen in Figure 6.4. Therefore, after all scans being made we had to manually remove the wire supporting the object from the scans. Each 3D point cloud is the result of one 360-degree sweep of the scanner around each toy.

Some poses feature a small deformation, for example: *Default, Open* and *Celebrating*; *Seated*, *Seated hands front* and *Hands front*. These are exactly the same poses across all classes, *i.e.*, it means the deformation of the models are the same. These poses hamper the correct model classification since they appear to be more similar than the intrinsic object's shape itself. Others were created to test the object's asymmetry: *Dance move* and *Walking*. These poses can vary depending on object's class, *i.e.*, the feet and arms can be either to the front or to the back depending on each class. *Tilted* has the intention of testing a tilted scan of the model, *i.e.*, different parts appear in the object scan when the object is set up in a different slope, since the scanner is not able to capture surfaces that are in the same orientation of the laser or occluded from the scan head. Finally, the pose *Straight* has the

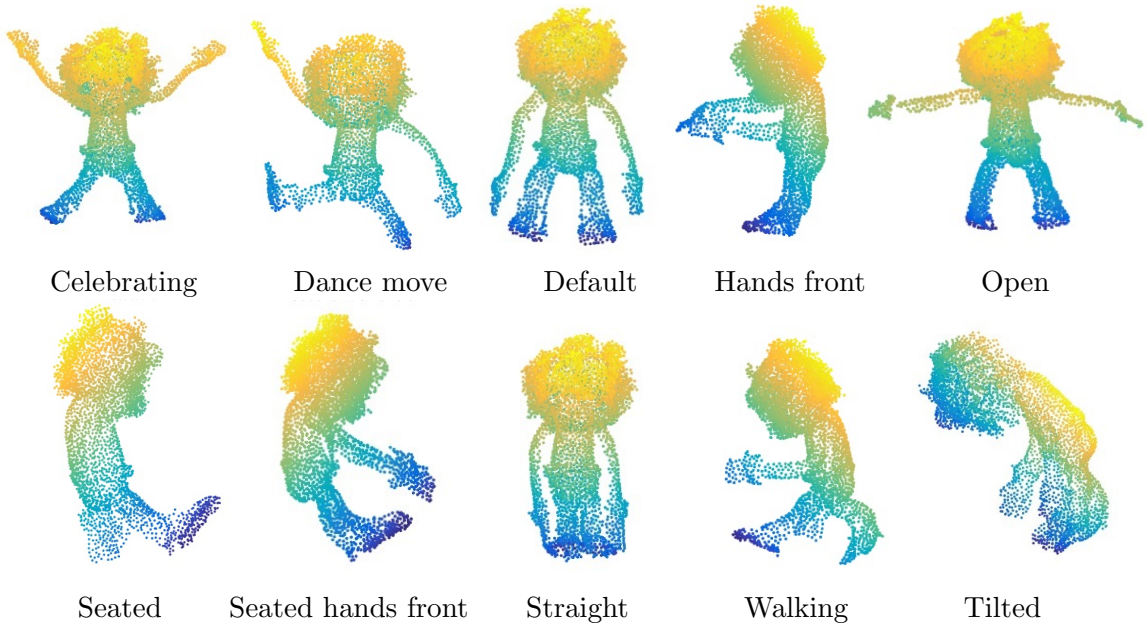


Figure 6.3: Different poses captured of the objects, showing model *Monster* as an example. Point clouds were coloured by Y and Z coordinates.

intention of testing major topology changes, since feet and arm points are very close to the body, leaving an uncertainty about their meaning in the scan as there is no connectivity information.

While different poses hamper the recognition of shape classes, some models are also very similar to each other like *Dog* and *Fox*; or *Teddy*, *Monkey* and *Sheep* (see Figure 6.1). Some are also quite distinct like *Rat*, *Tiger*, *Einstein* and *Monster*. Typically, these should be easier to retrieve since the discrepancies are greater.

It is also our intention to investigate what are the poses which are easier to retrieve and which are the more difficult to identify. This will give us an understanding of what deformations are the most difficult to detect and therefore we will be able point future research topics in shape retrieval to be investigated.

6.3 Evaluation

In this section, we detail our evaluation methodology used to rank algorithm performances. The result of a shape retrieval problem is a dissimilarity matrix which gives the difference between every model in the database. A dissimilarity matrix is a matrix of size $N \times N$,

(a) (b)

Figure 6.4: Object scanning. In (a) is shown the model *Dog* being scanned. See that the scanner captures distances over the red line and it captures all the object by sweeping the scan head over the 360 degrees. In (b) is shown the scanner of Cyberware doing the sweep and capturing the model *Monster*.

where N is the number of models of the dataset, where the position (i, j) in the matrix gives the difference between models i and j. Similarly to past SHREC benchmark contests, standard evaluation measures were computed to test the retrieval accuracy of the algorithms: Precision-and-Recall (PR) curve , mean Average Precision (mAP), E-Measure (E), Discounted Cumulative Gain (DCG), Nearest Neighbor (NN), First-Tier (FT) and Second-Tier (ST). These measures are described in detail in Section 4.4.2.

To summarize, mAP is given by the area below the PR curve. E-measure is given by the precision and recall of the first 32 retrieved models, even when classes have less than 32 models. NN gives a percentage of correct matches within all closest matches for each model. FT is given by the percentage of correct matches that appear within the top K matches, where $K = |C| - 1$ and $|C|$ is the number of models of each class. For ST, $K = 2 * (|C| - 1)$. DCG is a logarithmic statistic which weights correct matches based on their position on the retrieval list. More details of each evaluation measure can be find in Section 4.4.2.

6.4 List of the participant methods

In this section, we include the list of the participant methods that have been submitted to the PRoNTo benchmark in the SHREC'17. In total, 8 groups have participated in the contest with 31 valid submissions. The 31 submissions divided per group were:

1. *MFLO-FV-IWKS*, *MFLO-SV-IWKS*, *PCDL-FV-IWKS*, *PCDL-SV-IWKS*, *GL-FV-IWKS* and *GL-SV-IWKS* submitted by Frederico A. Limberger and Richard C. Wilson.

2. *BoW-RoPS-1*, *BoW-RoPS-2*, *BoW-RoPS-DMF-3*, *BoW-RoPS-DMF-4*, *BoW-RoPS-DMF-5* and *BoW-RoPS-DMF-6* submitted by Minh-Triet Tran, Viet-Khoi Pham, Hai-Dang Nguyen and Vinh-Tiep Nguyen.

3. *POHAPT* and *BPHAPT* submitted by Andrea Giachetti.

4. *CDSPF* submitted by Atsushi Tatsuma and Masaki Aono.

5. *SQFD(HKS)*, *SQFD(WKS)*, *SQFD(SIHKS)*, *SQFD(WKS-SIHKS)* and *SQFD(HKS-WKS-SIHKS)* submitted by Benjamin Bustos and Ivan Sipiran.

6. *SnapNet* submitted by Bertrand Le Saux, Nicolas Audebert and Alexandre Boulch.

7. *AlphaVol1*, *AlphaVol2*, *AlphaVol3* and *AlphaVol4* submitted by Santiago Velasco-Forero.

8. *m3DSH-1*, *m3DSH-2*, *m3DSH-3*, *m3DSH-4*, *m3DSH-5* and *m3DSH-6* submitted by Bo Li, Yijuan Lu and Afzal Godil.

To a more detailed explanation of the methods that participated in the PRoNTo contest we refer the reader to the SHREC'17 - PRoNTo paper [Limberger et al., 2017] or to the website [Limberger and Wilson, 2017] where you will find all the relevant information related to the track and the dataset.

6.5 Experiments

In this section, we perform an exhaustive evaluation of the participant's methods on the SHREC'17 - PRoNTo dataset. For this, we show evaluation performances highlighted by

the previous section. As mentioned previously, 8 groups participated with 31 dissimilarity matrices. The retrieval scores computed from these matrices represent the overall retrieval performance of each method, *i.e.*, how well they perform on retrieving all models from the same class when querying every model in the database.

It is worth pointing that these are preliminarily results since we do not have full knowledge of the participant algorithms. This means it would be cumbersome to implement and validate them. These algorithms can be considered undocumented until the time that this document was written. More information about each algorithm can be find in the benchmark paper [Limberger et al., 2017], however, for more precise information about them we recommend the reader to contact the algorithm's authors.

Table 6.1 shows the method performances of all 31 runs and Figure 6.5 shows the Precision-and-Recall curves of all participant methods. It is worth pointing out that some methods perform quite well on this database. By analysing particularly DCG, which is a very good and stable measure for evaluating shape retrieval methods [Lian et al., 2015], we can see that three methods have DCG greater than 0.900 (BoW-RoPS-DMF-3, BPHAPT and MFLO-FV-IWKS). Surprisingly, Tran's methods have DCG values greater than 0.990. The method clearly outperforms all other methods in the contest as evidenced by the Precision-and-Recall plot in Figure 6.6. BoW-RoPS can definitely capture the differences between classes and it seems robust to most of the non-rigid deformations presented in this database. Curiously, Tran's method uses asymmetric distance computation between descriptors, which leads to distances between models i and j being different from the distances between models j and i. This is clearly evidenced by their dissimilarity matrices.

Considering all groups that have participated in this contest, half of them (4) computes local features (MFLO-FV-IWKS, SQFD(WKS), CDSPF and BoW-RoPS-DMF-3) and the other half (4) computes global features (BPHAPT, SnapNet, m3DSH-3 and AlphaVol1). Our first guess was that local features would be more popular to represent non-rigid shapes, as evidenced by [Lian et al., 2015]. Our guess was based on the fact that ideally local features should be more similar than global features because same-class shapes were captured originally from the same 3D object, and locally they should be more similar than globally. For example, while a shape can be in a totally different pose, locally only joint regions are

Table 6.1: Six standard quantitative evaluation measures of all 31 runs computed for the PRoNTo dataset.

Participant	**Method**	**NN**	**FT**	**ST**	**E**	**DCG**	**mAP**
Boulch	SnapNet	0.8800	0.6633	0.8011	0.3985	0.8663	0.771
Giachetti	POHAPT	0.9400	0.8300	0.9144	0.4156	0.9419	0.900
	BPHAPT	0.9800	0.9111	0.9544	0.4273	0.9743	0.953
Li	m3DSH-1	0.4000	0.1656	0.2778	0.1824	0.4802	0.297
	m3DSH-2	0.4400	0.1867	0.2856	0.1932	0.4997	0.313
	m3DSH-3	0.4400	0.1767	0.2878	0.1917	0.5039	0.314
	m3DSH-4	0.4000	0.1511	0.2511	0.1712	0.4659	0.286
	m3DSH-5	0.4200	0.1722	0.2767	0.1815	0.4930	0.304
	m3DSH-6	0.4100	0.1700	0.2678	0.1712	0.4848	0.300
Limberger	GL-FV-IWKS	0.8200	0.5756	0.7244	0.3595	0.8046	0.702
	GL-SV-IWKS	0.7000	0.5267	0.6678	0.3327	0.7562	0.651
	MFLO-FV-IWKS	0.8900	0.7911	0.8589	0.4024	0.9038	0.858
	MFLO-SV-IWKS	0.9000	0.7100	0.7933	0.3702	0.8765	0.800
	PCDL-FV-IWKS	0.8200	0.6656	0.7978	0.3976	0.8447	0.764
	PCDL-SV-IWKS	0.8900	0.6656	0.7911	0.3732	0.8613	0.781
Sipiran	SQFD(HKS)	0.2900	0.2244	0.3322	0.2176	0.5226	0.344
	SQFD(WKS)	0.5400	0.3111	0.4467	0.2507	0.6032	0.427
	SQFD(SIHKS)	0.2900	0.2533	0.4133	0.2590	0.5441	0.377
	SQFD(WKS-SIHKS)	0.5000	0.3100	0.4500	0.2634	0.6000	0.425
	SQFD(HKS-WKS-SIHKS)	0.3900	0.2844	0.4389	0.2624	0.5722	0.403
Tatsuma	CDSPF	0.9200	0.6744	0.8156	0.4005	0.8851	0.794
Tran	BoW-RoPS-1	1.0000	0.9744	0.9967	0.4390	0.9979	0.995
	BoW-RoPS-2	1.0000	0.9778	0.9933	0.4385	0.9973	0.993
	BoW-RoPS-DMF-3	1.0000	0.9778	0.9978	0.4390	0.9979	0.995
	BoW-RoPS-DMF-4	1.0000	0.9778	0.9978	0.4390	0.9979	0.995
	BoW-RoPS-DMF-5	1.0000	0.9733	0.9978	0.4390	0.9979	0.995
	BoW-RoPS-DMF-6	1.0000	0.9733	0.9978	0.4390	0.9979	0.995
Velasco	AlphaVol1	0.7900	0.5878	0.7578	0.3980	0.8145	0.707
	AlphaVol2	0.7800	0.5122	0.6844	0.3751	0.7673	0.643
	AlphaVol3	0.7700	0.4567	0.6467	0.3629	0.7364	0.600
	AlphaVol4	0.7000	0.4356	0.6111	0.3454	0.7148	0.571

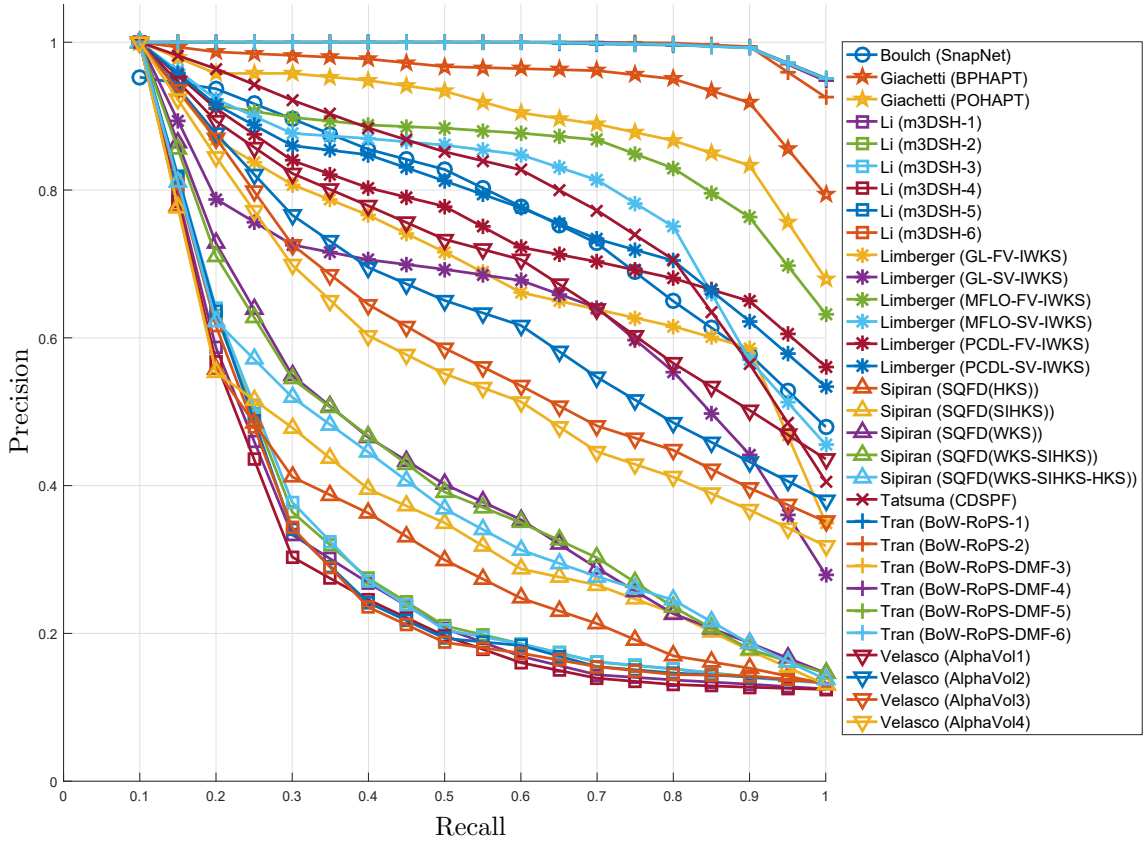


Figure 6.5: Precision-and-Recall curves of all methods that participated in the PRoNTo contest evaluated for the PRoNTo dataset.

deformed. However, we also need to consider local noise in the formula, which does not affect global methods in the same level.

Tran's method is in the first place and uses local features. Clearly, in the second place is Giachetti's method, which is based on global features from 3D meshes created from the point clouds. In total, 3 groups use meshing procedures before computing the descriptors (BPHAPT, SQFD(WKS) and SnapNet). Interestingly, two methods use quadratic form distance to compute dissimilarities between descriptors, one from a global descriptor (m3DSH-3) and other from a local descriptor SQFD(WKS).

Even though no training set was available in this track, Boulch's method uses a Convolutional Neural Network by employing an unsupervised learning architecture where every model is considered belonging to a different class. On the other hand, more methods also adopt unsupervised learning algorithms to create dictionaries using the Bag of Words encoding paradigm (BoW-RoPS-DMF-3 and MFLO-FV-IWKS) being these ranked first and

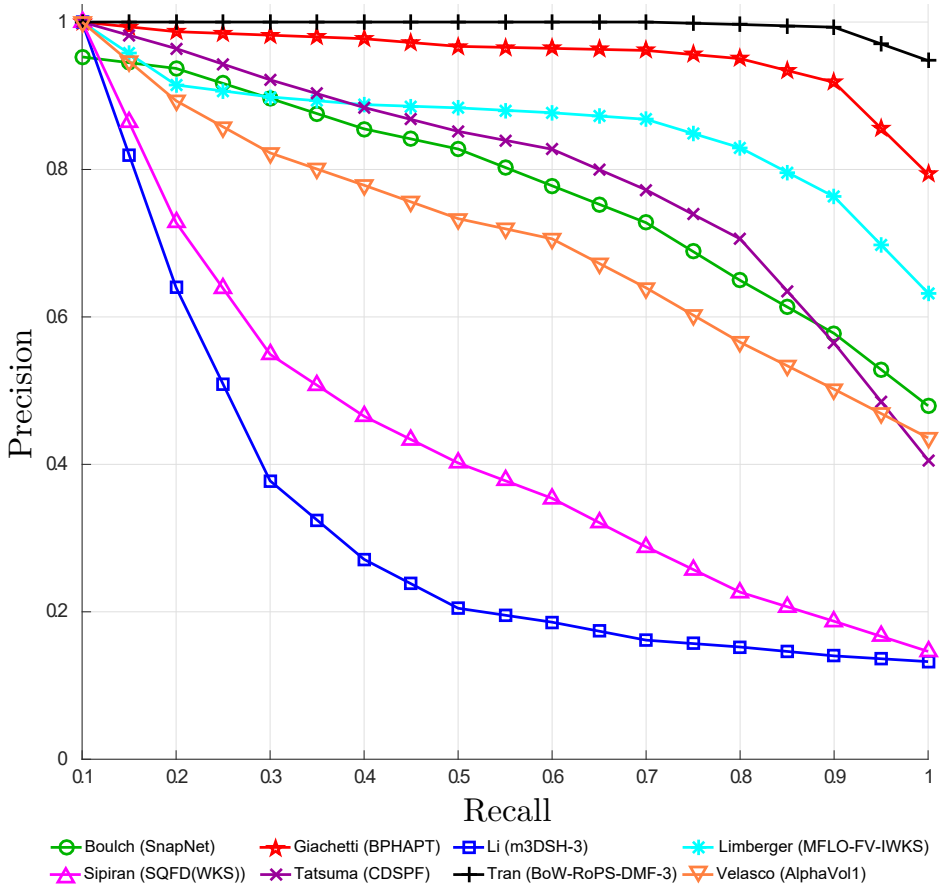


Figure 6.6: Precision-and-recall curves of the best runs of each group that participated in the PRoNTo contest evaluated for the PRoNTo dataset.

third on this contest, respectively, and showing that the BoW model is a good way of representing local features. Furthermore, two other methods use histogram encoding (vector quantization) to create a unique descriptor for each point cloud (BPHAPT and CDSPF).

We also observed a couple of new other ideas applied to PRoNTo dataset. For instance, Velasco uses alpha-shapes to represent point clouds; by varying the alpha-shape radius he compares models given their alpha-shape volume curve. Limberger's method uses a new formulation to compute the Laplace-Beltrami operator of point clouds, which leads to better results than the standard Graph Laplacian. Tatsuma computes additional statistics of point features in addition to the geometric feature proposed by [Wahl et al., 2003]. Two groups use matrix-fusion methods with different weights to improve the performance of their methods (Tran and Sipiran), however, these methods did not show a substantial improvement from the performance of the original descriptors.

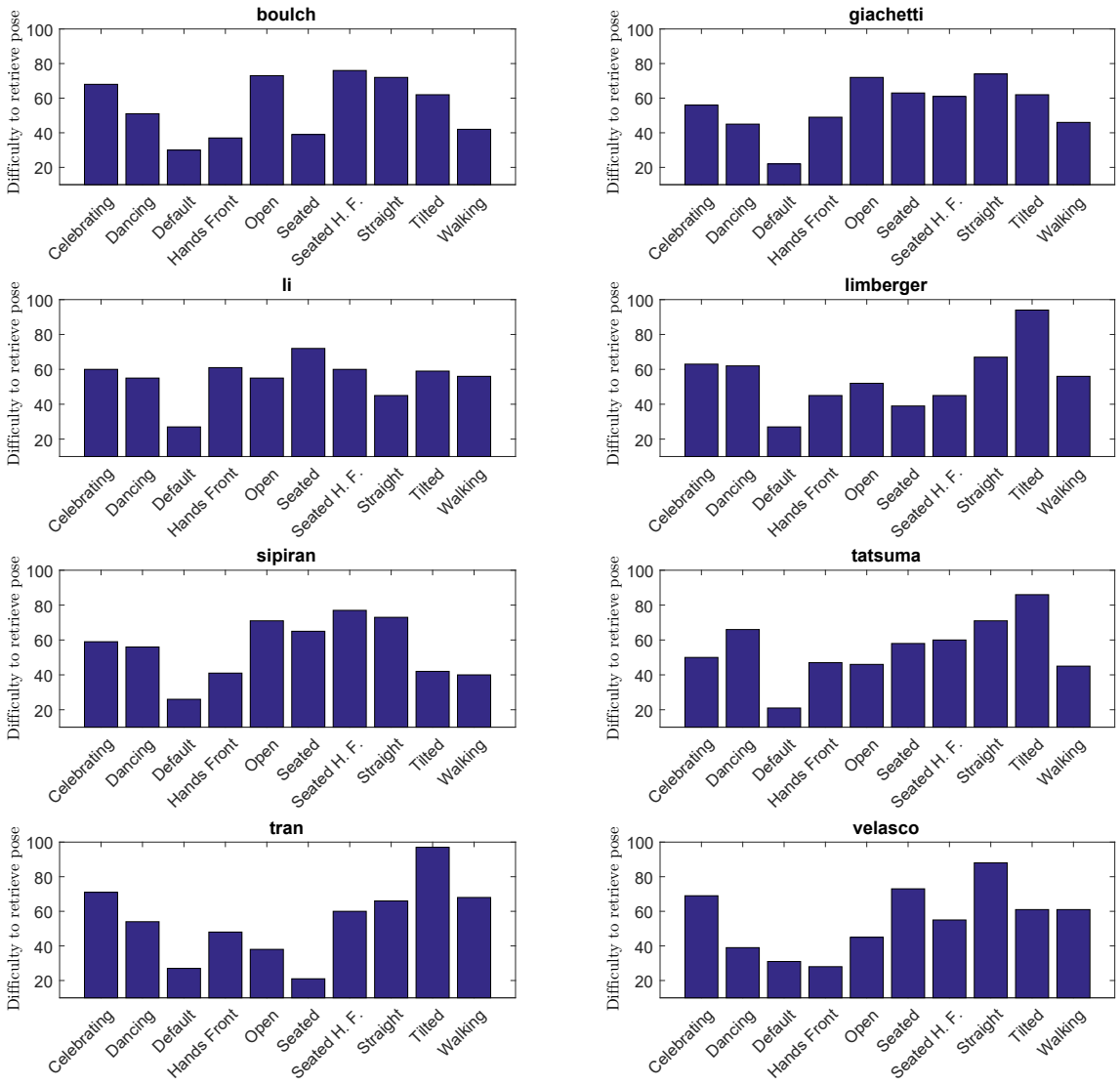


Figure 6.7: Histograms of rankings of each pose of the best methods submitted by each group that participated in the PRoNTo contest. The higher the histogram bins the further is the pose in the retrieving page, which means that, for example, for Tran's method the *Tilted* pose was the most difficult to retrieve.

Regarding spectral signatures, which is the focus of this thesis, we can observe that the MFLO submitted by us presents an improvement over the GL and the PCDL, which are methods to compute the Laplace-Beltrami operator of point clouds. Sipiran's methods also use spectral descriptors to represent local features on the shapes, however, they have used meshing procedures to compute the LBO. Although Sipirian claims that it is necessary to reconstruct the surface of the shape to guarantee the proper computation of the LBO, we have shown that the point cloud Laplace operators perform better on these experiments.

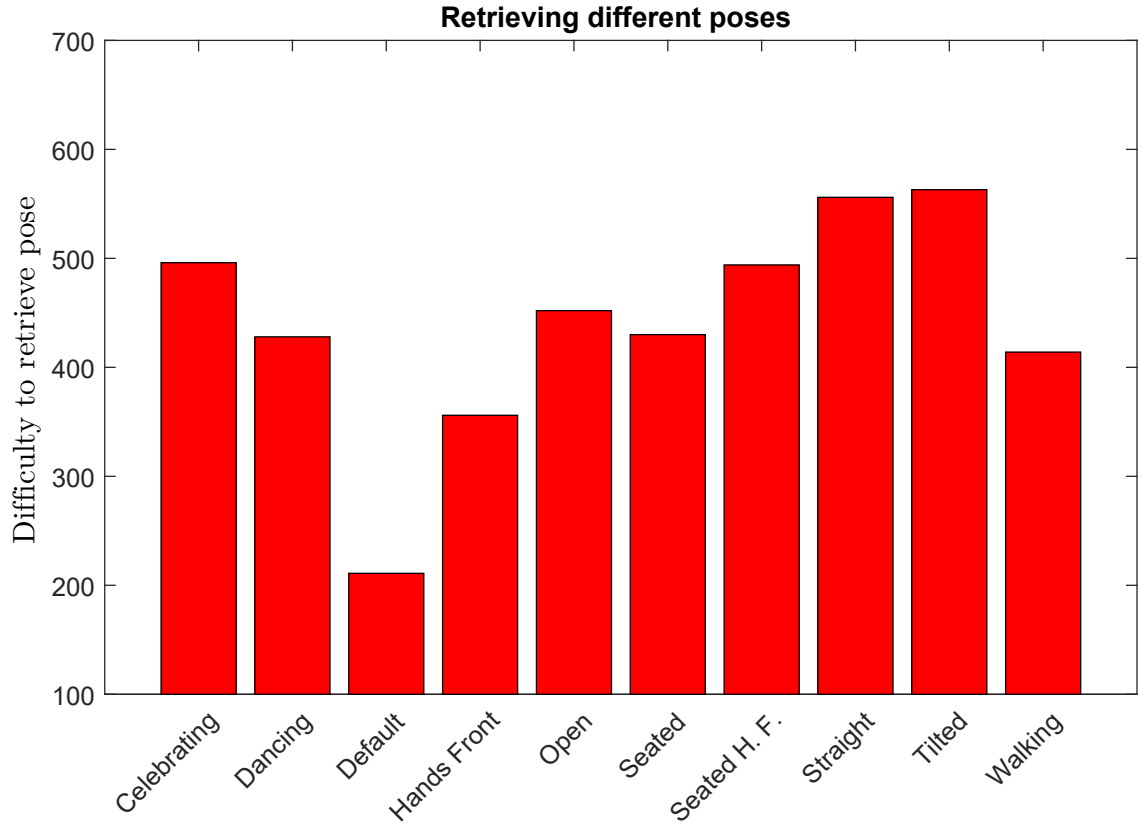


Figure 6.8: Histograms of rankings of each pose when we accumulate all the histograms from Figure 6.7. The higher is the histogram bin the more difficult is to retrieve that pose in general for all participant methods.

We also analyze the performance of each method in respect to the poses, *i.e.*, which poses are the easiest and hardest to retrieve. For this, we took the sum of the differences from one pose to the other poses in the same class and sorted them from in ascending order to analyze the pose ranking. Then, we gave a score from 1, to the closest pose, to 10, to the furthest pose. Doing this for all classes, we accumulate the scores on the entire dataset. Figure 6.7 shows the total numbers for each participant method. The higher the histogram bin the more difficult is to retrieve that pose, since it usually appears further in the retrieving page.

By analyzing the poses from all methods, it is possible to say that clearly the easiest pose to retrieve was the *Default* pose. *Hands Front* and *Seated* poses also had low scores for some methods (Boulch, Tran and Velasco). Furthermore, we accumulate all the poses from all methods to better visualize which poses are the easiest and the hardest to retrieve. The result of this experiment can be seen in Figure 6.8. As already mentioned, the *Default* pose

was the easiest pose to retrieve by far. The most difficult poses to retrieve were *Tilted* and *Straight*, which is evidenced by the majority of histograms from Figure 6.7. Both poses were expected to be difficult to retrieve since they have the most different topologies compared to the other poses. It is important to mention that these histograms (from Figure 6.7) do not say anything about how good these methods perform, since, in total, all poses receive the same number of scores ($(1 + 2 + 3 + \ldots + 10) \times 10 = 550$).

In addition to poses, we looked into classes. Figure 6.9 shows the Precision-and-Recall plots of the participant methods separated by classes. Some methods are able to retrieve some classes perfectly, however, there are clearly some classes that are more difficult to retrieve than others. In Figure 6.10, we show the errors of all methods for each class, thus we can rank the easiest and hardest classes to retrieve. As mentioned before (Section 6.2.1), *Sheep*, *Teddy* and *Monkey* classes have quite similar models. The results reflect this (ranked in 1st, 3rd and 5th) and show that these classes have larger errors than other classes. Furthermore, *Dog* and *Fox* also were mentioned as similar classes. Thus we can infer that, probably *Fox* is more seen as *Dog* than *Dog* is seen as *Fox* since the class *Fox* presents larger errors. Differently from what we expected, *Rat* is a very difficult class to retrieve. We believe that it is because of its thin legs and arms, which sometimes do not appear in the scans, therefore changing the topology of the shape. It is interesting to note that, the analysis of the classes in Section 6.2.1 was made before any sort of experiment using the participant methods.

For more information about the PRoNTo dataset, please refer to the official website [Limberger and Wilson, 2017] where the database, the corresponding evaluation code and classification file are available for academic use.

6.6 Summary

In this chapter, we have shown how we created a non-rigid point cloud dataset which is derived from real toy objects. In the beginning, we discussed the importance of this data to future researchers. Then, we explained the dataset characteristics and we showed how the evaluation was carried out. Afterwards, we introduced each one of the 8 groups and their methods which competed on the PRoNTo contest. In the end, we presented quantitative

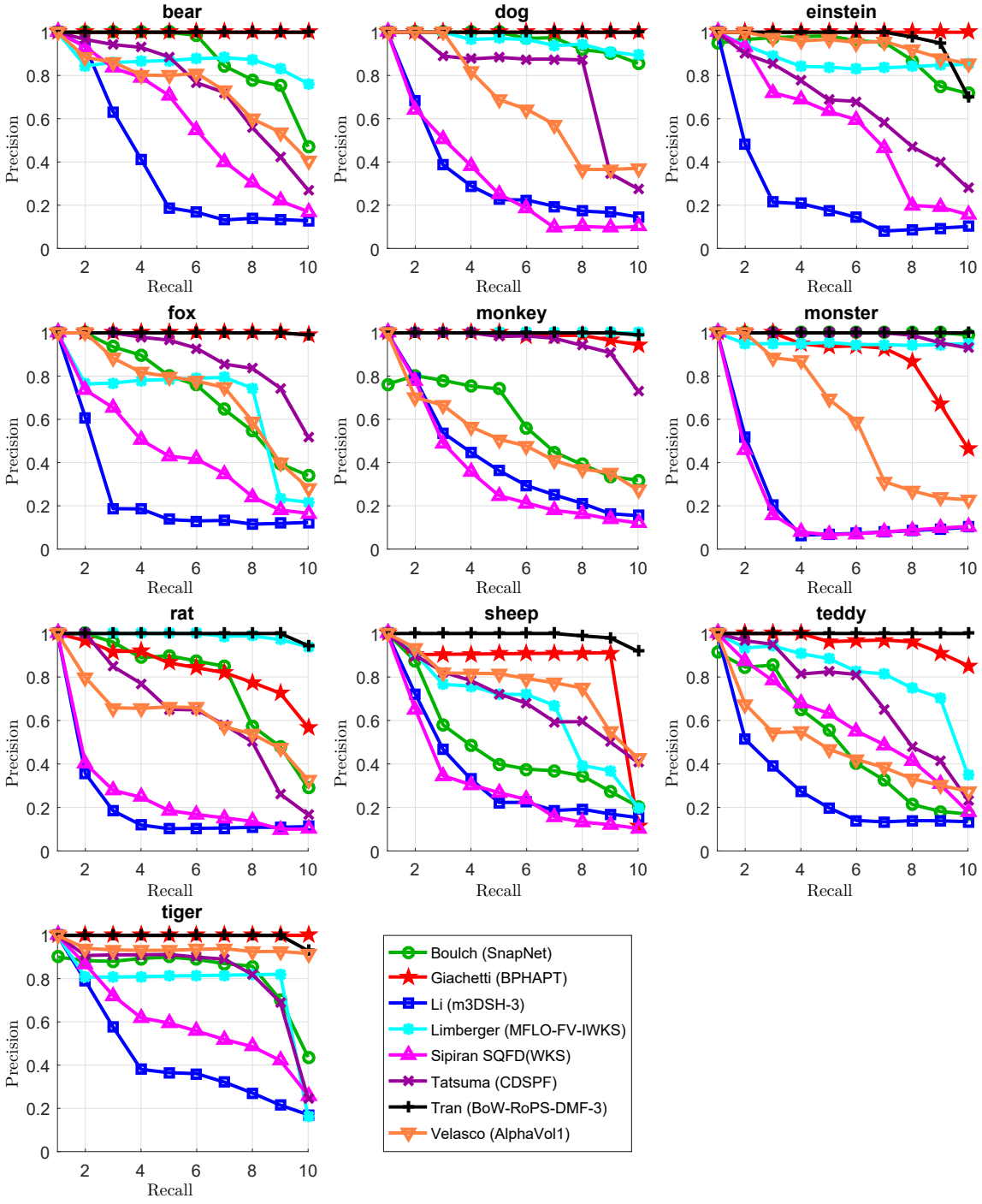


Figure 6.9: Precision-and-Recall curves per class of the best methods from each group.

measures of the 31 runs submitted by the contest participants and analysed their results.

The interest in non-rigid shape retrieval is overwhelming and evident by the previous SHREC tracks. The track created for this dataset was not different. It has attracted a large

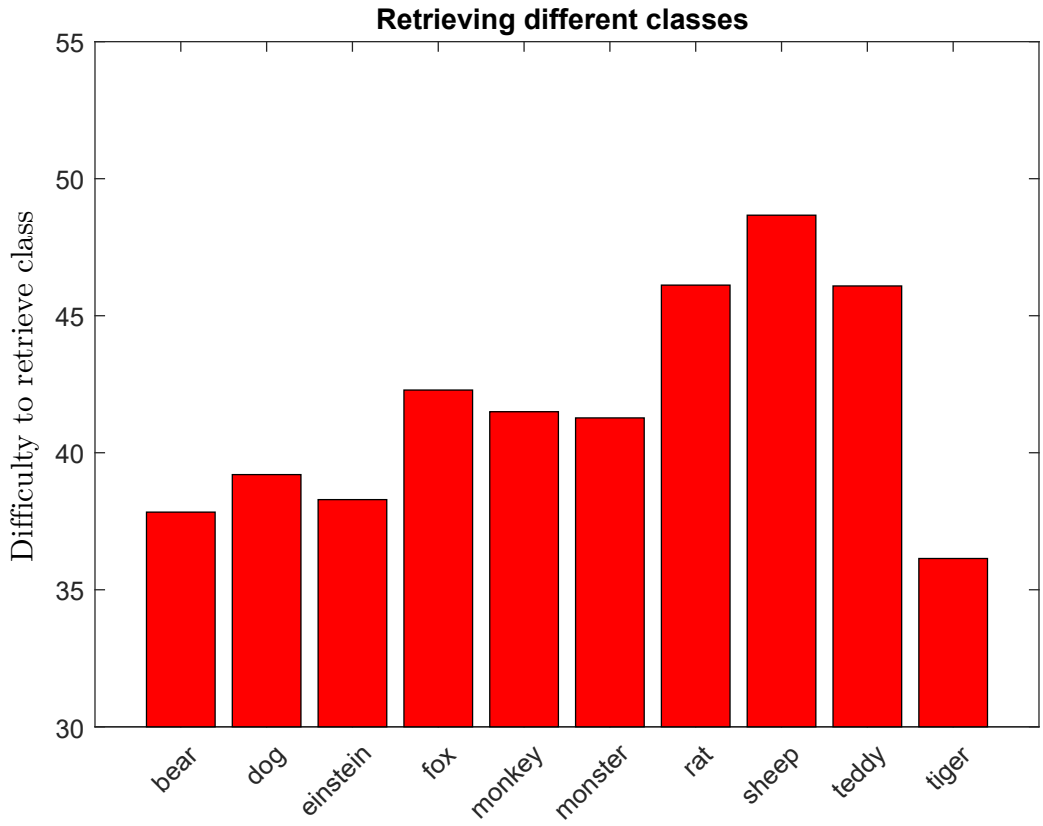


Figure 6.10: Histograms of errors from the Precision-and-Recall plots of all participant methods that competed in the PRoNTo contest. The higher is the histogram bin the more difficult is to retrieve that specific class for the participant methods in general.

number of participants (8 groups and 31 runs) given that it is the first time that a non-rigid point-cloud dataset is used in the SHREC contest. We believe that the organization of the track is just a beginning and it will encourage other researchers to further investigate this important research topic.

Several research directions in point-cloud shape retrieval can be pursuit from this work and are listed as follows: (1) Create a larger dataset which contains more types of objects (not only human shaped toys) to better evaluate shape signatures. (2) Create more discriminative local or global signatures for 3D point clouds. (3) Employ state-of-the-art Deep Learning techniques which do not depend on large training datasets. (4) Investigate signatures which are less variant to topology changes.

Chapter 7

Mesh-Free Point-Cloud Laplace Operator

This chapter describes the Mesh-Free Point-Cloud Laplace Operator (MFLO), which is a framework to compute the shape spectrum of point clouds, which do not have connectivity. It is able to approximate the Laplace-Beltrami operator without the need of creating a tangent space approximation around each point. For this, the point cloud must be sufficiently dense to estimate the underlying manifold around a local neighboring region. Our intention with this operator is to create a faster and more robust method to compute the spectrum of the Laplace-Beltrami operator on the underlying manifold, which is described discretely by a point cloud. With the shape spectrum, we can create spectral signatures for shape retrieval without the need of creating a triangle mesh.

The MFLO has participated in a shape retrieval contest in the *SHREC'17 track: Point-Cloud Shape Retrieval of Non-Rigid Toys* [Limberger et al., 2017], which we described in the previous Chapter. This chapter is organized in the following way: we discuss standard methods that aim to compute the discrete Laplace operator for point clouds in Section 7.1. Then, we show how to construct the Mesh-Free Laplace operator in Section 7.2. We show how to create spectral signatures from the eigendecomposition of the MFLO in Section 7.3. Finally, we show experiments of the new Laplace operator applied to non-rigid point-cloud shape retrieval benchmarks in Section 7.4.

7.1 Related Methods

The standard way to compute the Laplace operator of graphs is using the Graph Laplacian (GL). The GL matrix for a finite, simple and undirected graph G, with node set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set E, is given by

$$L = D - A, \tag{7.1}$$

where A is the adjacency matrix

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases} \tag{7.2}$$

and D is the degree matrix

$$D_{ij} = \begin{cases} \deg(x_i) & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{7.3}$$

If we consider the graph edges with some weight, it is possible to compute the weighted graph Laplacian of G [Belkin and Niyogi, 2008]

$$L(G) = D(G) - W(G) \tag{7.4}$$

where W(G) is the weighting adjacency matrix of G

$$W(G)_{\mathbf{ij}} = \begin{cases} w_{\mathbf{ij}} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise,} \end{cases} \tag{7.5}$$

where $w_{\mathbf{ij}}$ is commonly chosen as the isotropic kernel [Coifman and Lafon, 2006]

$$w_{\mathbf{ij}} = e^{-\frac{||x_i - x_j||^2}{t}} \tag{7.6}$$

for some carefully chosen parameter t which depends on point density. $D(G)$ is a diagonal

matrix where the i-th diagonal entry is the total edge weight of node i

$$D_i = \sum_j W_{ij} \tag{7.7}$$

There are two problems with constructing this operator for a point cloud; firstly we do not have a mesh and therefore we need to produce a connection scheme for the points to construct a graph. Secondly, some arbitrary connection scheme will not faithfully reproduce the LBO of the underlying surface. Although there exist efficient algorithms to deal with the problem of converting a point cloud to a surface in the 3-dimensional space [Amenta and Bern, 1998, Amenta et al., 2000], these algorithms can fail in recovering the entire object surface when this is not fully sampled by the point cloud. Furthermore, when the problem is not embedded in the three-dimensional space but in higher dimensions, the solution becomes much more difficult.

Other methods were proposed in the literature to deal with these problems. Belkin *et al.* [Belkin et al., 2008] created an algorithm for approximating the LBO for arbitrary point clouds embedded in a d-dimensional space. Their algorithm computes a tangent space approximation around each point p_i from the point cloud, followed by a local mesh construction using Delaunay triangulation. Finally, to approximate the LBO they compute an integral approximation around p_i. They show that their method converges to the LBO using a sufficiently dense point cloud.

Liu *et al.* [Liu et al., 2012] compute an approximation of the LBO in a similar way to [Belkin et al., 2008]. Their method, called PB-MHT, is also divided into three steps: first a tangent plane is approximated around each point; then, they approximate a Voronoi diagram and compute the area of the Voronoi cell; finally an integral approximation is computed to approximate the LBO on the respective location. The main difference to Belkin *et al.* 's method is that Liu *et al.* 's operator is symmetrizable, therefore it can be written as $L = A^{-1}W$ and generates only real eigenvalues.

In the next section, we detail the steps to construct our Point-Cloud Laplace-Beltrami opearator. It was designed to give an elegant solution to the problems of the Graph Laplacian while being at the same time simple to compute and mathematically accurate.

7.2 Constructing the Laplace operator

In this section, we introduce the framework to compute the Mesh-Free Laplace Operator (MFLO). Based on the problems mentioned in the previous section, we need to construct an edge connection scheme and a faithful weighting scheme to reproduce the underlying unknown manifold to compute accurate distances in this space.

To address these two problems, we begin by adopting weighted edge connections between the points and construct the corresponding Laplacian as in Equation 7.4. Our goal is to find an edge-weighting scheme which properly approximates the LBO and is easy to compute. For this reason, we look for a weighting scheme, that is different from Equation 7.6, which also only depends on inter-point distance

$$l_{ij} = ||x_{\mathbf{i}} - x_{\mathbf{j}}|| \tag{7.8}$$

and is able to properly approximate the underlying manifold.

Let $f(.)$ be some function of interest on the manifold, and let x_i be the manifold coordinates of point i. The discrete Laplacian on the graph formed from these points gives

$$Lf(x_i) = \sum_{j,(i,j)\in E} W_{ij}[f(x_i) - f(x_j)]. \tag{7.9}$$

Using the Taylor series of the function on the manifold in normal co-ordinates at point x_i we can find an expression for $f(x_i) - f(x_j)$ assuming $x_i - x_j$ is small

$$Lf = -\sum_j W(l_{ij})\nabla f \cdot (x_j - x_i) - \sum_j W(l_{ij})\frac{1}{2}(x_j - x_i)'H(x_j - x_i) \tag{7.10}$$

where we have assumed w_{ij} is only dependent on l_{ij}. If the point sampling is isotropic, then we expect the first term, dependent on a sum over $x_j - x_i$, to be small, and similarly the off-diagonal elements of the Hessian H will be small. This allows us to identify the graph Laplacian with the LBO (up to a constant) if

$$\sum_j W(l_{ij})l_{ij}^2 = \text{const.} \tag{7.11}$$

This result depends on a local approximation and so the connection scheme must be localized so it can preserve accuracy. For the connection scheme, we use a k-nearest neighbor graph (k-NNG). Since the Laplacian should be symmetric, we symmetrize the NNG by including edge (i, j) if j is a k-NN of i or i is a k-NN of j. In order to obtain a set of weights satisfying (7.11), we perform Sinkhorn-Knopp normalization on the adjacency matrix A of this graph to obtain a weight matrix W' which has constant row and column sums. Thus, we can find the weights

$$W(G)_{\mathbf{ij}} = \begin{cases} W'_{ij}/l^2_{\mathbf{ij}} & \text{if } (\mathbf{i},\mathbf{j}) \in E, \\ 0 & \text{otherwise,} \end{cases} \tag{7.12}$$

which satisfy (7.11).

Proof. We begin by employing normal coordinates (u, v) at point x_i on the object where we wish to compute the LBO. This gives us a simple form for both the LBO and Taylor series around this point (see a 2D diagram in Figure 7.1):

$$\Delta_{x_i} f = -\partial_u^2 f - \partial_v^2 f \tag{7.13}$$

$$f(x_j) - f(x_i) \simeq \delta u \partial_u f + \delta v \partial_v f + \frac{1}{2}\delta u^2 \partial_u^2 f + \frac{1}{2}\delta v^2 \partial_v^2 f + \delta u \delta v \partial_u \partial_v f + \dots \tag{7.14}$$

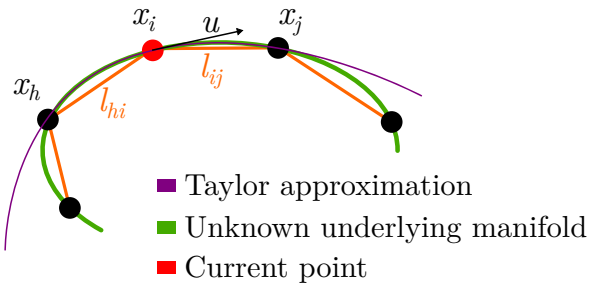


Figure 7.1: 2D diagram of Taylor series' approximation for the local point x_i. Taylor expansion gives a better approximation to the underlying manifold than just using an exponential distance (Equation 7.6).

If the graph Laplacian is sufficiently local, substituting 7.14 in 7.9 with the Taylor expansion we obtain

$$Lf \simeq -\sum_j W_{ij} \left[\delta u_j \partial_u f + \delta v_j \partial_v f + \frac{1}{2}\delta u_j^2 \partial_u^2 f + \frac{1}{2}\delta v_j^2 \partial_v^2 f + \delta u_i \delta v_j \partial_u \partial v f \right] \tag{7.15}$$

Here $W_{ij} = W(l_{ij})$, i.e. it is a function of distance between the points but not direction. We now assume that our points are isotropically sampled from the manifold and sufficiently dense so that the expectation

$$E[\delta u] \approxeq E[\delta v] \approxeq E[\delta u \delta v] \approxeq 0. \tag{7.16}$$

Similarly, $E[\delta u^2] \propto l^2$. Then we can identify the graph Laplacian with the LBO up to some multiplying constant if

$$\sum_j W_{ij} l_{ij}^2 = 2 \tag{7.17}$$

Finally, we set $W'_{ij} = W_{ij} l_{ij}^2$ so that $\sum_j W'_{ij} = 2$. If the graph is regular this is easily satisfied, but since we symmetrize the k-NNG, the resulting graph is only approximately regular. To obtain a more approximate solution, Sinkhorn normalization can be applied to obtain a doubly-stochastic matrix for W' from the adjacency A.

The Sinkhorn-Knopp algorithm [Sinkhorn and Knopp, 1967] normalizes the rows and columns of a nonnegative matrix up to a constant. It takes a matrix W and finds diagonal matrices L and R such that $W' = LWR$ is a matrix where the sum of each row and column is equal to 1 (constant). The term "doubly stochastic" comes from the fact that all rows and columns of the matrix sum to unity. These doubly stochastic matrices have interesting properties and many other applications, for example in communication theory or for page rankings.

By using Taylor series expansion, it is not necessary to have a very high resolution point cloud, since Taylor approximation will try to approximate the real surface instead of simply connecting dots based on their distance. Our method assumes a nearly isotropic sampling of the point cloud, which can be easily and quickly achieved by using, for example, Poisson-disk sampling [Corsini et al., 2012].

A summarized pipeline of the MFLO can be seen in Fig. 7.2. It starts with a point cloud, although it can also be sampled from meshes in a previous step. This is important because meshes are not always well shaped and the KLBO or other mesh Laplacians can fail on the correct estimation. In the second step, the k-NNG is computed. Then, we compute the

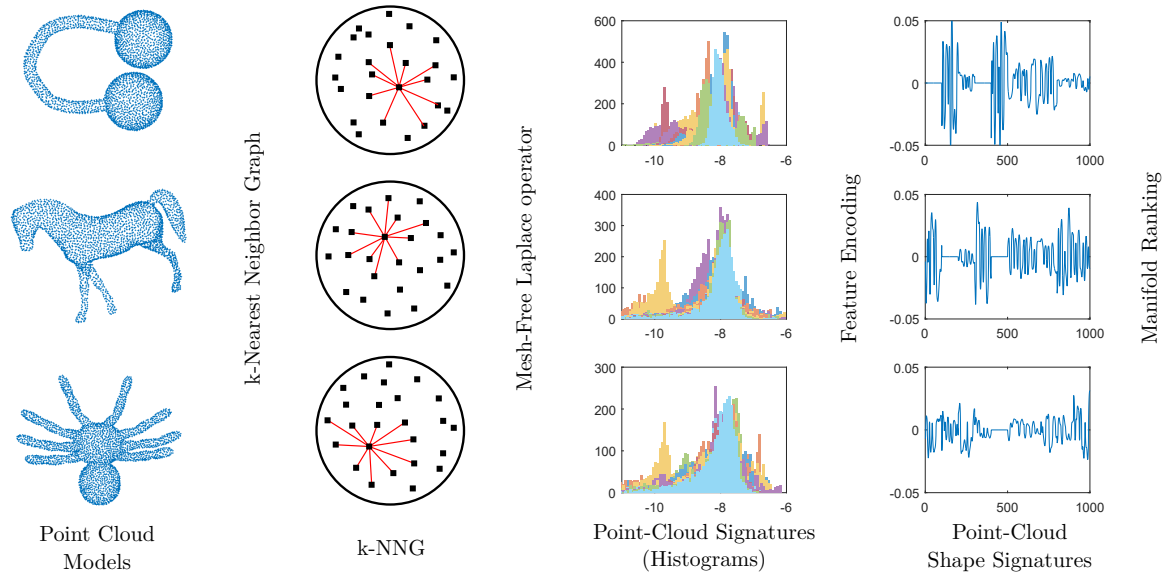


Figure 7.2: Pipeline proposed for the non-rigid shape retrieval problem for point clouds. By weighting the kinetic energy on the Euler-Lagrangian equation we reduce the effect of shape articulations, causing same-class shapes' signatures to be closer to each other. Then, by encoding the local kinetic signatures using either Fisher Vector or Super Vector we are able to compare shapes efficiently using Manifold Ranking technique.

Laplacian and its eigendecomposition to find the eigenvalues and eigenfunctions. From there, we compute local point-cloud signatures (third column). More details about the computation of spectral descriptors can be found in the next section. Finally, we encode the normalized descriptors into shape signatures (fourth column).

7.3 Creating signatures

After computing the shape spectra, we can create spectral descriptors using the same framework proposed in Section 4.2. We compute spectral descriptors using the same parameters described in Section 4.2.1. However, it was empirically observed that spectral descriptors, generated by the MFLO, produce values in a logarithmic distribution, differently from the LBO or the KLBO. In order to control this effect, we apply log normalization to the MFLO descriptors. Log normalization is a process that rescales values from a particular interval (usually close to zero when data is log normally distributed). In log normalization, maximums and minimums are maintained. This normalization is also applied to many other areas where data generated is in log space, for example, gamma ray logs, earthquake strength,

sound loudness, light intensity, etc. We show in Figure 7.3 the effect of applying log normalization to the spectral signatures. The SIHKS is the only signature where this effect is not observed thus we do not apply any normalization to the SIHKS.

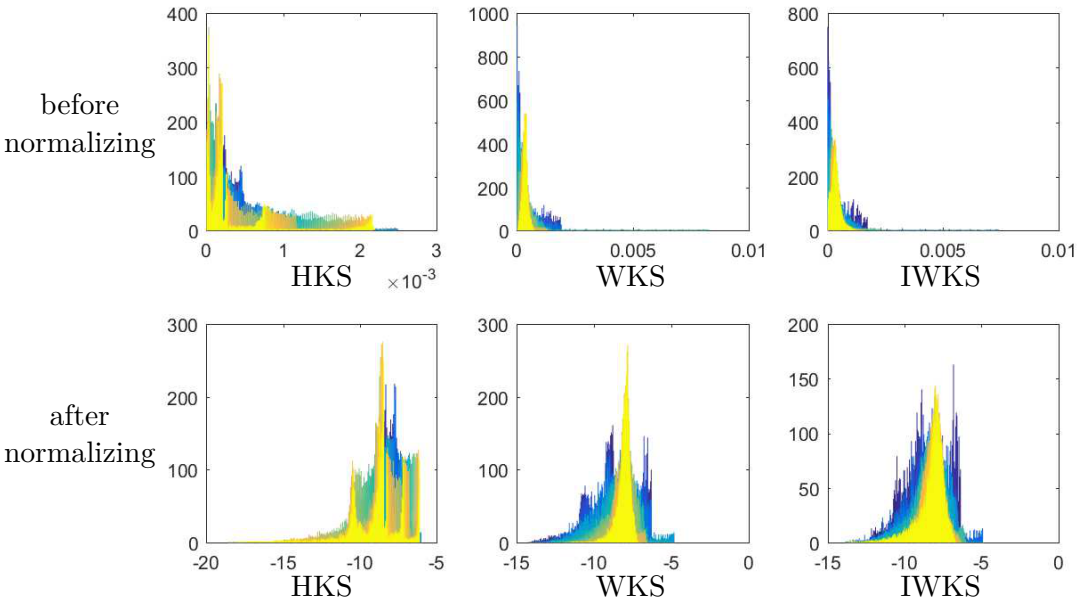


Figure 7.3: Histograms of the MFLO spectral descriptors before the log-normalization (first row) and after log-normalization (second row).

By first computing a log-normalization of the descriptors, the FV and the SV benefit from it. These encoding methods are able to construct better dictionaries using Gaussian mixture models and they are able to properly compute descriptor deviations from the vocabulary that are more discriminative since dictionary words are now more distinct than before.

Therefore, we encode the normalized descriptors using either the Fisher Vector or the Super Vector, according to Section 4.2.2. Following, we compute distances between descriptors using Efficient Manifold Ranking as described in Section 4.3.

7.4 Experiments

In this section, we show many experiments about the MFLO. First, we show that the MFLO converges to the Laplace-Beltrami operator on standard shapes like a rectangle and a sphere, where the manifold Laplacian can be computed explicitly and compared with the MFLO

eigenfunctions.

Secondly, we show experiments applied to the non-rigid point-cloud shape retrieval problem. In the same way that the KLBO is used to compute the Laplacian from meshes, we use the our framework to compute spectral descriptors from point clouds by using the MFLO instead. Therefore, we show that the MFLO descriptors can also be approximated by GMMs. To test signature performance, we tested the MFLO in the PRoNTo benchmark and compare against all other shape signatures. We also sampled three mesh datasets [Lian et al., 2010, Lian et al., 2011, Lian et al., 2015] into point clouds and compare against other point-cloud signatures and also mesh signatures.

7.4.1 Experiments on standard shapes

We begin by showing experiments on standard shapes to analyze the convergence behavior of the MFLO. We created 5 *rectangles* and 5 *spheres* with different samplings to compare the manifold Laplacian, which is explicitly computed from these shapes, with the MFLO eigenfunctions. Both the rectangle and the sphere were sampled using Poisson-disk sampling algorithm which tries to create an isotropic sampling of the manifold. Figure 7.4 shows the point clouds of the rectangle and sphere with 500 samples used to compute these errors.

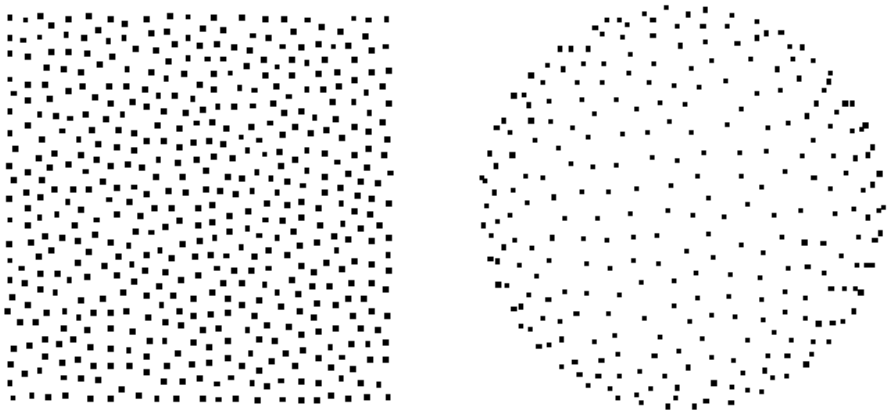


Figure 7.4: Rectangle and sphere with 500 points used to compute the manifold Laplacian errors. Only the front part of the sphere is shown above.

The *rectangles* were sampled in the range $(x, y) \in ([0, 1], [0, 1])$ with the respective number of samples described in Table 7.1. We compare the MFLO eigenfunctions errors with the respective PCDLaplace [Belkin et al., 2009] and the WGL [Lafon, 2004] eigenfunctions errors

on the shapes. To compute the errors we evaluate the manifold Laplacian Δ of the rectangle with side a and b using Neumann boundary conditions

$$\psi_{m,n} = \cos\left(\frac{m\pi}{a}x\right) * \cos\left(\frac{n\pi}{b}y\right) \quad m, n \in \{0, 1, 2, \ldots\}. \tag{7.18}$$

We compute the different Laplacian methods $\tilde{\Delta}$ and measure the difference between their eigenfunctions $\tilde{\Phi}$ to the manifold Laplacian eigenfunctions Φ using the normalized L_2 error

$$E_2 = \frac{||\Phi_t - \tilde{\Phi}_t||_2}{||\Phi_t||_2} \tag{7.19}$$

which in this case $\Phi = \psi_{m,n}$. Then, we average the errors of the 2nd, 3rd and 4th eigenfunctions and show in Table 7.1 the Laplacian errors for the rectangle with different number of samples.

Table 7.1: Laplacian errors for the *rectangle* with different number of samples.

Methdod	500	1000	2000	4000	5000
MFLO	0.0500	0.0566	0.0342	0.0214	0.0182
WGL	0.0691	0.0628	0.0417	0.0284	0.0235
PCDL	0.3495	0.3450	0.3495	0.3486	0.3448

We do the same thing for the *sphere*, however, the exact solutions of the manifold Laplacian on the sphere are given by the spherical harmonics of degree l and order m

$$Y_l^m(\theta, \phi) = Ne^{im\phi}P_l^m(\cos(\theta)) \tag{7.20}$$

where N is a normalization weight and P_l^m is a Legendre function. Again, we average the errors of the 2nd, 3rd and 4th eigenfunctions and show in Table 7.2 the Laplacian errors for the sphere with different number of samples. We use the same error function (Equation 7.19) where this time $\Phi = Y_l^m(\theta, \phi)$. Figure 7.5 shows the errors in plots to help visualizing the data values and the convergence of the methods.

It is possible to observe that in both the *rectangle* and the *sphere*, using isotropic sampling, the MFLO converges to the manifold Laplacian as we increase the number of samples

Table 7.2: Laplacian errors for the *sphere* with different number of samples.

Methdod	500	1000	2000	4000	5000
MFLO	0.0559	0.0424	0.0294	0.0378	0.0198
WGL	0.0594	0.0414	0.0278	0.0278	0.0214
PCDL	0.5139	0.0143	0.0060	0.0203	0.0178

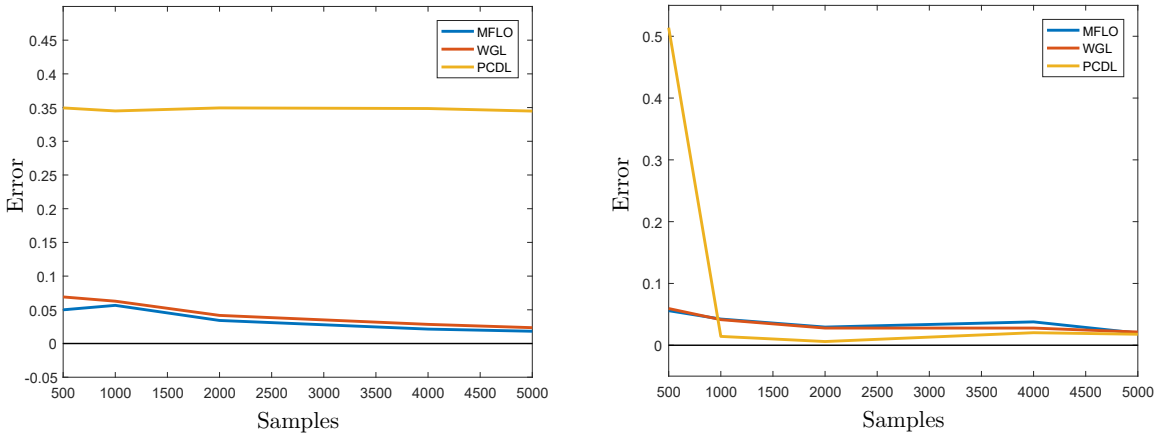


Figure 7.5: Laplacian errors on the **rectangles** (left) and on the **spheres** (right).

in the experiments. In the *sphere* case, PCDLaplace converge faster than the WGL and MFLO. The large error in the *sphere* case (point cloud with 500 points) for PCDLaplace is because it only generated two distinct functions on the 2nd 3rd and 4th eigenfunctions when it should generate three ($f = x, f = y$ and $f = z$). It is also interesting to note that the PCDLaplace does not work properly in the *rectangle* surface, giving large errors, while it looks like the PCDLaplace is specifically tailored for the *sphere*. This analysis was not investigated by Belkin *et al.* in their paper [Belkin et al., 2009], where they only test their algorithm against shapes without boundaries. This phenomenon is much more noticeable after the $4th$ eigenfunction (see Figure 7.6), those which we do not compare on the error experiments of Table 7.2. Perhaps, PCDLaplace does not work well on point clouds with boundaries but this is still a topic for further research. The same effect of Figure 7.6, where PCDLaplace shows eigenfunctions with many outlier values on the boundaries also occurs in the majority of models from PRoNTo, since they also have boundaries (holes and missing parts).

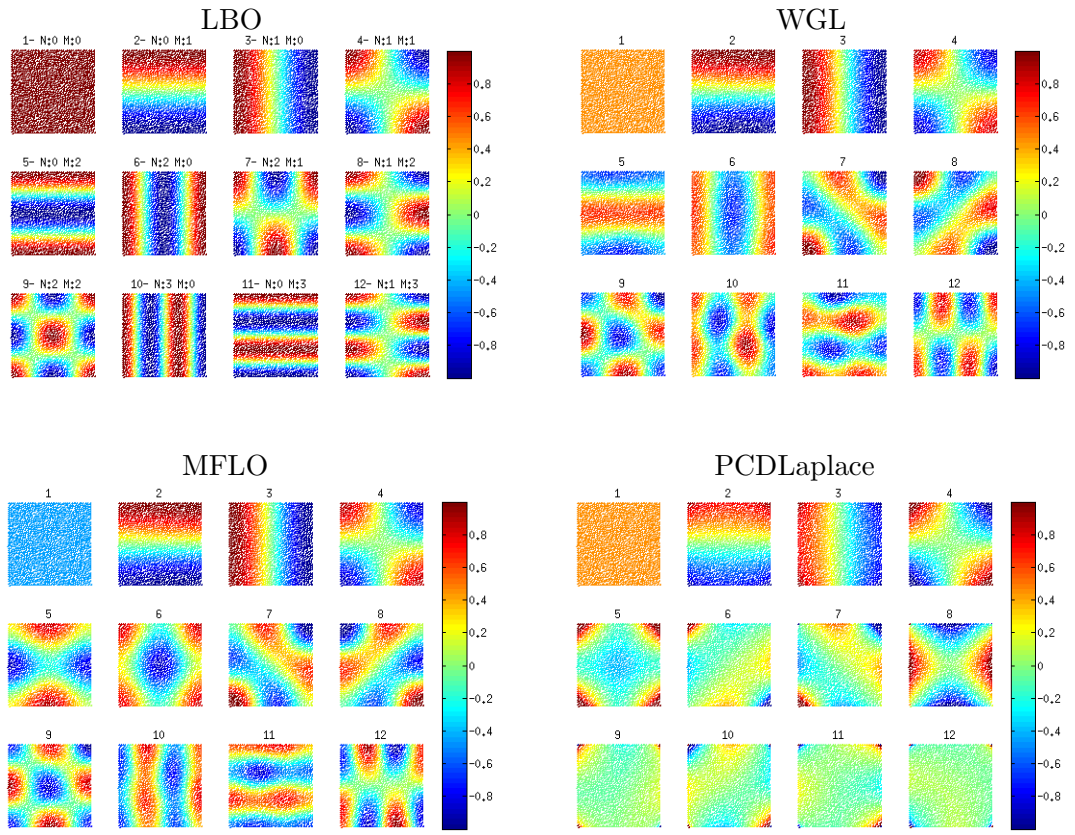


Figure 7.6: Eigenfunctions on the *rectangle* with $2K$ samples. Top left, the eigenfunctions computed analytically (LBO), top right, the WGL, bottom left, the MFLO and, bottom right, PCDLaplace. Note that the eigenfunctions of MFLO, WGL and PCDLaplace can be a linear combination of eigenfunctions of the LBO. These linear combinations are also valid solutions and they can happen in any higher dimensional eigenspaces than 1 dimension. These solution have been taken into account when computing the respective errors. It can be seen that, the WGL and the MFLO can successfully compute the eigenfunctions for the shape *rectangle*, while the PCDL fails in computing eigenfunctions, mainly after the $4th$ frequency.

Furthermore, the convergence of the MFLO seems to be similar to the convergence of the WGL. This happens because when the sampling is very dense the WGL and the MFLO are almost identical. On the other hand, when sampling is sparse MFLO shows a bigger advantage over the WGL in approximating the manifold Laplacian. We make another experiment (see Table 7.3) using the vertices of a regular *icosahedron* with 42, 162, 642 and 2562 samples to show that when the sampling is sparse the MFLO shows a smaller error compared to WGL when approximating the Laplacian of a sphere. The respective *icosahedron* point clouds are shown in Figure 7.7. Another time, the PCDLaplace failed in generating all eigenfunctions to the icosahedron with 162 samples.

Table 7.3: Laplacian errors for a regular *icosahedron* approximating a **sphere** with different number of samples.

Methdod	42	162	642	2562
MFLO	0.1108	0.0667	0.0663	0.0617
WGL	0.1468	0.0798	0.0739	0.0621
PCDL	0.1112	0.5068	0.0068	0.0315

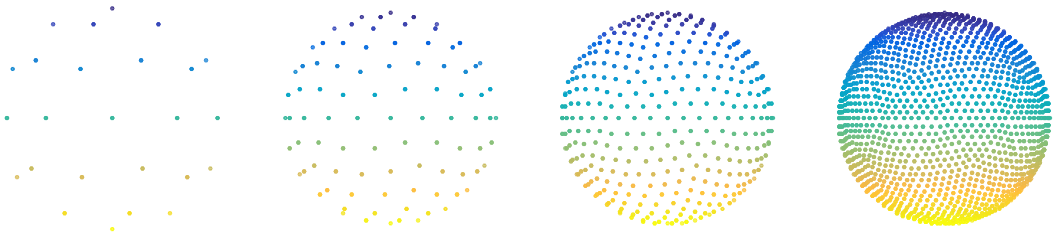


Figure 7.7: Regular *icosahedron* point clouds with number of samples 42, 162, 642 and 2562 used in the experiments of Table 7.3.

7.4.2 GMM dictionaries and MFLO descriptors

After showing experiments comparing the theoretical Laplacian on standard surfaces, we show that the MFLO descriptors can be also used with GMMs dictionaries and thus create signatures for non-rigid shape retrieval. Figure 7.8 shows 5 randomly chosen shape features computed with the MFLO and their respective GMMs approximated using the Expectation-maximization algorithm. Figure 7.9 shows the errors generated by the respective approximations. We can see that errors stabilize around 0.006 for three signatures (SIHKS, WKS and IWKS), while the HKS present the largest error, around 0.008.

7.4.3 Benchmark experiments

Secondly, we present experiments on the PRoNTo, PC-SHREC'10, PC-SHREC'11 and PC-SHREC'15 benchmarks. In this part, we show the same retrieval statistics as before in this thesis (NN, FT, ST, E, DCG, mAP).

In the PRoNTo benchmark, we compute MFLO signatures using different parameters to test signatures performance against different sample rates. We created a second dataset with models averaging 10K points each (PRoNTo 10K). Then, we compute the MFLO with the same number of nearest neighbours for constructing the k-NN graph that we used for the

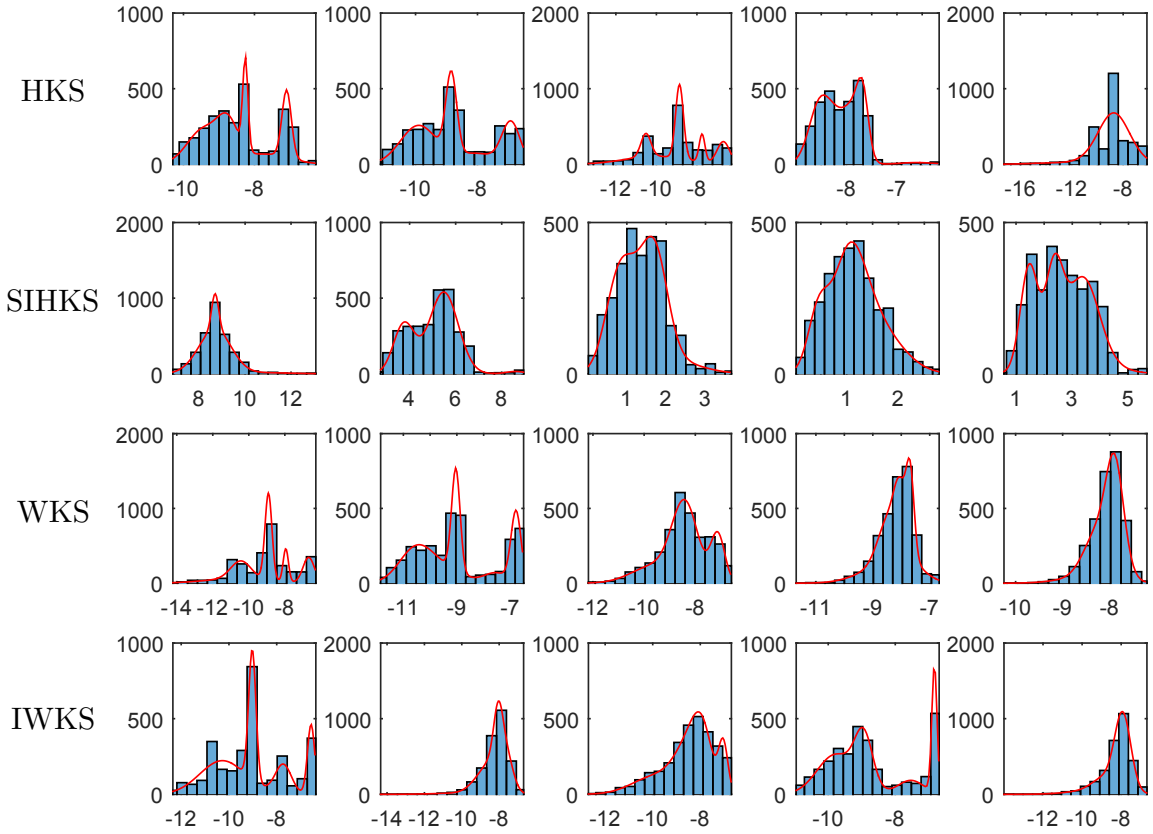


Figure 7.8: Histograms of randomly-chosen shape features plotted with the respective learned GMM with 5 components using EM algorithm for different MFLO signatures. Each row represent features from HKS, SIHKS, WKS and IWKS, respectively, from the first model of the database.

original dataset (PRoNTo 4K). Later, we increased the number of nearest neighbours up to 40 and computed the retrieval performance again in PRoNTo 10K.

In Table 7.4, we show the retrieval performance of the MFLO-15 (using 15 nearest neighbours) on the PRoNTo 4K. Results on the PRoNTo 4K can vary slightly to the results of previous chapter because here we average the results over 4 runs. Then, in Table 7.5, we show the performances of MFLO-15 on the PRoNTo 10K. Finally, in Table 7.6, we show the results of MFLO-40 on the PRoNTo 10K. In the following, Figures 7.10, 7.11 and 7.12 show the respective precision and recall plots of the previous experiments.

From these experiments, we see that, to achieve similar retrieval performance, the number of nearest neighbours chosen to construct the k-NN graph has to increase with the number of point-cloud samples. This has to be with the fact that the neighbourhood needs to have the same size to estimate a similar descriptor. In a denser point cloud, it is necessary to pick

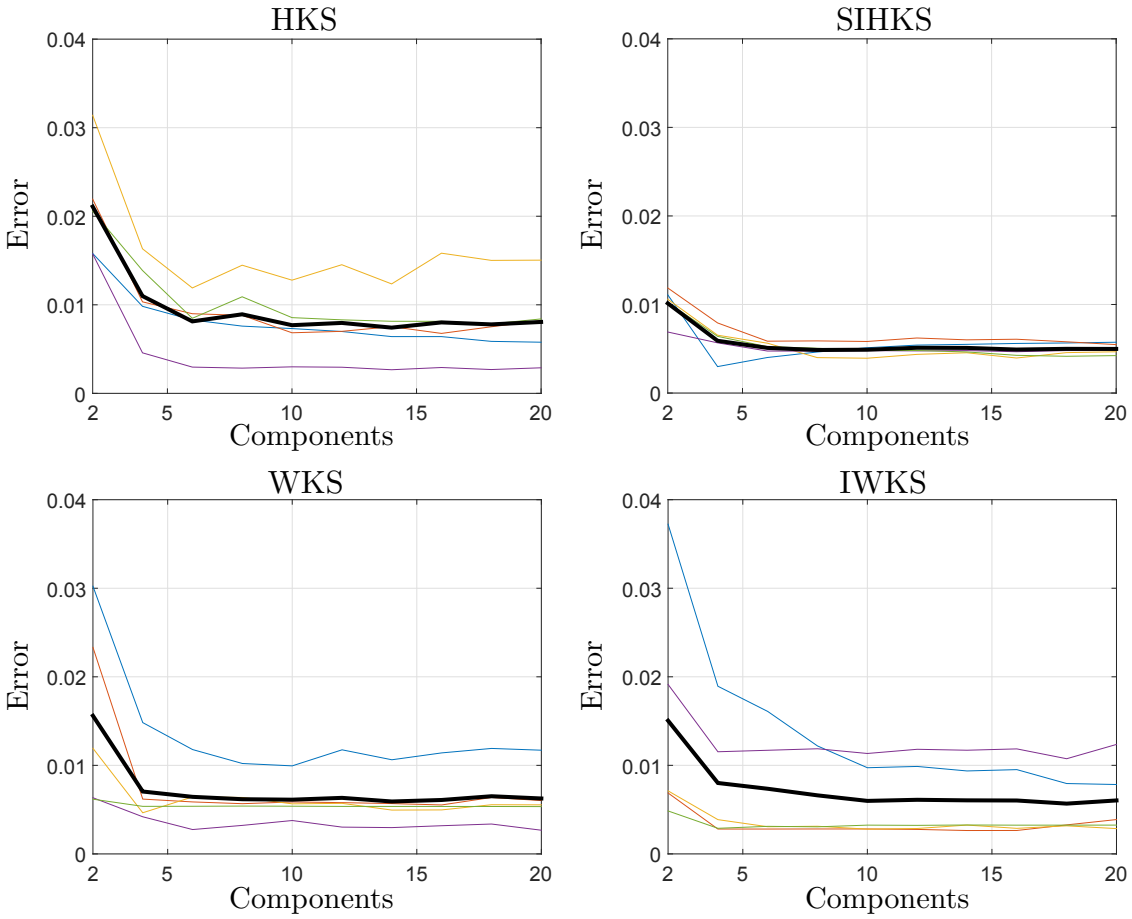


Figure 7.9: Residuals of the GMM fitting on the five feature histograms of Fig. 7.8 for each MFLO signature. The black line represents the average loss of the five histograms. As can be seen, the error stabilizes around 0.006 in most of cases when are used in average again 5 components or more.

more neighbour points to cover the same area that in a less dense point cloud. Therefore, it can be seen that the results of the MFLO-15 on PRonTo 4K are similar with the results of MFLO-40 on PRoNTo 10K, while the results of MFLO-15 on PRoNTo 10K are slightly worse. Figure 7.13 shows the plot of DCG for each spectral signature in the three tests mentioned above (MFLO-15 PRoNTo 4K; MFLO-15 PRoNTo 10K; MFLO-40 PRoNTo 10K).

In the next experiments, we compute point-cloud versions of SHREC'10, SHREC'11 and SHREC'15 using Poisson-disk algorithm [Corsini et al., 2012]. Then, we sampled all models to point clouds with in average 3K points each and computed spectral signatures from the models using the MFLO-15. Tables 7.7, 7.8 and 7.9 show the retrieval statistics of different spectral signatures using the MFLO-15 to compute the shape spectra. We can see from

Table 7.4: Retrieval performances of the spectral signatures computed with the **MFLO** combined with FV or SV applied to **PRoNTo 4K benchmark** using the **15 nearest neighbours** for constructing the k-NN graph. Dissimilarities are computed using **Manifold Ranking**. Bold values represent best retrieval performances for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.6825	0.4570	0.5653	0.2867	0.7195	0.543
FV-SIHKS	0.8550	0.6386	0.7561	0.3662	0.8412	0.780
FV-WKS	0.8175	0.6450	0.7242	0.3526	0.8374	0.743
FV-IWKS	**0.9050**	**0.7919**	**0.8578**	**0.3867**	**0.9086**	**0.858**
SV-HKS	0.6425	0.3722	0.4756	0.2557	0.6645	0.482
SV-SIHKS	0.7850	0.5850	0.7222	0.3644	0.8066	0.696
SV-WKS	0.7450	0.6086	0.7278	0.3501	0.8079	0.681
SV-IWKS	0.8900	0.7256	0.8108	0.3798	0.8860	0.800

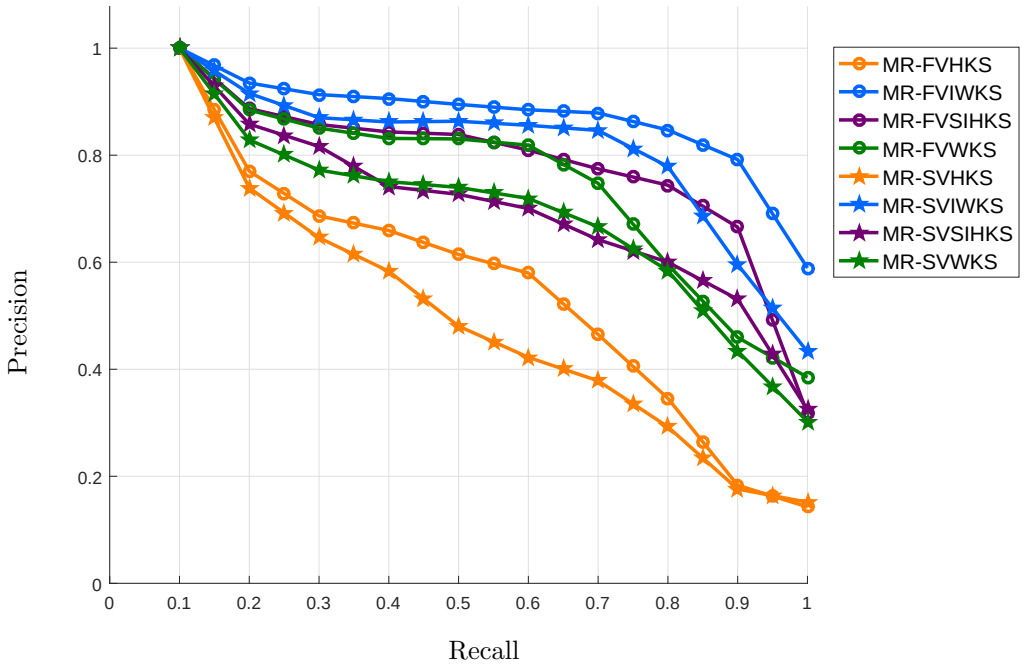


Figure 7.10: Precision and Recall plot of the spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV or SV encoding methods applied to the **PRoNTo 4K benchmark**. The shape spectra is computed by the **MFLO** using **15 nearest neighbours** for construction of the kNN graph. Distances are computed using **Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the two best performance in this benchmark (MR-FVIWKS and MR-SVIWKS) use our proposed descriptor.

Table 7.5: Retrieval performances of the spectral signatures computed with the **MFLO** combined with FV or SV applied to **PRoNTo 10K benchmark** using the **15 nearest neighbours** for constructing the k-NN graph. Dissimilarities are computed using **Manifold Ranking**. Bold values represent best retrieval performances for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.6000	0.3492	0.4511	0.2476	0.6465	0.513
FV-SIHKS	0.7475	0.5578	0.6586	0.3325	0.7819	0.677
FV-WKS	0.7725	0.6092	0.7056	0.3443	0.8135	0.715
FV-IWKS	0.8500	**0.7147**	0.8028	0.3732	**0.8670**	**0.817**
SV-HKS	0.5650	0.3095	0.4036	0.2311	0.6100	0.438
SV-SIHKS	0.7225	0.5345	0.6309	0.3194	0.7566	0.634
SV-WKS	0.7425	0.5514	0.6539	0.3282	0.7828	0.647
SV-IWKS	**0.8625**	0.7078	**0.8175**	**0.3850**	**0.8670**	0.795

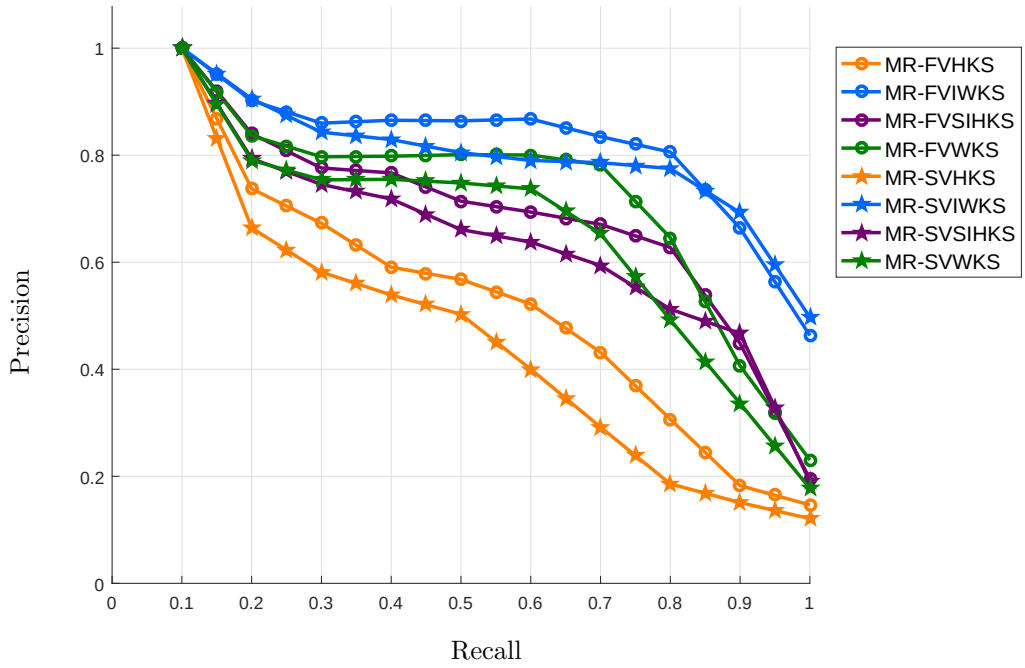


Figure 7.11: Precision and Recall plot of the spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV or SV encoding methods applied to the **PRoNTo 10K benchmark**. The shape spectra is computed by the **MFLO** using the **15 nearest neighbours** for construction of the kNN graph. Distances are computed using **Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the two best performance in this benchmark (MR-FVIWKS and MR-SVIWKS) use our proposed descriptor.

Table 7.6: Retrieval performances of spectral signatures computed with the **MFLO** combined with FV or SV applied to **PRoNTo 10K benchmark** using the **40 nearest neighbours** for constructing the k-NN graph. Dissimilarities are computed using **Manifold Ranking**. Bold values represent best retrieval performances for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.6575	0.4056	0.5225	0.2786	0.6933	0.536
FV-SIHKS	0.8000	0.5981	0.6958	0.3449	0.8065	0.701
FV-WKS	0.7775	0.6153	0.7203	0.3538	0.8221	0.735
FV-IWKS	**0.9150**	**0.8223**	**0.9109**	**0.4142**	**0.9292**	**0.884**
SV-HKS	0.6275	0.3683	0.4667	0.2500	0.6515	0.496
SV-SIHKS	0.7875	0.5686	0.6967	0.3530	0.7972	0.698
SV-WKS	0.7350	0.5672	0.6639	0.3271	0.7883	0.672
SV-IWKS	0.9100	0.7972	0.8744	0.4038	0.9162	0.872

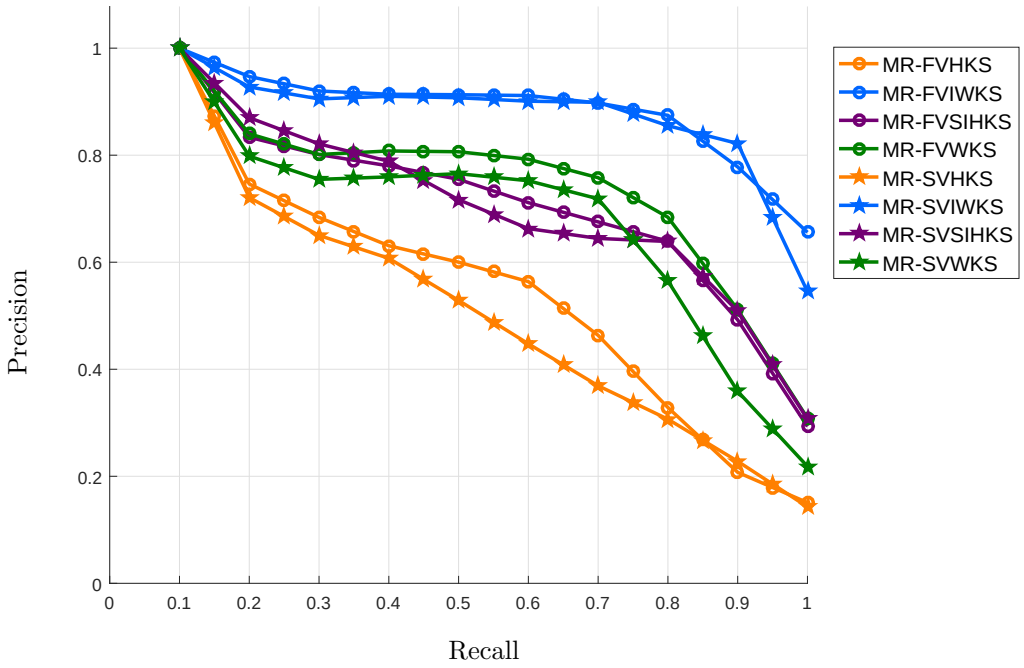


Figure 7.12: Precision and Recall plot of different spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **PRoNTo 10K benchmark**. The shape spectra is computed by the **MFLO** using the **40 nearest neighbours** for construction of the kNN graph. Distances are computed using **Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the two best performance in this benchmark (MR-FVIWKS and MR-SVIWKS) use our proposed descriptor.

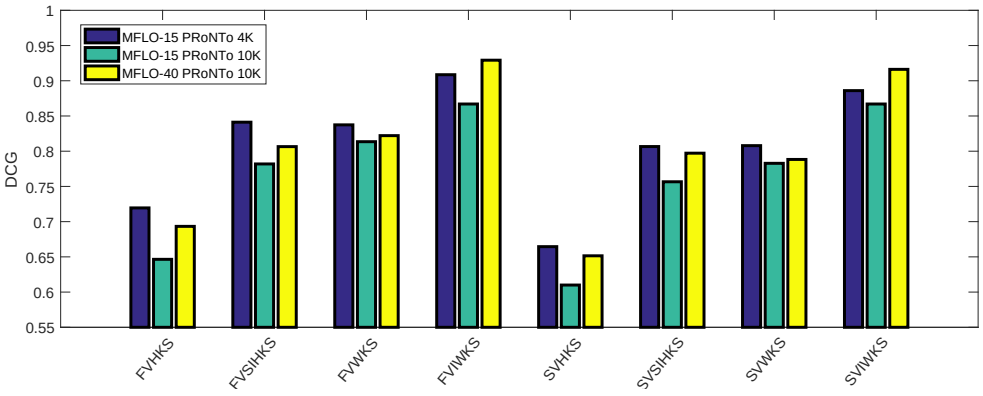


Figure 7.13: Comparison of different sampling rates for MFLO spectral descriptors. We show results of DCG over different spectral signatures of MFLO-15 applied to PRoNTo 4K and PRoNTo 10K and MFLO-40 applied to PRoNTo 10K.

these experiments that the MFLO-15 can compute a very discriminative shape spectra since descriptors achieve very high retrieval performances on all these datasets.

Table 7.7: Retrieval performances of spectral signatures computed with the **MFLO-15** combined with FV and SV applied to the **PC-SHREC'10 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent best retrieval performances for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9638	0.8712	0.9138	0.6728	0.9584	0.910
FV-SIHKS	0.9538	0.7950	0.9124	0.6639	0.9427	0.858
FV-WKS	0.9688	0.9085	0.9613	0.7090	0.9712	0.954
FV-IWKS	**0.9900**	**0.9667**	**0.9919**	**0.7360**	**0.9912**	**0.997**
SV-HKS	0.9500	0.8415	0.9120	0.6712	0.9472	0.888
SV-SIHKS	0.9438	0.7790	0.9155	0.6560	0.9385	0.874
SV-WKS	0.9838	0.9091	0.9525	0.7049	0.9755	0.938
SV-IWKS	0.9650	0.8724	0.9369	0.6865	0.9622	0.909

We also make comparisons with other Laplace operators. We compare our MFLO with the PCDLaplce [Belkin et al., 2009], the PB-MHT [Liu et al., 2012], the standard graph Laplacian and the weighted graph Laplacian [Belkin and Niyogi, 2008]. We compute descriptors in the same way we did in Chapter 4 for the LBO, but only changing the way we

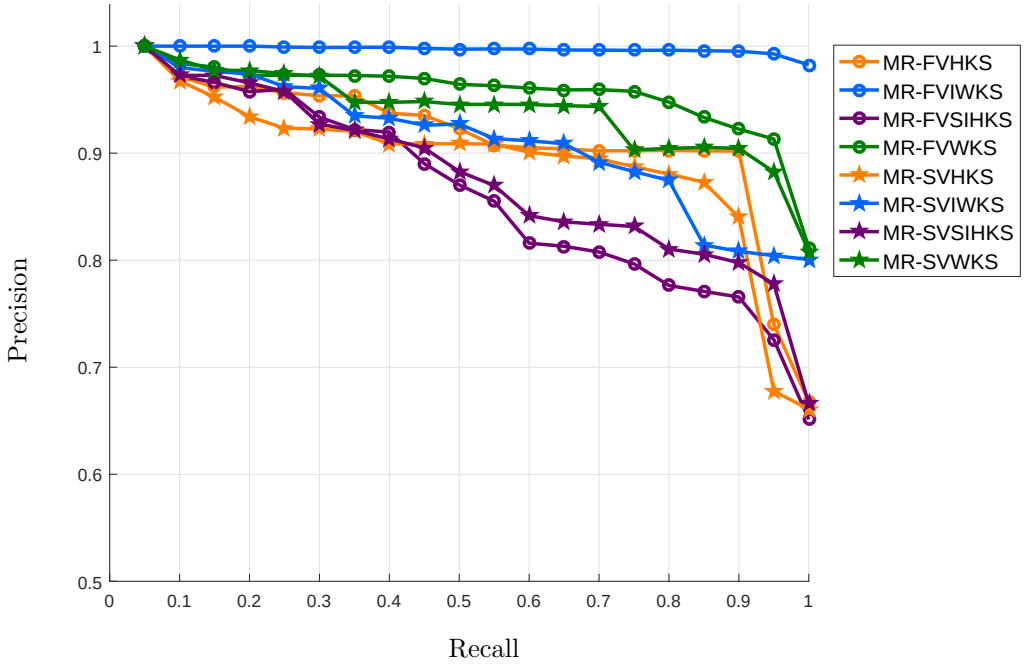


Figure 7.14: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **PC-SHREC'10 benchmark**. The shape spectra is computed by the **MFLO-15**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the best performance in this benchmark is MR-FVIWKS.

Table 7.8: Retrieval performances of spectral signatures computed with the **MFLO-15** combined with FV and SV applied to the **PC-SHREC'11 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent best retrieval performances for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9458	0.8494	0.9108	0.6655	0.9447	0.904
FV-SIHKS	0.9621	0.9042	0.9585	0.7022	0.9649	0.929
FV-WKS	0.9892	0.9682	0.9874	0.7307	0.9910	0.969
FV-IWKS	0.9904	0.9735	0.9838	0.7310	0.9902	0.982
SV-HKS	0.9329	0.8079	0.8931	0.6464	0.9310	0.863
SV-SIHKS	0.9629	0.8992	0.9602	0.6991	0.9674	0.946
SV-WKS	0.9867	0.9645	0.9820	0.7272	0.9881	0.983
SV-IWKS	**0.9979**	**0.9929**	**0.9985**	**0.7425**	**0.9984**	**0.997**

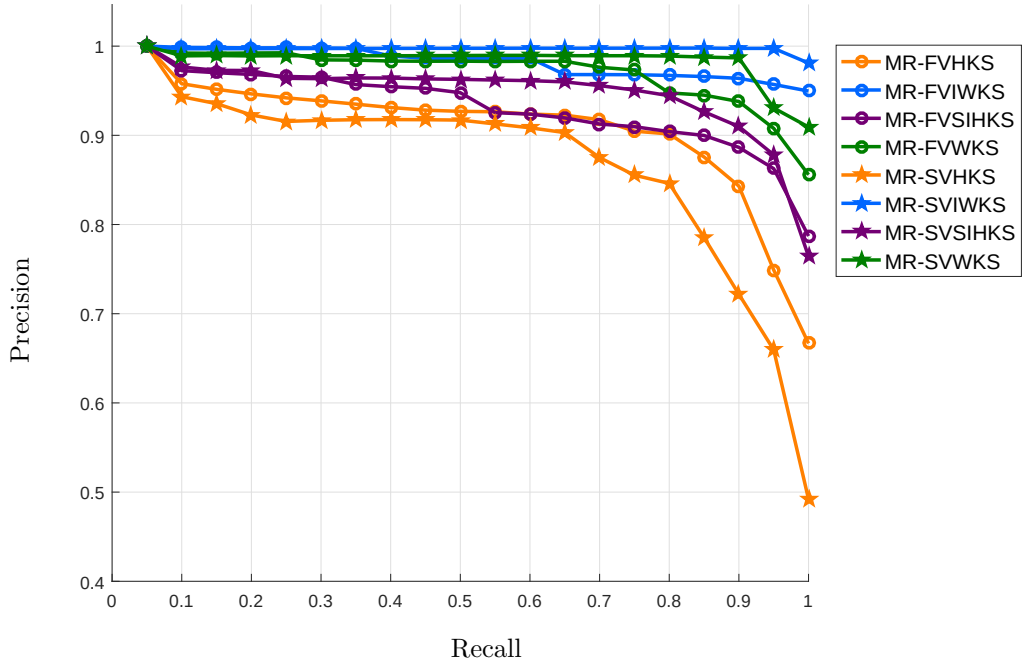


Figure 7.15: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **PC-SHREC'11 benchmark**. The shape spectra is computed by the **MFLO-15**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the best performance in this benchmark is MR-SVIWKS.

Table 7.9: Retrieval performances of spectral signatures computed with the **MFLO-15** combined with FV and SV applied to the **PC-SHREC'15 benchmark**. Dissimilarities are computed using **Efficient Manifold Ranking**. Bold values represent best retrieval performances for each evaluation measure.

Method	NN	FT	ST	E	DCG	mAP
FV-HKS	0.9288	0.7804	0.8382	0.6759	0.9140	0.825
FV-SIHKS	0.9533	0.8674	0.9130	0.7456	0.9488	0.895
FV-WKS	0.9852	0.9431	0.9593	0.7967	0.9795	0.953
FV-IWKS	**0.9946**	**0.9797**	**0.9862**	**0.8221**	**0.9929**	**0.988**
SV-HKS	0.9194	0.7470	0.8110	0.6525	0.8992	0.780
SV-SIHKS	0.9496	0.8542	0.9014	0.7348	0.9439	0.877
SV-WKS	0.9785	0.9225	0.9468	0.7814	0.9723	0.941
SV-IWKS	0.9911	0.9724	0.9819	0.8177	0.9904	0.972

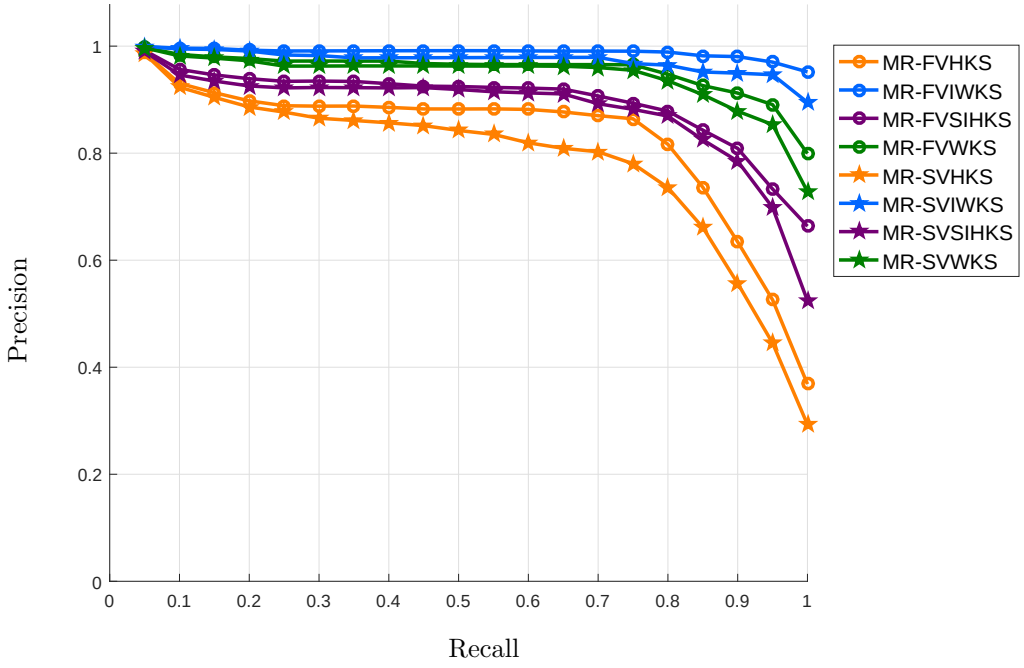


Figure 7.16: Precision and Recall plot of spectral signatures (HKS, SIHKS, WKS and IWKS) tested with FV and SV encoding methods applied to the **PC-SHREC'15 benchmark**. The shape spectra is computed by the **MFLO-15**. Distances are computed using **Efficient Manifold Ranking**. Equal colours represent the same local descriptor. As shown above, the two best performances in this benchmark are MR-FVIWKS and MR-SVIWKS.

compute the Laplace operator for each case. Thus, we generate spectral signatures using these three methods. For all Laplacian methods we use the same parameters and we also apply log normalization. Therefore, we use the 15 nearest neighbors to compute the respective Laplacian matrix. In Table 7.10, we show this comparison using e-measure as retrieval comparison score.

Finally, Tables 7.11, 7.13 and 7.14 show a full comparison of all point-cloud and mesh descriptors on the three non-rigid SHREC benchmarks. To make it clear, we used MFLO, PCDLaplace, GL and WGL to compute the spectrum of the point-cloud datasets, and the other results are computed from the original mesh datasets. We indicate, in the respective point-cloud descriptors, that they were computed using the point-cloud version of the dataset. The respective signatures presented in these tables were the best combinations of all spectral descriptors for each Laplace operator.

Table 7.10: Comparison of the MFLO, PCDLaplace and Graph Laplacian applied to different spectral descriptors on different point-cloud benchmarks. In bold are highlighted the best retrieval performances for each benchmark and method. All methods use Efficient Manifold Ranking to compute dissimilarities.

Benchmark	Descriptor	PCDL-15 (E)	GL-15 (E)	WGL-15 (E)	PBMHT (E)	MFLO-15 (E)
PC-SHREC'10	FVHKS	0.5274	0.6774	0.6679	0.3231	0.6728
	FVSIHKS	0.4865	0.7044	0.6955	0.3147	0.6639
	FVWKS	**0.6374**	0.7072	0.7126	0.4438	0.7090
	FVIWKS	0.6205	0.7053	0.7089	**0.5106**	**0.7360**
	SVHKS	0.5478	0.6573	0.6637	0.3265	0.6712
	SVSIHKS	0.4731	0.7061	**0.7171**	0.3403	0.6560
	SVWKS	0.6277	**0.7110**	0.7057	0.4553	0.7049
	SVIWKS	0.6128	0.7038	0.7128	0.4525	0.6865
PC-SHREC'11	FVHKS	0.4584	0.6245	0.6552	0.2337	0.6655
	FVSIHKS	0.5090	0.6775	0.7226	0.2242	0.7022
	FVWKS	0.6523	0.7145	0.7259	0.3426	0.7307
	FVIWKS	0.6734	0.7339	0.7365	**0.4682**	0.7310
	SVHKS	0.4584	0.6098	0.6403	0.2670	0.6464
	SVSIHKS	0.4963	0.6481	0.7106	0.2214	0.6991
	SVWKS	0.6491	0.7110	0.7223	0.3302	0.7272
	SVIWKS	**0.6844**	**0.7352**	**0.7402**	0.4421	**0.7425**
PC-SHREC'15	FVHKS	0.4407	0.5971	0.6713	0.1607	0.6759
	FVSIHKS	0.5027	0.6830	0.7747	0.1991	0.7456
	FVWKS	0.7121	0.7623	0.7943	0.3635	0.7967
	FVIWKS	0.7664	0.8149	0.8224	**0.4706**	**0.8221**
	SVHKS	0.4213	0.5720	0.6344	0.1690	0.6525
	SVSIHKS	0.4678	0.6811	0.7672	0.2125	0.7348
	SVWKS	0.6951	0.7585	0.7779	0.3293	0.7814
	SVIWKS	**0.7702**	**0.8159**	**0.8230**	0.4156	0.8177

7.5 Timing analysis

The MFLO has similar computational complexity to the LBO. It is comprises the computation of the matrix $L(G)$ and its eigendecomposition. We also have to compute Sinkhorn-Knopp normalization to compute $L(G)$. This step in linear in the number of matrix components. Like before, the computational complexity of computing $L(G)$ is dominated by its

Table 7.11: Retrieval performance comparison with state-of-the-art signatures on SHREC'10 sorted by mAP. We show best runs from the three groups that performed better in SHREC'10 (MR-BF-DSIFT-E, DMEVD_run1, CF) and other descriptors that outperformed those, against our descriptors. In bold are highlighted the best performances for each retrieval measure.

Descriptor	e-Measure	mAP [%]
MFLO-FVIWKS (point-cloud dataset)	**0.7360**	**99.7**
KLBO-SVWKS	**0.7328**	**99.1**
ConTopo++ [Sfikas et al., 2011]	0.7140	97.6
GL-SVWKS (point-cloud dataset)	0.7110	96.3
WGL-SVSIHKS (point-cloud dataset)	0.7171	94.7
MR-BF-DSIFT-E	0.7055	95.4
DMEVD_run1	0.7012	94.1
MDS-ZFDR [Li et al., 2014a]	-	94.1
PCDLaplace-FVWKS (point-cloud dataset)	0.6374	86.6
SV-IWKS [Limberger and Wilson, 2015]	0.5957	80.0
CF	0.5527	75.2
PBMHT-FVIWKS (point-cloud dataset)	0.5106	60.5

own eigendecomposition.

Table 7.12: Average computation times (in seconds) for computing one point-cloud signature for models with 3K vertices from each dataset. MFLO stands for computing the Laplace matrix and their eigenvalues and eigenfunctions. HKS, SIHKS, WKS and IWKS stand for time to compute respective signatures. FV and SV stand for computation times of Fisher Vector and Super Vector. EMR represents the time to perform retrieval of one model. Total times to compute signatures and retrieve all models using either FV or SV are shown in Total-FV and Total-SV columns. All values represent the average computational times for computing one point-cloud shape signature with 3K vertices.

Benchmark	MFLO	HKS	SIHKS	WKS	IWKS	FV	SV	EMR	**Total-FV**	**Total-SV**
PC-SHREC'10	4.12	0.02	1.34	0.01	0.01	0.11	1.41	0.01	**1,122**	**1,382**
PC-SHREC'11	4.12	0.02	1.34	0.01	0.01	0.10	1.61	0.01	**3,360**	**4,266**
PC-SHREC'15	4.12	0.02	1.34	0.01	0.01	0.23	1.72	0.02	**6,876**	**8,664**

The worst case scenario for computing the eigenfunctions is $O(|V|^3)$. Here, the MFLO is also sparse and it can take advantage of specific methods to solve this special eigendecomposition. Also, we only compute the first 100 eigenvalues and eigenfunctions which decreases even more the computation time. We show in Table 7.12 detailed running times to compute MFLO. In the table, we show average times to compute one model from each database.

Table 7.13: Retrieval performance comparison with state-of-the-art signatures on SHREC'11 sorted by mAP. We show the best runs from the six groups that performed better in SHREC'11 (SD_GDM-meshSIFT, MDS-CM-BOF, OrigM-n12-normA, FOG+MRR, BOGH, LSF) and other descriptors that outperformed those, against our descriptors. In bold are highlighted the best performances for each retrieval measure.

Descriptor	e-Measure	mAP [%]
KLBO-FVWKS/KLBO-SVIWKS/KLBO-SVSIHKS	**0.7451**	**100.0**
MFLO-SVIWKS (point-cloud dataset)	0.7425	99.7
WGL-SVIWKS (point-cloud dataset)	0.7402	99.3
3DVFF [Furuya and Ohbuchi, 2014]	-	99.1
GL-SVIWKS (point-cloud dataset)	0.7352	98.6
SD-GDM-meshSIFT	0.7358	98.5
FV-IWKS [Limberger and Wilson, 2015]	0.7318	98.0
MDS-ZFDR [Li et al., 2014a]	-	97.5
SV-DSIFT [Takahiko Furuya, 2015]	-	97.2
R-BiHDM-L23 [Ye and Yu, 2015]	0.7300	-
SGWC-BoF [Masoumi et al., 2016]	0.7290	-
SV-LSF_kpaca50 [Takahiko Furuya, 2015]	-	96.2
Geodesic Distances (LS) [Pickup et al., 2015]	0.7170	-
MDS-CM-BOF	0.7166	95.0
OrigM-n12-normA	0.7047	94.4
FOG+MRR	0.6958	91.8
ConTopo++ [Sfikas et al., 2011]	0.6950	94.7
PCDLaplace-SVIWKS (point-cloud dataset)	0.6844	90.1
BOGH	0.6469	86.7
LSF	0.6327	85.1
PBMHT-FVIWKS (point-cloud dataset)	0.4682	67.5

At the last two columns it is shown the total time to compute each of the benchmarks, using either FV or SV. As already said, FV is only faster than SV because we use VLfeat implementation [Vedaldi and Fulkerson, 2008] while SV is implemented in Matlab.

7.6 Summary

In this chapter, we have presented a new method for computing the Laplacian function of the underlying manifold of a point cloud shape. The Mesh-Free Laplace Operator is a method

Table 7.14: Retrieval performance comparison with state-of-the-art signatures on SHREC'15 sorted by mAP. We show best runs from the six groups that performed better in SHREC'15 (SV-LSF_kpaca50, HAPT_run1, SPH_SparseCoding_1024, CompactBoHHKS10D, FV-WKS, EDBCF_NW) and other descriptors in the literature against our descriptors. In bold are highlighted the best performances for each retrieval measure.

Descriptor	e-Measure	mAP [%]
SV-LSF_kpaca50 [Takahiko Furuya, 2015]	**0.8357**	**99.8**
KLBO-FVIWKS	0.8269	99.2
MFLO-FVIWKS (point-cloud dataset)	0.8221	98.8
WGL-FVIWKS (point-cloud dataset)	0.8224	98.6
GL-SVIWKS (point-cloud dataset)	0.8159	97.9
HAPT_run1	0.8150	97.7
FV-IWKS [Limberger and Wilson, 2015]	0.8102	96.9
SPH_SparseCoding_1024	0.8047	96.8
PCDLaplace-SVIWKS (point-cloud dataset)	0.7702	93.2
SGWC-BoF [Masoumi et al., 2016]	0.7470	-
CompactBoHHKS10D	0.7465	90.1
SRG [Mohamed and Hamza, 2016]	0.7390	-
FV-WKS	0.7242	87.5
EDBCF_NW	0.7076	85.0
PBMHT-FVIWKS (point-cloud dataset)	0.4706	57.8

which can describe precisely the characteristics of shapes, therefore it is useful for creating signatures for 3D models represented by point clouds. It is based in a robust weighting scheme and a localized edge connection scheme which preserves accuracy. The weighting is considered robust because it works in different sorts of standard shapes as showed in the course of this Chapter.

Convergence experiments show that the MFLO converges to the Laplace-Beltrami operator as we increase the number of samples of the models. This is true at least for the rectangle and sphere point clouds tested in this thesis with isotropic sampling.

We have also made experiments that show the gain in performance in most recent point-cloud benchmarks by using the MFLO instead of other Laplacian methods. In most of the benchmarks, the MFLO performs better than GL, WGL, PCDLaplace and PBMHT, using the same parameters for all methods in all benchmarks. We have also identified that its is necessary to create more challenging benchmarks since the best methods are already receiving

very good scores. The performance of the WGL is close to the MFLO for retrieving non-rigid shapes because when the sampling is dense enough to represent the underlying manifold both methods have similar errors, as stated in Section 7.4.1.

Furthermore, we have shown experiments on shapes by varying the number of samples of the point clouds. We show that, as long we increase the number of nearest neighbors of the k-NNG while we increase the number of samples, the performance of the MFLO is preserved, since more connections are necessary to cover the same area.

Chapter 8

Conclusion

This chapter summarizes the main contributions of this thesis as well as it discusses limitations and possible directions for future research.

8.1 Contributions

Our goal in this thesis was to construct state-of-the-art spectral descriptors for non-rigid shape retrieval. We begun by proposing an improved version of the wave kernel signature which is more informative and discriminative. We have also presented a framework for computing global signatures from local spectral descriptors. Then, we presented improved Laplace operators for both shape representations: meshes and point clouds. Furthermore, we proposed a non-rigid point-cloud benchmark which aims in evaluating point-cloud descriptors. We now summarize each of the contributions separately.

In this thesis, we first proposed a new spectral descriptor which is based on the WKS but has some advantages for non-rigid shape retrieval. We proposed a new power scaling, replacing the logarithmic scale, which better balance the eigenfunctions of the Laplace-Beltrami operator and creates a more informative descriptor. Moreover, we also aggregate principal curvatures to the descriptor in order to differentiate peculiar topologies of the shape, creating this time a more discriminative descriptor.

After, we proposed to use state-of-the-art encoding schemes with spectral signatures. We have shown that it is possible to use these methods since we can approximate local

spectral features with Gaussian mixture models even when represented by a few components. By describing these features with probabilistic distribution functions and also creating a dictionary of features, we were able to generate a set of statistics that represent the global shape signature by computing gradients of the local features in respect to the vocabulary. In the end, we show that the Fisher Vector and the Super Vector can increase the performance of shape retrieval combined with a meaningful distance computation.

Next, we decided to take a step forward and create an even better descriptor for non-rigid shape retrieval by undertaken a more fundamental approach. This way, we focused in the computation of the shape spectra to propose the Kinetic Laplace-Beltrami operator. It is based on a modification to the dynamic systems on the mesh (kinetic energy). By introducing a new curvature-based kinetic term, we were able to improve significantly the retrieval performance of spectral descriptors. This term naturally downweights the importance of joints in the descriptor and consequently improves non-rigid shape retrieval performance.

From this point, we started working with point-cloud shape representations. First, we identified a lack of benchmarks in this area. Therefore, to close this gap, we created a non-rigid point-cloud shape benchmark for shape retrieval. This benchmark aims testing shape signatures that are computed directly and efficiently from point clouds. The dataset was released in the 3D Object Retrieval Workshop in Eurographics 2017 with participation of 8 groups. Shape retrieval statistics were computed for each dissimilarity matrix submitted by the participants. In total 31 dissimilarity matrices were submitted. The evaluation results shown by this work suggest that researchers are moving in the right direction towards shape descriptors which can capture the main characteristics of 3D models, however, more tests still need to be made, since this was the first time we compared directly non-rigid signatures for point-cloud shape retrieval.

Our final contribution in this thesis was the the Mesh-Free Laplace operator (MFLO). The novelty here is the new way we compute the Laplace-Beltrami operator of point clouds. By using an isotropic edge-weighting scheme that takes into account the underlying manifold structure, we are able to characterize features on the shape more precisely in respect to the unknown manifold. Then, by computing spectral descriptors using the eigenvalues and eigenfunctions of the MFLO, we were able to achieve better performances than shape

descriptors that use other Laplacian-matrix formulations.

By combining either the KLBO or the MFLO with spectral signatures and computing robust distances between descriptors we clearly outperform the state-of-the-art in three recent non-rigid benchmarks [Lian et al., 2011, Lian et al., 2015, Biasotti et al., 2017]. We show that our methods are consistent over different kinds of data since they achieve good overall retrieval performances in all recent non-rigid databases. They are also as fast to compute as any other spectral method and straightforward to implement.

8.2 Limitations and future work

Although the methods proposed in this thesis outperform the state-of-the-art methods, they still have some limitations that need to be taken into consideration. In this section, we discuss these limitations and point possible directions for future research that can overcome some of these drawbacks.

The first limitation is related to all spectral analysis methods. Spectral descriptors, which depend on the computation of the eigendecomposition of the Laplace-Beltrami operator, can be very time consuming tasks once the size of the meshes or point clouds increase. The eigendecomposition is performed on a $V \times V$ matrix (L), where V is the number of vertices of the model. In the worst case scenario, the method has a cubic time complexity ($O(n^3)$). As the number of vertices hit the five digits (10,000) the computation time starts to increase considerably. However, this limitation is not an exclusivity of our methods. All spectral methods share this drawback. Possible ways of overcoming this situation include downsampling the model into fewer vertices and computing a more efficient eigendecomposition of the LBO (taking advantage of the fact that L is sparse).

One advantage of the Improved Wave Kernel Signature is that it was proposed to be more informative, *i.e.*, having more information about high spectrum frequencies. However, we have observed that when shapes are not actually the same, like mentioned in Chapter 5 about SHREC'10 [Lian et al., 2010], our signature fails in identifying shape classes because it gives more importance to fine details than the HKS and the WKS. Nonetheless, when models are the same and preserve fine details like in SHREC'11 [Lian et al., 2011] and SHREC'15

[Lian et al., 2015] our method can accurately distinguish shape classes. As future work, it would be interesting to create a spectral shape descriptor which can adaptively modifies the scaling of the spectrum filters for creating specific signatures for the problem at hand, similar to the method of [Litman and Bronstein, 2014] for shape correspondences. Also, it would be interesting to investigate how to compute spectral signatures for the comprehensive shape retrieval task.

Since the spectral descriptors are based on smooth filters of the shape spectra, some important information can be lost in order to account for the possible noisy surface. However, when the differences between classes being analyzed are very small, like on SHREC'14 [Pickup et al., 2014b], where all classes consist of human models, the spectral signatures are not very good at detecting the crucial differences between these classes. In this scenario, methods that learn from the data will perform better since it is very hard to code these small differences without prior knowledge of the classes.

In respect to the MFLO, it would be interesting to aggregate some of the KLBO ideas in its formulation, like the curvature weighting for improving the retrieval performance of non-rigid models. Analogously, computing the KLBO for point clouds is another possible direction of research.

Generally, in spectral shape retrieval, it should be interesting to test machine learning techniques, like it has been done in [Litman and Bronstein, 2014] and [Aflalo et al., 2012] where shape correspondences were the focus of these researches. By using the shape spectra to create global signatures the training can be very costly, depending on the encoding framework. This way, one has to design an efficient and discriminative framework to deal with this problem.

Furthermore, we have identified a lack of challenging benchmarks in the area of non-rigid shape retrieval (meshes and point clouds). We have created a new challenging benchmark for point-cloud non-rigid shape retrieval, however, one method had a very good performance on the data. We will continue our efforts to inspire others to create even more challenging benchmarks in the future for non-rigid shape retrieval.

Finally, the methodologies developed in this thesis can be applied to other fields of research. The KLBO can be modified to define other weightings on the graph and emphasize

other characteristics for a number of graph problems, for example in the fields of biology (brain, molecules), sociology (social networks) or communication (computer networks).

List of Symbols

A	Adjacency matrix
D	Degree matrix
W	Weighting adjacency matrix
$\mathcal{L}$	Manifold Laplacian
L	Discrete Laplacian
Δ	Laplacian operator
λ	Eigenvalues
ϕ	Eigenvectors
K	Size of the dictionary
k_1	Maximum principal curvature
k_2	Minimum principal curvature
T	Sample size
μ	Mean of points
G	Graph
H	Mean curvature
II	Second fundamental form matrix
V	Vertex set of a graph
E	Edge set of a graph
$C(\mathbf{x})$	Curvature function of vertice $\mathbf{x}$

Abbreviations

1SIFT	One Scale-Invariant Feature Transform
3D	Three-dimensional
BoF	Bag-of-Feature
DCG	Discounted cumulative gain
DSIFT	Dense Scale Invariant Feature Transform
E	E-measure
EM	Expectation Maximization
EUC	Euclidean distance
FT	First Tier
FV	Fisher Vector
GL	Graph Laplacian
GMM	Gaussian Mixture model
GPS	Global Point Signature
HKS	Heat Kernel Signature
IWKS	Improved Wave Kernel Signature
KLBO	Kinetic Laplace-Beltrami operator
LBO	Laplace-Beltrami operator
LL	Locality-constrained linear coding
mAP	mean Average Precision
MDS	Multidimensional scaling
MFLO	Mesh-Free Laplace Operator

MR	Manifold Ranking
MSB	McGill 3D Shape Benchmark
NN	Nearest neighbour
PB-MHT	Point-based Manifold Harmonic Transform
PR	Precision and Recall
PCDLaplace	Point-Cloud Data Laplace
PRoNTo	Point-Cloud Retrieval of Non-rigid Toys
RGB	Red, blue and green
SC	Shape Context
SHREC	Shape Retrieval Contest
SIFT	Scale-Invariant Feature Transform
SIHKS	Scale-invariant Heat Kernel Signature
ST	Second Tier
SURF	Speeded up robust features
SSIFT	Salient Scale-Invariant Feature Transform
SV	Super Vector
VQ	Vector Quantization
WGL	Weighted Graph Laplacian
WKS	Wave Kernel Signature

List of References

[Aflalo et al., 2012] Aflalo, Y., Bronstein, Alexander M. nd Bronstein, M. M., and Kimmel, R. (2012). *Deformable Shape Retrieval by Learning Diffusion Kernels*, pages 689–700. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Amenta and Bern, 1998] Amenta, N. and Bern, M. (1998). Surface reconstruction by voronoi filtering. In *Proceedings of the Fourteenth Annual Symposium on Computational Geometry*, SCG '98, pages 39–48, New York, NY, USA. ACM.

[Amenta et al., 2000] Amenta, N., Choi, S., Dey, T. K., and Leekha, N. (2000). A simple algorithm for homeomorphic surface reconstruction. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, SCG '00, pages 213–222, New York, NY, USA. ACM.

[Andreux et al., 2014] Andreux, M., Rodola, E., Aubry, M., and Cremers, D. (2014). Anisotropic laplace-beltrami operators for shape analysis. In *Sixth Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*.

[Aubry et al., 2011a] Aubry, M., Schlickewei, U., and Cremers, D. (2011a). Pose-consistent 3d shape segmentation based on a quantum mechanical feature descriptor. In *Proceedings of the 33rd International Conference on Pattern Recognition*, DAGM'11, pages 122–131, Berlin, Heidelberg. Springer-Verlag.

[Aubry et al., 2011b] Aubry, M., Schlickewei, U., and Cremers, D. (2011b). The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), IEEE International Conference on*, pages 1626–1633.

[Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):346–359.

[Belkin and Niyogi, 2008] Belkin, M. and Niyogi, P. (2008). Towards a theoretical foundation for laplacian-based manifold methods. *J. Comput. Syst. Sci.*, 74(8):1289–1308.

[Belkin et al., 2008] Belkin, M., Sun, J., and Wang, Y. (2008). Discrete laplace operator on meshed surfaces. In *Proceedings of the Twenty-fourth Annual Symposium on Computational Geometry*, SCG '08, pages 278–287, New York, NY, USA. ACM.

[Belkin et al., 2009] Belkin, M., Sun, J., and Wang, Y. (2009). Constructing laplace operator from point clouds in Rd. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 1031–1040, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

[Belongie and Malik, 2000] Belongie, S. and Malik, J. (2000). Matching with shape contexts. In *Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on*, pages 20–26.

[Belongie et al., 2000] Belongie, S., Malik, J., and Puzicha, J. (2000). Shape context: A new descriptor for shape matching and object recognition. In *Neural Information Processing Systems*, pages 831–837.

[Biasotti et al., 2014] Biasotti, S., Cerri, A., Abdelrahman, M., Aono, M., Hamza, A. B., El-Melegy, M. T., Farag, A. A., Garro, V., Giachetti, A., Giorgi, D., Godil, A., Li, C., Liu, Y., Martono, H. Y., Sanada, C., Tatsuma, A., Velasco-Forero, S., and Xu, C. (2014). Retrieval and classification on textured 3d models. In *3DOR 2014*, pages 111–120.

[Biasotti et al., 2017] Biasotti, S., Thompson, E. M., Aono, M., Hamza, A. B., Bustos, B., Dong, S., Du, B., Fehri, A., Li, H., Limberger, F. A., Masoumi, M., Rezaei, M., Sipiran, I., Sun, L., Tatsuma, A., Forero, S. V., Wilson, R. C., Wu, Y., Zhang, J., Zhao, T., Fornasa, F., and Giachetti, A. (2017). Shrec17 track: Retrieval of surfaces with similar relief patterns. In *10th Eurographics workshop on 3D Object retrieval*, Lyon, France.

[Boscaini et al., 2015] Boscaini, D., Masci, J., Melzi, S., Bronstein, M. M., Castellani, U., and Vandergheynst, P. (2015). Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. In *Proceedings of the Eurographics Symposium on Geometry Processing*, SGP '15, pages 13–23. Eurographics Association.

[Boscaini et al., 2016] Boscaini, D., Masci, J., Rodol, E., Bronstein, M. M., and Cremers, D. (2016). Anisotropic Diffusion Descriptors. *Computer Graphics Forum*.

[Bronstein et al., 2010a] Bronstein, A. M., Bronstein, M. M., Castellani, U., Dubrovina, A., Guibas, L. J., Horaud, R. P., Kimmel, R., Knossow, D., von Lavante, E., Mateus, D., Ovsjanikov, M., and Sharma, A. (2010a). SHREC 2010: robust correspondence benchmark. In *3DOR 2010*.

[Bronstein et al., 2010b] Bronstein, A. M., Bronstein, M. M., Castellani, U., Falcidieno, B., Fusiello, A., and Godil, A. (2010b). SHREC 2010 : robust large-scale shape retrieval benchmark. *3DOR 2010*.

[Bronstein et al., 2011] Bronstein, A. M., Bronstein, M. M., Guibas, L. J., and Ovsjanikov, M. (2011). Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Trans. Graph.*, 30(1):1:1–1:20.

[Bronstein et al., 2006] Bronstein, A. M., Bronstein, M. M., and Kimmel, R. (2006). Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comput.*, 28(5):1812–1836.

[Bronstein and Kokkinos, 2010] Bronstein, M. and Kokkinos, I. (2010). Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1704–1711.

[Castellani et al., 2011] Castellani, U., Mirtuono, P., Murino, V., Bellani, M., Rambaldelli, G., Tansella, M., and Brambilla, P. (2011). *A New Shape Diffusion Descriptor for Brain Classification*, pages 426–433. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Chatfield et al., 2011] Chatfield, K., Lempitsky, V. S., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods.

In *British Machine Vision Conference, BMVC 2011, Dundee, UK, 2011. Proceedings*, pages 1–12.

[Chen et al., 2009] Chen, X., Golovinskiy, A., and Funkhouser, T. (2009). A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3).

[Coifman and Lafon, 2006] Coifman, R. R. and Lafon, S. (2006). Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30.

[Corsini et al., 2012] Corsini, M., Cignoni, P., and Scopigno, R. (2012). Efficient and flexible sampling with blue noise properties of triangular meshes. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):914–924.

[Csurka and Perronnin, 2011] Csurka, G. and Perronnin, F. (2011). Fisher vectors: Beyond bag-of-visual-words image representations. In Richard, P. and Braz, J., editors, *Computer Vision, Imaging and Computer Graphics. Theory and Applications*, volume 229 of *Communications in Computer and Information Science*, pages 28–42. Springer Berlin Heidelberg.

[Desbrun et al., 1999] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. (1999). Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 317–324, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.

[Elad and Kimmel, 2003] Elad, A. and Kimmel, R. (2003). On bending invariant signatures for surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1285–1295.

[Flitton et al., 2010] Flitton, G., Breckon, T., and Megherbi Bouallagu, N. (2010). Object recognition using 3d sift in complex ct volumes. In *Proceedings of the British Machine Vision Conference*, pages 11.1–11.12. BMVA Press. doi:10.5244/C.24.11.

[Furuya and Ohbuchi, 2009] Furuya, T. and Ohbuchi, R. (2009). Dense sampling and fast encoding for 3d model retrieval using bag-of-visual features. In *Proceedings of the ACM*

International Conference on Image and Video Retrieval, CIVR '09, pages 26:1–26:8, New York, NY, USA. ACM.

[Furuya and Ohbuchi, 2013] Furuya, T. and Ohbuchi, R. (2013). Ranking on cross-domain manifold for sketch-based 3d model retrieval. In *2013 International Conference on Cyberworlds, Yokohama, Japan, October 21-23, 2013*, pages 274–281.

[Furuya and Ohbuchi, 2014] Furuya, T. and Ohbuchi, R. (2014). Fusing multiple features for shape-based 3d model retrieval. In *Proceedings of the British Machine Vision Conference*. BMVA Press.

[Gasparetto et al., 2015] Gasparetto, A., Minello, G., and Torsello, A. (2015). Non-parametric spectral model for shape retrieval. In *2015 International Conference on 3D Vision*, pages 344–352.

[Giachetti et al., 2016] Giachetti, A., Isaia, L., and Garro, V. (2016). Multiscale descriptors and metric learning for human body shape retrieval. *The Visual Computer*, 32(6):693–703.

[Giachetti and Lovato, 2012] Giachetti, A. and Lovato, C. (2012). Radial Symmetry Detection and Shape Characterization with the Multiscale Area Projection Transform. *Computer Graphics Forum*.

[Google, 2015] Google (2015). Google 3D Warehouse. https://3dwarehouse.sketchup.com/.

[Grigor'yan, 2006] Grigor'yan, A. (2006). Heat kernels on weighted manifolds and applications.

[Jaakkola and Haussler, 1998] Jaakkola, T. S. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 487–493. MIT Press.

[Johnson, 1997] Johnson, A. (1997). *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

[Knopp et al., 2010] Knopp, J., Prasad, M., Willems, G., Timofte, R., and Van Gool, L. (2010). Hough transform and 3d surf for robust three dimensional classification. In

Proceedings of the 11th European Conference on Computer Vision: Part VI, ECCV'10, pages 589–602, Berlin, Heidelberg. Springer-Verlag.

[Kokkinos et al., 2012] Kokkinos, I., Bronstein, M., Litman, R., and Bronstein, A. (2012). Intrinsic shape context descriptors for deformable shapes. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 159–166.

[Lafon, 2004] Lafon, S. (2004). *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University.

[Laptev et al., 2007] Laptev, I., Caputo, B., Schüldt, C., and Lindeberg, T. (2007). Local velocity-adapted motion events for spatio-temporal recognition. *Comput. Vis. Image Underst.*, 108(3):207–229.

[Laptev and Lindeberg, 2006] Laptev, I. and Lindeberg, T. (2006). Local descriptors for spatio-temporal recognition. In *Proceedings of the First International Conference on Spatial Coherence for Visual Motion Analysis*, SCVMA'04, pages 91–103, Berlin, Heidelberg. Springer-Verlag.

[Levy, 2006] Levy, B. (2006). Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*, SMI '06, page 13, Washington, DC, USA. IEEE Computer Society.

[Li et al., 2012] Li, B., Godil, A., and et al. (2012). Shrec'12 track: Generic 3d shape retrieval. In *Proceedings of the 5th Eurographics Conference on 3D Object Retrieval*, EG 3DOR'12, pages 119–126. Eurographics Assoc.

[Li et al., 2014a] Li, B., Godil, A., and Johan, H. (2014a). Hybrid shape descriptor and meta similarity generation for non-rigid and partial 3d model retrieval. *Multimedia Tools and Applications*, 72(2):1531–1560.

[Li et al., 2014b] Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., Aono, M., Fu, H., Furuya, T., Johan, H., Liu, J., Ohbuchi, R., Tatsuma, A., and Zou, C. (2014b). SHREC'14 track:

Extended large scale sketch-based 3D shape retrieval. In *Proc. EG Workshop on 3D Object Retrieval, 2014 (to appear)*.

[Li et al., 2014c] Li, B., Lu, Y., Li, C., Godil, A., Schreck, T., and et al. (2014c). Large scale comprehensive 3D shape retrieval. In *3DOR 2014*, pages 131–140.

[Li, 2013] Li, C. (2013). Spectral geometric methods for deformable 3d shape retrieval.

[Li and BenHamza, 2013] Li, C. and BenHamza, A. (2013). A multiresolution descriptor for deformable 3d shape retrieval. *The Visual Computer*, 29(6):513–524.

[Li et al., 2016] Li, P., Ma, H., and Ming, A. (2016). A non-rigid 3d model retrieval method based on scale-invariant heat kernel signature features. *Multimedia Tools and Applications*, pages 1–24.

[Lian et al., 2010] Lian, Z., Godil, A., and et al. (2010). SHREC'10 Track: Non-rigid 3D Shape Retrieval. In Daoudi, M. and Schreck, T., editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Assoc.

[Lian et al., 2011] Lian, Z., Godil, A., and et al. (2011). SHREC'11 track: shape retrieval on non-rigid 3D watertight meshes. In *3DOR 2011*, pages 79–88. Eurographics Assoc.

[Lian et al., 2013] Lian, Z., Godil, A., Sun, X., and Xiao, J. (2013). CM-BOF: visual similarity-based 3d shape retrieval using clock matching and bag-of-features. *Machine Vision and Applications*, 24(8):1685–1704.

[Lian et al., 2015] Lian, Z., Zhang, J., and et al. (2015). Non-rigid 3D Shape Retrieval. In *3DOR 2015*. The Eurographics Assoc.

[Limberger and Wilson, 2015] Limberger, F. A. and Wilson, R. C. (2015). Feature encoding of spectral signatures for 3D non-rigid shape retrieval. In *Proceedings of the British Machine Vision Conference*, pages 56.1–56.13. BMVA Press.

[Limberger and Wilson, 2017] Limberger, F. A. and Wilson, R. C. (2017). SHREC'17 - Point-Cloud Shape Retrieval of Non-Rigid Toys. URL: https://www.cs.york.ac.uk/cvpr/pronto/.

[Limberger et al., 2017] Limberger, F. A., Wilson, R. C., Aono, M., Audebert, N., Boulch, A., Bustos, B., Giachetti, A., Godil, A., Saux, B. L., Li, B., Lu, Y., Nguyen, H.-D., Nguyen, V.-T., Pham, V.-K., Sipiran, I., Tatsuma, A., Tran, M.-T., and Velasco-Forero., S. (2017). SHREC'17 Track: Point-Cloud Shape Retrieval of Non-Rigid Toys. In *10th Eurographics workshop on 3D Object retrieval*, pages 1 – 11, Lyon, France.

[Litman et al., 2014] Litman, R., Bronstein, A., Bronstein, M., and Castellani, U. (2014). Supervised learning of bag-of-features shape descriptors using sparse coding. *Symposium on Geometry Processing (SGP)*.

[Litman and Bronstein, 2014] Litman, R. and Bronstein, A. M. (2014). Learning spectral descriptors for deformable shape correspondence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(1):171–180.

[Liu et al., 2012] Liu, Y., Prabhakaran, B., and Guo, X. (2012). Point-based manifold harmonics. *IEEE Transactions on Visualization and Computer Graphics*, 18(10):1693–1703.

[Lowe, 1999] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA. IEEE Computer Society.

[Luo et al., 2009] Luo, C., Sun, J., and Wang, Y. (2009). Integral estimation from point cloud in d-dimensional space: A geometric view. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*, SCG '09, pages 116–124. ACM.

[Maalej et al., 2011] Maalej, A., Amor, B. B., Daoudi, M., Srivastava, A., and Berretti, S. (2011). Shape analysis of local facial patches for 3d facial expression recognition. *Pattern Recognition*, 44(8):1581 – 1589.

[Masoumi et al., 2016] Masoumi, M., Li, C., and Hamza, A. B. (2016). A spectral graph wavelet approach for nonrigid 3d shape retrieval. *Pattern Recognition Letters*.

[Meyer et al., 2003] Meyer, M., Desbrun, M., Schrder, P., and Barr, A. (2003). Discrete differential-geometry operators for triangulated 2-manifolds. In Hege, H.-C. and Polthier,

K., editors, *Visualization and Mathematics III*, Mathematics and Visualization, pages 35–57. Springer Berlin Heidelberg.

[Mohamed and Hamza, 2016] Mohamed, W. and Hamza, A. B. (2016). Deformable 3d shape retrieval using a spectral geometric descriptor. *Applied Intelligence*, pages 1–17.

[Ohbuchi and Furuya, 2010] Ohbuchi, R. and Furuya, T. (2010). Distance metric learning and feature combination for shape-based 3d model retrieval. In *Proceedings of the ACM Workshop on 3D Object Retrieval*, 3DOR '10, pages 63–68, New York, NY, USA. ACM.

[Ohbuchi et al., 2008] Ohbuchi, R., Osada, K., Furuya, T., and Banno, T. (2008). Salient local visual features for shape-based 3d model retrieval. In *2008 IEEE International Conference on Shape Modeling and Applications*, pages 93–102.

[Osada et al., 2002] Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D. (2002). Shape distributions. *ACM Trans. Graph.*, 21(4):807–832.

[Ovsjanikov et al., 2009] Ovsjanikov, M., Bronstein, A., Bronstein, M., and Guibas, L. (2009). Shape google: a computer vision approach to isometry invariant shape retrieval. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 320–327.

[Perronnin and Dance, 2007] Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition. IEEE Conference on*, pages 1–8.

[Perronnin et al., 2010a] Perronnin, F., Liu, Y., Sanchez, J., and Poirier, H. (2010a). Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391.

[Perronnin et al., 2010b] Perronnin, F., Sánchez, J., and Mensink, T. (2010b). Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 143–156, Berlin, Heidelberg. Springer-Verlag.

[Pickup et al., 2015] Pickup, D., Sun, X., Rosin, P. L., and Martin, R. R. (2015). Euclidean-distance-based canonical forms for non-rigid 3d shape retrieval. *Pattern Recognition*, 48(8):2500 – 2512.

[Pickup et al., 2016a] Pickup, D., Sun, X., Rosin, P. L., and Martin, R. R. (2016a). Skeleton-based canonical forms for non-rigid 3d shape retrieval. *Computational Visual Media*, pages 1–13.

[Pickup et al., 2014a] Pickup, D., Sun, X., Rosin, P. L., Martin, R. R., Cheng, Z., Lian, Z., Aono, M., Hamza, A. B., Bronstein, A. M., Bronstein, M. M., Bu, S., Castellani, U., Cheng, S., Garro, V., Giachetti, A., Godil, A., Han, J., Johan, H., Lai, L., Li, B., Li, C., Li, H., Litman, R., Liu, X., Liu, Z., Lu, Y., Tatsuma, A., and Ye, J. (2014a). Shape retrieval of non-rigid 3d human models. In *Eurographics Workshop on 3D Object Retrieval, Strasbourg, France, 2014. Proceedings*, pages 101–110.

[Pickup et al., 2014b] Pickup, D., Sun, X., Rosin, P. L., Martin, R. R., Cheng, Z., Lian, Z., and et al. (2014b). SHREC'14 track: Shape Retrieval of Non-Rigid 3D Human Models. In *Proceedings of the 7th Eurographics workshop on 3D Object Retrieval*, EG 3DOR'14. Eurographics Assoc.

[Pickup et al., 2016b] Pickup, D., Sun, X., Rosin, P. L., Martin, R. R., Cheng, Z., Lian, Z., and et al. (2016b). Shape retrieval of non-rigid 3d human models. *International Journal of Computer Vision*, 120(2):169–193.

[Pinkall et al., 1993] Pinkall, U., Juni, S. D., and Polthier, K. (1993). Computing discrete minimal surfaces and their conjugates. *Experimental Mathematics*, 2:15–36.

[Raviv et al., 2010] Raviv, D., Bronstein, M. M., Bronstein, A. M., and Kimmel, R. (2010). Volumetric heat kernel signatures. In *Proceedings of the ACM Workshop on 3D Object Retrieval*, 3DOR '10, pages 39–44, New York, NY, USA. ACM.

[Reuter et al., 2005] Reuter, M., Wolter, F.-E., and Peinecke, N. (2005). Laplace-spectra as fingerprints for shape matching. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, SPM '05, pages 101–106, New York, NY, USA. ACM.

[Reuter et al., 2006] Reuter, M., Wolter, F.-E., and Peinecke, N. (2006). Laplace-beltrami spectra as 'shape-dna' of surfaces and solids. *Comput. Aided Des.*, 38(4):342–366.

[Rusinkiewicz, 2004] Rusinkiewicz, S. (2004). Estimating curvatures and their derivatives on triangle meshes. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2Nd International Symposium*, 3DPVT '04, pages 486–493, Washington, DC, USA. IEEE Computer Society.

[Rustamov, 2007] Rustamov, R. M. (2007). Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 225–233. Eurographics Assoc.

[Samir et al., 2006] Samir, C., Srivastava, A., and Daoudi, M. (2006). Three-dimensional face recognition using shapes of facial curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1858–1863.

[Sanchez et al., 2013] Sanchez, J., Perronnin, F., Mensink, T., and Verbeek, J. (2013). Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision*, 105(3):222–245.

[Savva et al., 2016] Savva, M., Yu, F., Su, H., Aono, and et al. (2016). Large-Scale 3D Shape Retrieval from ShapeNet Core55. In Ferreira, A., Giachetti, A., and Giorgi, D., editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association.

[Schneider and Tuytelaars, 2014] Schneider, R. G. and Tuytelaars, T. (2014). Sketch classification and classification-driven analysis using fisher vectors. *ACM Trans. Graph.*, 33(6):174:1–174:9.

[Scovanner et al., 2007] Scovanner, P., Ali, S., and Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th International Conference on Multimedia*, MULTIMEDIA '07, pages 357–360, New York, NY, USA. ACM.

[Sfikas et al., 2011] Sfikas, K., Pratikakis, I., and Theoharis, T. (2011). Contopo: Non-rigid 3d object retrieval using topological information guided by conformal factors. In

Proceedings of the 4th Eurographics Conference on 3D Object Retrieval, 3DOR '11, pages 25–32. Eurographics Assoc.

[Shilane et al., 2004] Shilane, P., Min, P., Kazhdan, M., and Funkhouser, T. (2004). The princeton shape benchmark. In *Proceedings of the Shape Modeling International 2004*, SMI '04, pages 167–178, Washington, DC, USA. IEEE Computer Society.

[Siddiqi et al., 2008] Siddiqi, K., Zhang, J., Macrini, D., Shokoufandeh, A., Bouix, S., and Dickinson, S. (2008). Retrieving articulated 3-d models using medial surfaces. *Mach. Vision Appl.*, 19(4):261–275.

[Simonyan et al., 2013] Simonyan, K., Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2013). Fisher Vector Faces in the Wild. In *British Machine Vision Conference*.

[Sinkhorn and Knopp, 1967] Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific J. Math.*, 21(2):343–348.

[Sivic and Zisserman, 2003] Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 1470–1477.

[Sun et al., 2009] Sun, J., Ovsjanikov, M., and Guibas, L. (2009). A concise and provably informative multi-scale signature based on heat diffusion. In *Proceedings of the Symposium on Geometry Processing*, SGP '09, pages 1383–1392. Eurographics Association.

[Takahiko Furuya, 2015] Takahiko Furuya, R. O. (2015). Diffusion-on-manifold aggregation of local features for shape-based 3d model retrieval. *Proc. ACM International Conference on Multimedia Information Retrieval*.

[Taubin, 1995] Taubin, G. (1995). A signal processing approach to fair surface design. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 351–358, New York, NY, USA. ACM.

[TOSCA, 2009] TOSCA (2009). Project TOSCA. http://tosca.cs.technion.ac.il/. Accessed: 2016-11-02.

[Vedaldi and Fulkerson, 2008] Vedaldi, A. and Fulkerson, B. (2008). VLFeat: An open and portable library of comp. vision algorithms. http://www.vlfeat.org/.

[Vohra et al., 2002] Vohra, N., Vemuri, B., Rangarajan, A., Gilmore, R., Roper, S., and Leonard, C. (2002). Kernel fisher for shape based classification in epilepsy. In Dohi, T. and Kikinis, R., editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2002*, volume 2489 of *Lecture Notes in Computer Science*, pages 436–443.

[Wahl et al., 2003] Wahl, E., Hillenbrand, U., and Hirzinger, G. (2003). Surflet-pair-relation histograms: A statistical 3D-shape representation for rapid classification. In *Proceedings of the 4th International Conference on 3D Digital Imaging and Modeling*, 3DIM '03, pages 474–482.

[Wang et al., 2010] Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367.

[Xu et al., 2011] Xu, B., Bu, J., Chen, C., Cai, D., He, X., Liu, W., and Luo, J. (2011). Efficient manifold ranking for image retrieval. In *Proceedings of the 34th Annual International ACM SIGIR Conference (SIGIR'11)*.

[Xu, 2004] Xu, G. (2004). Discrete laplace–beltrami operators and their convergence. *Comput. Aided Geom. Des.*, 21(8):767–784.

[Xu, 2006] Xu, G. (2006). Discrete laplace-beltrami operator on sphere and optimal spherical triangulations. *Int. J. Comput. Geometry Appl*, 16:75–93.

[Ye et al., 2013] Ye, J., Yan, Z., and Yu, Y. (2013). Fast nonrigid 3d retrieval using modal space transform. In *Proceedings of the 3rd ACM Conference on International Conference on Multimedia Retrieval*, ICMR '13, pages 121–126, New York, NY, USA. ACM.

[Ye and Yu, 2015] Ye, J. and Yu, Y. (2015). A fast modal space transform for robust nonrigid shape retrieval. *The Visual Computer*, pages 1–16.

[Zhou et al., 2004] Zhou, D., Weston, J., Gretton, A., Bousquet, O., and Schölkopf, B. (2004). Ranking on data manifolds. In Thrun, S., Saul, L. K., and Schölkopf, B., editors, *Advances in Neural Information Processing Systems 16*, pages 169–176. MIT Press.

[Zhou et al., 2010] Zhou, X., Yu, K., Zhang, T., and Huang, T. S. (2010). Image classification using super-vector coding of local image descriptors. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pages 141–154, Berlin. Springer-Verlag.