Technische Universität München

TUM School of Computation, Information and Technology

Deformable Shape Correspondence Using Pointwise and Pairwise Features

Matthias Vestner

Vollständiger Abdruck der von der TUM School of Computation, Information and Technology der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr. Nils Thürey
Prüfer*innen der Dissertation:	1. Prof. Dr. Daniel Cremers
	2. Assoc. Prof. Dr. Justin Solomon
	3. Prof. Dr. Emanuele Rodolà

Die Dissertation wurde am 24.05.2022 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 27.01.2023 angenommen.

Abstract

This thesis investigates the problem of analyzing digitized threedimensional objects undergoing non-rigid deformations. The main focus lies on finding semantically meaningful dense correspondences between two triangle meshes. With these correspondences it is possible to transfer knowledge from one to the other, for instance a segmentation of a human into bodyparts from a template mesh to a mesh obtained from a 3D reconstruction pipeline.
We give an extensive introduction on the theory of non-rigid shape analysis, again with the focus on correspondences. In particular we introduce fundamental concepts of differential geometry and how to transfer them to shapes discretized as triangle meshes. We dedicate a compehensive section to make the reader familiar with the *Laplace Beltrami operator* (LBO). In a tutorial at SGP in 2014 the Laplace Beltrami operator was once referred to as the *Swiss army knife of geometry processing*. From pointwise feature descriptors that can describe shapes on different levels of locality over pairwise descriptors up to compressed representations of correspondences between shapes, a variety of tooling can be derived from the LBO which makes it a fundamental tool for non-rigid shape analysis as well. We introduce the LBO as a generalization of the well known Laplace operator on Euclidean domains, discuss its main properties and show that it's eigenfunctions can be seen as generalizations of *Fourier basis functions* and spherical harmonics. We further provide a detailed derivation of a discretization of the LBO and discuss possible alternatives. We revisit the rigid alignment of pointsets and introduce its non-rigid generalization of finding non-Euclidean isometries between shapes. Permutation matrices as discrete representations of bijective mappings are introduced and their practical limitations (and possible remedies) in the context of optimization problems are being discussed. Pointwise and pairwise descriptors are introduced, both as general concepts and with specific examples.
We show how machine learning techniques such as random forests can be utilized to find optimal pointwise descriptors for a given class of shapes and deformations. We also show how correspondences can be seen as submanifolds of the so called product manifold of two shapes and propose an iterative way to find the correspondence based on kernel density estimation in the product manifold. We finally relate this approach to solving optimization problems stemming from pairwise descriptors using the *difference of convex functions* algorithm and close the loop with the observation that pairwise descriptors based on the spectrum of the LBO, so called heat kernels, lead to correspondences with high regularity.

Zusammenfassung

Diese Dissertation untersucht das Problem der automatisierten Analyse von verformbaren dreidimensionalen Objekten. Der Fokus der Arbeit liegt im Finden von Korrespondenzen zwischen zwei Dreiecksmeshes, einer üblichen digitalen Representation von dreidimensionalen Objekten. Diese Korrespondenzen können beispielsweise genutzt werden um verschiedenste Arten von Informationen von einem Mesh auf ein anderes zu übertragen. Beispielsweise kann die Segmentierung eines menschlichen Körpers in Körperteile von einem synthetisch generierten Template Mesh auf einen 3D Scan übertragen werden.
Wir geben einen umfassenden Überblick über die Theorie des Forschungsbereichs *non-rigid shape analysis* mit dem Fokus auf dem Korrespondenz-Problem. Grundlegende Konzepte der Differentialgeometrie und deren diskrete Analoga werden eingeführt . Ein umfangreiches Kapitel widmet sich dem *Laplace Beltrami Operator* (LBO). Dieser wurde einmal als *schweizer Taschenmesser des Geometry Processing* (Erstellung und Bearbeitung von digitalen 3D Objekten) bezeichnet. Auch eine Vielzahl von Bausteinen für die Analyse von 3D Objekten (von punktweisen Feature Deskriptoren über paarweise Deskriptoren bis hin zu kompakten Representationen von Korrespondenzen) baut auf dem LBO auf. Das macht den LBO auch zu einem essentiellen Werkzeug für die Analyse von dreidimensionalen Objekten. Wir führen den LBO als nicht-euklidische Veralgemeinerung des bekannten Laplace Operators ein, geben einen Überblick über seine wesentlichen Eigenschaften und zeigen, dass seine Eigenfunktionen als Verallgmeinerungen von *Fourierbasen* interpretiert werden können. Weiter leiten wir eine diskrete Version des LBO her, die genutzt werden kann um all die oben genannten Bausteine für die Analyse von Dreiecksmeshes verfügbar zu machen. Nach einer kurzen Einführung des *iterative closest point* Algorithmus zum Registrieren von riden Objekten mittels euklidischen Transformationen (Rotation und Translation), führen wir das Finden von nicht-euklischen Isometrien als Generalisierung für Korrespondenzen zwischen nicht-rigiden Objekten ein. Eine übliche diskrete Representation von allgemeinen bijektiven Korrespondenzen sind *Permutationsmatritzen*. Wir formulieren Korrespondenzprobleme als Optimierungsprobleme über dem Raum der Permutationsmatritzen und diskutieren praktische Limitierungen und mögliche Auswege (so genannte Relaxierungen). Punktweise und paarweise Feature Deskriptoren werden eingeführt, sowohl als allgmeines Konzept als auch in Form von konkreten Beispielen.
Wir zeigen wir Konzepte aus dem Bereich des maschinellen Lernens, konkret *random decision forests* genutzt werden können um optimale Feature Deskriptoren zu

finden, wenn ein Datensatz von Objekten gegeben ist, in dem Art und Umfang der Deformationen kodiert sind. In gewisser Weise orthogonal dazu erläutern wir wie Korrespondenzen als Untermannigfaltigkeiten der so genannten Produktmannigfaltigkeiten von zwei Objekten interpretiert werden können und führen einen iterativen Algorithmus ein, der aufbauend auf dem Konzept der *Kerndichteschätzung*, Korresponzen mit hoher Regularität findet. Abschließend setzen wir dieses Verfahren mit paarweisen Deskriptoren in Zusammenhang und - in gewisser Weise um den Kreis zu schließen - zeigen, dass paarweise Deskriptoren, die auf dem Spektrum des LBO basieren (so genannte *Heat kernels*) zu Korrespondenzen mit hoher Regularität führen.

Acknowledgments

It has been a long journey! I want to take the opportunity to thank Prof. Daniel Cremers for giving me the opportunity to pursue my academic research under his supervision. Thank you for enabling trips to conferences, attendance at workshops, and most importantly the trust that this journey will eventually converge.
Moreover I want to thank Prof. Nils Thürey, Prof. Emanuele Rodolà, and Prof. Justin Solomon for serving as members of the examination committee.
Besides my supervisor, the second pillar of my academic career has undoubtedly been Prof. Emanuele Rodolà. He was the best postdoc any Ph.D. student could hope for. Thank you for stimulating discussions, increasing the pressure when there was the need to but also for all the fun times we had together.
It was a pleasure to work at the same office as Zorah Lähner, Thomas Windheuser, and Frank Schmidt (the *Shape office*), Mohammed Souoai and Robert Maier. I will never forget the moments we spent together, whether it was on the bike, in karaoke bars, while giving lectures, on road trips up Highway No. 1 or while attending workshops and research stays at different places on this planet.
Speaking of research stays, I must also express my gratitude to Prof. Alex Bronstein. He allowed me to visit his labs at Tel Aviv University and the Technion. It were these trips and the opportunity to collaborate with brilliant individuals during those times that greatly enhanced my academic profile. Thank you, Alex, for providing such an enriching experience!
In regards to brilliant people, I want to express my sincere appreciation to all my co-authors, including Amit Boyarski, Alex Bronstein, Michael Bronstein, Daniel Cremers, Ron Kimmel, Roee Litmann, Zorah Lähner, Or Litany, Tal Remez, Emanuele Rodolà, Samuel Rota Bulò, Ron Slossberg, Rudolph Triebel and Thomas Windheuser. I enjoyed the in-depth conversations we had and am proud of the publications that were the outcome of those discussions. I am sure everyone who has pursued a Ph.D. at the *Chair for Computer Vision & Artificial Intelligence* at TUM can agree that life would have been way harder without Sabine Wagner and Quirin Lohr. Thank you for all administrative and technical support!
I am grateful to Thomas Möllenhoff and René Ranftl for proofreading parts of my thesis and in René's case also for helping me prepare for my defense presentation.
Last but not least I want to thank my parents, my family, and friends for their unwavering belief in me - it was certainly not always easy.
There are pillars for an academic career, and there are pillars for life. The latter are without any doubt my brother Christian and my wife Nathalie. I love you!

Contents

Part I

Introduction and Overview

Chapter 1

Introduction

In recent years the ability of creating digital models of real world objects has experienced a dramatic increase. This is partialy due to a new family of depth sensors such as the Microsoft Kinect, Asus Xtion and Intel Realsense but also due to improvements on the software side including deep learning based approaches for 3D reconstruction. As a consequence, the creation of 3D scans has reached the consumer market. First smartphones such as the latest iPhone models come with built in depth sensors. With the increasing amount of 3D content comes the need for an automatic understanding thereof.

In this thesis we consider isolated 3D objects such as scans of humans. We refer to these objects as *non-rigid shapes* to emphasize that we may observe different shapes that can be seen as non-rigid deformations of each other where the deformation is not necessarily rigid. Examples include scans of the same subject in different poses (*e.g.* standing and running), at different times (possibly years apart), or even entirely different persons (possibly also in different poses and/or at different times). We may want to characterize the (dis-)similarity of a pair or a collection of such shapes or be able to transfer properties such as texture or facial expressions from one shape to another.

A fundamental requirement to achieve this is to put different shapes into *correspondence*. A correpondence identifies pairs of regions or points on both shapes that semantically describe the same thing, *e.g.* the two nose tips. While for rigid deformations the correspondence can be described using a small set of parameters, such a compact representation is not at hand for more general deformations such as different poses. Even worse: no finite number of parameters can describe all possible deformations and thus correspondences between continuouos surfaces. The common approach is to only describe where a finite subset of points sampled from the source objects' surface are mapped to on the target objects' surface This is either a coordinate in space in vicinity of the target objects' surface or - in most cases - a point on the target objects' surface. The literature distinguishes sparse and dense correspondences, depending on whether only a sparse set of landmark points (*e.g.*

Figure 1.1: Given a pair of non-rigid shapes the task is to find a dense correspondence between them. Typical visualizations of correspondences transfer colors or texture from one shape to the other via the given correspondence. The input coloring is often chosen to assign a unique color to each point, *e.g.* by translating the xyz coordinates to RBG values as in the left most visualization. Further insight can be obtained by transferring multiple colorings, *e.g.* x-, y- and z coordinates seperately. (Meshes from the FAUST dataset [11])

at joints) are described or the number of point-to-point correspondences is on the same magnitude as the discrete representation of the shapes. As a matter of fact not only must the correspondence be described by a finite number of parameters but also the shapes themselves are only available in a discretized form, for example as triangle meshes or point clouds. This thesis focuses on dense correspondences between triangle meshes, many concepts can however be applied to pointclouds as well.
A valid correspondence should be semantically meaningful. Similar regions, *e.g.* faces must be mapped to each other. At the same time it should be *regular*. It turns out to be difficult to characterize what exactly we mean by regularity. Semantics seem almost as hard in many cases. Intuitively the correspondence should be as *bijective* and *smooth* as possible. Each region that is visible on both shapes needs to be part of the correspondence, closeby points should be mapped to closeby points in most cases. In practice a tradoff between the requirements must be made. Consider for example the two shapes in Figure 1.1: the intuitively correct mapping (head to head, left arm to left arm and so forth) is continuous when considered as a mapping from the male subject to the female subject (closeby point get mapped to closeby points). However if we consider the inverse mapping, points that are close by at the female subject (*e.g.* finger tips or right foot/left knee) are mapped to far away points on the male subject. Moreover not every region on the first shape has a (visible) corresponding region on the second shape (*e.g.* sole of right female foot). An additional difficulty arises when one tries to translate the notions of continuity and smoothness to discreticed shapes and pointwise correspondences. We will adress these issues in this thesis.

Pointwise descriptors As in other fields of Computer Vision, such as image registration or optical flow, a crucial ingredient for shape correspondences are *pointwise feature descriptors*, also referred to as *Euclidean embeddings*. These are functions

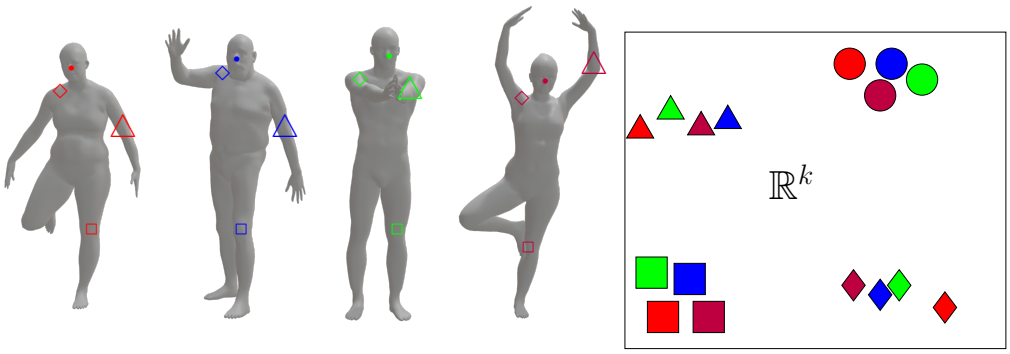

Figure 1.2: The descriptors of semantically similar points should be close in the Euclidean embedding space. On the contrary descriptors of dissimilar points should be distinct. (Meshes from the FAUST dataset [11])

that assign possibly high-dimensional vectors to the points on the shapes in a way that the (semanticaly) intuitive notion of similarity of points (*e.g.* the tips of the noses on two shapes) is reflected in similar (small distance) descriptors while dissimilar (*e.g.* nose and knee) points get assigned discriminative embeddings (see Figure 1.2). We will discuss popular pointwise descriptors such as the *heat kernel signature (HKS)* [52], the *wave kernel signature (WKS)* [6] and the *Signature of Histograms of OrienTations (SHOT)* [53]. In [4] we introduce methods to learn optimal pointiwse descriptors. Pointwise descriptors lead to *nearest neighbor problems* that are usually tackled using *kd trees* [7]. Assuming a consistent sampling of both shapes the discrete correspondence should be bijective, leading to *linear assignment problems* (LAPs). We will discuss both, kd trees and LAPs in the following sections.

Product manifolds We model the shapes as 2D manifolds embedded in $\mathbb{R}^3$ (the manifolds describe the boundary of the 3D object). We will recap the theory of manifolds (also called *Differential Geometry*) in the following sections. The graph of a correspondence is itself a manifold, a special case of a subset in the product of the two shapes, the so called *product manifold*. The authors of [57] observed that a regular correspondence has a regular graph and used this to enforce regularity by working directly in the product manifold. We will discuss pros and cons of this approach in the following sections. A main drawback lies in the computational intractability of the resultung integer linear program (ILP) which is NP-hard. In [3] we introduce an iterative method that circumvents the computational complexity and yields graphs of increasing regularity (and thus more regular correspondences) over the iterations.

Pairwise descriptors It turns out that the approach introduced in [3] is closely related to an instance of a *quadratic assignment problem* (QAP). We discuss this

connection and its implications in [2]. Quadratic assignment problems evolve from *pairwise descriptors* and are in general infeasible to optimize globally. We will present different choices of pairwise descriptors and approaches to tackle QAPs via relaxations in later sections.

Chapter 2

Outline of the Thesis

The present publication-based thesis is divided into three parts. In the following chapter we provide a summary for each part.

Part I:

Introduction

In this part we introduce mathematical concepts that are relevant in the field if non-rigid shape analysis.

Shapes: Manifolds and Triangle Meshes

We model non-rigid 3D shapes as two dimensional manifolds embedded in $\mathbb{R}^3$. In this chapter we recap important mathematical concepts about manifolds to make the reader familiar with concepts such as isometries, curvature and geodesic distances. We also discuss triangle meshes as discrete approximations of manifolds.

Laplace Beltrami Operator

'Laplace-Beltrami: The Swiss Army Knife of Geometry Processing' - the title of a SGP workshop in 2014 (held by Justin Solomon, Keenan Crane and Etienne Vouga) gives an impression on the importance of the Laplace Beltrami Operator when it comes to working with geometry. Not only is it a powerful tool for geometry processing but also for analysis of geometry, non-rigid shapes in particular. This chapter gives an extensive overview about its theory, discretization, and some applications.

Correspondences between Shapes

In this chapter we formaly introduce the problem we are addressing in this thesis, namely finding a correspondence between two non-rigid shapes. We extensively discuss permutations and their connection to correspondences between shapes. Global optimization over the space of permutations is in general a computationally intractable problem. We discuss popular relaxations such as *bistochastic matrices* and *functional maps*.
We also discuss pointwise and pairwise descriptors and make the reader familiar with the two requirements of invariance and discriminativity. We further present popular instances of pointwise descriptors such as the *heat kernel signature (HKS)* [52], the *wave kernel signature (WKS)* [6] and the *Signature of Histograms of Orientations (SHOT)* [53] and of pairwise descriptors, namely geodesic distances and heat kernels. We show how pointwise descriptors can be used to solve the correspondence problem via nearest neighbor search or linear assignment problems (LAPs) and how pairwise descriptors lead to quadratic assignment problems (QAPs).
Further the graph of a correspondence is considered and identified as a submanifold of the so called product manifold.

Part II:

This part includes a selection of peer reviewed research papers that were published during this thesis:

- Applying Random Forests to the Problem of Dense Non-rigid Shape Correspondence [4]
- Product Manifold Filter: Non-rigid Shape Correspondence via Kernel Density Estimation in the Product Space [3]
- Efficient Deformable Shape Correspondence via Kernel Matching [2]

Part III:

Concludes the thesis and proposes directions for possible future works.

Chapter 3

Shapes: Manifolds and Triangle Meshes

The most popular digital representation of 3D objects are triangle meshes. Those are approximations of the boundary of the object and can be seen as discrete approximations to *regular surfaces*, *i.e.* 2D submanifolds embedded in $\mathbb{R}^3$. In this chapter we give all necessary background to both, regular surfaces and triangle meshes. The following sections are closely linked to the text books [17] and [12] which we also suggest for further reading.
We start by formalizing the concept of a non-rigid shape. The object of interest in this thesis are *3D objects*, which are 'well behaving' subsets of $\mathbb{R}^3$.

Definition 1 An *object* of dimension d is an open and bounded subset $X \subset \mathbb{R}^d$ such that its boundary $\mathcal{X} := \partial X$ is a submanifold of dimension $d-1$.

We will only be working with the boundaries of objects and use the notation $\mathcal{O}^d$ for the space of all boundaries of objects of dimension d. All elements of $\mathcal{O}^d$ are compact orientable manifolds without boundary. We will specify what this means in the following section.

Definition 2 Given an equivalence relation $\sim$ on $\mathcal{O}^d$, the equivalence class $[O]_\sim$ of an element $O \in \mathcal{O}^d$ is its shape.

Typical equivalence relations are

$$O_1 \sim O_2 :\Leftrightarrow \quad \big[\exists (R,T) \in \mathrm{SO}(d) \times \mathbb{R}^d : \quad O_2 = R \cdot O_1 + T\big] \tag{3.1}$$

$$O_1 \sim O_2 :\Leftrightarrow \quad \big[\exists (R,T,\sigma) \in \mathrm{SO}(d) \times \mathbb{R}^d \times \mathbb{R}_+ : \quad O_2 = \sigma(R \cdot O_1 + T)\big] \tag{3.2}$$

The first relation defines two objects to have the same shape if they are related via a rigid motion, *i.e.* the shape does not change if the object is moved to a different location. The second relation additionaly allows changes in scale: a big circle and a small circle both have the shape of a circle.

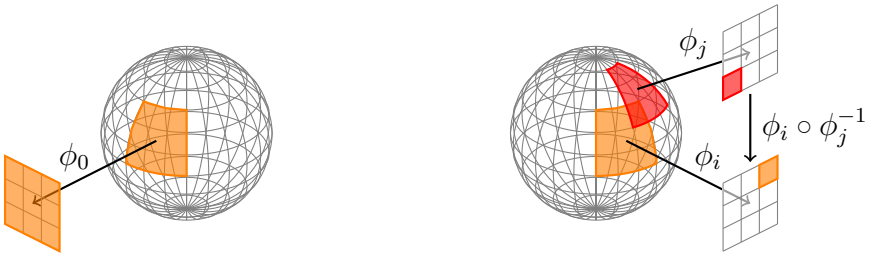

Figure 3.1: Most manifolds can not be covered by a single chart. Charts $(\mathcal{X}_i, \phi_i), (\mathcal{X}_j, \phi_j)$ covering the same subset of a manifold must be compatible, *i.e.* the concatenation $\phi_i \circ \phi_j^{-1} \colon \phi_j(\mathcal{X}_i \cap \mathcal{X}_j) \to \phi_i(\mathcal{X}_i \cap \mathcal{X}_j)$ must be a C^k-diffeomorphism

3.1 Manifolds

A tuple $(\mathcal{X}_0, \phi_0)$ is called an n-dimensional chart of the topologial space $\mathcal{X}$ iff $\phi_0 \colon \mathcal{X}_0 \to U_0$ is a homeomorphism between the open sets $\mathcal{X}_0 \subset \mathcal{X}$ and $U_0 \subset \mathbb{R}^n$.

Definition 3 A collection $(\mathcal{X}_i, \phi_i)_{i \in \mathcal{I}}$ of charts is called a C^k atlas iff $\mathcal{X} = \bigcup_{i \in \mathcal{I}} \mathcal{X}_i$ and for any two charts ϕ_i and ϕ_j, the mapping $\phi_i \circ \phi_j^{-1}$ is a C^k-diffeomorphism between $\phi_j(\mathcal{X}_i \cap \mathcal{X}_j)$ and $\phi_i(\mathcal{X}_i \cap \mathcal{X}_j)$.

Definition 4 A set $\mathcal{X}$ with a C^∞ atlas $(\mathcal{X}_i, \phi_i)_{i \in \mathcal{I}}$ is called a (smooth) manifold.

The terms *chart* and *atlas* create associations with geography. This is not a coincidence. The surface of the earth can be modelled as a sphere. We will soon see, that spheres are in fact examples of manifolds.
We like to think of n-dimensional manifolds as extensions of the linear space $\mathbb{R}^n$. In order to see this we will define the "manifold" $\mathbb{R}^n$.
Let $(\mathcal{X}_i)_{i \in \mathcal{I}}$ be a collection of open sets $\mathcal{X}_i \subset \mathbb{R}^n$ that cover $\mathbb{R}^n$. Possible choices are:

$$\mathcal{I} = \mathbb{Z}^n \qquad \mathcal{X}_{(i_1,\dots,i_n)} = \left\{ x \in \mathbb{R}^n \,\middle|\, \|x - i\| < \sqrt{n} \right\} \tag{3.3}$$

$$\mathcal{I} = \{0\} \qquad \mathcal{X}_0 = \mathbb{R}^n \tag{3.4}$$

Given these sets, the charts can be choosen as $(\mathcal{X}_i, \mathrm{id}_{\mathcal{X}_i})$ and for overlapping charts, we obtain the diffeomorphism

$$\phi_i \circ \phi_j^{-1} \colon \mathcal{X}_i \cap \mathcal{X}_j \to \mathcal{X}_i \cap \mathcal{X}_j \tag{3.5}$$

$$p \mapsto p \tag{3.6}$$

We are particularly interested in *submanifolds*, *i.e.* manifolds that "live" in an Euclidean embedding space.

Definition 5 A subset $\mathcal{X} \subset \mathbb{R}^n$ is a *(smooth) submanifold* of dimension m iff for every point $p \in \mathcal{X}$, there exists a chart (V, ϕ) of $\mathbb{R}^n$ such that

- $p \in V$.
- $\phi(\mathcal{X} \cap V) = \mathbb{R}^m \cap \phi(V)$.

We call $n - m$ the *co-dimension* of $\mathcal{X}$.

Notice that every submanifold is automatically a manifold. Before introducing two equivalent definitions of submanifolds we recapitulate the following theorem:

Theorem 1 *(Implicit Function Theorem)* Given a C^k-mapping $F\colon \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}^n$ and $(x_0, y_0) \in \mathbb{R}^m \times \mathbb{R}^n$ such that $F(x_0, y_0) = 0$ and $\frac{\partial}{\partial y} F(x_0, y_0)$ is invertible.
Then there exist open $U \subset \mathbb{R}^m$, $V \subset \mathbb{R}^n$ and $\phi\colon U \to V$ such that

- $x_0 \in U$, $y_0 \in V$ and $\phi(x_0) = y_0$.
- For all $x \in U$ we have $F(x, \phi(x)) = 0$.
- ϕ is a C^k-mapping.
- For all $x \in U$ we have $\frac{\partial}{\partial x}\phi(x) = -\left[\frac{\partial}{\partial y} F(x, \phi(x))\right]^{-1} \frac{\partial}{\partial x} F(x, \phi(x))$.

In other words, the implicitly defined set $\{(x, y) \in \mathbb{R}^{m \times n} | F(x, y) = 0\}$ can (locally) be interpreted as the graph of the function $\phi\colon U \to V$.
One important example of a submanifold is described by smooth coordinate mappings $x\colon U \to \mathbb{R}^n$:

Lemma 1 A subset $\mathcal{X} \subset \mathbb{R}^n$ together with smooth *coordinate mappings* $(x_i, U_i)_{i \in \mathcal{I}}$ is a smooth submanifold of dimension m if the following holds:

- All U_i are open subsets of $\mathbb{R}^m$.
- $\mathcal{X} = \bigcup_{i \in \mathcal{I}} x_i(U_i)$.
- For all $u \in U_i$, x_i is smooth and $Dx_i(u) \in \mathbb{R}^{n \times m}$ is of *maximal rank* m.

Proof. Given $p \in \mathcal{X}$, we choose $i \in \mathcal{I}$ and $\hat{u} \in \mathbb{R}^m$ such that $p = x_i(\hat{u})$. Since $Dx_i(\hat{u})$ is of maximal rank, we can find a matrix $A_0 \in \mathbb{R}^{n \times (n-m)}$ such that $A := \begin{pmatrix} Dx_i(\hat{u}) & A_0 \end{pmatrix} \in \mathbb{R}^{n \times n}$ is of maximal rank n, *i.e.*, invertible.
As a result, we can define the smooth function

$$\begin{aligned} \psi : U_i \times \mathbb{R}^{n-m} &\to \mathbb{R}^n \\ (u_1, \ldots, u_m, v_1, \ldots, v_{n-m}) &\mapsto x_i(u) + A_0 \cdot v, \end{aligned}$$

with $D\psi(\hat{u}, 0) = \begin{pmatrix} Dx_i(\hat{u}) & A_0 \end{pmatrix} = A$. Using Theorem 1 proves the Lemma for $\phi := \psi^{-1}$. □

In practice, it is often difficult to define different charts or coordinate functions. Instead, we like to define the manifold M by formulating certain constraints, *e.g.*,

$$\mathbb{S}^2 := \{x \in \mathbb{R}^3 | \, \|x\|^2 = 1\}. \tag{3.7}$$

Lemma 2 Given a function $f\colon \mathbb{R}^n \to \mathbb{R}^k$ and a *regular value* $c \in \mathbb{R}^k$, *i.e.*,

$$x \in f^{-1}(c) \qquad \Rightarrow \qquad \operatorname{rank}(Df(x)) = k. \tag{3.8}$$

Then, $\mathcal{X} := f^{-1}(c)$ is a submanifold of co-dimension k.

In the example above, f and c are given by $f\colon \mathbb{R}^n \to \mathbb{R}$ with $f(x) = \|x\|^2$ and $c = 1$.

Proof. Let $p \in \mathcal{X} \subset \mathbb{R}^n$. Since $Df(p)$ is of rank k, we can find k columns of $Df(p)$ that are linear independent. W.l.o.g. we assume that these k columns are the last k. Thus, the function $f\colon \mathbb{R}^{n-k} \times \mathbb{R}^k \to \mathbb{R}^k$ satisfies the *Implicit Function Theorem* with respect to $(x_0, y_0) = p$. The implicit function $\phi\colon \mathbb{R}^k \to \mathbb{R}^{n-k}$ defines a coordinate mapping $x\colon \mathbb{R}^k \to \mathbb{R}^n$ via $x(u) := (u, \phi(u))$, which proves the Lemma. □

Note that the implicit submanifold can be transformed into an explicit submanifold. Given a point $p \in \mathcal{X}$ the created coordinate mapping x satisfies

$$Dx(p) = \begin{pmatrix} \mathrm{id} \\ -\left[\frac{\partial f}{\partial y}(p)\right]^{-1} \frac{\partial f}{\partial x}(p) \end{pmatrix}. \tag{3.9}$$

3.1.1 Tangent Space

Smooth submanifolds $\mathcal{X}$ of dimension m can locally be approximated with a linear vector spaces of dimension m:

Definition 6 *(Tangent Space)* Let $\mathcal{X} \subset \mathbb{R}^n$ be a submanifold of dimension $m \leqslant n$ that is given via coordinate functions $(x_i, U_i)_{i \in \mathcal{I}}$. Given $i \in \mathcal{I}$ such that $p = x_i(u)$, we define the *tangent space* $T_p\mathcal{X}$ of $\mathcal{X}$ at the position p as

$$T_p\mathcal{X} := \{Dx_i(u) \cdot v | v \in \mathbb{R}^m\} \qquad [= \operatorname{im}(Dx_i(u))]\,. \tag{3.10}$$

The tangent space $T_pM \subset \mathbb{R}^n$ is a linear subspace and does not depend on the choice of the coordinate map (x_i, U_i). For implicitly defined manifolds defined via a smooth function f there is a neat connection between the tangent space and the differential Df:

Lemma 3 *(Tangent Space)* Let $f\colon \mathbb{R}^n \to \mathbb{R}^k$ be a smooth function with regular value $c \in \mathbb{R}^k$ and $\mathcal{X} := f^{-1}(c)$ the manifold with respect to this value. For every $p \in \mathcal{X}$ we have

$$T_p\mathcal{X} := \{v \in \mathbb{R}^n | Df(p) \cdot v = 0\} \qquad \big[= \ker(Df(p))\big]. \tag{3.11}$$

Given a point $p \in \mathcal{X}$ of a d-dimensional submanifold $\mathcal{X} \subset \mathbb{R}^n$, we can represent a tangent vector $v \in T_pM$ as a curve $c : (-\epsilon, \epsilon) \to \mathcal{X}$ with $c(0) = p$. To see this, let us look at the manifold from the point of view of a coordinate mapping $x \colon U \to \mathcal{X}$ with $0 \in U \subset \mathbb{R}^d$ and $x(0) = p$. Since $v \in T_pM = \mathrm{im}(Dx(0))$, we know that there is an $h \in \mathbb{R}^d$ such that $Dx(0)[h] = v$. Using

$$c \colon (-\epsilon, \epsilon) \to \mathcal{X} \qquad\qquad c(t) = x\left(t \cdot h\right), \tag{3.12}$$

we have

$$Dc(0) = Dx(0 \cdot h)[h] = v. \tag{3.13}$$

Given a point $p \in \mathcal{X}$ of a d-dimensional submanifold $\mathcal{X} \subset \mathbb{R}^n$, we define

$$\mathcal{C}_p\mathcal{X} := \{c \colon (-\epsilon, \epsilon) \to \mathcal{X} | \exists \epsilon > 0 : c \text{ is smooth and } c(0) = p\}. \tag{3.14}$$

The goal is to define $T_p\mathcal{X}$ by defining an equivalence relation on $\mathcal{C}_p\mathcal{X}$:

$$c_1 \sim c_2 \qquad\qquad :\Leftrightarrow \qquad\qquad Dc_1(0) = Dc_2(0), \tag{3.15}$$

It is easy to check that $\sim$ is indeed an equivalence relation (*i.e.* reflexive, symmetric and transitive). It turns out $T_pM = \mathcal{C}_pM/\sim$, which provides us with an alternative definition for the tangent space T_pM. The advantage of this rather technical definition is that for any $v \in T_p\mathcal{X}$ we can *choose a curve* $c \in v$ that passes through p and vice versa, *i.e.*, any curve c that passes through a point p defines a tangent vector $v := [c]$.

3.1.2 Functions on Manifolds

Given a manifold $\mathcal{X}$, a function $f \colon \mathcal{X} \to \mathbb{R}^d$ is called *smooth* iff for all charts $(\mathcal{X}_i, \phi_i)$, the function $f \circ \phi_i^{-1} \colon \phi_i(\mathcal{X}_i) \to \mathbb{R}^d$ is smooth. The "smooth functions" f on the "manifold" $\mathbb{R}^n$ are exactly the functions $C^\infty(\mathbb{R}^n)$.
Let us recap how the *differential* of a function $f : \mathbb{R}^m \to \mathbb{R}^n$ acting on the Euclidean space $\mathbb{R}^m$ is defined. The differential $Df(p) \colon \mathbb{R}^m \to \mathbb{R}^n$ at a position $p \in \mathbb{R}^m$ is uniquely defined as the *linear mapping*

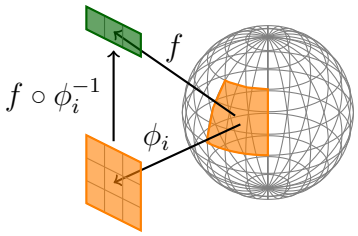

Figure 3.2: Smooth function on manifold

$$f(p + v) = f(p) + Df(p)[v] + r(v) \tag{3.16}$$

where $\lim_{v \to 0} \frac{r(v)}{\|v\|} = 0$. As a linear mapping, the differential $Df(p)$ has a matrix representation (using the canonical bases for $\mathbb{R}^m$ and $\mathbb{R}^n$) $J \in \mathbb{R}^{n \times m}$, such that

$Df(p)[v] = J \cdot p$. This matrix is called *Jacobian (of f at p)*. Its entries are given by $J_{i,j} = \partial_j f^i(p)$. Although the Jacobian depends on f and p one usually omits indicating this dependency with symbols. Equation (3.16) can be interpreted as following:

- p describes a *point* in the space on which f is defined,
- v describes the *direction* in which we change the point p
- $Df(p)[v]$ describes the *direction* in which f changes if we change the point p in the direction v.

For vector spaces, there is no distinction between points and directions. For manifolds $\mathcal{X}$, points p will be on the manifold and directions on the tangent space $T_p\mathcal{X}$. Given two submanifolds $\mathcal{X}$ and $\mathcal{Y}$ as well as a function $f\colon \mathcal{X} \to \mathcal{Y}$. For $p \in \mathcal{X}$ and $q = f(p) \in \mathcal{Y}$, the differential $Df(p)$ is the *push-forward*

$$Df(p)\colon T_p\mathcal{X} \to T_q\mathcal{Y} \tag{3.17}$$
$$[c] \mapsto [f \circ c] \tag{3.18}$$

Unlike the Euclidean spaces $\mathbb{R}^m$ and $\mathbb{R}^n$ for general submanifolds there are no canonical bases of the tangent planes and thus no canonical matrix representation of the push forward.

3.1.3 First Fundamental Form

Given a coordinate mapping $x\colon U \to \mathcal{X} \subset \mathbb{R}^n$, $U \subset \mathbb{R}^m$, we want to measure some quantities of $\mathcal{X}$ directly on U. Important geometric quantities that we like to measure are

Angles In practice, we are not so much interested in the angle α itself, but rather in $\cos(\alpha)$. In particular, we would like to determine whether two lines that pass through a point p are orthogonal to one another. In order to measure $\cos(\alpha)$ we need something like a scalar product.

Length We want to measure the length of curves on the manifold by computing a line integral on the domain U.

Area Usually integration is considered in the context of computing the size of certain areas. It is therefore natural to also transform a surface integral on $\mathcal{X}$ into a surface integral on U.

These problems can be addressed with the so called *First Fundamental Form*. It is also called the *Riemannian Metric* or the *metrical tensor*.

Given the coordinate mapping $x\colon U \to \mathcal{X} \subset \mathbb{R}^n$, $U \subset \mathbb{R}^m$, the *first fundamental form* is defined as

$$g\colon U \to \mathbb{R}^{m\times m} \qquad u \mapsto Dx(u)^\top Dx(u) \tag{3.19}$$

The matrix $g(u)$ is symmetric and positive-definite, *i.e.*,

$$g(u)^\top = g(u) \qquad \langle X, X\rangle_{g(u)} := \langle X, g(u)\cdot X\rangle > 0 \qquad \forall X \neq 0 \tag{3.20}$$

Therefore, the Riemannian metric can be seen as a continuously changing scalar product on the domain U which motivates the notation $\langle X, Y\rangle_{g(u)}$.
Let us assume the coordinate map $x\colon U \to \mathcal{X}$ and the first fundamental form $g\colon U \to \mathbb{R}^{m\times m}$. Further, let us assume that two curves $\gamma_{1,2}\colon (-\epsilon, \epsilon) \to U$ are given in the parameter domain U that pass through the same point $\gamma_1(0) = u = \gamma_2(0)$.
Now let $X := \gamma_1'(0) \in \mathbb{R}^d$ and $Y := \gamma_2'(0) \in \mathbb{R}^d$.
The curves γ_i define curves $c_i := x \circ \gamma_i$ on the manifold $\mathcal{X}$ and pass through the point $p = x(u) \in \mathcal{X}$. The curves define tangent vectors in $T_p\mathcal{X}$ via

$$v_i := c_i'(0) = Dx(u)\cdot\gamma_i'(0) \in T_p\mathcal{X} \subset \mathbb{R}^m \tag{3.21}$$

Instead of building the scalar product in the embedding space $\mathbb{R}^m$, we can move the computation back onto U

$$\langle v_1, v_2\rangle = \langle Dx(u)X, Dx(u)Y\rangle = \langle X, Y\rangle_{g(u)} \tag{3.22}$$

In summary, we can measure the cosine of the angle α between $c_1 = x \circ \gamma_1$ and $c_2 = x \circ \gamma_2$ in means of $X = \gamma_1'(0)$ and $Y = \gamma_2'(0)$:

$$\cos(\alpha) = \frac{\langle v_1, v_2\rangle}{\|v_1\|\,\|v_2\|} = \frac{\langle X, Y\rangle_{g(u)}}{\|X\|_{g(u)}\,\|Y\|_{g(u)}} \tag{3.23}$$

Thus, we are able to measure $\cos(\alpha)$ without explicitly looking at the coordinate map (embedding) x but by just making use of g. We call every quantity that can be measured in that way as *intrinsic*. They do not depend on the surrounding space, but only on measurements "inside" of the manifold. A coordinate mapping $x\colon U \to \mathcal{X}$ is called *conformal* if every angle measurement in U coincides with the angle measurement on $\mathcal{X}$. The Riemannian metric g of a conformal coordinate mapping $x\colon U \to \mathcal{X}$ is the identity matrix multiplied with a scalar $r\colon U \to \mathbb{R}$ that depends on the location $u \in U$ of the parametrization domain.
In the following we want to revisit the length computation of a curve. To this end, let $x\colon U \to \mathcal{X}$ be a coordinate map, $\gamma\colon [0;1] \to U$ a curve in the parametrization domain U and

$$c\colon [0;1] \to \mathcal{X} \qquad c(t) := x \circ \gamma(t) \tag{3.24}$$

the curve on the manifold $\mathcal{X}$ whose length we like to measure.

The length of c can be computed via

$$\text{length}(c) = \int_0^1 \|\dot{c}(t)\| \,\mathrm{dt} = \int_0^1 \|Dx(\gamma(t)) \cdot \dot{\gamma}(t)\| \,\mathrm{dt} = \int_0^1 \|\dot{\gamma}(t)\|_{g(\gamma(t))} \,\mathrm{dt} \tag{3.25}$$

We can express $\text{length}(c)$ with γ as long as we take the Riemannian metric g into account. Thus, $\text{length}(c)$ is an intrisic quantity of $\mathcal{X}$.
For two points $p, q \in \mathcal{X}$ their *geodesic distance* is given by the length of the shortest curve connecting them (red curve in Figure 3.3):

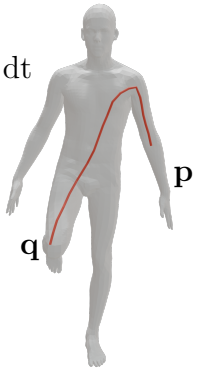

Figure 3.3: The red curve is the shortest curve connecting the points p and q. Its length is the *geodesic distance* of p and q. (Mesh from the FAUST dataset [11])

$$d(p, q) = \min_{c\colon [0,1]\to\mathcal{X}} \{\text{length}(c) | c[0] = p\,, c[1] = q\} \tag{3.26}$$

For *compact* manifolds a minimizer always exists (but may not be unique). The manifolds we are interested in are compact.
Next we want to find an expression for the gradient of a scalar function $f\colon \mathcal{X} \to \mathbb{R}$.

Definition 7 Let $f\colon \mathcal{X} \to \mathbb{R}$ be a differentiable function. The *gradient* $\nabla f(p)$ at $p \in \mathcal{X}$ is the unique element of $T_p\mathcal{X}$ such that

$$df(p)[v] = \langle \nabla f, p \rangle \tag{3.27}$$

Existence and uniqueness follow from the *Riesz representation theorem*.

Lemma 4 Let $x\colon U \to \mathcal{X}$ be a coordinate map with first fundamental form $g\colon U \to \mathbb{R}^{m\times m}$ and $\tilde{f} = f \circ x$. Then the coefficients $\alpha \in \mathbb{R}^m$ (local coordinates) of $\nabla f = Dx \cdot \alpha \in T_p\mathcal{X}$ are given by

$$\alpha = g^{-1}(u)\nabla\tilde{f}(u) \tag{3.28}$$

Proof. Let $\beta \in \mathbb{R}^m$ be the coefficients of $v \in T_p\mathcal{X}$. Then

$$df(p)[v] = \langle \nabla\tilde{f}(u), \beta \rangle = \langle \alpha, \beta \rangle_{g(u)} = \langle \nabla f, v \rangle \tag{3.29}$$

□

Finally integration of a function $f\colon \mathcal{X} \to \mathbb{R}$ can be moved to the parameter domain via

$$\int_{\mathcal{X}_i} f(p)dp = \int_{U_i} (f \circ x_i)(u) \cdot \sqrt{\det g(u)}du \tag{3.30}$$

where $x_i : U_i \to X_i = x_i(U_i) \subset \mathcal{X}$ is a coordinate map. While we do not provide a proof we do want to emphasize that $\sqrt{\det g(u)}$ is the volume of the (m-dimensional) parallelepiped spanned by the tangent vectors Dx_i, *e.g.* in two dimensions

$$\sqrt{\det g(u)} = \sqrt{\|Dx_1\| \, \|Dx_2\| - \langle Dx_1, Dx_2 \rangle} = \|Dx_1 \times Dx_2\| \tag{3.31}$$

Equation (3.30) is a generalization of substitution. Notice that the integral only depends on values of the function $f \circ x_i \colon U_i \to \mathbb{R}$ and the first fundamental form $g : U_i \to \mathbb{R}$. Thus integration of functions is intrinsic as well.
At this point we stop our excursion to the fascinating world of differential geometry and refer the interested reader to the de facto standard text book [17]. There further concepts like normals, curvatures, second fundamental forms and Christoffel symbols are being discussed. Just one last comment: the *mean curvature* is also an intrinsic property, *i.e.* it is invariant under isometries.

3.2 Triangle Meshes

When working with shapes on a digital computer with a finite amount of memory, shapes have to be represented in a discretized way. The standard representation are polyhedral meshes, while alternatives (signed distance fields, octrees, tetmeshes) exist. The most common choice are *triangle meshes*. Quadriliteral meshes are often used in disciplines such as architecture, but are not subject of this work. In this section we introduce traingle meshes and functions that are defined on them. A triangle mesh $\mathcal{X}$ is a pair $(\mathcal{V}, \mathcal{F})$ with $\mathcal{V} = \{v_1, \dots, v_{|\mathcal{V}|}\} \subset \mathbb{R}^3$ being the set of vertices and $\mathcal{T} = \{t_1, \dots, t_{|\mathcal{T}|}\}, \quad t_i \in \mathcal{V} \times \mathcal{V} \times \mathcal{V}$ being the set of triangles. Notice that the triangles come with an *orientation*, *i.e.* $t = (v_i, v_j, v_k) \neq (v_i, v_k, v_j) = \tilde{t}$. On the other hand two elements of $\mathcal{T}$ that differ only by a cyclic permutation of the vertices describe *the same* triangle:

$$(v_i, v_j, v_k) \sim (v_j, v_k, v_i) \sim (v_k, v_i, v_j) \tag{3.32}$$

Thus stictly speaking the set of triangles is the space $\mathcal{T} \setminus \sim$. We will however still denote it with $\mathcal{T}$ and write (v_i, v_j, v_k) instead of $[(v_i, v_j, v_k)]_\sim$. Following the *righthand rule* every triangle $t = (v_i, v_j, v_k)$ comes with a well defined *outward pointing normal* $n = \frac{(v_j - v_i) \times (v_k - v_i)}{\|(v_j - v_i) \times (v_k - v_i)\|}$. The triangles induce a set of (directed) edges $\mathcal{E} = \{(v_i, v_j) | \exists f \in \mathcal{F} : f = (v_i, v_j, v_k)\} \subset \mathcal{V} \times \mathcal{V}$ (up to cyclic shifts).
There are two ways to look at triangle meshes: as approximations of smooth manifolds or as (less smooth) C^0 manifolds (coordinate maps are only C^0 functions) themselves. A possible choice of coordinate maps is given by $\{(U_j, x_j)\}_{j=1}^{|T|}$ where each $U_j \subset \mathbb{R}^2$ contains the *reference triangle* T_r spanned by the vertices $(0,0), (1,0)$ and $(0,1)$ that is affinely mapped to the triangle t_j via x_j:

$$x_j(u)|_{T_r} = v_j^1 + u_1(v_j^2 - v_j^1) + u_2(v_j^3 - v_j^1) \tag{3.33}$$

In Computer Graphics this type of mapping is commonly referred to as *uv mapping*. Because the mapping (restricted to the T_r) is affine, the differential and consequently the first fundamental form are constant

$$Dx_j|_{T_r} = \begin{pmatrix} | & | \\ v_j^2 - v_j^1 & v_j^3 - v_j^1 \\ | & | \end{pmatrix} \tag{3.34}$$

$$g|_{T_r} = \begin{pmatrix} \left\|v_j^2 - v_j^1\right\|^2 & \left\langle v_j^2 - v_j^1, v_j^3 - v_j^1 \right\rangle \\ \left\langle v_j^2 - v_j^1, v_j^3 - v_j^1 \right\rangle & \left\|v_j^3 - v_j^1\right\|^2 \end{pmatrix} \tag{3.35}$$

The famous Euler formula states an interesting relation between the number of vertices $|V|$, edges $|E|$ and triangles $|T|$ in a closed and connected triangle mesh:

$$|V| - |E| + |T| = 2(1 - g) \tag{3.36}$$

where g denotes the *genus* (intuitively the number of holes) of the surface. For most practical applications the genus g is small compared to the number of vertices, faces, and edges. Further

- each triangle is bounded by three edges
- each edge is incident to two triangles.

This leads to the following approximations relating the number of vertices, edges and triangles:

- The number of triangles is approximately twice the number of faces $F \approx 2V$.
- The number of edges is approximately three times the number of vertices $E \approx 3V$.
- The average vertex valence (the number of incident edges) is 6.

3.2.1 Linear (affine) finite elements on triangle meshes

As we discretized two dimensional manifolds by a finite number $|V|$ of vertices and $|T|$ of triangles we are also only able to store a finite number of values to represent a function defined on a triangle mesh. Given a function $f : \mathcal{X} \to \mathbb{R}$ defined on a triangle mesh $\mathcal{X}$, a standard way to discretize it is to only store its values at the vertices:

$$\begin{pmatrix} f(v_1) & \dots & f(v_V) \end{pmatrix}^\top = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_{|V|} \end{pmatrix}^\top = \mathbf{f} \in \mathbb{R}^{|V|} \tag{3.37}$$

However many different functions f may have the same discretization.

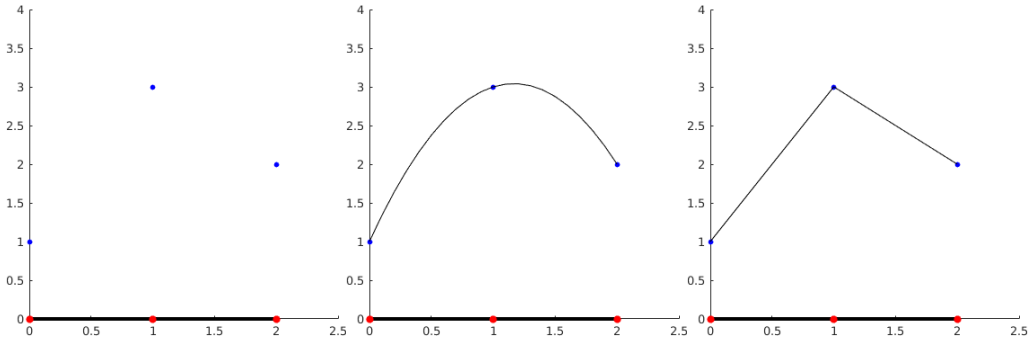

Figure 3.4: Sampling of functions

We will henceforth interpret vectors $\mathbf{f} \in \mathbb{R}^V$ as samplings of *piecewise linear (PL) functions*. The space of PL functions is a vector space. Adding (scaling) the representing vectors is equivalent to adding or scaling the represented functions. Notice that PL functions are not (classically) differentiable (at edges and vertices). As a $|\mathcal{V}|$-dimensional vectorspace we should be able to find $|\mathcal{V}|$ basis functions $\{\psi_1, \dots \psi_\mathcal{V}\}$ that span this space. A standard choice are the so called *hat functions* or *nodal basis functions*, defined via

$$\psi_i(v_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

and their property of being PL functions (*i.e.* linear inside each triangle). Sampled values coincide with coefficients in this basis:

$$f(x) = \sum_{i=1}^{|V|} \alpha_i \psi_i(x) = \sum_{i=1}^{|V|} f(v_i)\psi_i(x) = \sum_{i=1}^{|V|} \mathbf{f}_i \psi_i(x) \tag{3.38}$$

In addition to the easy interpretation of coefficients, these basis functions have the desirable property of having localized support (area where the function $\neq 0$). This will lead to *sparse matrices*. They are however lacking an other desirable property: the nodal basis functions *do not form an orthonormal basis*:

$$\langle \psi_i, \psi_j \rangle_{L^2(\mathcal{X})} = \int_\mathcal{X} \psi_i(p)\psi_j(p)dp \neq 0 \tag{3.39}$$

if v_i and v_j are connected with an edge. As a consequence the L^2-product of two PL functions f and g is not the same as the (standard) scalarproduct of their coefficients

$$\langle f, g \rangle_{L^2(\mathcal{X})} \neq \langle \mathbf{f}, \mathbf{g} \rangle \tag{3.40}$$

Instead:

$$\langle f, g\rangle_{L^2(\mathcal{X})} = \int_{\mathcal{X}} (\sum_{i=1}^{|V|} \mathbf{f_i}\psi_i(p))(\sum_{=1}^{|V|} \mathbf{g_j}\psi_j(p))dp \tag{3.41}$$

$$= \sum_{i=1}^{|V|}\sum_{j=1}^{|V|} \mathbf{f}_i\mathbf{g}_j \int_{\mathcal{X}} \psi_i(p)\psi_j(p)dp \tag{3.42}$$

$$= \langle \mathbf{f}, \mathbf{M}\mathbf{g}\rangle =: \langle \mathbf{f}, \mathbf{g}\rangle_{\mathbf{M}} \tag{3.43}$$

The matrix $\mathbf{M} \in \mathbb{R}^{|V|\times|V|}$ with entries $\mathbf{M}_{ij} = \int_{\mathcal{X}} \psi_i(p)\psi_j(p)dp$ is called *mass matrix*. We discuss properties and an explicit formula for its entries in the following subsection.

3.2.2 The mass matrix

The mass matrix is a symmetric positive definite matrix and thus induces an *alternative inner product* on $\mathbb{R}^{|V|}$.

We will now derive explicit expressions for the entries M_{ij}.

We can decompose the integral $\int_{\mathcal{X}} \psi_i(p)\psi_j(p)dp$ over the mesh $\mathcal{X}$ into the sum of integrals over all triangles:

$$\mathbf{M}_{ij} = \int_{\mathcal{X}} \psi_i(p)\psi_j(p)dp \tag{3.44}$$

$$= \sum_{k=1}^{|T|} \int_{t_k} \psi_i(p)\psi_j(p)dp. \tag{3.45}$$

Note that the integral over the triangle t_k is zero if neither vertex v_i nor v_j is a corner of t_k. Thus $M_{ij} = 0$ if $v_i \neq v_j, (v_i, v_j) \notin \mathcal{E}$. This leaves two cases: $v_i = v_j$ or $(v_i, v_j) \in \mathcal{E}$. Let first $v_i \neq v_j, (v_i, v_k) \in \mathcal{E}$. There are exactly two triangles having v_i and v_j as corners. Let $t_k = (v_l, v_i, v_j)$ be one of them. With the parametrization

$$x_k(u)|_{T_r} = v_l + u_1(v_i - v_l) + u_2(v_j - v_l) \tag{3.46}$$

it holds that $(\psi_i \circ x_k)(u) = u_1$, $(\psi_j \circ x_k)(u) = u_2$ which together with equations 3.30,

3.34 gives

$$\int_{t_k} \psi_i(p)\psi_j(p)dp = \int_{t_r} u_1 u_2 \sqrt{\|e_{li}\| \, \|e_{lj}\| - \langle e_{li}, e_{lj} \rangle} du \tag{3.47}$$

$$= 2|t_k| \int_0^1 \int_0^{1-u_1} u_1 u_2 du_2 Du_1 \tag{3.48}$$

$$= |t_k| \int_0^1 u_1 (1-u_1)^2 du_1 \tag{3.49}$$

$$= |t_k| \int_0^1 u_1 - 2u_1^2 + u_1^3 du_1 \tag{3.50}$$

$$= |t_k| \left(\frac{1}{2} - \frac{2}{3} + \frac{1}{4} \right) \tag{3.51}$$

$$= \frac{1}{12}|t_k| \tag{3.52}$$

where $e_{li} = (v_i - v_l)$, $e_{lj} = (v_j - v_l)$ and $|t_k| = \frac{1}{2}\sqrt{\|e_{li}\| \, \|e_{lj}\| - \langle e_{li}, e_{lj} \rangle} = \frac{1}{2} \|e_{li} \times e_{lj}\|$ is the area of the triangle t_k (*c.f.* Equation (3.31)). Consequently

$$\mathbf{M}_{ij} = \frac{1}{12}(|t_k| + |t_{k'}|) \tag{3.53}$$

with t_k and $t_{k'}$ being the two triangles sharing the edge (v_i, v_j). An analogous derivation shows that the diagonal entries $\mathbf{M}_{ii}$ are given as the sixth part of the sum of areas of all triangles having v_i as a corner.

$$\mathbf{M}_{ii} = \frac{1}{6} \sum_{v_i \in t_k} |t_k| \tag{3.54}$$

In particular

- $\mathbf{M}$ is a symmetric and *sparse matrix* with support "on the edges of the mesh"
- all entries of $\mathbf{M}$ are positive
- the diagonal entries of $\mathbf{M}$ equal the sum of the non-diagonal entries $\mathbf{M}_{ii} = \sum_{j \neq i} \mathbf{M}_{ij}$
- $\mathbf{M}$ is positive definite.

The mass matrix is often approximated with a diagonal matrix $\mathbf{A}$ usually referred to as the *matrix of area elements*. Its entries are given by $\mathbf{A}_{ii} = \sum_j \mathbf{M}_{ij}$. As the name suggests the entries have a neat interpretation: For well behaving meshes (no obtuse triangles) the values coincide with the area of the *Voronoi cells* around the

corresponding vertices. In order to integrate a function f given by the coefficients $(\mathbf{f}_1, \ldots, \mathbf{f}_{|V|})$ one can simply build the weighted sum of those coefficients:

$$\int_{\mathcal{X}} f(p)dp = \sum_i \mathbf{A}_{ii}\mathbf{f}_i \tag{3.55}$$

$$= \mathbf{1}^\top \mathbf{A}\mathbf{f} \tag{3.56}$$

$$= \mathbf{1}^\top \mathbf{M}\mathbf{f} \tag{3.57}$$

$$= \int_{\mathcal{X}} 1 \cdot f(p)dp \tag{3.58}$$

where $\mathbf{1} = \begin{pmatrix} 1 & \ldots & 1 \end{pmatrix}^\top$ is the $|V|$-dimensional vector with 1's everywhere. This approach however fails when integrating function that are not linear inside the triangles such as products of PL functions (which are piecewise quadratic), *i.e.* for general $f = \sum_i \mathbf{f}_i\psi_i$ and $g = \sum_i \mathbf{g}_j\psi_j$

$$\int_{\mathcal{X}} f(p)g(p)dp \neq \mathbf{f}^\top \mathbf{A}\mathbf{g} = \sum_i \mathbf{f}_i\mathbf{g}_i\mathbf{A}_{ii} \tag{3.59}$$

In particular evident is the case of the L^2 product of two hat functions which suddenly vanishes even when the corresponding vertices are adjacent. A remedy of this inconsistency is to interpret the coefficients in a different function space: the space of *piecewise constant functions* where the basis functions are *indicator functions* on the Voronoi cells. The aforementioned inconsistency will be omitted not only when dealing with products of two functions but also for products of an arbitrary number of functions since the space of piecewise constant functions with disjoint support is closed under pointwise multiplication. Yet another inconsistency will arise when introducing the *discrete Laplacian* as piecewise constant functions are not (even weakly) differentiable and the construction of the *stiffness matrix* (that will be introduced in the next chapter) will thus fail. That being said, we must admit that we are not aware of any practical implication this inconsistency causes. However, we prefer to work with the "real" mass matrix instead of the matrix of area elements.

Chapter 4

Laplace Beltrami Operator

This chapter is dedicated to the probably most famous tool used in non-rigid shape analysis during the last decade. The *Laplace Beltrami operator* is used for a wide variety of tasks, such as

- regularization of shapes
- interpolation of shapes
- solving of physics-based PDEs on shapes
- definition of pointwise and pairwise descriptors for non-rigid shapes
- compressed representation of functions on and correspondences between shapes
- efficient derivation of geodesic distances on shapes

We will get back to some of these applications in the following chapters. In this chapter we focus on the mathematical properties and the discretization of the Laplace Beltrami operater (LBO).
For C^2-functions $f\colon \mathbb{R}^m \supset U \to \mathbb{R}$ the Laplace Beltrami operator is just the well known Laplace operator

$$\triangle f(u) = \sum_{i=1}^{m} \frac{\partial f^2}{\partial x_i^2} \tag{4.1}$$

$$\Rightarrow \triangle f = \operatorname{div}(\nabla f) \tag{4.2}$$

The Laplace Beltrami operator can however be applied to a broader class of functions (so called *weakly differentiable functions*) and domains. In fact it acts on functions $f\colon \mathcal{X} \to \mathbb{R}$ defined on submanifolds $\mathcal{X}$. We will next introduce the spaces of *weakly differentiable functions*, the so called *Sobolev spaces* which will allow the proper definition of the Laplacian.

4.1 Sobolev Spaces

This - admittedly technical - section introduces the notion of *weak derivatives* and *Sobolev spaces*. We follow notation and structure of Chapter 5 in [20]. Let U be either an open subset of $\mathbb{R}^m$ or an open subset of a m-dimensional manifold. We denote the space of *test functions* on U by

$$C_0^\infty(U) = \{h \in C^\infty(U) | h \text{ has compact support in } U\}. \tag{4.3}$$

For a better intuition we provide an example of a test function on $U = B_2(0) = \{u \in \mathbb{R}^2 | \, \|u\| < 2\}$:

$$h(u) = \begin{cases} \exp(-\frac{1}{1-\|u\|^2}) & \|u\| \leqslant 1 \\ 0 & \|u\| > 1 \end{cases} \tag{4.4}$$

the support of h is given by $\operatorname{supp} h = \overline{\{u \in U | h(u) \neq 0\}} = \overline{\{u | \, \|u\| < 1\}} = \{u | \, \|u\| \leqslant 1\}$ which is a compact subset of U.
For more details regarding test functions we refer to standard text books of functional analysis or partial differential equations, such as [20] and will just emphasize the properties of test functions that are important for the rest of this chapter:

- A test function h is infinitely often continuously differentiable.
- When evaluating h or any of its derivatives at the boundary ∂U of U, the value is 0.
- The derivative h' of a test function h is again a test function.
- Given a test function $h \in C_0^\infty(U)$, $U \subset \mathbb{R}^m$ and a coordinate map $x \colon U \to \mathcal{X}$ of a submanifold $\mathcal{X}$, the function $\psi = h \circ x^{-1} \colon x(U) \to \mathbb{R}$ is a test function on $x(U)$ (and by setting $\psi(p) = 0 \forall p \notin x(U)$ on $\mathcal{X}$).
- For a compact manifold $\mathcal{X}$ without boundary ($\partial X = \varnothing$), every smooth function is a test function, *i.e.* $C_0^\infty(\mathcal{X}) = C^\infty(\mathcal{X})$.

Let $f \colon \mathbb{R}^m \supset U \to \mathbb{R}$ be *locally integrable*, (*i.e.* $\int_K |f(u)| du < \infty$ for all compact $K \subset U$) and consequently $\int_U |f(u)h(u)| du < \infty$ for all test functions h. Let further $\alpha = \begin{pmatrix} \alpha_1 & \dots & \alpha_m \end{pmatrix} \in \mathbb{N}^m$ be a multi-index and

$$D^\alpha = \frac{\partial^{\alpha_1}}{\partial u_1^{\alpha_1}} \cdots \frac{\partial^{\alpha_m}}{\partial u_m^{\alpha_m}} \tag{4.5}$$

$$|\alpha| = \sum_i \alpha_i \tag{4.6}$$

. If there exists a locally integrable function g with

$$\int_U f(u)D^\alpha h(u)du = (-1)^{|\alpha|}\int_U g(u)h(u)du \qquad \forall h \in C_0^\infty(U) \tag{4.7}$$

we call g the *α-th weak partial derivative* of f. It is easy to see that for every $C^{|\alpha|}$-function weak and classical derivatives coincide. Moreover, weak derivatives are unique (up to sets of measure 0). As a simple example for a function that is weakly differentiable but not classically differentiable consider $f\colon (a,b) \to \mathbb{R}$ that is continuous on (a,b) and (classically) differentiable on (a,u_0) and on (u_0,b) but is not differentiable at $u_0 \in (a,b)$ (*i.e.* $\lim_{u\uparrow u_0} f'(u) \neq \lim_{u\downarrow u_0} f'(u)$). It holds

$$\begin{aligned}
\int_a^b f(u)h'(u)du &= \int_a^{u_0} f(u)h'(u)du + \int_{u_0}^b f(u)h'(u)du && (4.8)\\
&= -\int_a^{u_0} f'(u)h(u)du + [f(u_0)h(u_0) - f(a)h(a)] && (4.9)\\
&\quad -\int_{u_0}^b f'(u)h(u)du + [f(b)h(b) - f(u_0,h(u_0))] && (4.10)\\
&= -\int_a^{u_0} f'(u)h(u)du + -\int_{u_0}^b f'(u)h(u)du && (4.11)\\
&= -\int_a^b g(u)h(u)du && (4.12)
\end{aligned}$$

Thus the weak derivative coincides with the classical derivative where it exists. However g is not continuous since $\lim_{u\uparrow u_0} f'(u) \neq \lim_{u\downarrow u_0} f'(u)$. The value $g(u_0)$ can be chosen arbitrarily ($\{u_0\}$ has measure 0). This example of continuous and piecewise differentiable functions generalizes to higher dimensions and in particular applies to the piecewise linear function on triangle meshes that we introduced in the previous chapter. Following [20] (5.1.1) we define

Definition 8 The *Sobolev space* $W^{k,p}(U)$ consists of all locally summable function $f\colon U \to \mathbb{R}$ such that for each multi-index α with $|\alpha| \leqslant k$, $D^\alpha f$ exists (in the weak sense) and belongs to $L^p(\Omega)$.

Together with the norm

$$\|f\|_{W^{k,p}} = \begin{cases} \left(\sum_{|\alpha|\leqslant k}\int_U |D^\alpha f(u)|^p du\right)^{\frac{1}{p}} & 1 \leqslant p < \infty \\ \sum_{|\alpha|\leqslant k} \operatorname{ess\,sup}_U |D^\alpha f(u)| & p = \infty \end{cases} \tag{4.13}$$

We are mainly interested in the (Hilbert-) spaces $H^k(U) = W^{k,2}(U)$ and in particular $H^1(U)$.

For manifolds $\mathcal{X}$ partial derivatives do not make sense (since there is no canonical basis of the tangent space $T_p\mathcal{X}$). However we can also define *weak directional derivatives* via the equation

$$\int_{\mathcal{X}} df(p)[v]h(p)dp = -\int_{\mathcal{X}} f(p)dh(p)[v]dp \tag{4.14}$$

which together with $df(p)[v] = \langle \nabla f, v \rangle$ defines weak gradients of functions on manifolds and thus the space $H^1(\mathcal{X})$.
Finally the space

$$W_0^{k,p} = \overline{C_0^\infty}^{\|\cdot\|_{W^{k,p}}} \tag{4.15}$$

can be interpreted as the space of all $W^{k,p}(U)$ (and $W^{k,p}(\mathcal{X})$) functions with vanishing derivatives (up to $k-1$) at the boundary of U (in the L^p sense) (and $\mathcal{X}$ respectively).

The relevant function spaces for the rest of this chapter are $H_0^1(\mathcal{X})$ and its *dual space* $H^{-1}(\mathcal{X})$.
For a more precise and detailed introduction to Sobolev spaces we refer to [20].

4.2 The Laplacian acting on H_0^1

Definition 9 Let $f \in H_0^1(\mathcal{X})$, $\mathcal{X}$ being a m-dimensional manifold (possibly a flat subset of $\mathbb{R}^m$). The Laplacian of f is the unique element $\triangle f \in H^{-1}(\mathcal{X})$ such that

$$\int_{\mathcal{X}} \triangle f(p)h(p)dp = -\int_{\mathcal{X}} \langle \nabla f, \nabla h \rangle \, dp \quad \forall h \in H_0^1\mathcal{X} \tag{4.16}$$

It is equivalent to only "test" with functions $g \in C_0^\infty(\mathcal{X})$ since every element of H_0^1 is the limit of test functions (*c.f.* Equation (4.15)). We will next derive explicit representations for C^2 functions f on Euclidean spaces and manifolds. Afterwards we will present some properties of the Laplacian.
Let $U \subset \mathbb{R}^m$, $f \in C^2(U)$ and $h \in C_0^\infty(U)$. Then

$$\int_U \langle \nabla f(u), \nabla h(u) \rangle \, du = \int \sum_i \frac{\partial f(u)}{\partial u_i} \frac{\partial h(u)}{\partial u_i} \tag{4.17}$$

$$= -\int \sum_i \frac{\partial^2 f}{\partial u_i^2} h(u)du \tag{4.18}$$

$$\Rightarrow \triangle f = \sum_i \frac{\partial^2 f}{\partial u_i^2} \tag{4.19}$$

Thus the Laplacian as defined in Definition 9 generalizes the well known operator on C^2 functions.
Let next $f\colon \mathcal{X} \to \mathbb{R}$ be a C^2-function defined on a submanifold $\mathcal{X}$. We will derive an expression for $\triangle f$ in local coordinates and discover that the Laplacian is an *intrinsic operator*. The Laplacian of a C^2 function is a continuous function and it does make sense to ask for the value $\triangle f(p)$ at a point $p \in \mathcal{X}$. Let $x\colon U \to V \subset \mathcal{X}$ be a coordinate map with $p \in V = x(U)$ and $h \in C_0^\infty(V)$. Further $g\colon U \to \mathbb{R}^{m\times m}$ is the first fundamentalform with inverse $g^{-1} = (g^{ij})_{i,j=1\dots m}$ and $\tilde{f} = f \circ x^{-1}$, $\tilde{h} = h \circ x^{-1}$. Making use of Equation (3.28) and Equation (3.30) we get

$$\int_V \langle \nabla f, \nabla h \rangle dp = \int_U \left\langle g^{-1}(u)\nabla\tilde{f}(u), g^{-1}(u)\nabla\tilde{h}(u) \right\rangle_{g(u)} \sqrt{\det g(u)}du \tag{4.20}$$

$$= \int_U \left\langle \nabla\tilde{f}(u), g^{-1}(u)\nabla\tilde{h}(u) \right\rangle \sqrt{\det g(u)}du \tag{4.21}$$

$$= \int_U \left(\sum_{i,j=1}^m \frac{\partial\tilde{f}(u)}{\partial u_i} g^{ij}(u) \frac{\partial\tilde{h}(u)}{\partial u_j} \right) \sqrt{\det g(u)}du \tag{4.22}$$

$$= \int_U \left(\sum_{i,j=1}^m \sqrt{\det g(u)} \frac{\partial\tilde{f}(u)}{\partial u_i} g^{ij}(u) \frac{\partial\tilde{h}(u)}{\partial u_j} \right) du \tag{4.23}$$

Integration by parts gives:

$$= -\int_U \sum_{i,j=1}^m \frac{\partial}{\partial u_j} \left(\sqrt{\det g(u)} \frac{\partial\tilde{f}(u)}{\partial u_i} g^{ij}(u) \right) \tilde{h}(u)du \tag{4.24}$$

$$= -\int_U \frac{1}{\sqrt{\det g}} \sum_{i,j=1}^m \frac{\partial}{\partial u_j} \left(\sqrt{\det g} \frac{\partial\tilde{f}}{\partial u_i} g^{ij} \right) \tilde{h}\sqrt{\det g}du \tag{4.25}$$

Thus (in combination with the *Fundamental Lemma of the calculus of variations* $\triangle f$) has a representation solely in terms of derivatives if $\tilde{f}$ and g:

$$-\frac{1}{\sqrt{\det g(u)}} \sum_{i,j=1}^m \frac{\partial}{\partial u_j} \left(\sqrt{\det g(u)} \frac{\partial\tilde{f}(u)}{\partial u_i} g^{ij}(u) \right) \tag{4.26}$$

The Laplacian is therefor an *intrinsic* operator. Whenever g is the identity matrix (as for Euclidean spaces) we recover the well known Equation (4.1).

4.3 Properties of the Laplacian

In this section we summarize some properties of the Laplacian that are essential for the applications (*e.g.* heat equation or functional maps) presented in the following chapters.

4.3.1 Linearity

First notice that due to the linearity of integration and derivatives, also the Laplacian is a *linear operator*, *i.e.*

$$\triangle(f_1 + \alpha f_2) = \triangle f_1 + \alpha \triangle f_2 \tag{4.27}$$

for all $f_1, f_2 \in H_0^1$, $\alpha \in \mathbb{R}$.

4.3.2 Formal self adjointness

Further the Laplacian is *formally self-adjoint*

$$\langle \triangle f, h \rangle = \langle f, \triangle h \rangle \tag{4.28}$$

for every pair of functions f, h where the expression makes sense. Since H_0^1 is not a Hilbert space (in particular self dual) we cannot say that $\triangle$ is *self adjoint* but only call aboves property *formal self adjointness*.

4.3.3 Locality

The Laplacian is a *local operator*, *i.e.* changing the value $f(y)$ at any point $x \neq y$ does not affect the value of $\triangle f$. This can be seen by "testing" with a test function h with $y \notin \operatorname{supp} h$.

4.3.4 Mean Value formula

The value of the Laplacian $\triangle f(p)$ at a point $p \in \mathcal{X}$ has a neat interpretation: it measures how much the function value $f(p)$ differs from the values in its neighborhood.

$$\triangle f(p) = \lim_{r \to 0} \fint_{B_r(p)} f(q)dq - f(p) \tag{4.29}$$

where $B_r(p) = \{q \in \mathcal{X} | d(p,q) < r\}$ denotes a geodesic ball with radius r centered at p and $\fint_{B_r(p)} = \frac{1}{|B_r(p)|} \int_{B_r(p)}$ is the integral normaliced by the area of the integration domain. We only provide a proof for Euclidean domains $\mathcal{X} = U \subset \mathbb{R}^m$. The proof makes use of the first of *Greens' formulas* (see *e.g.* [20] for a proof).

Theorem 2 *(Green's formulas)* Let $U \subset \mathbb{R}^n$ be a bounded open subset of $\mathbb{R}^n$ with boundary $\partial U \in C^1$. For $f, h \in C^2(\bar{U})$

1. $\int_U \triangle f(u)du = \int_{\partial U} \frac{\partial f(p)}{\partial \nu} dp$

2. $\int_U Df(u) \cdot Dh(u)du = \int_U f(u) \triangle h(u)du + \int_{\partial U} \frac{\partial h(p)}{\partial \nu} f(p)dp$
3. $\int_U f(u) \triangle h(u) - h(u) \triangle f(u)du = \int_{\partial U} \frac{\partial h(p)}{\partial \nu} f(p) - \frac{\partial f(p)}{\partial \nu} h(p)dp$

where $\nu \colon \partial U \to \mathbb{R}^m$ is the outward pointing normal of U.

Notice that aboves formulas to do not assume f and h to vanish at the boundaries of U.
Let next $B_r(x)$ denote a ball of radius r around x and

$$V(r) = \fint_{\partial B_r(x)} f(p)dp \tag{4.30}$$

A first order Taylor expansion of V around $r = 0$ gives

$$\begin{aligned} V(r) - V(0) &\approx \frac{\partial}{\partial r} \fint_{\partial B_r(x)} f(q)dq && (4.31)\\ &= \frac{\partial}{\partial r} \fint_{\partial B_1(0)} f(x + rq)dq && (4.32)\\ &= \fint_{\partial B_1(0)} Df(x + rq) \cdot qdq && (4.33)\\ &= \fint_{\partial B_r(p)} Df(q) \cdot \frac{q - p}{r} dq && (4.34)\\ &= \fint_{\partial B_r(p)} \frac{\partial f(q)}{\partial \nu} dq && (4.35)\\ &= \fint_{\partial B_r(p)} \triangle f(q)dq && (4.36) \end{aligned}$$

So in the limit $r \to 0$ we get $\lim_{r \to 0} \fint_{\partial B_r(p)} f(q)dq - f(p) = \triangle f(p)$. Equation (4.29) can be obtained from $B_r(p) = \bigcup_{r' \leqslant 0} \partial B_{r'}(p)$.

4.3.5 Maximum principle

A function $f \in C^2(\mathcal{X})$ on a manifold $\mathcal{X}$ (which is open) is called *harmonic* if $\triangle f = 0$. As a consequence of of the *mean value formula* f does not provide a maximum in $\mathcal{X}$. In case $f \in C^2(\mathcal{X}) \cap C(\bar{\mathcal{X}})$:

$$\max_{\bar{\mathcal{X}}} f = \max_{\partial \mathcal{X}} f \tag{4.37}$$

Due to the linearity of the Laplacian the same holds for minima. If $\partial \mathcal{X} = \varnothing$ each harmonic function is constant.

4.3.6 Negative semidefinitness

The Laplacian is a negative semi-definite operator:

$$\langle \triangle f, f \rangle = -\int_{\mathcal{X}} \langle \nabla f, \nabla f \rangle \, dp \leqslant 0 \tag{4.38}$$

with equality iff f is constant.

4.4 Spectral decomposition of the Laplacian

In this section we consider the equation

$$\triangle f = \lambda f \tag{4.39}$$

in particular for functions defined on manifolds without boundary. We thus avoid the necessity of *boundary conditions*.
It can be shown that the Laplacian has a countable number of *eigen pairs* (λ_k, ϕ_k):

$$\triangle \phi_k = \lambda_k \phi_k \tag{4.40}$$

It is easy to see that all eigenvalues are real and non-positive:

$$\lambda_k \langle \phi_k, \phi_k \rangle_{\geqslant 0} = \langle \triangle \phi_k, \phi_k \rangle = -\int_{\mathcal{X}} \langle \nabla \phi_k, \nabla \phi_k \rangle \, dp \leqslant 0 \tag{4.41}$$

One can further show that $0 = \lambda_0 > \lambda_1 \geqslant ... \to -\infty$. Let further ϕ_k, ϕ_l be two eigenfunctions with distinct eigenvalues λ_k, λ_l. Then

$$\lambda_k \langle \phi_k, \phi_l \rangle = \langle \triangle \phi_k, \phi_l \rangle = \langle \phi_k, \triangle \phi_l \rangle = \langle \phi_k, \lambda_l \phi_l \rangle = \lambda_l \langle \phi_k, \phi_l \rangle \tag{4.42}$$

and thus $\langle \phi_k, \phi_l \rangle = 0$. With a bit of extra work (dimension of eigenspaces with repeating eigenvalues) we get the following Lemma:

Lemma 5 Let $\mathcal{X}$ be a compact manifold with a single connected component. There exists a sequence $\{\lambda_k\}_{k \in \mathbb{N}}$, $0 = \lambda_0 > \lambda_1 \geqslant ... \to -\infty$ and *orthonormal functions* $\phi_k \colon \mathcal{X} \to \mathbb{R}$ such that

$$\triangle \phi_k = \lambda_k \phi_k \tag{4.43}$$

The eigenfunctions ϕ_k of the Laplacian are also called *harmonics* and can be seen as generalizations of the *Fourier basis functions* $\exp(ikx)$. For functions defined on the real line, the Laplacian becomes the second derivative and it holds that

$$\frac{\partial^2}{\partial x^2} \exp(i\sqrt{\omega}x) = -\omega \exp(i\sqrt{\omega}x) \tag{4.44}$$

For compact domains $[a, b] \subset \mathbb{R}$ one gets a countable number of *eigenfrequencies* ω due to periodic boundary conditions ($\exp(i\sqrt{\omega}a) \stackrel{!}{=} \exp(i\sqrt{\omega}b)$). This is the analogon to the harmonics on compact manifolds.
An alternative way to define the eigenfunctions is as minimizers of the *Dirichlet energy* E_D:

$$\phi_k = \arg\min_{\|\phi\|=1} E_D(\phi) = \arg\min_{\|\phi\|=1} \int_{\mathcal{X}} \|\nabla\phi\|^2 \, dp \tag{4.45}$$

$$\text{s.t.} \langle \phi_k, \phi_l \rangle = 0 \qquad \forall l < k \tag{4.46}$$

The Dirichlet energy of an eigenfunction equals the absolut value of the corresponding eigenvalue.
Notice that the eigendecomposition is not unique. Let λ be an eigenvalue with corresponding eigenvector ϕ. Then also every vector $\alpha\phi$ ($\alpha \neq 0$) is an eigenvector to λ. Even by constraining the eigenvectors to have unit length there is still an ambiguity between the vectors ϕ and $-\phi$. The term *sign flip* is often used to address this ambiguity. In case of higher dimensional eigenspaces (repeating eigenvalues) every linear combination of the eigenvectors is again an eigenvector to the same eigenvalue. Again, constraining the eigenvectors to be orthonormal does not solve this ambiguity: Applying an orthogonal matrix R that is acting on the *eigenspace* spanned by the eigenvectors gives an alternative set of eigenvectors that are still orthonormal. Since the harmonics form an orthonormal basis every function $f \in L^2(\mathcal{X})$ can be written in that basis:

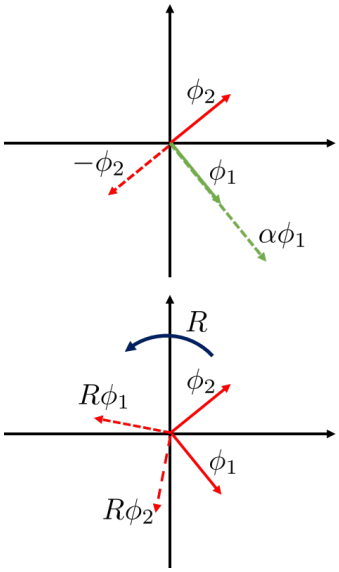

Figure 4.1: Ambiguity of eigenfunctions

$$f(p) = \sum_{k=0}^{\infty} c_k \phi_k(p) = \sum_{k=0}^{\infty} \langle f, \phi_k \rangle \phi_k(p) \tag{4.47}$$

where the coefficients are given by the products $\langle f, \phi_k \rangle = \int_{\mathcal{X}} f(p)\phi(p)dp$.
We have seen that the harmonics form an orthonormal basis and are ordered based on their Dirichlet energy which is a measure of variability. The harmonics and the spectrum (set of eigenvalues) of the Laplacian have some more interesting properties. First note that integrating the Laplacian of any function over the manifold gives

$$\int_{\mathcal{X}} \triangle f = \int_{\mathcal{X}} 1 \triangle f = -\int_{\mathcal{X}} \langle \nabla 1, \nabla f \rangle = 0 \tag{4.48}$$

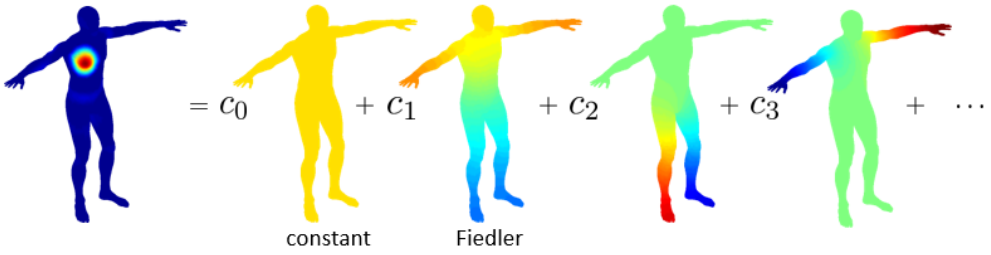

Figure 4.2: Every function $f \in L^2(\mathcal{X})$ can be written in the Laplace-Beltrami-Eigen-basis: $f(p) = \sum_{k=0}^{\infty} c_k \phi_k(p)$. For compact manifolds the first eigenfunction ϕ_0 corresponding to the eigenvalue $\lambda_0 = 0$ is constant. The eigenfunction ϕ_1 corresponding to the smallest (as absolute value) eigenvalue is also called *Fiedler vector*. (Mesh from the TOSCA dataset [14])

which implies that the mean value of every eigenfunction ϕ_k with corresponding eigenvalue $\lambda_k \neq 0$ is zero:

$$\int_{\mathcal{X}} \phi_k = \frac{1}{\lambda_k} \int_{\mathcal{X}} \Delta \phi_k = 0$$

The eigenfunction ϕ_0 with corresponding eigenvalue $\lambda_0 = 0$ is constant, $\phi_0(p) = c$. Due to the normalization we get

$$1 = \int_{\mathcal{X}} \phi_0^2(p) dp = c^2 \text{area}(\mathcal{X}) \Leftrightarrow c = \pm \frac{1}{\sqrt{\text{area}(\mathcal{X})}}$$

Let $\tilde{\mathcal{X}} = \alpha \mathcal{X}$ be a scaled version of $\mathcal{X}$ and (λ, ϕ) be an eigenpair of the Laplacian $\triangle_{\mathcal{X}}$ on $\mathcal{X}$. Then the pair $(\tilde{\lambda}, \tilde{\phi})$ with $\tilde{\lambda} = \frac{1}{\alpha^2}\lambda$ and $\tilde{\phi}(y) = \frac{1}{\alpha}\phi(\frac{y}{\alpha})$ is an eigenpair of $\triangle_{\tilde{\mathcal{X}}}$:

$$\int_{\tilde{\mathcal{X}}} \langle \triangle \tilde{\phi}(y), \tilde{h}(y) \rangle dy = -\int_{\mathcal{X}} \langle \nabla \tilde{\phi}(y), \nabla \tilde{h}(y) \rangle dy \tag{4.49}$$

$$= -\int_{\mathcal{X}} \langle \frac{1}{\alpha^2} \nabla \phi(x), \nabla \frac{1}{\alpha} \tilde{h}(\alpha x) \rangle \alpha^2 dx \tag{4.50}$$

$$= \frac{1}{\alpha} \int_{\mathcal{X}} \langle \triangle \phi(x), \tilde{h}(\alpha x) \rangle dx \tag{4.51}$$

$$= \frac{\lambda}{\alpha} \int_{\mathcal{X}} \langle \phi(x), \tilde{h}(\alpha x) \rangle dx \tag{4.52}$$

$$= \frac{\lambda}{\alpha} \int_{\tilde{\mathcal{X}}} \langle \alpha \tilde{\phi}(y), \tilde{h}(y) \rangle \frac{1}{\alpha^2} dy \tag{4.53}$$

$$= \tilde{\lambda} \int_{\tilde{\mathcal{X}}} \langle \tilde{\phi}(y), \tilde{h}(y) \rangle dy \tag{4.54}$$

The function $\tilde{\phi}(y)$ also has unit norm:

$$\int_{\tilde{\mathcal{X}}} \left\| \tilde{\phi}(y) \right\|^2 dy = \int_{\mathcal{X}} \left\| \frac{1}{\alpha} \phi(x) \right\|^2 \sqrt{\det g(x)} dx = \int_{\mathcal{X}} \left\| \tilde{\phi}(x) \right\|^2 dy = 1 \tag{4.55}$$

As a consequence of *Weyl's law* [55] or more precisely its extension to compact closed manifolds by Levitan [31] we also get the following asymptotic approximation for the eigenvalues:

$$|\lambda_j| \approx \frac{\pi}{\text{area}(\mathcal{X}) j} \quad (j \to \infty) \tag{4.56}$$

Finally remember that the Laplacian is an intrinsic operator. As a consequence both its spectrum an its eigenspaces are consistent across a pair of *isometric shapes*. We will elaborate on this in the following chapter.

4.5 Discretizing the Laplacian

We want to define a *discrete Laplace operator* acting on functions defined on triangle meshes (*c.f.* Section 3.2). Let us first review the definition of the Laplacian on a compact manifold manfold $\mathcal{X}$ without boundary:

$$\int_{\mathcal{X}} \triangle f(p) h(p) dp = -\int_{\mathcal{X}} \langle \nabla f, \nabla h \rangle \, dp \tag{4.57}$$

for all $f, h \in H^1(\mathcal{X})$ (which equals $H_0^1(\mathcal{X})$ since $\partial \mathcal{X} = \varnothing$). The approach is based on the *Ritz-Galerkin method* [43]: instead of demanding Equation (4.57) for all

functions of the *infinite dimensional space* $H^1(\mathcal{X})$ we only ask it to hold on a finite dimensional *Ansatz space* $\mathcal{A}$ (where f is defined) and *Test space* $\mathcal{T}$ (where h is defined):

$$\int_{\mathcal{X}} \triangle f(p)h(p)dp = -\int_{\mathcal{X}} \langle \nabla f, \nabla h \rangle \, dp \quad \forall f \in \mathcal{A},\, h \in \mathcal{T} \tag{4.58}$$

The Ritz-Galerkin method sets $\mathcal{A} = \mathcal{T}$. By increasing the dimensions and under some additional regularity requirements, one can show that the discrete operator converges to the smooth Laplacian. We refer to any text book on numerics of PDEs (finite element methods) for more details on convergence. In the context of triangle meshes we choose $\mathcal{A}$ and $\mathcal{T}$ to be the space of piecewise linear functions as defined in Section 3.2.1. Let now f and h be two piecewise linear functions on a triangle mesh $\mathcal{X}$ given by their coefficients $\mathbf{f}, \mathbf{h} \in \mathbb{R}^{|V|}$. The discrete Laplacian is an $|V| \times |V|$ matrix $\mathbf{L}$ such that $\mathbf{Lf} = \mathbf{k} \in \mathbb{R}^{|V|}$ are the coefficients of the (again piecewise linear) function k satisfying

$$\int_{\mathcal{X}} k(p)h(p) = -\int_{\mathcal{X}} \langle \nabla f, \nabla h \rangle \, dp \tag{4.59}$$

We have already seen that we can write the left term using the *mass matrix* $\mathbf{M}$ (*c.f.* Equation (3.41)):

$$\int_{\mathcal{X}} k(p)h(p) = \mathbf{hMk} \tag{4.60}$$

with $\mathbf{M}_{ij} = \int_{\mathcal{X}} \psi_i(p)\psi_j(p)dp$. It further holds

$$\int_{\mathcal{X}} \langle \nabla k, \nabla h \rangle \, dp = \mathbf{hSf} \tag{4.61}$$

$$\mathbf{S}_{ij} = \int_{\mathcal{X}} \langle \nabla \psi_i, \nabla \psi_j \rangle \, dp \tag{4.62}$$

$$\tag{4.63}$$

We call the matrix $\mathbf{S}$ *stiffness matrix*. Cobining aboves equations we see

$$\mathbf{hMk} = -\mathbf{hSf} \qquad \forall \mathbf{h} \in \mathbb{R}^{|V|} \tag{4.64}$$

$$\Rightarrow \mathbf{k} = -\mathbf{M}^{-1}\mathbf{Sf} \tag{4.65}$$

$$\Rightarrow \mathbf{L} = -\mathbf{M}^{-1}\mathbf{S} \tag{4.66}$$

The stiffness matrix is often referred to as *cotangent matrix*, the following derivation shows why. As in Section 3.2.2 we decompose the integral $\mathbf{S}_{ij} = \int_{\mathcal{X}} \langle \nabla \psi_i, \nabla \psi_j \rangle \, dp$ into the sum of integrals over all triangles and see that

$$\mathbf{S}_{ij} = \int_{\mathcal{X}} \langle \nabla \psi_i, \nabla \psi_j \rangle \, dp = \int_{t_k} \langle \nabla \psi_i, \nabla \psi_j \rangle \, dp + \int_{t_{k'}} \langle \nabla \psi_i, \nabla \psi_j \rangle \, dp \tag{4.67}$$

where t_k and $t_{k'}$ are the two triangles sharing the edge (v_i, v_j). Let $t_k = (v_l, v_i, v_j)$ be one of them and the parametrization be as in Equation (3.46) (*c.f.* Figure 4.3 for notation). Then

$$\int_{t_k} \langle \nabla\psi_i, \nabla\psi_j \rangle \, dp = \int_0^1 \int_0^{1-u_1} \left\langle g^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix}, g^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\rangle_g \sqrt{\det(g)} du \tag{4.68}$$

$$= \frac{1}{2} \begin{pmatrix} 1 & 0 \end{pmatrix} \frac{1}{\det(g)} \begin{pmatrix} \|e_{jl}\|^2 & -\langle e_{il}, e_{jl} \rangle \\ -\langle e_{il}, e_{jl} \rangle & \|e_{il}\|^2 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \sqrt{\det(g)} \tag{4.69}$$

$$= -\frac{1}{2} \frac{\langle e_{il}, e_{jl} \rangle}{\sqrt{\det(g)}} = -\frac{1}{2} \frac{\|e_{il}\| \, \|e_{jl}\| \cos(\alpha_{ilj})}{\|e_{il}\| \, \|e_{jl}\| \sin(\alpha_{ilj})} \tag{4.70}$$

$$= -\frac{1}{2} \cot(\alpha_{ilj}) \tag{4.71}$$

and consequently

$$\tag{4.72}$$

$$\mathbf{S}_{ij} = \begin{cases} 0 & v_i \neq v_j \hat{(} v_i, v_j) \notin \mathcal{E} \\ -\frac{1}{2} \left(\cot(\alpha_{ilj}) + \cot(\alpha_{il'j}) \right) & (v_i, v_j) \in \mathcal{E} \\ -\sum_{l \neq i} S_{il} & v_i = v_j \end{cases} \tag{4.73}$$

The last equality can be derived analogously but also follows directly from

$$\mathbf{S}_{ii} = \langle \nabla\psi_i, \nabla\psi_i \rangle \tag{4.74}$$

$$= \left\langle \nabla(1 - \sum_{l \neq i} \psi_l), \nabla\psi_i \right\rangle \tag{4.75}$$

$$= -\sum_{l \neq i} \langle \nabla\psi_l, \nabla\psi_i \rangle = -\sum_{l \neq i} S_{il} \tag{4.76}$$

Let us enumerate some of the important properties of the stiffness matrix $\mathbf{S}$:

1. The stiffness matrix is sparse, with support on the edges (just as the mass matrix).

2. The stiffness matrix is symmetric.

3. The stiffness matrix is positive semidefinit with a onedimensional kernel spanned by the contant vector $\mathbf{1} \in \mathbb{R}^{|V|}$.

We want to address concerns regarding possibly surprising properties of the discrete Laplacian $\mathbf{L}$

1. The matrix $\mathbf{M}^{-1}$ is dense (support everwhere) and consequently is $\mathbf{L}$

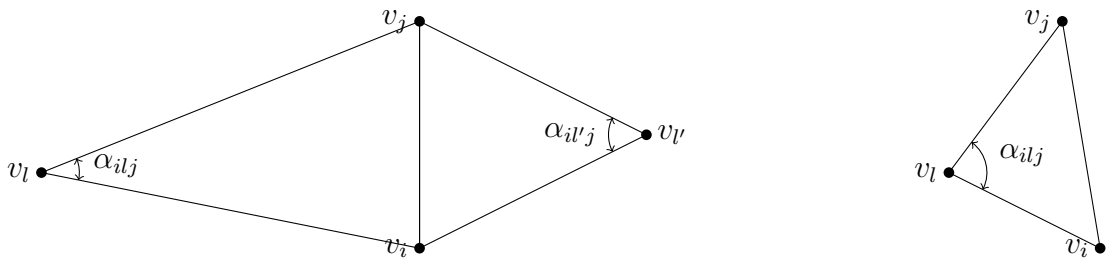

Figure 4.3: Left: The stiffness matrix is a symmetric sparse matrix. $\mathbf{S}_{ij} = \mathbf{S}_{ji}$ is non-zero only if the vertices v_i and v_j share an edge. In this case the value equals the negative mean of cotangents of the angles opposing the edge: $S_{ij} = -\frac{1}{2}\left(\cot(\alpha_{ilj}) + \cot(\alpha_{il'j})\right)$. This is why the stiffness matrix is commonly referred to as the *cotangent matrix*. Right: If the edge is a boundary edge there is only one angle opposing it and the sum reduces to a single term.

2. $\mathbf{L}$ is *not* symmetric. This even holds when approximating $\mathbf{M}$ with a diagonal matrix (area elements)

The main concern regarding the first property is that the application of the Laplacian to a function becomes a dense matrix vector product and thus expensive ($O(|V|^2)$). This would in particular make algorithms slow that have to apply the Laplacian multiple times (such as power iterations to solve eigenvalue problems). However in most applications one can just multiply the arising equations with $\mathbf{M}$

$$\mathbf{Lf} = \mathbf{k} \Leftrightarrow -\mathbf{Sf} = \mathbf{Mk} \tag{4.77}$$

and thus transform the dense matrix vector product into two sparse matrix vector products. In particular we can rewrite the eigenvalue problem as a *generalized eigenvalue problem*:

$$\mathbf{L}\boldsymbol{\phi} = \lambda\boldsymbol{\phi} \Leftrightarrow -\mathbf{S}\boldsymbol{\phi} = \lambda\mathbf{M}\boldsymbol{\phi} \tag{4.78}$$

for which dedicated solvers exist [30, 51] and are available in computing toolboxes such as MATLAB and SciPy. The resulting eigenvectors $\boldsymbol{\phi}_k$ will not be orthogonal with respect to the standard inner product. This is a consequence of $\mathbf{L}$ not being symmetric. However the vectors will be orthogonal with respect to the $\mathbf{M}$-inner product

$$\begin{aligned}
\lambda_k \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_l \rangle_{\mathbf{M}} &= \langle \lambda_k \mathbf{M}\boldsymbol{\phi}_k, \boldsymbol{\phi}_l \rangle && (4.79)\\
&= \langle -\mathbf{S}\boldsymbol{\phi}_k, \boldsymbol{\phi}_l \rangle && (4.80)\\
&= \langle \boldsymbol{\phi}_k, -\mathbf{S}\boldsymbol{\phi}_l \rangle && (4.81)\\
&= \lambda_l \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_l \rangle_{\mathbf{M}} && (4.82)\\
\Rightarrow \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_l \rangle_{\mathbf{M}} &= 0. && (4.83)
\end{aligned}$$

for $\lambda_k \neq \lambda_l$. (The proof for real and non-positive eigenvalues goes along the same lines.) This however is exactly what we want as $\mathbf{M}$-orthogonal *eigenvectors* correspond to orthogonal *eigenfunctions* ϕ_k and ϕ_l:

$$\langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_l \rangle_{\mathbf{M}} = \int_{\mathcal{X}} \phi_k(p)\phi_l(p)dp = 0 \tag{4.84}$$

In matrix notation the $\mathbf{M}$-orthogonality reads

$$\boldsymbol{\Phi}^T \mathbf{M} \boldsymbol{\Phi} = \boldsymbol{I} \tag{4.85}$$

where $\boldsymbol{\Phi}$ stores all eigenfunctions as columns and $\boldsymbol{I} \in \mathbb{R}^{|V|\times|V|}$ is the identity matrix. Let $\tilde{X}$ be the triangle mesh obtained by scaling $\mathcal{X}$ with α. The elements of the mass matrix $\tilde{\mathbf{M}}$ of $\tilde{\mathcal{X}}$ are scaled with the factor α^2 (since they depend on the area of the triangles), the elements of the new stiffness matrix $\tilde{\mathbf{S}}$ remain the same (since the angles do not change). Thus for every eigenpair (λ_k, ϕ_k) of the discrete Laplacian on $\mathcal{X}$ there is an eigenpair $(\tilde{\lambda}_k, \tilde{\phi}_k)$ on $\tilde{\mathcal{X}}$ with

$$\tilde{\lambda}_k = \frac{1}{\alpha^2}\lambda_k \tilde{\phi}_k = \frac{1}{\alpha}\phi_k \tag{4.86}$$

This is consistent with the behaviour of the continuous Laplacian.
Alternative discretizations of the Laplacian can be obtained by using higher order finite element approaches [42], mixed finite element methods or considering the relation of the Laplacian with the curvature of a surface [35]. The variety of discretizations is tightly connected to the observation that for general triangle meshes there is no discrete Laplacian satisfying a set of desireable properties [10, 54]:

NULL $\triangle f = 0$ whenever f is constant

SYM Symmetry: $\langle \triangle f, h \rangle = \langle f, \triangle h \rangle$ whenever f and h are sufficiently smooth and vanish along the boundary of $\mathcal{X}$.

LOC Local support: for any pair $p \neq q$ of points, $\triangle f(p)$ is independent of $f(q)$. Altering the function value at a distant point will not affect the action of the Laplacian locally

LIN Linear precision: $\triangle f = 0$ whenever $\mathcal{X}$ is part of the Euclidean plane, and $f = ax + by + c$ is a linear function on the plane.

MAX Maximum principle: harmonic functions (those for which $\triangle f = 0$ in the interior of $\mathcal{X}$) have no local maxima (or minima) at interior points.

NSD Negative semi-definiteness: the Dirichlet energy, $E_D(f) = \int_{\mathcal{X}} \|\nabla f\|^2 \, dp$, is non-negative. By our choice of sign for $\triangle$, we obtain $E_D(f) = -\langle \triangle f, f \rangle \leqslant 0$ whenever f is sufficiently smooth and vanishes along the boundary of $\mathcal{X}$.

Our discretization in particular lacks the property **LOC** (non-zero entries only at edges) due to the non-diagonal structure of the mass matrix $\mathbf{M}$.

4.6 The heat equation

Apart from the *Helmholtz equation* (4.39) the most prominent partial differential equation is the *heat equation*. Again, we avoid a discussion regarding boundary conditions by considering compact manifolds without boundary as the domain. In this case the heat equation is given by:

$$\partial_t u = c \triangle u \tag{4.87}$$

$$u(0, \cdot) = u_0 \tag{4.88}$$

where $u = u(t, x)$ is a function defined on $\mathbb{R}^{\geqslant 0} \times \mathcal{X}$ (the first argument is interpreted as time), $c > 0$ is a constant and the Laplacian $\triangle$ is acting on the latter argument as described in the previous sections. If the function $u_0 \colon \mathcal{X} \to \mathbb{R}^{\geqslant 0}$ described the initial distribution of heat on $\mathcal{X}$, $u(t, \cdot)$ defines the distribution of heat at time t. From Equation (4.29) we know that the Laplacian $\triangle u(t, x)$ measures the difference of the average of $u(t, \cdot)$ in its neighborhood to the value at $x \in \mathcal{X}$ itself. Thus if $\triangle u(t, x) < 0$ the temperature at x is higher than the average temperature in its neighborhood and thus the temperature at this point is decreasing: $\partial_t u = c \triangle u < 0$. We want to point out that (4.87) is a simplified approximation of the real diffusion process: Analyzing (4.87) shows that heat diffuses with infinite speed: $u(t, \cdot) > 0 \, \forall t > 0$ which is physically not possible.
We will now derive an explicit expression for the solution of the heat equation in terms of eigenfunctions and eigenvalues of the Laplacian. Just for the sake of simplicity we assume $c = 1$ but all what follows generalizes to arbitrary values of c in a straight forward manner. We can write

$$u(x, t) = \sum_k c_k(t) \phi_k(x) \tag{4.89}$$

as for each t, the function $u(\cdot, t) \in L^2(\mathcal{X})$. The coefficients c_k vary with t. Applying the differential operators ∂_t and $\triangle$ to (4.89) yields

$$\partial_t u(x, t) = \partial_t \sum_k c_k(t) \phi_k(x) \tag{4.90}$$

$$= \sum_k c_k'(t) \phi_k(x) \tag{4.91}$$

$$\triangle u(x, t) = \triangle \sum_k c_k(t) \phi_k(x) \tag{4.92}$$

$$= \sum_k c_k(t) \triangle \phi_k(x) \tag{4.93}$$

$$= \sum_k c_k(t) \lambda_k \phi_k(x) \tag{4.94}$$

and consequently by plugging this into (4.87)

$$\sum_k c_k'(t)\phi_k(x) = \sum_k c_k(t)\lambda_k\phi_k(x) \quad \forall t \in \mathbb{R}^{\geqslant 0}\, \forall x \in \mathcal{X} \tag{4.95}$$

Consequently by comparison of the coefficients:

$$c_k'(t) = \lambda_k c_k(t) \forall t \tag{4.96}$$

$$\Rightarrow c_k(t) = c_k(0)\exp(\lambda_k t) \tag{4.97}$$

The initial coefficients $c_k(0)$ are given by the initial distribution of heat via

$$c_k(0) = \langle u_0, \phi_k \rangle \tag{4.98}$$

$$= \int_{\mathcal{X}} u_0(x)\phi_k(x)dx \tag{4.99}$$

This follows directly from the orthogonality of the ϕ_k's and the previous formulas:

$$\int_{\mathcal{X}} u_0(x)\phi_k(x)dx = \int_{\mathcal{X}} \left(\sum_l c_l(0)\phi_l(x)\exp(\lambda_l \cdot 0) \right) \phi_k(x)dx \tag{4.100}$$

$$= \sum_l c_l(0) \left(\int_{\mathcal{X}} \phi_l(x)\phi_k(x)dx \right) = c_k(0) \tag{4.101}$$

4.6.1 Heat kernels

There is yet an other way to express the solution of the heat equation:

$$u(t,x) = \int_{\mathcal{X}} k_t(x,y)u_0(y)dy \tag{4.102}$$

where k_t are the so called *heat kernels*. Intuitively $k_t(x,y)$ describes what fraction of heat is transported from location $y \in \mathcal{X}$ to $x \in \mathcal{X}$ at time t. The total amount of heat at $x \in \mathcal{X}$ is then given by integrating over all sources $y \in \mathcal{X}$ weighted by the initial amount of heat. The heat kernel can also be thought as the solution of the heat equation with initial distribution $u_0 = \delta_y$. $\delta_y = \delta_0(\cdot - y)$ is no classical function but the *dirac distribution* [16] satisfying

$$\langle f, \delta_y \rangle = f(y) \quad \forall f \in C^1(\mathcal{X}) \tag{4.103}$$

It is easy to see that

$$k_t(x,y) = \sum_k \exp(\lambda_k t)\phi_k(x)\phi_k(y) \tag{4.104}$$

using the formulas from the previous section. We will come back to heat kernels and their applications as *descriptors* in Sections 5.3 and 5.4. For more theory and solid math we refer the reader to [22].

Chapter 5

Correspondences between Shapes

"Shape matching", "Shape correspondence", "Registration of shapes" - all these terms refer to the task of finding a mapping $m\colon \mathcal{X} \to \mathcal{Y}$ between two non-rigid shapes. The mapping should map semantically similar regions to each other. In some scenarios such a mapping does not exist as some regions that are visible on $\mathcal{X}$ are not present on $\mathcal{Y}$. In this chapter we assume *consistent visibility*, *i.e.* for each region on $\mathcal{X}$ there exists a corresponding region on $\mathcal{Y}$ and vice versa. In other words the sought mapping $m\colon \mathcal{X} \to \mathcal{Y}$ is a *bijection*. Moreover, m and m^{-1} should be *regular*, *i.e.* map closeby points to closeby points. Formally we want them to be C^1 functions. To summarize, the correpondence $m\colon \mathcal{X} \to \mathcal{Y}$ should satisfy three properties:

1. It should be bijective.
2. Both, m and m^{-1} should be differentiable.
3. m should map semantically meaningful points (*e.g.* the tips of the noses) onto each other.

The first two properties can be summarized as m being a *diffeomorphism*. It turns out to be difficult to translate the concept of diffeomorphisms to the discrete world where only a finite amount of numbers are available to represent a correspondence. We will introduce *permutations* as discrete counterparts to bijections and present an approach to discretize the concept of differentiability.
Correspondence problems are most commonly phrased as optimization problems. The requirement of bijectivity turns out to be a computational bootleneck. We will present some variants of *relaxations* as possible way outs. In addition to all aforementioned difficulties it is not clear, how to formalize the third requirement: How can we translate "semantically similar" into mathematical terms? This is still an unanswered question, nowadays commonly approached via supervised (deep) machine learning methods [18, 32]. In scenarios without access to labeled ground truth data, a common assumption is that the considered shapes are "near isometric",

i.e. m is an almost isometry. Points on $\mathcal{X}$ and $\mathcal{Y}$ can thus be compared using descriptors that are *invariant under isometries*. We will provide details regarding isometries in the following section and examples for isometry invariant features in the next chapter.

5.1 Isometries

A common approach for non-rigid shape correspondence is to look for a mapping that is approximately an isometry:

Definition 10 A bijective mapping $m\colon \mathcal{X} \to \mathcal{Y}$ between to metric spaces $\mathcal{X}$ and $\mathcal{Y}$ is called an *isometry* if it preserves all pairwise distances:

$$D^{\mathcal{X}}(x_1, x_2) = D^{\mathcal{Y}}(m(x_1), m(x_2)) \quad \forall x_1, x_2 \in \mathcal{X} \tag{5.1}$$

If such a mapping exists, the spaces $\mathcal{X}$, $\mathcal{Y}$ are called *isometric*.

Articulated bodies (of animals or humans) are often "almost isometric" and one is thus seeking a mapping m such that

$$D^{\mathcal{X}}(x_1, x_2) \approx D^{\mathcal{Y}}(m(x_1), m(x_2)) \tag{5.2}$$

.

5.1.1 Iterative Closest Point algorithm for rigid alignment of point sets

An important example (not directly applicable to non-rigid shapes though) are *Euclidean isometries*. Consider two subsets $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^n$ of a Euclidean space together with the *Euclidean metrics* $d_X(x_1, x_2) = \|x_1 - x_2\|$, $d_Y(x_1, x_2) = \|y_1 - y_2\|$. Each isometry between $\mathcal{X}$ and $\mathcal{Y}$ is a *rigid motion*:

$$m(x) = \mathbf{R}x + \mathbf{t} \qquad \mathbf{R} \in \mathbf{O(n)},\, \mathbf{t} \in \mathbb{R}^{\mathbf{n}} \tag{5.3}$$

Given two sets $\mathcal{X} = \{x_1, \ldots, x_{m_\mathcal{X}}\}$, $\mathcal{Y} = \{y_1, \ldots, y_{m_\mathcal{Y}}\}$ finding the optimal rotation and translation to align those two is commonly tackled using the *iterative closest point* (ICP) algorithm [5, 9]:

- For each point x_i find closest point in $\mathcal{Y}$ and call it z_i
- Find the optimal pair $\mathbf{R}$, $\mathbf{t}$ that minimizes $\sum_i \|\mathbf{R}x_i + t - z_i\|^2$
- transform $\mathcal{X}$ using the obtained $\mathbf{R}$, $\mathbf{t}$
- iterate the process

Notice that in the ICP algorithm every (transformed) element of $\mathcal{X}$ is querying it's nearest neighbor in every iteration. A naive implementation of the nearest neighbor search would thus lead to a complexity of $O(\#\text{iter} \times m_\mathcal{X} \times m_\mathcal{Y})$. An efficient data structure for nearest neighbor searches are *kd trees* [7]. They are binary trees, where each non-leaf node encodes a hyperplane, dividing the samples from the training set (the set where we will later query for nearest neighbors, $\mathcal{Y}$ above) represented by that node into two sets of (almost) same size. In Figure 5.1 we depict an example for a kd tree in two dimensions. kd trees are constructed by

- assigning the set $\mathcal{Y}$ to the root node
- recursively
  - pick one of the dimensions $1 \leqslant j \leqslant n$ (in the original paper the space dimension n was denoted k, thus the name).
  - create two child nodes and assign every sample with j-value smaller or equal to the median (in dimension j) to the left child and the others to the right child (this guarantees a balanced tree in the end)
- stop the iteration once the number of the assigned samples reaches a certain threshold (usually 1)

This construction is of complexity $O(m_\mathcal{Y} \times \log m_\mathcal{Y})$ and only happens once. In many cases finding the nearest neighbor of $x \in \mathcal{X}$ can be performed in $O(\log m_\mathcal{Y})$ by traversing down the tree, dependent on the values of the jth entry of x in every node, possibly checking a small set of neighboring branches. In degenerate cases one may need to backtrack and check many or even all branches of the tree, leading to $O(m_\mathcal{Y})$ complexity. This, however, can be neglected in practive such that the total complexity of querying nearest of all elements of $\mathcal{X}$ in $\mathcal{Y}$ is considered to be $O(m_\mathcal{X} \times \log m_\mathcal{Y})$ What does have a major influence of the complexity though is the space dimension n. As a consequence kd trees are only efficient for $m \gg 2^n$. While in classical rigid alignment this should most often be the case ($n = 3$) it may not hold true when aligning point sets in higher dimensional *feature spaces*.
We will next show that there exists a closed form solution for the optimal rigid motion aligning two point sets with given correspondences. The following Lemma will help us finding the optimal rotation.

Lemma 6 Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a symmetric, positive definite matrix. Then

$$\arg\max_{\tilde{R} \in O(n)} \operatorname{tr}(\tilde{\mathbf{R}}^\top \mathbf{A}) = \mathbf{I} \tag{5.4}$$

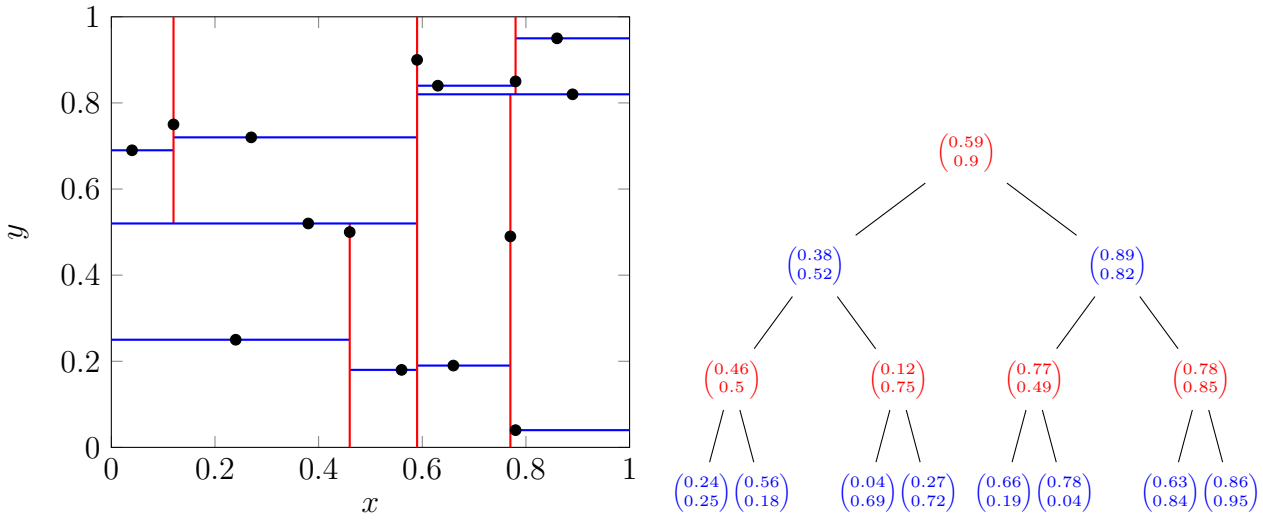

Figure 5.1: Example of a kd Tree in two dimensions. Each node of the tree (right) represents one hyperplane (left). The colors indicate the splitting dimensions. For a given query point, *e.g.* $\begin{pmatrix}0.2 & 0.2\end{pmatrix}^\top$ traversing down the tree already gives good candidates for possible nearest neighbors. There may however be cases where the actual nearest neighbor is in a very different branch than expected. The query point $\begin{pmatrix}0.6 & 0.2\end{pmatrix}^\top$ would land in the 5th leave while the actual nearest neighbor corresponds to the 4th leave. Backtracking from one to the other is almost as expensive as doing the naive all-to-all comparison. However this case happens very seldom and in practice approximate nearest neighbors are often sufficient.

Proof. Let us write $\mathbf{A}$ in terms of its *positive* eigenvalues λ_i and *orthogonal* eigenvectors ϕ_i and see that

$$\begin{aligned}
\operatorname{tr}(\tilde{\mathbf{R}}^\top \mathbf{A}) &= \operatorname{tr}(\tilde{\mathbf{R}}^\top \sum_i \lambda_i \phi_i \phi_i^\top) && (5.5)\\
&= \sum_i \lambda_i \operatorname{tr}(\tilde{\mathbf{R}}^\top \phi_i \phi_i^\top) && (5.6)\\
&= \sum_i \lambda_i \underbrace{(\tilde{\mathbf{R}}\phi_i)^\top \phi_i}_{\leqslant 1} && (5.7)\\
&\leqslant \sum_i \lambda_i = \operatorname{tr}(\mathbf{A}) && (5.8)
\end{aligned}$$

In case $\mathbf{A}$ is just positive semi definite (does not have full rank), there are equivalent minimizers (product of eigenvectors corresponding to 0 eigenvalue does not affect the sum) but the identity is still one of them. □

Theorem 3 Let $\{x_1, \ldots, x_m\} \subset \mathbb{R}^n$ and $\{z_1, \ldots, z_m\} \subset \mathbb{R}^n$ be two point sets. Then a solution of

$$(R^*, \mathbf{t}^*) = \underset{\mathbf{R}\in O(n), \mathbf{t}\in\mathbb{R}^n}{\arg\min} \sum_{i=1}^{m} \|\mathbf{R}x_i + t - z_i\|^2 \tag{5.9}$$

is given by

$$\mathbf{t}^* = \bar{z} - \mathbf{R}^*\bar{x} \tag{5.10}$$

$$\mathbf{R}^* = \mathbf{U}\mathbf{V}^\top \tag{5.11}$$

where $\bar{x} = \frac{1}{m}\sum x_i$ and $\bar{z} = \frac{1}{m}\sum z_i$ are the respective means of the point sets and $\mathbf{M} = \mathbf{U}\mathbf{D}\mathbf{V}^\top$ is the singular value decomposition of $\mathbf{M} = \sum_i (z_i - \bar{z})(x_i - \bar{x})^\top$. If $\mathbf{M}$ has full rank the solution is unique.

Proof. Let us first assume $\mathbf{R}^*$ to be given and solve for $\mathbf{t}^*$ by only using the orthogonality of $\mathbf{R}^*$. We start by adding a zero and introducing $\tilde{t} = \mathbf{R}^{*\top}(\bar{z} - t)$:

$$\begin{aligned}
\sum_{i=1}^{m} \|\mathbf{R}^* x_i + t - z_i\|^2 &= \sum_{i=1}^{m} \|\mathbf{R}^* x_i - \bar{z} + t - z_i + \bar{z}\|^2 && (5.12)\\
&= \sum_{i=1}^{m} \left\|\mathbf{R}^*(x_i - \mathbf{R}^{*\top}(\bar{z} - t)) - (z_i - \bar{z})\right\|^2 && (5.13)\\
&= \sum_{i=1}^{m} \left\|\mathbf{R}^*(x_i - \tilde{t}) - (z_i - \bar{z})\right\|^2 && (5.14)\\
&= \sum_{i=1}^{m} \left\|x_i - \tilde{t}\right\|^2 - 2\sum_i (z_i - \bar{z})^\top \mathbf{R}^*(x_i - \tilde{t}) + \sum_{i=1}^{m} \|z_i - \bar{z}\|^2 && (5.15)
\end{aligned}$$

We solve for $\tilde{t}$ (and thus for $\mathbf{t}^*$) by taking the gradient with respect to $\tilde{t}$ and setting it to zero:

$$0 = 2\sum_{i=1}^{m}(x_i - \tilde{t}) - 2\sum_{i=1}^{m}\mathbf{R}^{*\top}(z_i - \bar{z}) \tag{5.16}$$

$$= 2\sum_{i=1}^{m}(x_i - \tilde{t}) - 2\mathbf{R}^{*\top}\sum_{i=1}^{m}(z_i - \bar{z}) \tag{5.17}$$

$$= 2\sum_{i=1}^{m}(x_i - \tilde{t}) - 2\mathbf{R}^{*\top}\left(\sum_{i=1}^{m} z_i - m\bar{z}\right) \tag{5.18}$$

$$= 2\sum_{i=1}^{m}(x_i - \tilde{t}) \tag{5.19}$$

$$\Rightarrow \tilde{t} = \frac{1}{m}\sum_{i=1}^{m} x_i \quad \Leftrightarrow \mathbf{t}^* = \bar{z} - \mathbf{R}^*\bar{x} \tag{5.20}$$

For finding the optimal $\mathbf{R}^*$ we rewrite

$$\sum_{i=1}^{m}\|\mathbf{R}x_i + t^* - z_i\|^2 = \sum_{i=1}^{m}\|\mathbf{R}(x_i - \bar{x}) - (z_i - \bar{z})\|^2 \tag{5.21}$$

$$= \sum_{i=1}^{m}\|x_i - \bar{x}\|^2 + \sum_{i=1}^{m}\|z_i - \bar{z}\|^2 - 2\sum_{i=1}^{m}(x_i - \bar{x})^\top\mathbf{R}^\top(z_i - \bar{z}) \tag{5.22}$$

The first two terms are independent of $\mathbf{R}$ and we thus are looking for $\mathbf{R}$ maximizing the last sum. Using the cycle property of the trace operator, its linearity, and the fact that the trace of a scalar is the scalar itself we can write

$$\sum_{i=1}^{m}(x_i - \bar{x})^\top\mathbf{R}^\top(z_i - \bar{z}) = \mathrm{tr}(\sum_{i=1}^{m}(x_i - \bar{x})^\top\mathbf{R}^\top(z_i - \bar{z})) \tag{5.23}$$

$$= \sum_{i=1}^{m}\mathrm{tr}((x_i - \bar{x})^\top\mathbf{R}^\top(z_i - \bar{z})) \tag{5.24}$$

$$= \sum_{i=1}^{m}\mathrm{tr}(\mathbf{R}^\top(z_i - \bar{z})(x_i - \bar{x})^\top) \tag{5.25}$$

$$= \mathrm{tr}(\mathbf{R}^\top\sum_{i=1}^{m}(z_i - \bar{z})(x_i - \bar{x})^\top) \tag{5.26}$$

$$= \mathrm{tr}(\mathbf{R}^\top\mathbf{M}) \tag{5.27}$$

$$= \mathrm{tr}(\mathbf{R}^\top\mathbf{U}\mathbf{D}\mathbf{V}^\top) \tag{5.28}$$

$$= \mathrm{tr}(\mathbf{V}^\top\mathbf{R}^\top\mathbf{U}\mathbf{D}) \tag{5.29}$$

From Lemma 6 ($\mathbf{D}$ is a diagonal matrix with non-negative values) it follows that $\mathbf{V}^\top \mathbf{R}^{*\top} \mathbf{U} = \mathbf{I}$ which is equivalent to $\mathbf{R}^* = \mathbf{U}\mathbf{V}^\top$ if $\mathbf{M}$ (and thus $\mathbf{D}$) has full rank. If $\mathbf{M}$ does not have full rank equivalent solutions exist. This could *e.g.* be the case if there are only a few known correspondences ($m < n$) or the correspondences are not independent. □

We have seen that there are efficient ways to find nearest neighbors in spaces with moderate dimensions and that given corresponding points the calculation of an optimal rigid alignment can be done closed form. Thus - given a good intialization - the ICP algorithm is very efficient in finding *rigid* isometries. We want to point out that without either a good initial alignment or correspondences, ICP usually gets trapped in local optima.

5.1.2 Intrinsic isometries and multidimensional scaling

While being very efficient, the ICP algorithm not only relies on a good initialization but is also limited to rigid shapes, *i.e.* Euclidean isometries. When dealing with non-rigid shapes $\mathcal{X}$ and $\mathcal{Y}$ the chosen metrics $D^{\mathcal{X}}$ and $D^{\mathcal{Y}}$ are the geodesic distances, *i.e.*

$$d(p,q) = \min_{c\colon [0,1]\to\mathcal{X}} \{\text{length}(c) | c[0] = p\,, c[1] = q\} \tag{5.30}$$

In this case the set of isometries can not be described by a low dimensional space as in the Euclidean setting. All approaches that will be presented in the following sections (in particular Section 5.3 and Section 5.6) can in one way or an other be interpreted as mapping the shapes into a Euclidean space and then perform ICP-like methods there. A noticable exception are approaches based on pairwise descriptors (Section 5.4) which can be seen as variants of *generalized mutlidimensional scaling* [13].
The most straight forward way to turn the non-rigid isometry problem into a Euclidean one is called *multidimensional scaling* (MDS). The idea is to find a mapping $\phi_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}^N$ (N not necessarily related to dimension of $\mathcal{X}$) such that all distances are preserved:

$$d_{\mathcal{X}}(p,q) = d_{\mathbb{R}^N}(\phi(p), \phi(q)) = \|\phi(p) - \phi(q)\| \quad \forall p, q \in \mathcal{X} \tag{5.31}$$

If such a mapping exists for $\mathcal{X}$ there is also a $\phi_{\mathcal{Y}} : \mathcal{Y} \to \mathbb{R}^N$ preserving all distances of $\mathcal{Y}$. Thus we can get a correspondence between $\mathcal{X}$ and $\mathcal{Y}$ by rigidly aligning $\phi_{\mathcal{X}}(\mathcal{X})$ with $\phi_{\mathcal{Y}}(\mathcal{Y})$. However there are two major drawbacks to this approach:

1. The mappings $\phi_{\mathcal{X}}, \phi_{\mathcal{Y}}$ will never be unique: concatenating them with any rigid motion will not change the Euclidean distances. This means that we are not guaranteed to have a good initial alignment between $\mathcal{X}$ and $\mathcal{Y}$ in $\mathbb{R}^N$ even if we manage to satisfy 5.31 exactly. This however is essential for ICP to converge to the global optimum.

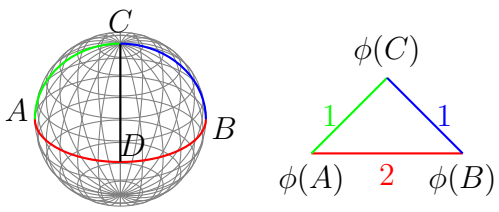

Figure 5.2: There is no isometric embedding ϕ of the sphere to any Euclidean space. Consider the points A, B, C and D. $\phi(C)$ would need to lie on the middle of the line connecting $\phi(A)$ and $\phi(B)$ (since the two smaller edge lengths sum up to the large one). The same however holds for $\phi(D)$, thus $\phi(C) = \phi(D)$ and their distance (1) would *not* be preserved.

2. Even with a very large dimension N (which already makes nearest neighbor search in ICP inefficient) there may not be any mapping satisfying 5.31 (*c.f.* Figure 5.2). Thus one can only try to minimize 5.31, very likely resulting in different/incompatible local minima for $\mathcal{X}$ and $\mathcal{Y}$.

We will in the following present ways to make use of the isometry of $\mathcal{X}$ and $\mathcal{Y}$ in other ways. Definition 10 is defined in global terms by considering *all pairs of points* on $\mathcal{X}$. The following Lemma gives a more local persective on isometries.

Lemma 7 Let $m\colon \mathcal{X} \to \mathcal{Y}$ be a diffeomorphism and Dm be the corresponding *push forward* (*c.f.* Equation (3.17))

$$Dm(p)\colon T_p\mathcal{X} \to T_q\mathcal{Y} \tag{5.32}$$
$$[c] \mapsto [m \circ c] \tag{5.33}$$

m is an isometry iff it preserves all angles:

$$\langle v, w\rangle_{T_p\mathcal{X}} = \langle Dm_p v, Dm_p w\rangle_{T_q\mathcal{Y}} \tag{5.34}$$

with $q = m(p)$

Proof. We first show that Equation (5.30) follows from Equation (5.34). From Equation (5.34) it is easy to see that the length of *all* curves is preserved under m. Let $c\colon [0,1] \to \mathcal{X}$ be a curve connecting $p \in \mathcal{X}$ and $q \in \mathcal{X}$. Then the curve $d = m \circ c\colon [0,1] \to \mathcal{Y}$ connects $m(p)$ with $m(q)$ and has length

$$L(d) = \int_0^1 \left\| \frac{d}{dt}(m \circ c(t)) \right\| dt = \int_0^1 \left\| Dm_{c(t)}\dot{c}(t)) \right\| dt = \int_0^1 \|\dot{c}(t))\| \, dt = L(c) \tag{5.35}$$

This in particular holds for the shortest curve c^* between p and q and $d^* = m \circ c^*$. And if there was a shorter curve $\tilde{d}$ between $m(p)$ and $m(q)$ then $\tilde{c} = m^{-1} \circ \tilde{d}$ would also be shorter than c^*.

For the other direction we make use of the continuity of the metric tensor. Let $\langle v, w\rangle_{T_p\mathcal{X}} \neq \langle Dm_p v, Dm_p w\rangle_{T_q\mathcal{Y}}$ for some $p \in \mathcal{X}$, let's say $\langle v, w\rangle_{T_p\mathcal{X}} > \langle Dm_p v, Dm_p w\rangle_{T_q\mathcal{Y}}$ (otherwise exchange the roles of $\mathcal{X}$ and $\mathcal{Y}$). Then there is a geodesic ball $B_\varepsilon(p) = \{\tilde{p} \in \mathcal{X} | d(p, \tilde{p}) < \varepsilon\}$ around p where

$$\langle v, w\rangle_{T_{\tilde{p}}\mathcal{X}} > \langle Dm_{\tilde{p}} v, Dm_{\tilde{p}} w\rangle_{T_{\tilde{q}}\mathcal{Y}} \, \forall \tilde{p} \in B_\varepsilon(p) \tag{5.36}$$

Let now $q \in B_\varepsilon(p)$, $c\colon [0,1] \to \mathcal{X}$ be the shortest curve connecting p and q. Additionaly, let $\tilde{d}\colon [0,1] \to \mathcal{Y}$ be the shortest curve connecting $m(p)$, and let $m(q)$ and $\tilde{c} = m^{-1}\tilde{d}$ be the corresponding curve on $\mathcal{X}$. We decompose $\tilde{c} = c_1 \cup c_2$ with $c_1 = c \cap B_\varepsilon(p)$ and $c_2 = c \cap \overline{B_\varepsilon(p)}$ and $d_1 = m \circ c_1, d_2 = m \circ c_2$. It follows

$$L(\tilde{d}) = L(d_1) + L(d_2) \geqslant L(d_1) > L(c_1) \geqslant L(c) \tag{5.37}$$

The inequality $L(d_1) > L(c_1)$ follows from Equation (5.36). The final inequality follows from the fact that c_1 either connects p with q and thus must be at least as long as the shortest path c, or it leaves $B_\varepsilon(p)$ at least once which requires $L(c_1) \geqslant \varepsilon > L(c)$. □

Consequently, isometric shapes come with *the same Riemannian metric* and thus all intrinsic properties are invariant under m. Some examples:

1. Mean curvature

$$\kappa(p) = \kappa(m(p)) \tag{5.38}$$

2. Areas and integrals

$$\int_{X_0} f(p)dp = \int_{m(X_0)} f(m^{-1}(p))dq \tag{5.39}$$

3. The value of the Laplacian $\triangle_\mathcal{X}$ applied to a function $f\colon \mathcal{X} \to \mathbb{R}$ at $p \in \mathcal{X}$ equals the value of the Laplacian $\triangle_\mathcal{Y}$ applied to $\tilde{f} = f \circ m^{-1}$ at $q = m(p)$:

$$\triangle_\mathcal{X} f(p) = \triangle_\mathcal{Y} \tilde{f}(q), \quad q = m(p) \tag{5.40}$$

The last property particularly applies to eigenfunctions of the Laplacian:

Lemma 8 Let $m\colon \mathcal{X} \to \mathcal{Y}$ be an isomorphism and $\triangle_\mathcal{X}\phi_k = \lambda_k\phi_k$. Then the function $\tilde{\phi} = \phi \circ m^{-1}\colon \mathcal{Y} \to \mathbb{R}$ is an eigenfunction of $\triangle_\mathcal{Y}$ with the same eigenvalue:

$$\triangle_\mathcal{Y}\tilde{\phi}_k = \lambda_k\tilde{\phi}_k \tag{5.41}$$

Moreover $\left\|\tilde{\phi}_k\right\| = \|\phi_k\| = 1$.

Proof. This is a direct consequence of the first fundamental form g being the identity for isometries:

$$\int_{\mathcal{Y}} \triangle\tilde{\Phi}_k(y)h(y)dy = -\int_{\mathcal{Y}} \langle\nabla\tilde{\Phi}_k(y)), \nabla h(y)\rangle dy \tag{5.42}$$

$$= -\int_{\mathcal{X}} \langle g^{-1}\nabla\Phi_k(x), g^{-1}\nabla h(m(x))\rangle\sqrt{\det(g)}dx \tag{5.43}$$

$$= -\int_{\mathcal{X}} \langle\nabla\Phi_k(x), \nabla h(m(x))\rangle dx \tag{5.44}$$

$$= \lambda_k \int_{\mathcal{X}} \Phi_k(x)h(m(x))dx \tag{5.45}$$

$$= \lambda_k \int_{\mathcal{Y}} \tilde{\Phi}_k(y)h(y)\frac{1}{\sqrt{\det(g)}}dy \tag{5.46}$$

$$= \lambda_k \int_{\mathcal{Y}} \tilde{\Phi}_k(y)h(y)dy \tag{5.47}$$

The preservation of the norm can be shown even easier using the same argument. □

Consequently, two isometric shapes have an identical spectrum (set of eigenvalues). The reverse statement however is not true: there exist iso-spectral shapes which are not isometric [24]. Still, there are approaches to reconstruct shapes from their spectrum [15] with shape compression as a possible application.
Let us now have a closer look at the eigenfunctions. Let λ be an eigenvalue of $\triangle_{\mathcal{X}}$ (and thus also of $\triangle_{\mathcal{Y}}$). If the isometry m is given we can construct a corresponding eigenfunction of $\triangle_{\mathcal{Y}}$ via $\tilde{f} \circ m^{-1}$. However this eigenfunction is not unique (even after normalization) - any rigid motion of $\tilde{f}$ inside the eigenspace of λ yields an alternative eigenfunction. For one dimensional eigenspaces, the only rigid motion is a *sign flip* $\tilde{f} \mapsto -\tilde{f}$. Thus independent eigendecompositions of $\triangle_{\mathcal{X}}$ and $\triangle_{\mathcal{Y}}$ give compatible eigenfunctions up to rigid motions, see Figure 5.3 (signflip in second column).
Notice that many shapes come with at least one intrinsic symmetry $s\colon \mathcal{X} \to \mathcal{X}$:

$$D^{\mathcal{X}}(x_1, x_2) = D^{\mathcal{X}}(s(x_1), s(x_2)) \tag{5.48}$$

(*e.g.* the left-right symmetry of human bodies). As a consequence isometries $m\colon \mathcal{X} \to \mathcal{Y}$ are not unique (since $m \circ s$ is also an isometry).

5.2 Permutation as discrete bijections

In this thesis we focus on "dense vertex-to-vertex" correspondences. This means that the discrete representation of a matching between two triangle meshes $\mathcal{X} = (V_{\mathcal{X}}, T_{\mathcal{X}}), \mathcal{Y} = (V_{\mathcal{Y}}, T_{\mathcal{Y}})$ is given by a mapping $p\colon V_{\mathcal{X}} \to V_{\mathcal{Y}}$.

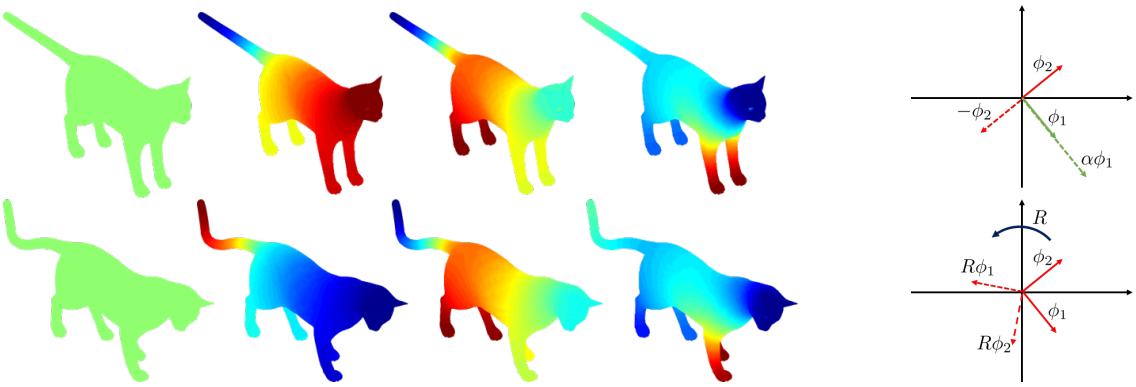

Figure 5.3: The Laplace Beltrami operator is an intrinsic operator and therefore invariant to isometric deformations. However the eigenfunctions are not uniquely defined (sign flips, rigid motions in higher dimensional eigenspaces). (Meshes from the TOSCA dataset [14])

A common criticism (*e.g.* [21]) of this approach is that for general meshes the "correct" matching of a vertex $v \in V_{\mathcal{X}}$ may not be a vertex on $\mathcal{Y}$ but rather lie in the interior of a triangle. We argue that

- the error induced by the vertex-to-vertex assumption lies in the same range as the approximation error of the surfaces themselves
- The two meshes $\mathcal{X}$ and $\mathcal{Y}$ should be treated symmetrically - a matching from $\mathcal{X}$ to $\mathcal{Y}$ should not be structurally different to one in the opposite direction

The last point raises another requirement: As a mapping p assigns each vertex on $\mathcal{X}$ exactly on vertex on $\mathcal{Y}$, it should also do so in the other direction, *i.e.* p is supposed to be a *bijection* between the vertices $V_{\mathcal{X}}$ on $\mathcal{X}$ and the ones on $\mathcal{Y}$. This of course requires a *consistent sampling* of the two meshes, in particular the same number of vertices. In many applications (such as template registration) this requirement is not satisfied. In these cases we suggest to create a consistent subsampling of the vertices on both meshes and then aim for a bijection between those subsets of vertices. *Farthest point sampling* (FPS [19]) guarantees a uniform distribution of the subsampled vertices. In particular for isometric shapes this leads to a consistent sampling on $\mathcal{X}$ and $\mathcal{Y}$. For another introduction to FPS for non-rigid shapes see Section 3 in [14]. *Remeshing* is another possibility to obtain the same number of vertices, however it would change the meshes themselves and in its simple variants (*e.g.* edge collapse) cannot guarantee the consistency of the sampling.
For simplicity we henceforth assume a pair of consistently discretized meshes such that the sought correspondence $p\colon V_{\mathcal{X}} \to V_{\mathcal{Y}}$ is a bijection and denote with $n = |V_{\mathcal{X}}| = |V_{\mathcal{Y}}|$ the (consistent) number of vertices. The concepts presented in the remainder of this chapter however extend to subsampled meshes as well.
A common representation of a bijection $p\colon V_{\mathcal{X}} \to V_{\mathcal{Y}}$ is a *permutation matrix*. We

denote the set of permutation matrices of size n as

$$\Pi_n = \{\mathbf{P} \in \{0,1\}^{n\times n} | \mathbf{P}1 = 1,\, \mathbf{P}^\top 1 = 1\} \tag{5.49}$$

and will drop the dependency on n whenever clear from the context. In other words, in each row and each column of a permutation matrix there is exactly one 1 and all other entries are 0. We use the following convention to encode the bijection $p\colon \mathcal{X} \to \mathcal{Y}$ as the permutation matrix $\mathbf{P} \in \Pi_n$:

$$p(v_i) = v_j \Leftrightarrow \mathbf{P}_{ji} = 1 \tag{5.50}$$

We want to emphasize that there are equivalent definitions that will lead to different relaxations later on. For instance permutation matrices can also be described as the $\{0,1\}$-valued matrices that are orthogonal:

$$\Pi_n = \{\mathbf{P} \in \{0,1\}^{n\times n} | \mathbf{P}\mathbf{P}^\top = \mathbf{I}\} \tag{5.51}$$

Finally finding the correct matching is phrased as an optimization problem

$$\arg\min_{\mathbf{P}\in\Pi_n} E(\mathbf{P}) \tag{5.52}$$

over the space of permutations. Unfortunatly problems of this type are combinatorial and in most cases NP-hard. An important exception are linear energies E which lead to *linear assignment problems* that can be solved in polynomial time. We will see that *pointwise descriptors* lead to linear energies.

5.3 Pointwise descriptors and linear assignment

A *pointwise descriptor* (or *feature*) is a mapping $f\colon \mathcal{X} \to \mathbb{R}^k$ [1] assigning each point on a shape (or vertex on a mesh) a Euclidean vector that describes the point.
Simple examples are the coordinates, the normal, and the mean curvature of the respective point. A good pointwise descriptor assigns similar (in the Euclidean sense) values to semantically similar points on different shapes while (semantically) dissimilar points are assigned descriptors with a large Euclidean distance (*c.f.* Figure 1.2). We will discuss examples of pointwise feature descriptors at the end of this section and will for now assume to be given some descriptor. In that case the mapping $p\colon \mathcal{X} \to \mathcal{Y}$ should satisfy

$$f(p(x)) \approx f(x) \quad \forall x \in \mathcal{X} \tag{5.53}$$

[1] In fact a feature is defined on all shapes, i.e. $f\colon \bigcup \mathcal{X} \to \mathbb{R}^k$. We choose this (over-) simplified notation for easier understanding. However we will later on also evaluate f on $\mathcal{Y}$ and other shapes.

In the discrete case where $\mathcal{X}, \mathcal{Y}$ are triangle meshes we store per-vertex wise features in matrices $F \in \mathbb{R}^{|V_X| \times k}$, $G \in \mathbb{R}^{|V_Y| \times k}$. Interpreting (5.53) in the l^2 sense and letting p be a bijection between the respective vertices (thus $|V_X| = |V_Y|$) this leads to the following energy

$$E(p) = \sum_{i \leqslant |V_X|} \sum_k (F_{ik} - G_{p(i)k})^2 \tag{5.54}$$

$$\tag{5.55}$$

Rewriting in terms of the matrix representation P of p:

$$E(\mathbf{P}) = \|\mathbf{P}F - G\|^2 \tag{5.56}$$

since from (5.50) it follows

$$\|F - \mathbf{P}^\top G\|^2 = \sum_i \sum_k \left(F_{ik} - (\mathbf{P}^\top G)_{ik}\right)^2 \tag{5.57}$$

$$= \sum_i \sum_k \left(F_{ik} - (\sum_j \mathbf{P}_{ji} G_{jk})\right)^2 \tag{5.58}$$

$$= \sum_i \sum_k (F_{ik} - G_{p(i)k})^2 \tag{5.59}$$

and the orthogonality of $\mathbf{P}$ gives

$$\|F - \mathbf{P}^\top G\|^2 = \|\mathbf{P}(F - \mathbf{P}^\top G)\|^2 \tag{5.60}$$

$$= \|\mathbf{P}F - G\|^2 \tag{5.61}$$

.

Again using the orthogonality we see that minimizing E as in (5.56) is equivalent to maximizing a linear function:

$$\arg\min_P \|\mathbf{P}F - G\|^2 = \arg\min_{\mathbf{P}} \langle \mathbf{P}F - G, \mathbf{P}F - G \rangle \tag{5.62}$$

$$= \arg\min_{\mathbf{P}} \langle \mathbf{P}F, \mathbf{P}F \rangle + \langle G, G \rangle - 2\langle \mathbf{P}F, G \rangle \tag{5.63}$$

$$= \arg\min_{\mathbf{P}} \langle F, F \rangle + \langle G, G \rangle - 2\langle \mathbf{P}F, G \rangle \tag{5.64}$$

$$= \arg\max_{\mathbf{P}} \langle \mathbf{P}F, G \rangle = \arg\max_{\mathbf{P}} \langle \mathbf{P}, GF^\top \rangle \tag{5.65}$$

where $\langle A, B \rangle = \mathrm{tr}(A^\top B)$ denotes the Frobenius inner product. Note that the equivalence only holds for orthogonal $\mathbf{P}$'s, *e.g.* when optimization is performed of the space of permutations. In this case Equation (5.65) is an instance of a *linear assignment problem* (LAP) for which polynomial time solvers exist [8, 26].

5.3.1 Examples of pointwise descriptors

In this section we want to briefly introduce a number of important feature descriptors. We will start with what are often called *spectral descriptors*. They have in common that they can be written solely in terms of the eigendecomposition of the Laplacian (cf. Section 4). The fact that they can be expressed that way makes them *isometry invariant* whenever they are well defined. Two isometric shapes have the same Laplacian. Up to the ambiguity of the eigendecomposition they also have the same set of eigenfunctions and -values. A well defined descriptor must be invariant to this ambiguity. As an example of a spectral descriptor where the ambiguity is *not* tackled consider the *Global point signature (GPS)* [46], given by:

$$GPS(x) = \begin{pmatrix} \frac{\phi_1(x)}{\sqrt{\lambda_1}} & \frac{\phi_2(x)}{\sqrt{\lambda_2}} & \dots \end{pmatrix} \tag{5.66}$$

Even when evaluated at the exact same point of the exact same shape, the values may differ due to sign flips of some of the harmonics coming from two different eigendecompositions. The GPS is thus not well defined and in particular a bad choice for a feature descriptor.
As a better example consider the *heat kernel signature (HKS)* [52]. It is defined in terms of the heat kernels discussed in Section 4.6.1. The heat kernel signature is a sampling of the diagonal elements $k_t(x,x)$ of the heat kernel at different values of t:

$$HKS(x) = \begin{pmatrix} k_{t_0}(x,x) & k_{t_1}(x,x) & \dots & k_{t_l}(x,x) \end{pmatrix} \tag{5.67}$$

From Equation (4.104) we know that $k_t(x,x) = \sum_k \exp(-\lambda_k t)\phi_k^2(x)$. It is obvious that the problem of sign flips is overcome due to the squaring of function values. Let us now consider the case of higher dimensional eigen spaces. Let λ be an eigenvalue with d-dimensional eigenspace, $\lambda_l = \dots = \lambda_{l+d} = \lambda$ and $\{\phi_l, \dots \phi_{l+d}\}$ and $\{\tilde{\phi}_l, \dots \tilde{\phi}_{l+d}\}$ be two different orthonormal bases of that eigenspace. Let us fix $x \in \mathcal{X}$ and introduce

$$\psi(x) = \begin{pmatrix} \phi_l & \dots & \phi_{l+d} \end{pmatrix} \tag{5.68}$$

$$\tilde{\psi}(x) = \begin{pmatrix} \tilde{\phi}_l & \dots & \tilde{\phi}_{l+d} \end{pmatrix} \tag{5.69}$$

We know that there is an orthogonal matrix $\mathbf{R} \in \mathbb{R}^{d\times d}$ such that $\psi(x) = \mathbf{R}\tilde{\psi}(x)$. For the affected part of the sum this yields

$$\sum_{k=l}^{l+d} \exp(\lambda_k t)\phi_k^2(x) = \exp(\lambda_l t) \sum_{k=l}^{l+d} \phi_k^2(x) \tag{5.70}$$

$$= \exp(\lambda_l t)\langle \psi(x), \psi(x) \rangle \tag{5.71}$$

$$= \exp(\lambda_l t)\langle \mathbf{R}\psi(x), \mathbf{R}\psi(x) \rangle \tag{5.72}$$

$$= \sum_{k=l}^{l+d} \exp(\lambda_k t)\tilde{\phi}_k^2(x) \tag{5.73}$$

Thus the value $k_t(x,x)$ is independent on the chosen eigendecomposition of the Laplacian. The same reasoning holds for all descriptors of the form $\sum_k f(\lambda_k;\theta)\phi_k^2(x)$ with parametrizable coefficients f (in case of the HKS $\theta = t$). Before we come to an other instance, we want to point out the multiscale property of the heat kernel signature and its possible applications. The diagonal entries of the heat kernel $k_t(x,x)$ describe how much heat remains at location $x \in \mathcal{X}$ after time t. For small times the value is mainly determined by the geometry of the local neighborhood of x:

$$h_t(x,x) = \frac{1}{4\pi t} + \frac{s(x)}{12\pi} + O(t) \tag{5.74}$$

with $s(x)$ being the scalar curvature. The bigger the time the more global information is captured. In the extreme case as $t \to \infty$ the heat is uniformly distributed on the shape, $k_t(x,x)$ thus converges to the inverse of the surface area which equals the value of the constant eigenfunction ϕ_0 (Note that $\exp(\lambda_0 t) = 1\,\forall t$). This multi scale property allows to compare shapes that are only locally isometric.
Another descriptor of the form $\sum_k f(\lambda_k;\theta)\phi_k^2(x)$ is given by the *wave kernel signature (WKS)* [6].

$$f(\lambda;\theta) = C_\theta \exp(-\frac{(\log(\lambda)-\theta)^2}{2\sigma^2}) \tag{5.75}$$

with $C_\theta = \left(\sum_k \exp(-\frac{(\log(\lambda_k)-\theta)^2}{2\sigma^2})\right)^{-1}$ being a normalization factor. In the original paper the authors suggest to use a variant of the L^1 distance to measure similarity of descriptors, however most subsequent works use the Euclidean distance. The functions $f(\lambda;\theta)$ can be interpreted as filter responses, see Figure 5.4. In [4] we suggest to learn the optimal parameters of the WKS from data. The authors of [33] suggest to learn generic filter responses.
A very popular descriptor both for rigid and non-rigid shape description is the *Unique signatures of histograms for surface and texture description (SHOT)* [53]. In a nutshell, SHOT is measuring histograms of surface normals around a given point. In contrast to all previous discussed desciptors, SHOT is *not* intrinsic but can change under isometric (non-rigid) deformations. This however can be seen as a feature: it can *e.g.* help to distinguish intrinsic symmetries of shapes. A popular approach is to combine SHOT with other descriptors such as HKS and WKS.

5.4 Pairwise descriptors and quadratic assigment

Pointwise descriptors are well established in many fields of Computer Vision and are popular due to the relatively simple optimization problems induced by them (nearest neighbour search, LAP). However in most scenarios pointwise descriptors are either

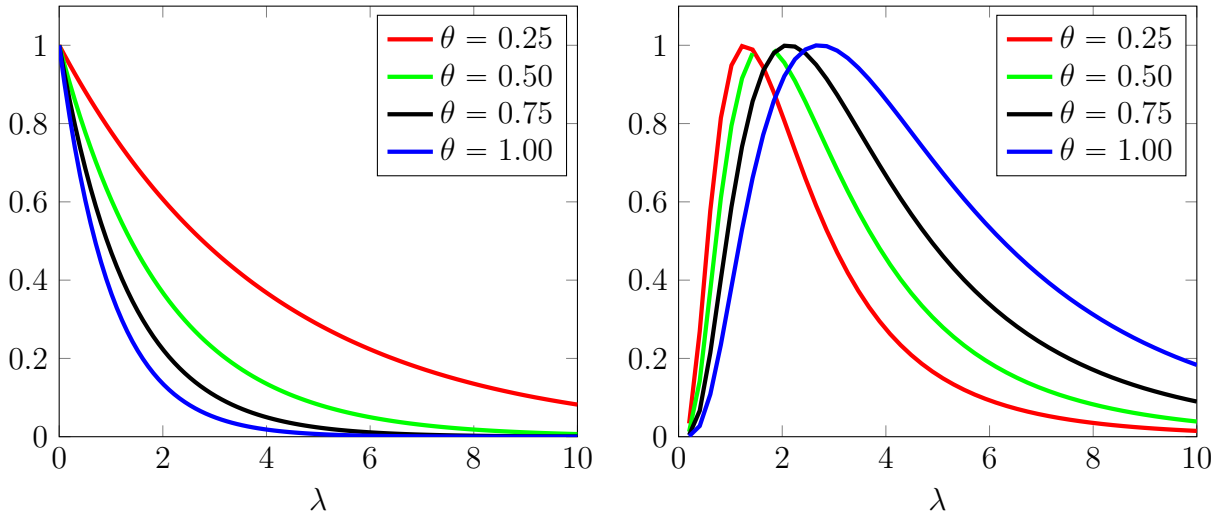

Figure 5.4: We plot the coefficient functions $f(\lambda;\theta)$ of spectral descriptors of the form $\sum_k f(\lambda_k;\theta)\phi_k^2(x)$. Different colors indicate different choices of parameters θ. The filter response of the heat kernel signature (left) act as *low pass filters* (emphasize always on eigenfunctions with low frequencies), the ones of the wave kernel signature (right, unnormalized) as *band pass filters* (the emphasis can be guided through choice of θ).

not sufficiently discriminative or invariant with respect to the observed type of deformations between shapes. The resulting assignment problems lead to inconsistent and irregular correspondences, see second column of Figure 5.5. A major reason for the irregularity, in particular resulting from unconstraint nearest neighbor formulations, is that all descriptors are calculated and treated individually. Constraining the correspondence to be a one-to-one mapping as in the LAP formulation already enforces some kind of communication between the points and results in more regular correspondences, see right column of Figure 5.6. However there is still no guarantee that close by points on the source shape will be mapped to close by points on the target shape, *i.e.* that the discrete mapping follows an intuitive notion of smoothness. Following the idea of mapping close by points on the source shape to close by points on the target shape leads to the concept of *pairwise descriptors*.
A *pairwise descriptor* is a mapping $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}^k$. The literature mainly focuses on scalar ($k = 1$) pairwise descriptors. Probably the most prominent example in the context of non-rigid shape correspondence is the *geodesic distance* (*c.f.* Eq.(3.26)). In case of *isometric shapes* (*c.f.* Definition 10) the sought mapping is an isometry, *i.e.* it preserves all geodesic distances. Since the considered shapes are never exactly isometric we get

$$D^{\mathcal{X}}_{ij} \approx D^{\mathcal{Y}}_{p(i)p(j)} \quad \forall i,j = 1 \dots n \tag{5.76}$$

when storing all pairwise distances in a quadratic $n \times n$ matrix with entries $D^{\mathcal{X}}_{ij} =$

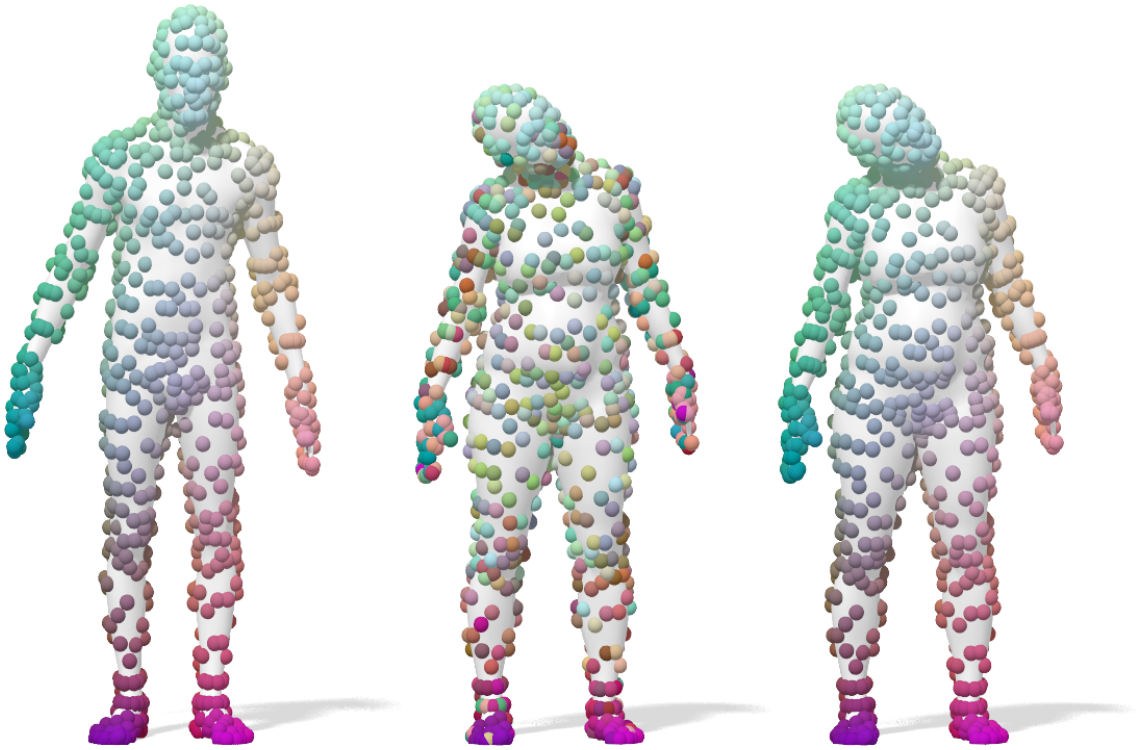

Figure 5.5: Pointwise descriptors lead to irregular correspondences: The second column shows the correspondence obtained using state of the art pointwise descriptors (WKS, SHOT). While mapping many points to the desired location, the correspondence suffers from a lack of regularity, mainly induced by the intrinsic symmetry of the shapes. Even if a majority of points on the left arm of the source shape is mapped to the left arm of the target shape, there is no way to communicate this choice to the few points who are mapped to the wrong side (right arm). Using pairwise descriptors to regularize the correspondence allows for more communication between the points and leads to a more regular correspondence (third column). In this conseptual illustration blue spheres depict points on the source shape, red spheres points on the target shape, and arrows describe the correspondence. (Meshes from the FAUST dataset [11])

Figure 5.6: Nearest neighbor vs. LAP: enforcing the mapping to be one-to-one leads to computationally more heavy optimization problems (LAPs) compared to plain nearest neighbor problems. Enforcing the bijectivity, however, can fix embeddings that do not place descriptors optimally.

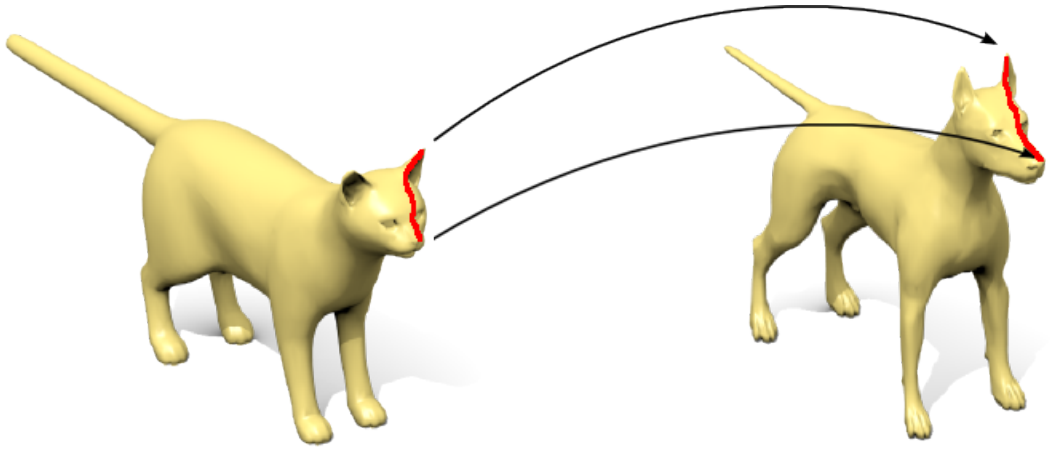

Figure 5.7: Geodesic distances are an example of pariwise descriptors. Each pair of points on the source shape, *e.g.* the nose and the right ear, gets assigned a scalar value. This value is approximately preserved under the desired mapping for all possible pairs of points. (Meshes from the Tosca dataset [14])

$D^{\mathcal{X}}(x_i, x_j)$ ($D^{\mathcal{Y}}$ is constructed similarly). In this case (and in most others) the matrices are symmetric, however this is not a necessary condition for a pairwise descriptor.

A straightforward way to turn Equation (5.76) into an optimization problem is by interpreting it in the L^2 sense:

$$E(\mathbf{P}) = \sum_{i,j}(D^{\mathcal{X}}_{ij} - D^{\mathcal{Y}}_{p(i)p(j)})^2 = \sum_{i,j}(D^{\mathcal{X}}_{ij} - (\sum_{k}\mathbf{P}_{ki}D^{\mathcal{Y}}_{kp(j)}))^2 \tag{5.77}$$

$$= \sum_{i,j}(D^{\mathcal{X}}_{ij} - (\sum_{k,l}\mathbf{P}_{ki}D^{\mathcal{Y}}_{kl}\mathbf{P}_{lj}))^2 \tag{5.78}$$

$$= \sum_{i,j}(D^{\mathcal{X}}_{ij} - (\mathbf{P}^\top D^{\mathcal{Y}}\mathbf{P})_{ij})^2 \tag{5.79}$$

$$= \left\| D^{\mathcal{X}} - \mathbf{P}^\top D^{\mathcal{Y}}\mathbf{P} \right\|^2 \tag{5.80}$$

We again make use of $\mathbf{P}$'s orthogonality to show that minimizing this energy over

the space of permutations is equivalent to a *quadratic assigment problem* (QAP):

$$\arg\min_{\mathbf{P}} \left\| D^{\mathcal{X}} - \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P} \right\|^2 \tag{5.81}$$

$$= \arg\min_{\mathbf{P}} \left\langle D^{\mathcal{X}}, D^{\mathcal{X}} \right\rangle - \left\langle D^{\mathcal{X}} + (D^{\mathcal{X}})^\top, \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P} \right\rangle + \left\langle \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P}, \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P} \right\rangle \tag{5.82}$$

$$= \arg\min_{\mathbf{P}} \left\langle D^{\mathcal{X}}, D^{\mathcal{X}} \right\rangle - \left\langle D^{\mathcal{X}} + (D^{\mathcal{X}})^\top, \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P} \right\rangle + \left\langle D^{\mathcal{Y}}, D^{\mathcal{Y}} \right\rangle \tag{5.83}$$

$$= \arg\max_{\mathbf{P}} \left\langle D^{\mathcal{X}} + (D^{\mathcal{X}})^\top, \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P} \right\rangle \tag{5.84}$$

$$= \arg\max_{\mathbf{P}} \operatorname{tr} \left(((D^{\mathcal{X}})^\top + D^{\mathcal{X}}) \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P} \right) \tag{5.85}$$

$$= \arg\max_{\mathbf{P}} \operatorname{tr} \left(D^{\mathcal{X}} \mathbf{P}^\top D^{\mathcal{Y}} \mathbf{P} \right) \quad \text{(for symmetric } D^{\mathcal{X}}\text{)} \tag{5.86}$$

Unlike *linear assignment problems* (LAP's) there are no known polynomial time algorithms to solve QAPs as they are NP-hard [29]. Thus, unless very small instances are considered, there we need to relax the problem in order to solve it in practice. We want to emphasize that while all of the above formulations are equivalent when evaluated solely on permutation matrices, their behaviour can dramatically differ when considered as functions of general matrices $\mathbf{P}$. Let us elaborate on this by introducing yet an other 'equivalent' formulation:

$$\arg\min_{\mathbf{P}} \left\| \mathbf{P} D^{\mathcal{X}} - D^{\mathcal{Y}} \mathbf{P} \right\|^2 \tag{5.87}$$

is a convex function, independent of the properties of $D^{\mathcal{X}}$ and $D^{\mathcal{Y}}$ (composition of linear and quadratic) with a unique minimizer over the space of all $n \times n$ matrices. The convexity of (5.86) however depends on the eigenvalues of $D^{\mathcal{X}}$ and $D^{\mathcal{Y}}$.

Notice that there are many different ways to write Equation (5.76) as an optimization problem. For instance every function $t : \mathbb{R} \to \mathbb{R}$ (not necessarily monotocally increasing or injective) gives a related problem $t(D^{\mathcal{X}}_{ij}) \approx t(D^{\mathcal{Y}}_{p(i)p(j)})$. A popular example of such a transformation is $t(d) = \exp(-\frac{d^2}{2\sigma^2})$. The choice of t influences the spectrum of the matrices $\tilde{D} = t(D)$ and consequently the properties of the resulting optimization problem. Equation (5.76) almost never holds with equality. The (global) optimum and possible local optima vary when choosing different transformations t.

We also want to emphasize that there are many alternatives to the geodesic distance and derived quantities. In particular, one could replace the geodesic distance with the distance in (pointwise) descriptor space: $D_{ij} = \|f(x_i) - f(x_j)\|$ for any given pointwise descriptor f. In [2] we suggest to align *heat kernels*: $D_{ij} = k_t(x_i, x_j)$ (*c.f.* Equation (4.104))

5.5 Relaxations

In the last sections we have seen that shape correspondence problems can be formulated as combinatorial optimization problems where the domain is the set of permutations. We have pointed out that quadratic assignment problems arising from pairwise descriptors are in general NP-hard and consequently not solvable in reasonable time even for moderatly sized problems (meshes with as few as 1000 vertices). Even in the simpler case of linear assignment problems methods such as the Hungarian algorithm [26] or the auction algortihm [8] need several minutes to hours of computation time when applied to problems in the magnitude of 10^3-10^4. Let us recap the linear assignment problem arising from pointwise decriptors:

$$\arg\max_{\mathbf{P}} \langle \mathbf{P}, \mathbf{H} \rangle \tag{5.88}$$

with $\mathbf{H} = \mathbf{G}\mathbf{F}^\top$. Since we constrain $\mathbf{P}$ to be a permutation matrix, its entries are either 0 or 1 with exactly one non-zero entry per row and column. Thus for each row i the assignment problem is allowed to pick exactly one entry in order to maximize the objective. Dropping the bijectivity constraint leads to independent optimization problems per row:

$$j^*(i) = \arg\max_j \mathbf{H}_{ij} \tag{5.89}$$

In most cases this approach will not result in bijections as the different rows do not 'communicate'. In particular one obtains inconsistent correspondences when switching the roles of source and target shape (considering columns instead of rows). While the described approach can be applied to general linear assignement problems with payoff matrix $\mathbf{H}$, in our case, we have access to the *low rank decomposition* $\mathbf{H} = \mathbf{G}\mathbf{F}^\top$, $\mathbf{F}$, $\mathbf{G} \in \mathbb{R}^{n \times k}$, stemming from the k dimensional feature space. Solving Equation (5.89) corresponds to finding the *nearest neighbor* of $\mathbf{F}_{i\cdot}$ among all columns of $\mathbf{G}$:

$$j^*(i) = \arg\max_j \mathbf{H}_{ij} = \arg\max_j \langle \mathbf{F}_{i\cdot}, \mathbf{G}_{j\cdot} \rangle \tag{5.90}$$

$$= \arg\min_j \|\mathbf{F}_{i\cdot} - \mathbf{G}_{j\cdot}\| \tag{5.91}$$

assuming that the columns of $\mathbf{G}$ are normalized. Finding the nearest neighbor of a k dimensional vector out of n canditates with $k \ll n$ can efficiently be done using kd-trees [7]. Finding all nearest neightbors for n points thus has a runtime complexity of $O(n \log(n))$ and can furhter be sped up using parallelization. This is a major boost compared to $O(n^3)$ complexity of the Hungarian algorithm for solving the linear assignment problem. Consequently this relaxation to a non-bijective nearest neighbor problem is the most common approach to tackle correspondence

problems in general. In addition to the computational benefits, the nearest neighbor problem is generic. It does not constrain source and target shape to be consistently discretized, *e.g.* it allows to match a uniformly discretized low resolution template to a high resolution, irregularly discretized 3d-scan.

For more complicated objectives, *e.g.* the ones resulting from pairwise descriptors, the optimization problem is usually simplified by a *convex relaxation* of the constraints. A main reason for the difficulty of combinatorial optimization problems is that the feasible set of solutions is disconnected and thus gradient based methods are not applicable. In fact there is no such thing as a gradient of a function defined on the set of permutation matrices since this set does not have a tangent space (where a gradient is ought to live). Another desirable property of the domain over which one is optimizing is *convexity*. Convexity of the domain allows to apply methods such as projected gradient descent (projection on non-convex domains is not well defined). The smallest convex set containing another set is known as its *convex hull*. The convex hull of the set of permutation matrices is called the *Birkhoff polytope* and consists of the set of bistochastic matrices

$$\begin{aligned}\mathcal{B}_n &= \{\mathbf{B} \in \mathbb{R}_+^{n\times n} | \mathbf{B}1 = 1,\, \mathbf{B}^\top 1 = 1\} \\ &= \{\mathbf{B} \in [0,1]^{n\times n} | \mathbf{B}1 = 1,\, \mathbf{B}^\top 1 = 1\}\end{aligned} \tag{5.92}$$

i.e. matrices with non-negative entries where each row and each column sums up to one (obviously the entries are bounded by 1). While optimization over the Birkhoff polytope is significantly easier than optimizing over the set of permutations it comes with a price: the optimizer does in general not encode a pointwise correspondence (as a permutation would do), thus post processing is necessary in case a pointwise correspondence is sought. One obvious post processing method is to look for the closest permutation matrix, which again results in a linear assignment problem:

$$\arg\max_{\mathbf{P}\in\Pi_n} \langle \mathbf{P}, \mathbf{B} \rangle \tag{5.93}$$

or, following what we described before, just pick the maximum entry in each row (if bijectivity is not critical).

The term *relaxation* describes the process of weakening some of the constraints that define the domain of a problem. The resulting (relaxed) problem has a larger domain in which optimization is easier. *Convex relaxations* as described above (domain replaced by a convex superset) are the most popular as they allow to apply a variety of well established optimization methods based on (possibly weak) derivatives. However for a given problem there exist many different relaxations, *e.g.* derived from different equations defining the original domain.

Let us recap the two definitions for the set of permutation matrices from Equa-

tion (5.49) and Equation (5.51):

$$\Pi_n = \{\mathbf{P} \in \{0,1\}^{n\times n} | \mathbf{P}1 = 1,\, \mathbf{P}^\top 1 = 1\} \tag{5.49 revisited}$$
$$\Pi_n = \{\mathbf{P} \in \{0,1\}^{n\times n} | \mathbf{P}\mathbf{P}^\top = \mathbf{I}\} \tag{5.51 revisited}$$

Relaxing the integer constraints in Equation (5.49) to

$$\{\mathbf{B} \in [0,1]^{n\times n} | \mathbf{B}1 = 1,\, \mathbf{B}^\top 1 = 1\} \tag{5.92 revisited}$$

leads to the convex relaxation described above. Relaxing them in Equation (5.51) makes the new domain the space of *orthogonal matrices*

$$O_n = \{\mathbf{B} \in \mathbb{R}^{n\times n} | \mathbf{P}\mathbf{P}^\top = \mathbf{I}\}. \tag{5.94}$$

While the space of orthogonal matrices as defined in Equation (5.94) is not convex, optimization on it is a well understood problem in many disciplines, including but not limited to Computer Vision. For instance whenever the poses (location and orientation) of objects or a cameras need to be estimated optimization is performed over the space of Euclidean transformations $E_n = O_n \times \mathbb{R}^3$. The main reason why optimization on O_n (or E_n) is feasible (at least local optima can be found efficiently) is due to $(O_n, \cdot)$ (with $\cdot$ being the usual matrix product) being a *Lie group*. A Lie group $(G, \cdot)$ is a group (existence of neutral element, for each element the inverse element is also element of the group, associativity) that additionally is a smooth manifold (see Definition 4). Additionaly the group operations (multiplication and inversion) must be smooth operations. We already know that the manifold property allows local parametrizations with parameters from Euclidean spaces (and thus many tools from optimization on Euclidean spaces can be carried over). The additional group property allows to turn the investigation of any element (and its neighborhood) to the investigation of the identity element and its neighborhood. In particular all tangent spaces are the same, the so called *Lie algebra* associated to the Lie group.
One thing to note though is that O_n consists of two disconnected components $SO_n = \{B \in O_n | \det(B) = 1\}$ and $O_n \backslash SO_n = \{B \in O_n | \det(B) = -1\}$. In many applications on is interested in *orientation preserving* orthogonal matrices with $\det(B) = 1$.
We have already discussed the probably most popular optimization problem over the space of orthogonal matrices, namely the *orthogonal Procrustes problem*

$$B^* = \operatorname*{arg\,min}_{B\in O_n} \|BX - Y\|_F^2 = \operatorname*{arg\,min}_{B\in O_n} \sum_{i=1}^{k} \|Bx_i - y_i\|^2 \tag{5.95}$$

where one is looking for the orthogonal matrix aligning the two tupels $(x_i)_{i=1}^k \subset \mathbb{R}^n$ and $(y_i)_{i=1}^k \subset \mathbb{R}^n$, i.e. bringing x_i as close as possible to y_i for all i and how to solve it using the *iterative closest point* algorithm.

5.6 Functional maps

Let us have a closer look at the convex relaxation we introduced in the previous section (Equation (5.92)). A possible interpretation for a matrix $\mathbf{B} \in \mathcal{B}_n$ can be obtained by multiplying an *unit vector* $e_j = \big(0, \dots, 1, \dots, 0\big)^\top$ with it (from the right). The result is the j-th column of $\mathbf{B}$ which is an n-dimensional vector of non-negative number summing to one. Each of the entries $\mathbf{B}_{ij}$ can thus be interpreted as the *probablity* that a vertex j from the source shape being mapped to a vertex i of the target shape. In fact multiplying $\mathbf{B}$ with any vector v keeps the sum of its elements the same $\sum_i (\mathbf{B}v)_i = \sum_i \sum_k \mathbf{B}_{ik} v_k = \sum_k \sum_i \mathbf{B}_{ik} v_k = \sum_k v_k$. Thus $\mathbf{B}$ can be (and often is) interpreted as a discrete *coupling* as in the Kantorovich formulation of *optimal transport* [25]. For a detailed introduction to optimal transport see [41]. For some applications of optimal transport in non-rigid shape analysis and geometry processing see [28, 50].

At this point we need to point out a major inconsistency in the literature (including our own publications). In the previous sections we have associated n-dimensional vectors with functions defined on triangle meshes (with n vertices). The association was done by interpreting the entries of the vector as coefficients of certain basis functions, namely *hat functions* (Equation (3.38)). The optimal transport interpretation of $\mathbf{B} \in \mathcal{B}_n$ however only works if the basis "functions" are chosen to be *dirac distributions* centered at the respective vertices. We want to provide a rectified version of Equation (5.92) that is more consistent with the notion of hat functions. Let $f(x) = \sum_i \mathbf{f}_i \psi_i(x)$ be a function on the source mesh $\mathcal{X}$ and $g(x) = \sum_i \mathbf{g}_i \psi_i(x)$ be the function on the target mesh $\mathcal{Y}$ with coefficient vector $\mathbf{g}$ given by $\mathbf{g} = \mathbf{B}\mathbf{f}$. We want to find constraints on $\mathbf{B}$ such that $\int_\mathcal{X} f(x)dx = \int_\mathcal{Y} g(x)dx$ for all possible input functions f. We derive

$$\int_\mathcal{X} f(x)dx = \int_\mathcal{X} \sum_i \mathbf{f}_i \psi_i(x)dx \tag{5.96}$$

$$= \sum_i \mathbf{f}_i \int_\mathcal{X} \psi_i(x)dx \tag{5.97}$$

$$\int_\mathcal{Y} g(x)dx = \int_\mathcal{Y} \sum_j \mathbf{g}_j \psi_j(x)dx \tag{5.98}$$

$$= \int_\mathcal{Y} \sum_j \sum_i \mathbf{B}_{ji} \mathbf{f}_i \psi_j(x)dx \tag{5.99}$$

$$= \sum_i \mathbf{f}_i \sum_j \mathbf{B}_{ji} \int_\mathcal{Y} \psi_j(x)dx \tag{5.100}$$

and see that the integrals are the same if and only if

$$\int_{\mathcal{X}} \psi_i(x)dx = \sum_j \mathbf{B}_{ji} \int_{\mathcal{Y}} \psi_j(x)dx \quad \forall i \tag{5.101}$$

$$\Leftrightarrow \mathbf{B}^\top \mathbf{a}^{\mathcal{Y}} = \mathbf{a}^{\mathcal{X}} \tag{5.102}$$

with $a^{\mathcal{X}} = (\int_{\mathcal{X}} \psi_i(x)dx)_{i=1}^n$, $\mathbf{a}^{\mathcal{Y}} = (\int_{\mathcal{Y}} \psi_i(x)dx)_{i=1}^n$ being the vectors of area elements (area of Voronoi cells around each vertex) on $\mathcal{X}$ and $\mathcal{N}$ respectively. Since the mass preservation should hold in both directions we end up with the constraints

$$\mathbf{B}^\top \mathbf{a}^{\mathcal{Y}} = \mathbf{a}^{\mathcal{X}} \tag{5.103}$$

$$\mathbf{B}\mathbf{a}^{\mathcal{X}} = \mathbf{a}^{\mathcal{Y}} \tag{5.104}$$

which are only equivalent to Equation (5.92) when the vertices are uniformly distributed over the mesh (which is almost never the case). This type of discretization inconsistency is widely seen in the literature. A thorough analysis of the consequences of this inaccuracy is yet outstanding. Most papers tend to evaluate on datasets with consistent topology (*e.g.* fitted templates for the FAUST dataset [11]). This could hide problems stemming from the discretization issue and possibly even exploit the consistency (*e.g.* by implicetely solving a graph isomorphism problem). We thus encourage every researcher in the field to evaluate their methods on datasets of independently remeshed shapes.

Let us move the focus from the constraints on the matrix representation of a correspondence back to the fundamental observation that this matrix can be interpreted as a mapping between function spaces. In [38] the authors have taken this observation one step further and established the term *functional maps* that has since become omnipresent in the literature on non-rigid shape correspondence and analysis. When researchers speak about *functional maps* they most often refer to the fact that the mapping between function spaces has more than one (discrete) representation and one of those representations is particularly appealing.

First let us note that transfering functions from one shape to the other using matrix-vector products makes the mapping between the function spaces, encoded by this matrix, *linear* (independently of additional constraints such as 5.92 or 5.94). A well known fact from linear algebra is that linear mappings have different representations depending on the chosen bases of domain and co-domain. Given a (possibly infinite but countable) basis $\{\Phi_i^{\mathcal{X}}\}_i$ of functions defined on $\mathcal{X}$ and $\{\Phi_i^{\mathcal{Y}}\}_i$ for functions on $\mathcal{Y}$ respectively, the matrix representation $\mathbf{C}$ of a linear mapping T between the two function spaces is given by the (unique) coefficients of the images of $\Phi_i^{\mathcal{X}}$ in the basis $\{\Phi_i^{\mathcal{Y}}\}_i$:

$$T(\Phi_i^{\mathcal{X}}) = \sum_j C_{ij} \Phi_j^{\mathcal{Y}} \tag{5.105}$$

In case of infinite dimension of the function spaces (*e.g.* $L^2(\mathcal{X})$ and $L^2(\mathcal{Y})$) the matrix has infinite many rows and columns. If we are interested in the coefficients of $T(f)$ or an aribtrary function $f = \sum_i a_i \Phi_i^{\mathcal{X}}$, we get them by multiplication of the vector $\mathbf{a} = \left(a_0, \ldots, a_{n_{\mathcal{X}}-1}\right)^\top$ with $\mathbf{C}$:

$$T(f) = T(\sum_i a_i \Phi_i^{\mathcal{X}}) \tag{5.106}$$

$$= \sum_i a_i T(\Phi_i^{\mathcal{X}}) \tag{5.107}$$

$$= \sum_i a_i \sum_j C_{ij} \Phi_j^{\mathcal{Y}} \tag{5.108}$$

$$= \sum_j \underbrace{\sum_i a_i C_{ij}}_{b_j} \Phi_j^{\mathcal{Y}} \tag{5.109}$$

$$\Rightarrow \mathbf{b} = \mathbf{C}^\top \mathbf{a} \tag{5.110}$$

The representation as *bistochastic matrix* 5.92 or its rectified version 5.103 corresponds to choosing *hat functions* (or possibly *diracs*) as bases for both function spaces. These representations consequently only make sense in a discrete setup, *i.e.* when the considered function spaces are the (finite dimensional) spaces of *piecewise linear* functions on two triangle meshes.

A desirable property of the chosen bases is that the possibly high dimensional (for hat functions the size is $|V_{\mathcal{X}}| \times |V_{\mathcal{Y}}|$) or even infinite matrix representation $\mathbf{C}$ can be well approximated by a finite dimensional submatrix $\tilde{\mathbf{C}} \in R^{k_{\mathcal{X}} \times k_{\mathcal{Y}}}$, $\mathbf{C}_{ij} = \tilde{\mathbf{C}}_{ij} \forall i < k_{\mathcal{X}}, \forall j < k_{\mathcal{Y}}$ with $k_{\mathcal{X}} \ll |V_{\mathcal{X}}|$, $k_{\mathcal{Y}} \ll |V_{\mathcal{Y}}|$.

In Section 4 we learned about the eigenfunctions of the Laplace Beltrami operator as an alternative basis for functions defined on manifolds. We have seen that they come with a natural order given by the magnitude of the corresponding eigenvalues which equals their *Dirichlet energy*, a measure for their variability or frequency. Thus functions can be well approximated (in the L^2 sense) by a linear combination of the first Laplace Beltrami eigenfunctions. This is exactly the property we were looking for as it allows to approximate the (full) matrix representation of the functional map by a relatively small submatrix. The submatrix of course corresponds to a different functional map than the full matrix. It can be shown that the new functional map is a low pass approximation of the original correspondence and that the same holds for the two graphs of the (soft) correspondence in the so called *product manifold* [45] [2]. The next section will provide more details about the product manifold.

Another neat property of the Laplace Beltrami eigenfunctions (which we will abbriviate as eigenfunctions in the following) is their ortonormality, *i.e.* $\langle \phi_j^{\mathcal{Y}}, \phi_k^{\mathcal{Y}} \rangle = \delta_{jk}$ (for $\mathcal{X}$ accordingly). This leads to the observation that the values $\mathbf{C}_{ij}$ equal the inner

product of the image of the basis function $\phi_i^{\mathcal{X}}$ (under T) with the basis function $\phi_j^{\mathcal{Y}}$:

$$\langle T(\phi_i^{\mathcal{X}}), \phi_j^{\mathcal{Y}} \rangle = \langle \sum_k \mathbf{C}_{ik} \phi_k^{\mathcal{Y}}, \phi_j^{\mathcal{Y}} \rangle \tag{5.111}$$

$$= \sum_k \mathbf{C}_{ik} \langle \phi_k^{\mathcal{Y}}, \phi_j^{\mathcal{Y}} \rangle \tag{5.112}$$

$$= \mathbf{C}_{ij} \tag{5.113}$$

It needs to be pointed out that the eigendecompositions of $\triangle_{\mathcal{X}}$ and $\triangle_{\mathcal{Y}}$ are not unique (signflips, multidimensional eigenspaces). Depending on the particular choice if eigenfunctions the matrix $\mathbf{C}$ varies, however some properties of $\mathbf{C}$ are independent of the particular choice of eigenfunctions.

Lemma 9 Let $m\colon \mathcal{X} \to \mathcal{Y}$ be an isometry with corresponding functional map T. Let $\mathbf{C}$ be the (any) matrix representation of T obtained by choosing the (one choice of) Laplace Beltrami eigenfunctions as bases on both shapes. Then

1. If all eigenspaces on $\mathcal{X}$ are one-dimensional (*i.e.* no repeating eigenvalues), C is a diagonal matrix with values 1 and -1 on the diagonal.

2. For general eigenspaces C is block diagonal with each block being an orthogonal matrix of the size of the corresponding eigenspace.

Proof. First note that $\triangle_{\mathcal{X}}$ and $\triangle_{\mathcal{Y}}$ have the same spectrum (set of eigenvalues) (*c.f.* Lemma 8). Let now $(\lambda_i, \phi_i^{\mathcal{X}})$ be an eigenpair of $\triangle_{\mathcal{X}}$. Again from Lemma 8 we recall that $T(\phi_i)$ (what we called $\tilde{\phi}_i^{\mathcal{X}}$) is an eigenfunction of $\triangle_{\mathcal{Y}}$ with eigenvalue λ_i. Let now $(\lambda_j, \phi_j^{\mathcal{Y}})$ be an eigenpair of $\triangle_{\mathcal{Y}}$. If $\lambda_i \neq \lambda_j$ the two eigenfunctions $T(\phi_i^{\mathcal{X}})$ and $\phi_j^{\mathcal{Y}}$ belong to different eigenspaces and consequently (since eigenspaces are orthogonal) $C_{ij} = \langle T(\phi_i^{\mathcal{X}}), \phi_j^{\mathcal{Y}} \rangle = 0$. In the case of one dimensional eigenspaces this applies to all non-diagonal entries of $\mathbf{C}$, since $i = j \Leftrightarrow \lambda_i = \lambda_j$. For the diagonal entries we know that since the eigenspaces are one dimensional there are only two possibilities to obtain an element with unit norm, thus $T(\phi_i^{\mathcal{X}}) = \phi_i^{\mathcal{Y}}$ or $T(\phi_i^{\mathcal{X}}) = -\phi_i^{\mathcal{Y}}$ and consequently $C_{ii} = \pm 1$.
Let us now consider a n_j dimensional eigenspace corresponding to the eigenvalue λ_j. To simplify notation we start the indexing of the eigenvectors at 1 and thus have the two sets of eigenfunctions $\{\phi_1^{\mathcal{Y}}, \ldots, \phi_{n_j}^{\mathcal{Y}}\}$ and $\{T(\phi_1^{\mathcal{X}}), \ldots, T(\phi_{n_j}^{\mathcal{X}})\} = \{\tilde{\phi}_1^{\mathcal{Y}}, \ldots, \tilde{\phi}_{n_j}^{\mathcal{Y}}\}$ of $\triangle^{\mathcal{Y}}$. We recall that the eigenspaces are unique up to an orthogonal transformation, such there is an orthogonal matrix $R_j = (r_{tp})_{t,p \leqslant n_j} \in O(n_j)$ with

$$\tilde{\phi}_p^{\mathcal{Y}}(y) = \sum_{t=1}^{n_j} r_{tp} \phi_t^{\mathcal{Y}}(y) \quad \forall 1 \leqslant p \leqslant n_j \tag{5.114}$$

For the entries C_{pq} of the $n_j \times n_j$ submatrix of C we thus get

$$C_{pq} = \int_{\mathcal{Y}} \tilde{\phi}_p(y)\phi_q(y)dy \tag{5.115}$$

$$= \int_{\mathcal{Y}} \sum_{t=1}^{n_j} r_{tp}\phi_t^{\mathcal{Y}}(y)\phi_q(y)dy \tag{5.116}$$

$$= \sum_{t=1}^{n_j} r_{tp} \int_{\mathcal{Y}} \phi_t^{\mathcal{Y}}(y)\phi_q(y)dy \tag{5.117}$$

$$= r_{qp} \tag{5.118}$$

□

As a consequence the full matrix C is orthogonal and its support (non-zero entries) are concentrated around the diagonal. These properties approximately hold for near isometries as well. The original paper thus proposes to enforce orthogonality of C in the optimization, follow up papers (*e.g.* [44]) further add priors on the non-diagonal entries.

5.7 The product manifold and the graph of a correspondence

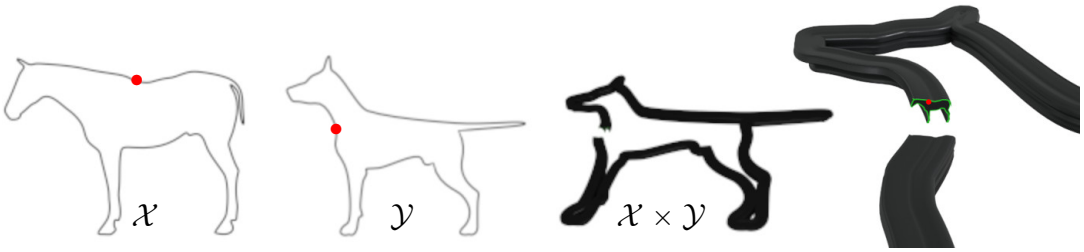

Figure 5.8: The productspace $\mathcal{X} \times \mathcal{Y}$ of two contours $\mathcal{X}$ (the horse contour) and $\mathcal{Y}$ (the dog contour) can be conceptually visualized as a two dimensional manifold embedded in $\mathbb{R}^3$. Each horizontal cut through it corresponds to a copy of $\mathcal{X}$, every vertical cut to a copy of $\mathcal{Y}$. Each point on the *product manifold* thus corresponds to a pair of points on the two contours (red circles).

Let $\mathcal{X}, \mathcal{Y}$ be manifolds of dimensions $m_{\mathcal{X}} \leqslant m_{\mathcal{Y}}$ respectively. Each mapping $m\colon \mathcal{X} \to \mathcal{Y}$ has an associated graph

$$\Gamma_m = \{(x, m(x)) | x \in \mathcal{X}\} \subset \mathcal{X} \times \mathcal{Y} \tag{5.119}$$

The space $\mathcal{X} \times \mathcal{Y}$ is itself a manifold of dimension $m_\mathcal{X} + m_\mathcal{Y}$ with metric tensor given by

$$g_{\mathcal{X}\times\mathcal{Y}} = \begin{pmatrix} g_\mathcal{X} & \\ & g_\mathcal{Y} \end{pmatrix} \tag{5.120}$$

The graph Γ_m is an $m_\mathcal{X}$-dimensional submanifold of $\mathcal{X} \times \mathcal{Y}$. Figure 5.8 provides a conceptual illustration for two 1-dimensional shapes.
There is a close link between the regularity of the matching m and its graph Γ_m: The matching is continuous iff its graph does not have jumps, *i.e.* Γ_m is a manifold without boundary, see Figure 5.9 for an illustration. In this thesis we focus on the matching of three dimensional shapes which are modelled as two dimensional manifolds. In this setup, the product manifold is four dimensional and the sought graph is a two dimensional submanifold. The authors of [47] formulate the matching problem as a constraint optimization problem where the constraints enforce the regularity of the graph and therefore the regularity of the correspondence:

- The graph should cover both the domain $\mathcal{X}$ and the subdomain $\mathcal{Y}$, *i.e.* $\{x|(x,y) \in \Gamma_m\} = \mathcal{X}$ and *i.e.* $\{y|(x,y) \in \Gamma_m\} = \mathcal{Y}$.
- The graph should be closed, *i.e.* not have a boundary (the authors introduce a so called boundary operator to formalize this property).

They additionally equip the product space with a (non negative) cost function that is accumulated over the graph of the correspondence. From the positivity of the cost function it is intuitivly clear that the graph will avoid unnecessary turns, in particular each point in $\mathcal{X}$ and in $\mathcal{Y}$ will be covered exactly once.
The cost of each element $(x, y) \in \Gamma_m$ is basically the difference of local properties of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ and models how much bending and stretching needs to be applied to make x and y be similar. The particular choice of the properties is again based on quantities from differential geometry, we refer to the reader to the original paper for details. We want to emphasize though that the formulation of the problem is basically the one we have seen when introducing *pointwise descriptors* in Section 5.3. In particular it leads to a *linear* costfunction with constraints, similar to the linear assignment problem in Equation (5.65)[2]. In fact the first of the two constraints above can be discretized to the requirement that the discrete representation of the graph should be a permutation matrix. The second constraint (closed graph) however makes the resulting *integer linear program* much more complicated (in fact it is NP-hard). The underlying mathematical reason for the higher complexity lies in the fact that the constraint matrix is not *totally unimodular* (as it

[2]In contrast to all the approaches we have discussed so far the discrete correspondence between two triangular meshes does not map vertices to vertices but (generalized) triangles to (generalized) triangles.

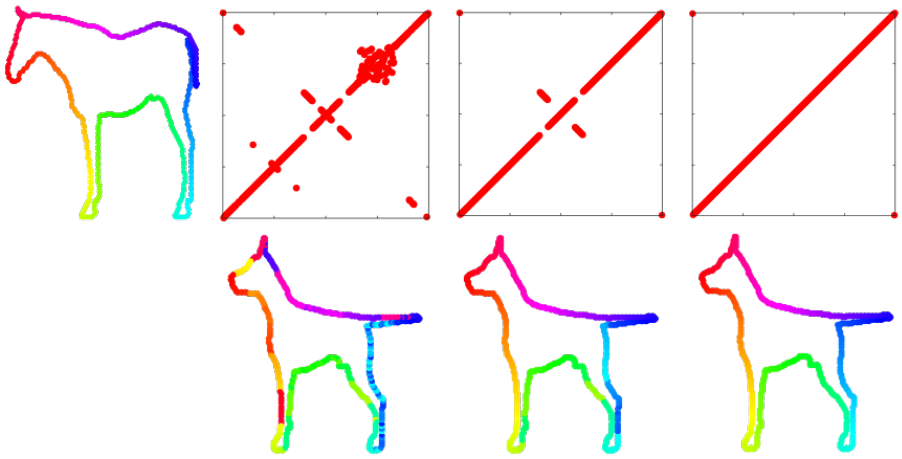

Figure 5.9: The graph of a continuous matching between two contours is a closed curve on the two dimensional product space. Discontinuities of the correspondence (color coded) can be identified with discontinuities of its graph. We have "unfolded" the product space (which topologically is a torus, see Figure 5.8) for a simpler visualization.

is for LAPs). Totally unimodular constraint matrices lead to significantly easier optimization problems, see for example [48]. The authors address this problem by relaxing the integer constraints to *stochasticity* constraints similar to what we have seen with the *bistochastic* matrices (Birkhoff polytope). This leads to a classical linear program with a unique optimum that can efficiently be calculated. The price, however, is that the solution is not integral and thus needs post processing to be converted to a classical correspondence.
For lower dimensional scenarios, where *e.g.* $\mathcal{X}$ is one dimensional, the problem corresponds to a *shortest path* problem which can be solved efficiently. The authors of [47] and [27] exploit this for finding a correspondence between two planar shapes (one dimensional contours as in Figure 5.9) and correspondences between one dimensional contours and two dimensional surfaces, respectively. An alternative approach to relax minimal surface problems with higher dimensions (*e.g.* four dimensional product space and two dimensional correspondence graph) to a convex approximation was introduced in [56]. While the application in the paper is image matching, the introduced tools and techniques can potentially be used for shape matching as well.

Part II

A Selection of Own Publications

In the following, we present the three peer-reviewed publications that form the cumulative content of this thesis. Following a short summary of the publication, highlighting my individual contributions we include a copy of the accepted version of each paper.

Chapter 6

Applying Random Forests to the Problem of Dense Non-rigid Shape Correspondence

Obtaining a meaningful correspondence between two shapes requires some kind of prior on the type of deformation one of the shapes needs to undergo to become the other one. The de facto standard prior is that this deformation is an (intrinsic) isometry. This prior motivates the use of isometry invariant descriptors (pointwise and pairwise) as introduced in the previous chapter. In practice the isometry assumption is never exactly met but as long as the descriptors are stable with respect to some deviation from isometries, the resulting optimization problems still lead to satisfactory results. In many scenarios the deformations can however not even be considered a near-isometric. An obvious example are changes in topology, *e.g.* a person in *T-pose* (sphere topology) and a person with one of the hands touching parts of the body (torus topology). Unfortunatly the language of differential geometry does not provide another class of deformations that would on the one hand cover this type of deformations while on the other hand still leading to meaningful correspondences.

In this paper we assume the class of deformations to be given in terms of an example dataset (triangular meshes with correspondences) rather than a rigorous mathematical property. We pass the responsibility of introducing invariance with respect to this class of deformations to a machine learning model. This is a common practice in related discipline such as image classification or image segmentation. We treat the correspondence problem as a classification problem of the individual vertices to a label set which corresponds to a reference mesh. During training the structure of a set of decision trees (a so called *random decision forests*) is created. At inference time each vertex is individually routed through all of the trees leading to a probablity distribution on the set of possible labels (vertices on the reference shape). We further introduce a regularizer based on metric distortion and functional maps.

Publication:	Matthias Vestner, Emanuele Rodolá, Thomas Windheuser, Samuel Buló, and Daniel Cremers. "Applying Random Forests to the problem of Dense Non-Rigid Shape Correspondence". In: *Breuß, M., Bruckstein, A., Maragos, P., Wuhrer, S. (eds) Perspectives in Shape Analysis. Mathematics and Visualization. Springer, Cham. DOI: 10.1007/978-3-319-24726-7_11*

Individual contribution		
	Problem definition	significantly contributed
	Literature survey	significantly contributed
	Implementation	contributed
	Experimental evaluation	significantly contributed
	Preparation of the manuscript	significantly contributed

Applying Random Forests to the Problem of Dense Non-Rigid Shape Correspondence

Matthias Vestner, Emanuele Rodolà, Thomas Windheuser, Samuel Rota Bulò, and Daniel Cremers

Abstract We introduce a novel dense shape matching method for deformable, three-dimensional shapes. Differently from most existing techniques, our approach is general in that it allows the shapes to undergo deformations that are far from being isometric. We do this in a supervised learning framework which makes use of training data as represented by a small set of example shapes. From this set, we learn an implicit representation of a shape descriptor capturing the variability of the deformations in the given class. The learning paradigm we choose for this task is a random forest classifier. With the additional help of a spatial regularizer, the proposed method achieves significant improvements over the baseline approach and obtains state-of-the-art results while keeping a low computational cost.

1 Introduction

Matching three-dimensional shapes is a pervasive problem in computer vision, computer graphics and several other fields. Nevertheless, while the advances made by works such as [14, 4, 23, 29, 2, 10] have been dramatic, the problem is far from being solved.

Many methods in shape matching use a notion of similarity that is defined on a very general set of possible shapes. Due to the highly ill-posed nature of the shape matching problem, it is very unlikely that a general method will reliably find good matchings between arbitrary shapes. In fact, while many matching methods (such as methods based on metric distortion [22, 4, 20] and eigen-decomposition of the Laplacian [23, 29, 2]) mostly capture near-isometric deformations, others might consider too general deformations which are not consistent with the human intu-

Matthias Vestner, Emanuele Rodolà, Thomas Windheuser, Daniel Cremers
Technische Universität München
Samuel Rota Bulò
Fondazione Bruno Kessler, Trento

ition of correspondence. In applications where the class of encountered shapes is in-between, adapting the matching methods at hand is often very tedious.

In this paper we try to bridge the gap between general shape matching methods and application-specific algorithms by taking a learning-by-examples approach.

In our scenario, we assume to have a set of training shapes which are equivalent up to some class of non-isometric deformations. Our goal is to learn from these examples how to match two shapes falling in the equivalence class represented by the training set. To this end, we treat the shape matching problem as a classification problem, where input samples are points on the shape manifold and the output class is an element of a canonical label set, which might e.g. coincide with the manifold of one of the shapes in the training set. A first contribution of this paper consists in a new random forest classifier, which can tackle this unconventional classification problem in an efficient and effective way, starting from a general parametrizable shape descriptor. Our classifier is designed in a way to randomly explore the descriptor's parametrization space and find the most discriminative features that properly recover the transformation map characterizing the shape category at hand. In this work, we consider the *wave kernel signature* (WKS) [2] as the shape descriptor. This descriptor is known to be invariant to isometric transformations, but the forest can effectively exploit it to match shapes that undergo non-rigid and non-isometric deformations.

In some sense, the output of the random forest can be seen as a new descriptor by itself that is tuned to the shapes and deformations appearing in the training set. In this respect, the proposed method is complementary to existing shape descriptors as it can improve the performance of a given descriptor [11, 12, 32]. Early attempts to apply machine learning techniques to the problem of non-rigid correspondence ([28], [25]) consider shapes represented by signed distance functions. We follow the intrinsic view point, considering shapes given by their boundary surface, seen as a Riemannian manifold.

One of the main benefits of our approach is the fact that the random forest classifier gives for each point on the shape an ordered set of matching candidates, hence delivering a dense point-to-point matching. Since such a descriptor does not include any spatial regularity, we propose to use a regularization technique along the lines of the *functional maps framework* [16]. We experimentally validate that the proposed learning approach improves the underlying general descriptor significantly, and it performs better than other state-of-the-art matching algorithms on equivalent benchmarks.

An earlier version of this work was published in [21].

1.1 Intrinsic point descriptors

We consider 3D shapes that are represented by their boundary surface, a two-dimensional Riemannian manifold (M,g) without boundary. A point descriptor is a function ϕ that assigns to each point on the surface an element of a metric space

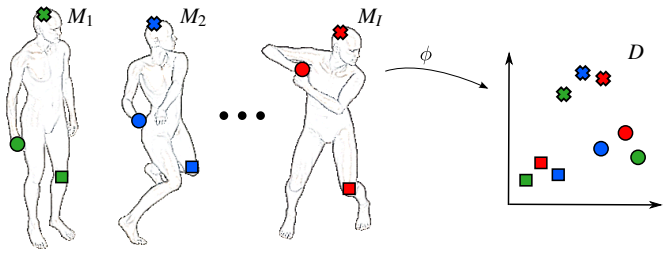

Fig. 1 A good point descriptor should at the same time assign similar values to corresponding points on deformed shapes and dissimilar values to non-corresponding points

D, the descriptor space. A good point descriptor should satisfy two competing properties (Figure 1):

- **deformation-invariance:** it should assign similar values to corresponding points on deformed shapes
- **discriminativity:** it should well distinguish non-corresponding points

While it is in principle possible to construct a descriptor that is invariant under an arbitrary large class of deformations (e.g. the constant function), it is evident that there will always be a tradeoff between deformation-invariance and discriminativity.

The descriptors we consider are based on the spectrum of the *Laplace-Beltrami operator* $\Delta_M = -\mathrm{div}_M(\nabla_M)$. Being a symmetric operator the spectrum of Δ_M consists of real eigenvalues $\lambda_1, \lambda_2, \ldots$ and the corresponding eigenfunctions $\gamma_1, \gamma_2, \ldots$ can be chosen to be real valued and orthonormal. Moreover, Δ_M is a non-negative operator with a one-dimensional kernel and a compact pseudo-inverse, so we can order the eigenvalues $0 = \lambda_1 < \lambda_2 \leq \ldots$ and assign to each point $x \in M$ a vector $p \in \mathbb{R}^{2K}$, $p = (\lambda_1, \ldots, \lambda_K, \gamma_1(x), \ldots, \gamma_K(x))$. The Laplace Beltrami Operator is purely intrinsic as it is uniquely determined by the metric tensor $g = (g_{ij})_{i,j=1}^2$ (respectively its inverse $(g^{ij})_{i,j=1}^2$):

$$\Delta_M = \frac{1}{\sqrt{\det g}} \sum_{i,j=1}^{2} \frac{\partial}{\partial x_i} \left(g^{ij} \sqrt{\det g} \frac{\partial}{\partial x_j} \right). \tag{1}$$

As a consequence the eigenvalues λ_k as well as the corresponding eigenspaces do not change whenever a shape undergoes an isometric deformation. The eigenbases however are not uniquely determined, even in the case of one dimensional eigenspaces the normalized eigenvectors are only unique up to sign. Nevertheless from the representation p it is possible to construct descriptors that are invariant under isometric deformations. Given a collection $(t_i)_{i=1}^n$ of positive numbers, the Heat Kernel Signature (HKS)

$$HKS(p) = \left(\sum_k \exp(-\lambda_k t_i) \gamma_k(x)^2 \right)_{i=1}^n \in \mathbb{R}^n \tag{2}$$

is a n-dimensional intrinsic point-descriptor [29]. From a physical point of view each component tells us how much heat $u(x,t_i)$ remains at point x after time t_i when the initial distribution of heat is a unit heat source at the very same point:

$$\Delta u = u_t \tag{3}$$
$$u(0,\cdot) = \delta_x \tag{4}$$

Since the class of isometric deformations includes reflections, any intrinsic descriptor will assign identical values to a point and its symmetric counterpart, whenever shapes exhibit bilateral intrinsic symmetries. Using information about the symmetry [18] or making use of extrinsic information as in [27] would overcome this problem.

From a signal processing viewpoint HKS can be seen as a collection of low-pass filters and thus it is not appropriate to localize features, see figure 2. Motivated by this observation Aubry et al. [2] introduced the Wave Kernel Signature (WKS), a descriptor where the low-pass filters are replaced by band pass filters:

$$WKS(p) = \left(\sum_k f_{(e_i,\sigma_i^2)}(\lambda_k)^2 \gamma_k(x)^2 \right)_{i=1}^{n} \in \mathbb{R}^n \tag{5}$$

Here the parameters (e_i, σ_i^2) correspond to mean and variance of the log-normal energy distributions

$$f_{(e,\sigma^2)}(\lambda) \propto \exp(-\frac{(\log e - \log \lambda)^2}{2\sigma^2}) \tag{6}$$

The authors propose fixed values for the parameters (e_i, σ_i) depending on the truncated spectrum of the Laplace-Beltrami-operator. Moreover they equip the descripor with a metric related to the L^1-distance.

In this work the parameters will be learned from training data, a distance function between vector valued descriptors is unneeded since descriptors are compared component wise in a hierarchical manner (2.1.1, 2.1.3).

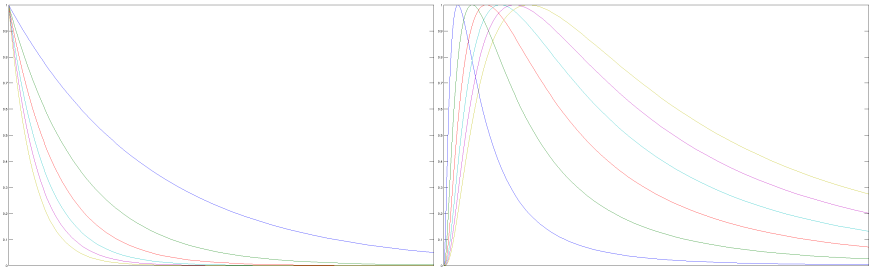

Fig. 2 The weighting functions of the heat kernel signature (left) can be seen as low-pass filters, the ones of the wave kernel signature (right) in contrary behave like band-pass filters.

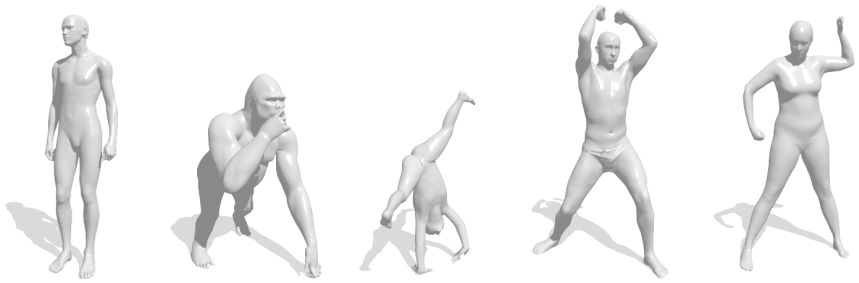

Fig. 3 Finding a correspondence between shapes should be feasible even if they are far from being isometric.

Both, HKS and WKS, are invariant under isometric deformations. However the human notion of similarity by far exceeds the class of isometries. Asking for a correspondence between an adult and a child or even an animal like a gorilla is a feasible task for us. Figure (3) shows examples of shapes taken from different datasets ([5],[21],[19],[1]) that could in principle be put into correspondence. By choosing application dependent parameters one can achieve descriptors that are less sensitive to the type of deformation one is interested in. In this work we implicitly determine optimal parameters when the deformation class is represented by a set of training shapes with known ground truth correspondence.

1.2 Discretized surfaces and operators

In practice the shapes are given as triangular meshes $M = (V_M, F_M)$. We will henceforth identify a shape M by the set of it vertices V_M. A one-to-one correspondence between two shapes can then be represented by a permutation matrix, a fuzzy correspondence, i.e. a function that assigns to each point a probability distribution over the other shape, respectively as a left-stochastic matrix. Functions defined on a shape become vectors and linear operators acting on them, e.g. the Laplace-Beltrami operator can be written as matrices. Inner products between functions are calculated via an area-weighted inner product between the vectors representing them. We chose the popular cotangent scheme [15] as the discretization of the Laplacian.

2 Dense Correspondence Using Random-Forests

In this work we treat the shape matching problem as a classification problem, where input samples are points on the shape and the output class is an element of a canonical label set, which might e.g. coincide with one of the shapes in the training set (the

reference shapes). The classifier we choose is a Random forest, designed in a way to randomly explore the descriptor's parametrization space and find the most discriminative features that properly recover the transformation map characterizing the shape category at hand. In this work, we consider the *wave kernel signature* (WKS) as the parametrizable point descriptor (weak classifier). In general other choices of parametrizable descriptors, e.g. HKS, are possible. As mentioned in section 1.1 any classifier based on isometry-invariant point descriptors can not distinguish a point from its symmetric counterpart. Thus the fuzzy outcome of the Random forest classifier has to be regularized in order to get a consistent correspondence.

2.1 Learning and Inference Using Random Forests

Random forests [3] are ensembles of decision trees that have become very popular in the computer vision community to solve both classification and regression problems with applications ranging from object detection, tracking and action recognition [9] to semantic image segmentation and categorization [26], and 3D pose estimation [30], to name just a few. The forest classifier is particularly appealing because its trees can be trained efficiently and techniques like bagging and randomized feature selection allow to limit the correlation among trees and thus ensure good generalization. We refer to [7] for a detailed review.

2.1.1 Inference.

In the context of shape matching, a decision tree comprised by the forest routes a point m of a test shape M from the root of the tree to a leaf node, where a probability distribution defined on a discrete label set L is assigned to the point. The path from the root to a leaf node is determined by means of binary decision functions called *split functions* located at the internal nodes, which given a shape point return $\mathtt{L}$ or $\mathtt{R}$ depending on whether the point should be forwarded to the left or to the right with respect to the current node. According to this inference procedure, each tree $t \in \mathscr{F}$ of a forest $\mathscr{F}$ provides a posterior probability $\mathrm{P}(\ell|m,t)$ of label $\ell \in L$, given a point $m \in M$ in a test shape M.

This probability measure is the one associated with the leaf of tree $t \in \mathscr{F}$ that the shape point would reach. The prediction of the whole forest $\mathscr{F}$ is finally obtained by averaging the predictions of the single trees:

$$\mathrm{P}(\ell|m,\mathscr{F}) = \frac{1}{|\mathscr{F}|} \sum_{t \in \mathscr{F}} \mathrm{P}(\ell|m,t)\,. \tag{7}$$

The outcome of the prediction over an entire shape M can be represented as a left-stochastic matrix $\mathtt{X}_M$ encoding the probabilistic canonical transformation, where

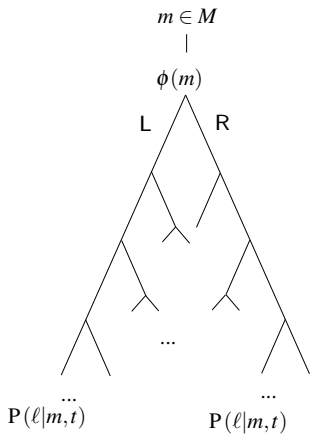

Fig. 4 At each inner node of a decision tree a binary split function is evaluated. Depending on the result the point m is either routed to the left or to the right. Leafs of the tree correspond to probability distributions in the label space. A random forest is a collection of mulitple decision trees.

$$(\mathtt{X}_M)_{ij} = \mathrm{P}(\ell_i | m_j, \mathscr{F}) \tag{8}$$

for each $\ell_i \in L$ and $m_j \in M$. Using Bayes' theorem we can further construct a fuzzy correspondence between two previously unseen shapes (i.e. no members of the training set).

2.1.2 Learning.

During the learning phase, the structure of the trees, the split functions and the leaf posteriors are determined from a training set. Let $\{(R_i, T_i)\}_{i=1}^{\mathsf{m}}$ be a set of m reference shapes R_i each equipped with a canonical transformation, i.e. a bijection $T_i : R_i \to L$ between the vertex set of the reference shape and the label set L. A training set $\mathbb{T}$ for the random forest is given by the union of the graphs of the mappings T_i, i.e.

$$\mathbb{T} = \{(r, T_i(r)) : r \in R_i,\, 1 \leq i \leq \mathsf{m}\}\,. \tag{9}$$

The learning phase that creates each tree forming the forest consists in a recursive procedure that starting from the root iteratively splits the actual terminal nodes. During this process each shape point of the training set is routed through the tree in a way to partition the whole training set across the terminal nodes. The decision whether a terminal node has to be further split and how the splitting will take place is purely local as it involves exclusively the shape points that have reached that node. A terminal node typically becomes a leaf of the tree if the depth of the node exceeds a given limit, if the size of the subset of training samples reaching the node is small enough, or if the entropy of the sample's label distribution is low enough. If this is the case, then the leaf node is assigned the label distribution of subset $\mathbb{S}$ of training samples that have reached the leaf, i.e.

$$\mathrm{P}(\ell|\mathbb{S}) = \frac{|\{(r,\ell) \in \mathbb{S}\}|}{|\mathbb{S}|} . \tag{10}$$

The probability distribution $\mathrm{P}(\cdot|\mathbb{S})$ will become the posterior probability during inference for every shape point reaching the leaf. Consider now the case where the terminal node is split. In this case, we have to select a proper split function $\psi(r) \in \{\mathsf{L},\mathsf{R}\}$ that will route a point r reaching the node to the left or right branch. An easy and effective strategy for guiding this selection consists in generating a finite pool Ψ of random split functions and retaining the one maximizing the information gain with respect to the label space L. The information gain $\mathrm{IG}(\psi)$ due to split function $\psi \in \Psi$ is given by the difference between the entropy of the node posterior probability defined as in (10) before and after having performed the split. In detail, if $\mathbb{S} \subseteq \mathbb{T}$ is the subset of the training set that has reached the node to be split and $\mathbb{S}^{\mathsf{L}}$, $\mathbb{S}^{\mathsf{R}}$ is the partition of $\mathbb{S}$ induced by the split function ψ then $\mathrm{IG}(\psi)$ is given by

$$\mathrm{IG}(\psi) = \mathrm{H}(\mathrm{P}(\cdot|\mathbb{S})) - \mathrm{H}(\mathrm{P}(\cdot|\mathbb{S})\,|\,\psi) , \tag{11}$$

where $\mathrm{H}(\cdot)$ denotes the entropy and

$$\mathrm{H}(\mathrm{P}(\cdot|\mathbb{S})\,|\,\psi) = \frac{|\mathbb{S}^{\mathsf{L}}|}{|\mathbb{S}|}\mathrm{H}\left(\mathrm{P}\left(\cdot|\mathbb{S}^{\mathsf{L}}\right)\right) + \frac{|\mathbb{S}^{\mathsf{R}}|}{|\mathbb{S}|}\mathrm{H}\left(\mathrm{P}\left(\cdot|\mathbb{S}^{\mathsf{R}}\right)\right) . \tag{12}$$

Intuitively the information gain of a split function is higher, the better it seperates members belonging to different classes (see figure 5).

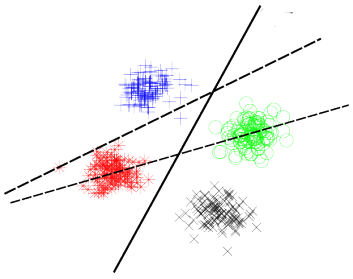

Fig. 5 The split function visualized as a solid line has the highest information gain (IG) among the three candidates.

2.1.3 Choice of Decision Functions

During the build up of the forest the randomized training approach allows us to vary the parametrization of the shape descriptor for each point of the shape. In fact, we can in principle let the forest automatically determine the optimal discriminative features of the chosen descriptor for the matching problem at hand. In this work we have chosen the Wave Kernel Signature (WKS) but as mentioned above, in principle any parametrizable feature descriptor (e.g. HKS) can be considered. From a practi-

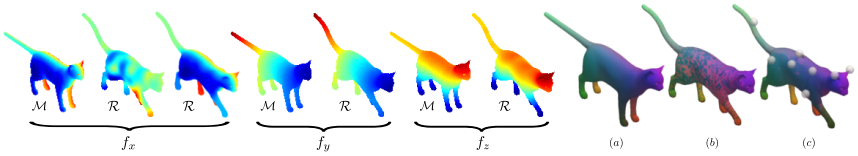

Fig. 6 The coordinate functions from a test shape M (standing cat) are transferred to a reference shape R (walking cat) via the functional map $T_{X_{M,R}}$ induced by the forest prediction. Most of the ambiguities arise in f_x, and are due to the global intrinsic symmetry of the cat. The first column shows the map f_x on the test cat, while the second and third columns are obtained by mapping f_x without and with regularization respectively. The remaining four columns show the mappings of f_y and f_z *without* regularization. The symmetric ambiguities disappear as a result of the regularization process (columns (a)-(c), matches encoded by color).

cal perspective, it can be shown [2] that the sum in (5) can be restricted to the first $\bar{k} < \infty$ components. We make explicit in (5) the dependency on $\bar{k}$ by writing:

$$p(m;e,\bar{k}) = \sum_{k=1}^{\bar{k}} f_e^2(\nu_k)\phi_k^2(m)\,. \tag{13}$$

We are now in the position of generating at each node of a tree during the training phase a pool of randomized split functions by sampling an energy level e_i, a number of eigenpairs $\bar{k}_i$ and a threshold τ_i. Accordingly, the split functions will take the form:

$$\psi_i(m) = \begin{cases} \mathsf{L} & \text{if } p(m;e_i,\bar{k}_i) > \tau_i \\ \mathsf{R} & \text{otherwise}\,. \end{cases} \tag{14}$$

2.2 Interpretation and regularization of the forests prediction

The simplest way to infer a correspondence from a forest prediction consists in assigning each point $m \in M$ to the most likely label according to its final distribution, i.e., the label maximizing $\mathrm{P}(\ell|m,\mathscr{F})$. If we are also given a reference shape R from the training set, the maximum a posteriori estimate of ℓ can be transformed into a point-to-point correspondence from M to R via the known bijection $T : R \to L$. Figures 6(a)(b) show an example of this approach. The resulting correspondence is exact for about 50% of the points, whereas it induces a large metric distortion on the rest of the shape. However, this is not a consequence of the particular criterion we adopted when applying the prediction. Indeed, the training process can not distinguish symmetric points and is oblivious to the underlying manifolds as it is only based on pointwise information: the correspondence estimates are taken *independently* for each point and thus the metric structure of the test shape is not taken into account during the regression. Nevertheless, as we shall see, the predicted distributions carry enough information that can be exploited to obtain a consistent matching.

2.2.1 Functional Maps

Multiplying X_M (as defined in (8)) from the left with the permutation matrix associated to the known bijection $T : L \to R$ between the label space L and a reference shape R gives raise to another left-stochastic matrix $X_{M,R}$. As pointed out in [16] this (fuzzy) correspondence $X_{M,R}$ can be interpreted as a linear map $T_{X_{M,R}} : L^2(M) \to L^2(R)$. In Figure 6 (first 7 columns) we use such a construction to map the coordinate functions $f_i : M \to \mathbb{R}$ (where $i \in \{x,y,z\}$) to scalar functions on R. Specifically, we plot f_i and their reconstructions $g_i = T_{X_{M,R}} f_i$. Note that the reference shape is axis-aligned, so that the x coordinates of its points grow from the right side (blue) to the left side of the model (red).

As in [16] from now on we consider $T_{X_{M,R}}$ in the truncated harmonic bases on the resprective shapes and by that dramatically reduce the size of the problem. Since the LB-eigenfunctions are chosen to form orthonormal bases, the norms considered in the following section are invariant under this basis-transform. For simplicity we will still denote the associated matrix by $X_{M,R}$.

2.2.2 Metric distortion using functional maps

The plots we show in Figure 6 tell us that most of the error in the correspondence arises from the (global) intrinsic symmetries of the shape. As mentioned previously, this is to be expected since the training process does not exploit any kind of structural information about the manifolds.

This suggests the possibility to regularize the prediction by introducing metric constraints on the correspondence. Specifically, we consider an objective of the form

$$E(\mathtt{X}) = c(\mathtt{X}_{M,R}, \mathtt{X}) + \rho(\mathtt{X}), \tag{15}$$

where $\mathtt{X}$ is a correspondence between shapes M and R. The first term (or *cost*) ensures closeness to the prediction given by the forest, while the second term is a regularizer giving preference to geometrically consistent solutions.

A functional map is assumed to be geometrically consistent if it approximately preserves distance maps. Suppose for the moment we are given a sparse collection of matches $O \subset M \times R$. Then for each $(p,q) \in O$ we can define the two distance maps $d_p : M \to \mathbb{R}$ and $d_q : R \to \mathbb{R}$ as

$$d_p(x) = d_M(p,x), \qquad d_q(y) = d_R(q,y). \tag{16}$$

With these definitions, we can express the regularity term $\rho(\mathtt{C})$

$$\rho(\mathtt{C}) = \sum_{(p,q)\in O} \omega_{pq} \| X_{M,R} d_p - d_q \|_2^2, \tag{17}$$

with weights $\omega_{pq} \in [0,1]$.

In order for the regularization to work as expected, the provided collection of matches should constrain well the solution, in the sense that it should help to disambiguate the intrinsic symmetries of the shape. For example, matches along the tail of the cat would bring little to no information on what solution to prefer. In practice, we can seek for a few matches that cover the whole shape and be as accurate as possible. To this end, we generate evenly distributed samples $V_{\text{fps}} \subset M$ on the test shape via farthest point sampling [13] by using the extrinsic Euclidean metric. Then, we construct a matching problem restricted to the set of *predicted* matches

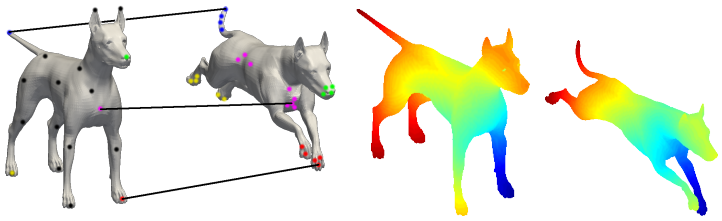

Fig. 7 In the regularization step first a coarse subsampling of the shape is constructed via euclidean farthest point sampling (dots on the left shape). In the small set of predicted matches O (cross product of dots on the two shapes) a sparse correspondence is obtained using an l^1 constrained optimazation technique. We expect a consistent correspondence to approximately preserve the distance maps d_p.

$$O = \{(m,r) \in V_{\text{fps}} \times R \,|\, (\mathtt{X}_{M,R})_{rm} > 0\}\,. \tag{18}$$

In practice this set is expected to be small, since the prediction given by the forest is very sparse and we select around 50 farthest samples per test shape (≈0.2% of the total number of points on the adopted datasets). This results in a small matching problem that we solve via game-theoretic matching [20], a ℓ_1-regularized technique that allows to obtain sparse, yet very accurate solutions in an efficient manner. Once a sparse set of matches is obtained, we solve (15) as the weighted least-squares problem

$$\min_{\mathtt{X}} \|\mathtt{X}_{M,R} - \mathtt{X}\|_F^2 + \sum_{(p,q)\in O} \omega_{pq} \|\mathtt{X} d_p - d_q\|_2^2\,, \tag{19}$$

where $\omega_{pq} \in [0,1]$ are weights (provided by the game-theoretic matcher) giving a measure of confidence for each match $(p,q) \in O$. Figure 6(c) shows the result of the regularization performed using 25 sparse matches (indicated by small spheres).

Notice that the distance between functional maps is yet not well understood. The authors of [6] suggest to replace the Frobenius norm in (19) with a regularized l^0 norm of the vector of singular values:

$$\|A\|_\varepsilon = \sum_i \frac{\sigma(A)_i^2}{\sigma(A)_i^2 + \varepsilon} \tag{20}$$

Assuming the shapes to be (nearly) isometric one can expect the Laplace Beltrami operators on the shapes to commute with the functional map, i.e. (in the harmonic bases):

$$X\Lambda_M = \Lambda_R X \tag{21}$$

where Λ_M and Λ_R are the diagonal matrices of the singular values. A measure of deviation from (21) can be used as an alternative regularity cost.

2.3 Experimental results

In all our experiments we used the WKS as pointwise descriptor for the training process. As in [16], we limited the size of the bases on the shapes to the first 100 eigenfunctions of the Laplace-Beltrami operator, computed using the cotangent scheme [15].

2.3.1 Comparison with dense methods

In this set of experiments we compare with the state of the art techniques in (dense) non-rigid shape matching, namely the functional maps pipeline [16], blended intrinsic maps (BIM) [10], and the coarse-to-fine combinatorial approach of [24]. We perform these comparisons on the TOSCA high-resolution dataset [5]. The dataset consists of 80 shapes belonging to different classes, with resolutions ranging in 4K-52K points. Shapes within the same class have the same connectivity and undergo nearly-isometric deformations. Ground-truth point mapping among shapes from the same class is available. In particular, given a predicted map $f : M \to N$ and the corresponding ground-truth $g : M \to N$, we define the *error* of f as

$$\varepsilon(f,g) = \sum_{m \in M} d_N(f(m), g(m)), \tag{22}$$

where d_N is the geodesic metric on N, normalized by $\sqrt{Area(N)}$ to allow inter-class comparisons. Similarly, we define the average (pointwise) geodesic error as $\frac{\varepsilon(f,g)}{|M|}$.

Although the methods considered in these experiments do not rely on any prior learning, the comparison is still meaningful as it gives an indication of the level of accuracy that our approach can attain in this class of problems. The experiments were designed on the same benchmark and following a procedure similar to the one reported in [10, 16]. Specifically, for each model M of a class (e.g., the class of dogs), we randomly picked other 6 models from the same class (not including M), and trained a random forest with them (thus, we only considered classes with at least 6 shapes). Then we predicted a dense correspondence for M according to the technique described in Section 2.2.

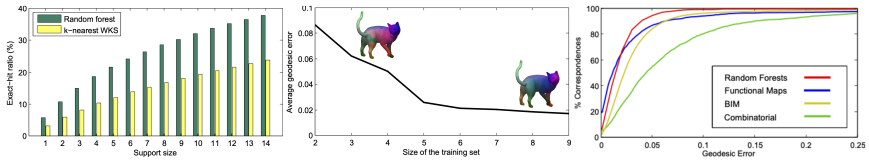

Fig. 8 Left: Fraction of exact matches predicted by a random forest vs. maximum support size of the probability distributions on a test shape. The forest was trained with 9 shapes. Middle: Sensitivity to number of shapes used in the training set. Note how the correspondence predicted using little training data (top-left model) is only partially regularized. Right: Comparison with the state-of-the-art methods on nearly-isometric shapes (TOSCA). Symmetric correspondences are considered correct solutions for all methods.

We show the results of this experiment in Fig. 8 (right). Each curve depicts the percentage of matches that attain an error below the threshold given on the x-axis. Our method (red line) detects 90% correct correspondences within a geodesic error of 0.05. Almost all correct matches are detected within an error of 0.1. This is compatible with and even improves the results given by the other methods on the same data. Note that our training process only makes use of pointwise information (namely, the WKS); in contrast, the functional maps pipeline (blue line) adopts several heuristics (WKS preservation constraints in addition to orthogonality of $\mathtt{C}$, region-wise features, etc.) in order to constrain the solution optimally [16]. Upon visual inspection, we observed that most of the errors in our method were due to the poor choice of points made in the regularization step. This is analogous to what is reported for the BIM method [10]. Typically, we observed that around 20 well-distributed points are sufficient to obtain accurate results.

2.4 Sensitivity to training parameters

We performed a sensitivity analysis of our method with respect to the parameters used in the training process, namely the size of the training set and the number of trees in the forest. In these experiments we employed the cat models from the TOSCA dataset (28K vertices) with the corresponding ground-truth.

In Fig. 8 (middle) we plot the average geodesic error obtained by a test shape (depicted along the curve) as we varied the number of shapes in the training set. The geodesic error of the correspondence stabilizes when at least 6 shapes are used for training. This means that only 6 samples per label are sufficient in order to determine an optimal parametrization of the nearly-isometric deformations occurring on the shape. This result contrasts the common setting in which random forests are trained with copious amounts of data [30, 8], making the approach rather practical when only limited training data is available.

Figure 8 (left) shows the change in accuracy as we increase the number of trees in the forest. Note that increasing the number of trees directly induces a larger support of the probability distributions over L. In other words, each point of the test shape

Fig. 9 Comparison between our method and an approach based on WKS affinity using shapes from the dataset of Vlasic et al. Columns one to four show the predicted and regularized solutions for both approaches. The last three columns show how the indicator function at one point gets functionally mapped to a second shape, by using the (non-regularized) X obtained from the forest, and by X_{WKS}.

receives more candidate matches if the forest is trained with more trees (see Eq. (7)). The hit ratio in the bar plot is defined as the fraction of *exact* predictions given by the forest over the entire test shape. We compare the results with the hit ratio obtained by looking for k-nearest neighbors in WKS descriptor space, with k equal to the maximum support size employed by the forest at each level. From this plot we see that the forest predictions are twice as accurate as WKS predictions for equal support sizes. In particular, random forest predicts the *exact* match for almost half (around 14K points) of the shape when trained with 15 trees.

Finally, in Fig. 9 we show a qualitative comparison between our method and an approach based on WKS. The rationale of this experiment is to show that the prediction given by the forest gives better results than what can be obtained without prior learning within the same pipeline (i.e., prediction followed by regularization). Specifically, for each point in one shape we construct a probability distribution on the other shape based on a measure of descriptor affinity in WKS space. We then estimated a functional map $\mathsf{C}_{\mathrm{WKS}}$ from the resulting set of constraints, and plotted a final correspondence before and after regularization.

2.5 Learning non-isometric deformations

In this section we consider a scenario in which the shapes to be matched may undergo more general (i.e., far from isometric) deformations. Examples of such deformations include local and global changes in scale, topological changes, resampling, partiality, and so forth. Until now, few methods have attempted to tackle this class of problems. Most dense approaches [10, 16, 24, 17] are well-defined in the quasi-isometric and conformal cases only; instances of inter-class matching were considered in [10], but the success of the method depends on the specific choice of (usually hand-picked) feature points used in the subsequent optimization. Sparse methods considering the general setting from a metric perspective [20, 4, 22] attempt to formalize the problem by using the language of quadratic optimization, leading to difficult and highly non-convex formulations. An exception to the gen-

eral trend was given in [31], where the matching is formulated as a linear program in the product space of manifolds. The method allows to obtain dense correspondences for more general deformations, but it assumes consistent topologies and is computationally expensive ($\sim$2 hours to match around 10K vertices). Another recent approach [11] attempts to model deviation from isometry in the framework of functional maps, by seeking compatible harmonic bases among two shapes. However, it relies on a (sparse) set of matches being given as input and it shares with [31] the high computational cost.

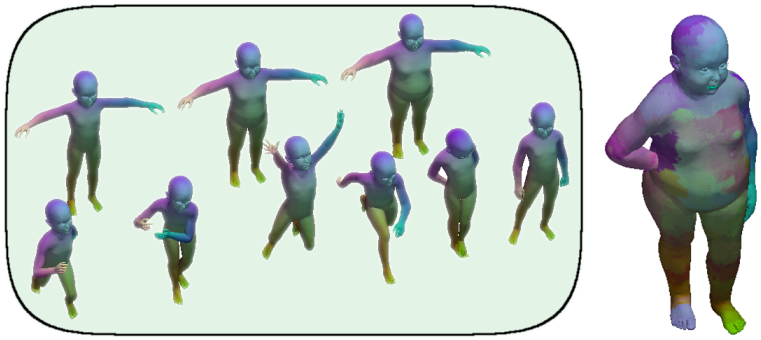

Fig. 10 Example of dense shape matching using random forests under non-isometric deformations. Shapes in the shaded area are a subset of the training set. The forest is trained with wave kernel descriptors and consists of 80K training classes with 19 samples per class. Matches are encoded by color.

As described in Section 2, the forest does not contain any explicit knowledge of the type of deformations it is asked to parametrize. This means that, in principle, one could feed the learning process with training data coming from any collection of shapes, with virtually no restrictions on the transformations that the shapes are allowed to undergo. Clearly, an appropriate choice of the pointwise descriptor should be made in order for the forest to provide a concise and discriminative model. To test this scenario, we constructed a synthetic dataset consisting of 8 high-resolution (80K vertices) models of a kid under different poses (quasi-isometries), and 11 additional models of increasingly corpulent variants of the same kid (local scale deformations) with a fixed pose (see Fig. 10). The shapes have equal number of points and point-to-point ground-truth is available. We test the trained random forest with a plump kid having a previously unseen pose.

Note that the result is reasonably accurate if we keep in mind the noisy setting: the forest was trained with WKS descriptors, which are originally designed for quasi-isometric deformations, and thus not expected to work well in the more general setting [12]. Despite being just a qualitative evaluation, this experiment demonstrates the generality of our approach. The matching process we described can still be employed in general non-rigid scenarios if provided with limited, yet sufficiently discriminative training data.

3 Conclusions

In this article, we showed how the random forest learning paradigm can be employed for problems of dense correspondence among deformable 3D shapes. To our knowledge, this is among the first attempts at introducing a statistical learning view on this family of problems. The effectiveness of our approach is demonstrated on a standard benchmark, where we obtain comparable results with respect to the state of the art, and very low prediction times for shapes with tens of thousands of vertices. The approach is flexible in that it provides a means to model deformations which are far from isometric, and it consistently obtains high predictive performance on all tested scenarios.

References

1. D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.
2. M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops*, 2011.
3. L. Breiman. Random forests. In *Machine Learning*, volume 45, 2001.
4. A.M. Bronstein, M.M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *PNAS*, 2006.
5. A.M. Bronstein, M.M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer Publishing Company, Incorporated, 1 edition, 2008.
6. É. Corman, M. Ovsjanikov, and A. Chambolle. Supervised descriptor learning for non-rigid shape matching.
7. A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. In *Found. and Trends in Comput. Graph. Vis.*, 2012.
8. G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, June 2011.
9. J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *PAMI*, 33(11), 2011.
10. V.G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *SIGGRAPH 2011*, 2011.
11. A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. *Comput. Graph. Forum*, 32(2pt4), 2013.
12. R. Litman and A.M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *TPAMI*, 2013.
13. F. Mémoli. Gromov-Wasserstein distances and the metric approach to object matching. *Found. Comput. Math.*, 11, 2011.
14. F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Found. of Comput. Math.*, 5(3), 2005.
15. M. Meyer, M. Desbrun, P. Schröder, and A.H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Proc. VisMath*, 2002.
16. M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4):30:1–30:11, July 2012.
17. M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One point isometric matching with the heat kernel. *Comput. Graph. Forum*, 29(5):1555–1564, 2010.

18. M. Ovsjanikov, J. Sun, and L. Guibas. Global intrinsic symmetries of shapes. In *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library, 2008.
19. D. Pickup, X. Sun, P. Rosin, R. Martin, Z. Cheng, Z. Lian, M. Aono, A. Ben Hamza, A. Bronstein, M. Bronstein, et al. Shrec14 track: Shape retrieval of non-rigid 3d human models. *3DOR*, 4(7):8, 2014.
20. E. Rodolà, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *CVPR*, 2012.
21. E. Rodolá, S. Rota Bulo, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *CVPR*, pages 4177–4184, June 2014.
22. E. Rodolà, A. Torsello, T. Harada, T. Kuniyoshi, and D. Cremers. Elastic net constraints for shape matching. In *ICCV*, 2013.
23. R.M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *SGP*. Eurographics Association, 2007.
24. Y. Sahillioğlu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Comput. Graph. Forum*, 30(5), 2011.
25. B. Schölkopf, F. Steinke, and V. Blanz. Object correspondence as a machine learning problem. In *ICML*, pages 776–783. ACM, 2005.
26. J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.
27. A. Shtern and R. Kimmel. Matching lbo eigenspace of non-rigid shapes via high order statistics. *arXiv preprint arXiv:1310.4459*, 2013.
28. F. Steinke, V. Blanz, and B. Schölkopf. Learning dense 3d correspondence. In *Advances in Neural Information Processing Systems*, pages 1313–1320, 2006.
29. J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *SGP*. Eurographics Association, 2009.
30. J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *CVPR*, 2012.
31. T. Windheuser, U. Schlickewei, F.R. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *ICCV*, 2011.
32. T. Windheuser, M. Vestner, E. Rodola, R. Triebel, and D. Cremers. Optimal intrinsic descriptors for non-rigid shape analysis. In *BMVC*. BMVA Press, 2014.

Chapter 7

Product Manifold Filter: Non-rigid shape correspondence via kernel density estimation in the product space

This paper approaches the correspondence problem as an optimization problem over the graph of the correspondence as introduced in Section **??**. As mentioned there, the regularity of the correspondence it tightly coupled to the regularity of its graph. We propose an iterative method that increases the regularity of the graph and thus of the correspondence itself in every iteration. In each iteration the current (graph of the) correspondence is first blurred and then sharpened again. We show how this approach is related to a minimal surface problem in the product space. Given very different types of initial correspondeces such as very sparse correspondences consisting of only two point-to-point corerspondences, noisy dense correspondences or even fuzzy correspondences (eg. arising from functional maps), we can obtain bijective dense correspondences of high regularity that are semantically meaningful. Since the bijectivity comes with a price (one needs to solve linear assignment problems in each iteration) we also propose a multiscale variant of the algorithm that allows matching of triangular meshes with a high number of vertices. While not being the focus of this paper we also show how the method can be used to match shapes of different dimensions, *e.g.* a countour (1D manifold) to a 3D shape (2D manifold).

Publication:	Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex Bronstein and Daniel Cremers. "Product Manifold Filter: Non-rigid shape correspondence via kernel density estimation in the product space". In: *International Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017. DOI: 10.1109/cvpr.2017.707

Individual contribution		
	Problem definition	significantly contributed
	Literature survey	significantly contributed
	Implementation	significantly contributed
	Experimental evaluation	contributed
	Preparation of the manuscript	significantly contributed

Product Manifold Filter: Non-Rigid Shape Correspondence via Kernel Density Estimation in the Product Space

Matthias Vestner
Technical University Munich

Roee Litman
Tel-Aviv University

Emanuele Rodolà
USI Lugano

Alex Bronstein
Technion, Israel Institute of Technology
Perceptual Computing Group, Intel, Israel

Daniel Cremers
Technical University Munich

Abstract

Many algorithms for the computation of correspondences between deformable shapes rely on some variant of nearest neighbor matching in a descriptor space. Such are, for example, various point-wise correspondence recovery algorithms used as a post-processing stage in the functional correspondence framework. Such frequently used techniques implicitly make restrictive assumptions (e.g., near-isometry) on the considered shapes and in practice suffer from lack of accuracy and result in poor surjectivity. We propose an alternative recovery technique capable of guaranteeing a bijective correspondence and producing significantly higher accuracy and smoothness. Unlike other methods our approach does not depend on the assumption that the analyzed shapes are isometric. We derive the proposed method from the statistical framework of kernel density estimation and demonstrate its performance on several challenging deformable 3D shape matching datasets.

1. Introduction

Estimating the correspondence between 3D shapes is among the fundamental problems in computer vision, geometry processing and graphics with a wide spectrum of applications ranging from 3D scene understanding to texture mapping and animation. Of particular interest is the case in which the objects are allowed to deform non-rigidly. In this setting, research has mainly focused on minimizing a measure of distortion between the input shapes, reaching in recent years very high levels of accuracy [43]. However, point-wise accuracy often comes under restricting requirements (isometry assumption), or at the price of a lack of useful properties on the computed map, namely *bijectivity* (each point on either shape should have exactly one corresponding point on the other) and *smoothness* (nearby points should match to nearby points).

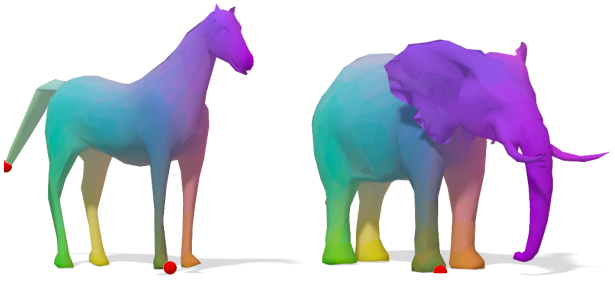

Figure 1. Our method can be used to recover a dense, smooth, bijective correspondence between highly non-isometric shapes from minimal input information. In this example, we initialize our algorithm with just two hand-picked matches (red spheres on tail and front leg). Correspondence quality is visualized by transferring colors from horse to elephant via the recovered map.

In this paper, we introduce a novel method to recover smooth bijective maps between deformable shapes. Contrarily to previous approaches, we do not rely on the assumption that the two shapes are isometric. We phrase our matching problem by using the language of statistical inference, whereas the input to our algorithm is either 1) a sparse collection of point-wise matches (as few as two) which are used as landmark constraints to recover the complete map, or 2) a dense, noisy, possibly non-surjective and non-smooth map which is converted to a better map with higher accuracy and the aforementioned properties.

1.1. Related works

A traditional approach to correspondence problems is finding a *point-wise* matching between (a subset of) the points on two or more shapes. *Minimum-distortion methods* establish the matching by minimizing some structure distortion, which can include similarity of local features [32, 14, 8, 45], geodesic [29, 13, 15] or diffusion distances [17], or a combination thereof [41].

Typically, the computational complexity of such meth-

ods is high, and there have been several attempts to alleviate the computational complexity using hierarchical [37] or subsampling [40] methods. Several approaches formulate the correspondence problem as quadratic assignment and employ different relaxations thereof [42, 24, 34, 2, 15, 19]. Algorithms in this category typically produce guaranteed bijective correspondences between a sparse set of points, or a dense correspondence suffering from poor surjectivity.

Embedding methods try to exploit some assumption on the correspondence (e.g. approximate isometry) in order to parametrize the correspondence problem with a few degrees of freedom. Elad and Kimmel [18] used multi-dimensional scaling to embed the geodesic metric of the matched shapes into a low-dimensional Euclidean space, where alignment of the resulting "canonical forms" is then performed by simple rigid matching (ICP) [16, 10]. The works of [27, 38] used the eigenfunctions of the Laplace-Beltrami operator as embedding coordinates and performed matching in the eigenspace. Lipman *et al.* [25, 20, 21] used conformal embeddings into disks and spheres to parametrize correspondences between homeomorphic surfaces as Möbius transformations. Despite their overall good performance, the majority of the matching procedures performed in the embedding space often produces noisy correspondences at fine scales, and suffers from poor surjectivity. More recently, in [6, 5] the authors obtain a bijective correspondence by first computing compatible embeddings of the two shapes, and then aligning the embeddings through the use of sparse input correspondences. As opposed to point-wise correspondence methods, *soft correspondence* approaches assign a point on one shape to more than one point on the other. Several methods formulated soft correspondence as a mass-transportation problem [28, 39]. Ovsjanikov *et al.* [31] introduced the *functional correspondence* framework, modeling the correspondence as a linear operator between spaces of functions on two shapes, which has an efficient representation in the Laplacian eigenbases. This approach was extended in several follow-up works [33, 22, 3, 35] . A point-wise map is typically recovered from a low-rank approximation of the functional correspondence by a matching procedure in the representation basis, which also suffers from poor surjectivity. A third class of matching methods formulates the correspondence problem as an optimization problem in the *product space* of the considered shapes. Windheuser *et al.* [44] seek for a two-dimensional minimal surface in the four-dimensional product space of the two input surfaces; this was later extended to a 2D-to-3D setting by Lähner *et al*. [23]. Making use of the graph structure of the considered shapes, the discretization leads to an integer linear program on the product mesh where desirable properties of the matching such as smoothness and surjectivity become linear constraints. However, the computational complexity is prohibitive even for a modestly-sized problem.

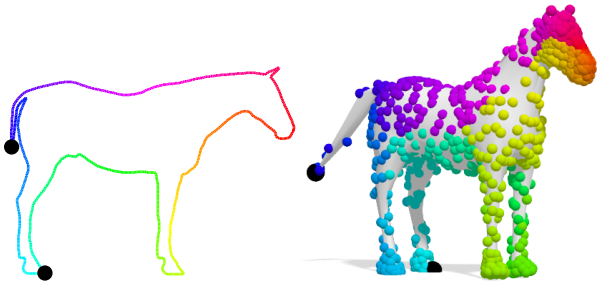

Figure 2. The Product Manifold Filter (PMF) can be applied to a variety of problems that are aiming for bijective, smooth mappings between metric spaces. Here we map a 2D shape (contour) to a 3D shape. We initialize the PMF with two semantically meaningful matches (black spheres) and obtain a dense semantically meaningful bijection.

1.2. Main contributions

Many of the works mentioned above provide a matching that is neither bijective nor smooth. In some cases the matching is only available as a sparse set of points in the product space of the two shapes. We treat these matchings as corrupted versions of the latent correspondence and propose the *Product Manifold Filter (PMF)*, a framework that increases the quality of the input mapping. We show that the considered filter leads to a linear assignment problem (LAP) guaranteeing bijective correspondence between the shapes. Despite the common wisdom, we demonstrate that the problem is efficiently solvable for relatively densely sampled shapes by means of the well-established auction algorithm [9] and a simple multi-scale approach. Unlike many of the previously mentioned techniques that assume the shapes to be (nearly) isometric, we allow them to undergo more general deformations (Figure 1) or even have different dimensionality (Figure 2).

Finally, we present a significant amount of empirical evidence that the proposed smoothing procedure consistently improves the quality of the input correspondence coming from different algorithms, including point-wise recovery methods from functional map pipelines. We also show the performance of PMF as an interpolator of sparse input correspondences.

2. A probabilistic framework

We consider a pair of three-dimensional shapes that are represented by their boundaries $\mathcal{X}$ and $\mathcal{Y}$, two-dimensional manifolds embedded in $\mathbb{R}^3$ and thus equipped with intrinsic metrics $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$. Our goal is to find a semantically meaningful correspondence between $\mathcal{X}$ and $\mathcal{Y}$. A correspondence is a *diffeomorphism* $\pi : \mathcal{X} \to \mathcal{Y}$, i.e., a smooth mapping with a smooth inverse. We do not make any other assumptions such as isometry. The correspondence π can be represented as a two-dimensional manifold Π in the four-dimensional product space $\mathcal{X} \times \mathcal{Y}$: a pair (x, y) belongs to Π iff $\pi(x) = y$. We henceforth assume that the true correspon-

dence π between $\mathcal{X}$ and $\mathcal{Y}$ and the manifold Π representing it are latent.

Let $\{(x_k, y_k)\}_{k\in\mathcal{K}} \subset \Pi$ be a possibly sparse sample of the said manifold. For example, these can be pairs of corresponding points on $\mathcal{X}$ and $\mathcal{Y}$ computed using a feature detector followed by descriptor matching. In practice, we only have access to a noisy realization of these points, $\{(\xi_k, \eta_k)\}_{k\in\mathcal{K}}$, which we assume to admit a separable i.i.d. Gaussian density, $f(\xi_k, \eta_k) \propto K(d_{\mathcal{X}}(x_k, \xi_k))K(d_{\mathcal{Y}}(y_k, \eta_k))$, where

$$K(d) = \exp\left(-\frac{d^2}{2\sigma^2}\right)$$

is an unnormalized Gaussian kernel with the parameter σ^2. Note that the density on the manifolds is expressed in terms of the intrinsic metrics $d_{\mathcal{X}}$ and $d_{\mathcal{Y}}$.

Given the set of noisy corresponding points $\{(\xi_k, \eta_k)\}_{k\in\mathcal{K}}$ as the input, our goal is to produce a faithful estimate of the correspondence π. We propose to estimate the latent manifold Π via kernel density estimation in the product space $\mathcal{X} \times \mathcal{Y}$. To that end, we estimate the density function using the Parzen sum

$$f(x, y) \quad \propto \quad \sum_{k\in\mathcal{K}} K(d_{\mathcal{X}}(x, \xi_k))\, K(d_{\mathcal{Y}}(y, \eta_k)). \quad (1)$$

For every point $x \in \mathcal{X}$, an estimate of $\pi(x)$ is given by a point y maximizing $f(x, y)$,

$$\hat{\pi}(x) = \arg\max_y f(x, y). \quad (2)$$

One can further impose bijectivity of $\hat{\pi} : \mathcal{X} \to \mathcal{Y}$ as a constraint, obtaining the following estimator of the entire map

$$\hat{\pi} = \arg \max_{\hat{\pi}:\mathcal{X}\overset{1:1}{\to}\mathcal{Y}} \int_{\mathcal{X}} f(x, \hat{\pi}(x))dx. \quad (3)$$

The process can be iterated as shown in the one-dimensional illustration in Figure 3.

Procedures (2) or (3) have an area reduction effect on the manifold Π producing a more regular version thereof and thus a more regular correspondence π. We interpret (3) as a filter of correspondences and will henceforth refer to it a *product manifold filter* (PMF). While we defer the rigorous proof of the area reduction property to the extended version of the paper, in what follows, we illustrate it by a simple one-dimensional example.

One dimensional illustration. Let us consider a configuration of three points $\{x_-, x, x_+\}$ and the corresponding noisy points $\{y_-, y, y_+\}$ on a pair of one-dimensional manifolds $\mathcal{X}$ and $\mathcal{Y}$ like those depicted in Figure 3. We assume that the points are directly given in arclength parametrization, such that $d_{\mathcal{X}}(x, x_\pm) = |x - x_\pm| = b$, $d_{\mathcal{Y}}(y, y_-) = |y - y_-| = a$, and $d_{\mathcal{Y}}(y, y_+) = |y - y_+| = a + \delta$. For convenience, we henceforth denote $x = y = 0$, $x_\pm = \pm b$, $y_- = -a$ and $y_+ = a + \delta$. In this setting, the one-dimensional manifold Π_0 representing the input correspondence in the product space comprises two segments connecting $(-b, -a)$, $(0, 0)$, and $(b, a + \delta)$, and its length is given by $L(\Pi_0) = \sqrt{b^2 + a^2} + \sqrt{b^2 + (a + \delta)^2}$.

PMF maximizes the density function

$$\begin{aligned} h(\hat{y}) &= f(0, \hat{y}) = K(0)K(\hat{y}) + \\ &\quad K(b)K(\hat{y} + a) + K(b)K(\hat{y} - a - \delta) \\ &= K(\hat{y}) + K(b)(K(\hat{y} + a) + K(\hat{y} - a - \delta)) \end{aligned} \quad (4)$$

over the values $\hat{y}$ for the point y. First, we observe that since $K(b) > 0$, the global maximum of $h(\hat{y})$ has to be around $\hat{y} = 0$. For $\hat{y} = 0$ and $\delta = 0$, one has

$$\frac{dh}{d\hat{y}} = K'(0) + K(b)(K'(a) + K'(-a))$$

and

$$\frac{d^2h}{d\hat{y}^2} = K''(0) + K(b)(K''(a) + K''(-a))$$

Since $K'(0) = 0$ and $K'(-a) = -K'(a)$, the first derivative vanishes, while the fact that $K''(0) < 0$ and $K''(-a) = K''(a)$ implies that $\hat{y} = 0$ is the maximum of h.

Next, we perform perturbation analysis of the above maximizer by invoking the first-order Taylor expansion of h around $(\delta, \hat{y}) = (0, 0)$:

$$\frac{\partial h}{\partial \hat{y}} \approx \left.\frac{\partial h}{\partial \hat{y}}\right|_{\hat{y}=0,\delta=0} + \hat{y}\left.\frac{\partial^2 h}{\partial \hat{y}^2}\right|_{\hat{y}=0,\delta=0} + \delta\left.\frac{\partial^2 h}{\partial \hat{y}\partial\delta}\right|_{\hat{y}=0,\delta=0}.$$

Demanding equality to zero yields the maximizer of the perturbed problem

$$\hat{y} \quad \approx \quad \frac{K(b)K''(a)\delta}{2K(b)K''(a) + K''(0)} = \frac{\delta}{2 + \frac{K''(0)}{K(b)K''(a)}} = c\delta.$$

For $a < \frac{\sigma}{\sqrt{2}}$ the ratio in the denominator is positive and consequently $c \in (0, \frac{1}{2})$.

The length of the estimated manifold $\hat{\Pi}$ can be obtained using a series of first-order Taylor approximations,

$$\begin{aligned} L(\hat{\Pi}) &= \sqrt{b^2 + (a + c\delta)^2} + \sqrt{b^2 + (a + \delta - c\delta)^2} \\ &\approx L(\Pi_0) + \frac{ac\delta}{\sqrt{b^2 + a^2}} - \frac{(a + \delta)c\delta}{\sqrt{b^2 + (a + \delta)^2}} \\ &\approx L(\Pi_0) - \frac{cb^2}{(b^2 + a^2)^{3/2}}\,\delta^2 < L(\Pi_0), \end{aligned} \quad (5)$$

which manifests the length reducing effect of the PMF.

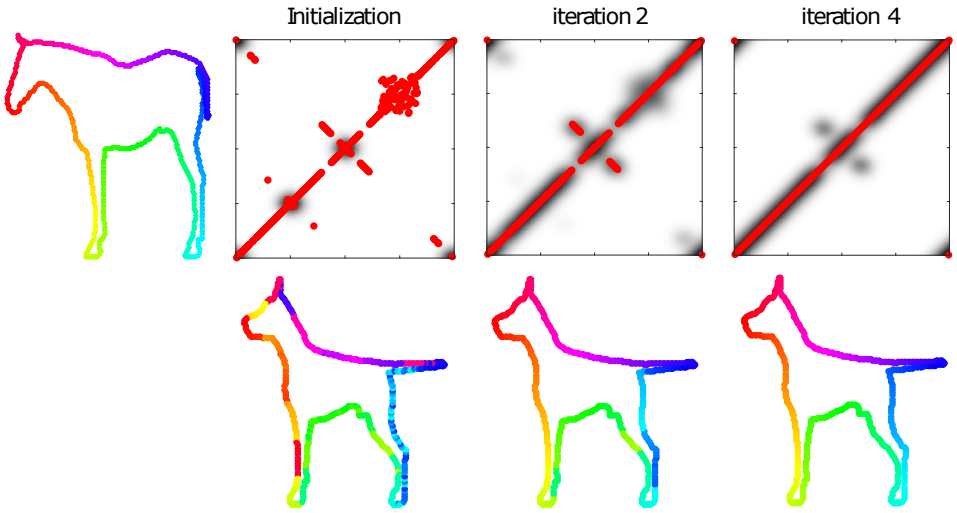

Figure 3. Conceptual illustration of our method on one-dimensional manifolds. Shown are iterations of PMF ($|\mathcal{K}| = 3$ sparse matches as initialization). Top: Kernel density estimation $f(x, y)$ as defined in (1) in the product space of the two shapes $\mathcal{X}$ and $\mathcal{Y}$. Dark areas correspond to higher density. According to (3), consistently maximizing $f(x, \cdot)$ gives a bijective and smoothed matching (red curve in product space) which is used to derive the density estimate in the next iteration. Bottom: matching visualized via color transfer. Shapes are parametrized counter-clockwise with the origin of the product space corresponding to the noses of horse and dog. Note the circular boundary conditions of the product space.

2.1. Discretization

In what follows, we consider a discretization of problem (3). We assume the shape $\mathcal{X}$ to be discretized at n points $\{x_i\}_{i=1}^n$ and the pairwise geodesic distances are stored in the matrix $\mathbf{D}_\mathcal{X} \in \mathbb{R}^{n\times n}$. Similarly, the shape $\mathcal{Y}$ is discretized as $\{y_i\}_{i=1}^n$ and its pairwise distance matrix is denoted by $\mathbf{D}_\mathcal{Y} \in \mathbb{R}^{n\times n}$. Given a (possibly sparse) collection of input correspondences $\{(\xi_k, \eta_k)\}_{k=1}^m$ the unnormalized kernel density estimation can be written as an $n \times n$ matrix

$$\mathbf{F} \quad = \quad \mathbf{K}_\mathcal{X}\mathbf{K}_\mathcal{Y}^T \tag{6}$$

with the matrices $\mathbf{K}_\mathcal{X} \in \mathbb{R}^{n\times m}$ and $\mathbf{K}_\mathcal{Y} \in \mathbb{R}^{n\times m}$ given by

$$(\mathbf{K}_\mathcal{X})_{ik} \quad = \quad K(d_\mathcal{X}(x_i, \xi_k)) \tag{7}$$

$$(\mathbf{K}_\mathcal{Y})_{ik} \quad = \quad K(d_\mathcal{Y}(y_i, \eta_k)). \tag{8}$$

The objective in (3) thus becomes

$$\begin{aligned}\int_\mathcal{X} f(x, \pi(x))dx \quad &= \quad \int_{\mathcal{X}\times\mathcal{Y}} f(x, y)\delta_{\pi(x)}(y)dydx \\ &\approx \quad \sum_{i,j=1}^n \mathbf{F}_{ij}\mathbf{P}_{\mathbf{ji}} = \langle \mathbf{P}, \mathbf{F}\rangle \end{aligned} \tag{9}$$

with $\mathbf{P} \in \{0, 1\}^{n\times n}$ being a permutation matrix representing a bijection between $\{x_i\}_{i=1}^n$ and $\{y_i\}_{i=1}^n$. At some points it will be convenient to use the vector representation $p \in \{1, \ldots, n\}^n$ of $\mathbf{P}$. Estimating the bijective correspondence as in (3) thus turns out to be a linear assignment problem (LAP) of the form

$$\hat{\mathbf{P}} = \arg\max_\mathbf{P} \langle \mathbf{P}, \mathbf{F}\rangle \tag{10}$$

where the optimization is performed over the space of all $n \times n$ permutation matrices.

2.2. Multiscale

While linear assignment problems like (10) can be solved in polynomial time, the memory consumption is quadratic in the vertex set size n. To alleviate this burden, we propose a multi-scale technique based on the assumption of local regularity of the manifold Π.

Given two shapes discretized at n points each, we perform farthest point sampling to obtain a hierarchy of p multiscale representations consisting of $n_1 < n_2 < \ldots < n_p = n$ points. Each of the samplings comes with a sequence of sampling radii, $r_1^\mathcal{X} > r_2^\mathcal{X} > \ldots > r_p^\mathcal{X}$ and $r_1^\mathcal{Y} > r_2^\mathcal{Y} > \ldots > r_p^\mathcal{Y}$, respectively.

For sufficiently large shapes, the $n \times n$ pairwise distance matrices $\mathbf{D}_\mathcal{X}$ and $\mathbf{D}_\mathcal{Y}$ can be no more stored entirely in memory. We follow [4, 26] and store only the projection of the latter matrices on the first r eigenfunctions of the Laplacian resulting in an $n \times r$ matrix. The original distances are reconstructed on-demand, with negligible error as shown in [1, 26].

We recursively apply a variant of the PMF to the sparse set of input matches obtained by the coarser scale:

$$\mathbf{P}_{i+1} \quad = \quad \arg\max_{\mathbf{P}\in\{0,1\}^{n_{i+1}^2}} \langle \mathbf{P}, \mathbf{F}_i \rangle \tag{11}$$

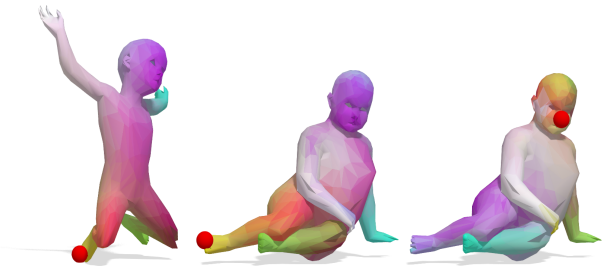

Figure 4. Our method finds smooth bijective maps between non-isometric shapes even when one single match is given as input (marked as small red spheres). Note that the map remains smooth even if the initial match is wrong (rightmost column).

where

$$\mathbf{F}_i(s,t) = \mathbf{W}(s,t) \sum_{k=1}^{n_i} \mathbf{K}_{\mathcal{X}}(s,k)\mathbf{K}_{\mathcal{Y}}(t,p_i(k))\,. \qquad (12)$$

The weighting matrix $\mathbf{W} \in \{0,1\}^{(n_{i+1}^2 \times n_i^2)}$ assures that the image of a point x_s being in the vicinity of x_k^i is constrained to be mapped to a point in the vicinity of $p(k)$ and vice versa (i.e., the matching and its inverse are supposed to be smooth):

$$\mathbf{W}(s,t) = \qquad (13)$$
$$\begin{cases} 0 & \text{if } \exists k : \mathbf{D}_{\mathcal{X}}(s,k) < r_i^{\mathcal{X}} \text{ and } \mathbf{D}_{\mathcal{Y}}(t,p_i(k)) > 2r_i^{\mathcal{Y}} \\ 0 & \text{if } \exists k : \mathbf{D}_{\mathcal{Y}}(t,p_i(k)) < r_i^{\mathcal{Y}} \text{ and } \mathbf{D}_{\mathcal{X}}(s,k) > 2r_i^{\mathcal{X}} \\ 1 & \text{otherwise} \end{cases}$$

This construction leads to a sparse payoff matrix corresponding to a smaller space of feasible permutations, so that the corresponding LAP can be solved efficiently. Note the factor 2 in (13). Since we cannot guarantee the Voronoi cells on the two shapes to have the same number of points and we want to be able to remove errors from the coarser scale, we permit moving a point to an adjacent Voronoi cell.

3. Experiments

While our method can be applied to a variety of problems aiming at bijective and smooth mappings between metric spaces (see Figure 2 for an extreme case), here we focus on the recovery of a correspondence between non-rigid and possibly non-isometric 3D shapes. We show the performance of our method in two very different scenarios, namely refinement of noisy dense correspondence, and completion of sparse correspondence. We additionally demonstrate the performance of our multi-scale technique by recovering bijective correspondences between high resolution shapes.

3.1. Recovery from sparse correspondences

In our first set of experiments we consider a scenario in which the input shapes come with a (possibly very sparse) collection of initial matches. These, in turn, can be obtained by a sparse non-rigid matching technique such as [34] or be hand-picked, depending on the application. In these experiments we compare PMF with the Tutte embedding approach recently introduced in [5]. Similarly to PMF, this approach produces guaranteed bijective and smooth maps starting from a sparse set of point-wise matches; to our knowledge, this method represents the state-of-the-art for this class of problems.

The results of this comparison are shown in Figure 5. The input matches were obtained by mapping farthest point samples on a reference shape via the ground-truth correspondence to the target shape, and are visualized by transferring a texture from reference to target via the recovered *dense* map. As we can read from the plots, our approach yields maps of better quality when fewer than ten matches are provided as the input, and maps of comparable quality when more matches are available. It is important to note that while our method still produces meaningful solutions when just one or two matches are given as the input (see Figures 1, 4), the approach of [5] has the theoretical minimum of five matches; furthermore, the latter approach gives different solutions depending on the specific ordering of the inputs, while our method is invariant to their permutations. Finally, as we demonstrate in the next section, a key ability of our method is being able to recover correct maps from noisy inputs, while the Tutte approach requires exact input.

3.2. Recovery from noisy input

In this set of experiments we assume to be given a low-rank approximation of the latent correspondence $\mathbf{P}$ in terms of a functional map

$$\mathbf{C} \quad = \quad \mathbf{\Psi}^T\mathbf{P}\mathbf{\Phi} \in \mathbb{R}^{r\times r}, \qquad (14)$$

where $\mathbf{\Phi}, \mathbf{\Psi} \in \mathbb{R}^{n\times r}$ are truncated orthonormal bases on $\mathcal{X}$ and $\mathcal{Y}$. We refer the reader to the original paper [31] for details and allow ourselves to condense its ideas to the above equation.

While a plurality of methods for finding $\mathbf{C}$ have been proposed in the last years, there currently exist only three approaches to recover a point-wise correspondence matrix $\mathbf{P}$ from it. In [31] the authors proposed to recover a pointwise correspondence between $\mathcal{X}$ and $\mathcal{Y}$ by solving the nearest-neighbor problem (NN)

$$\min_{\mathbf{P}\in\{0,1\}^{n\times n}} \|\mathbf{C}\mathbf{\Phi}^{\mathrm{T}} - \mathbf{\Psi}^{\mathrm{T}}\mathbf{P}\|_{\mathrm{F}}^2 \quad \text{s.t. } \mathbf{P}^{\mathrm{T}}\mathbf{1} = \mathbf{1}\,. \qquad (15)$$

alternated with an orthogonality-enforcing refinement of $\mathbf{C}$ (ICP). A variant is its bijective version (Bij. NN)

$$\min_{\mathbf{P}\in\{0,1\}^{n\times n}} \|\mathbf{C}\mathbf{\Phi}^{\mathrm{T}} - \mathbf{\Psi}^{\mathrm{T}}\mathbf{P}\|_{\mathrm{F}}^2 \quad \text{s.t. } \mathbf{P}^{\mathrm{T}}\mathbf{1} = \mathbf{1}\,,\ \mathbf{P}\mathbf{1} = \mathbf{1}\,. \qquad (16)$$

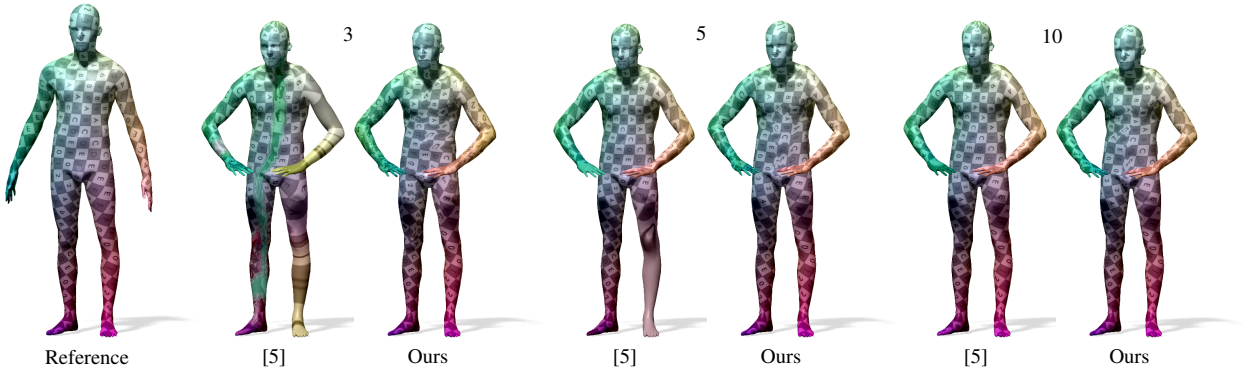

Figure 5. Comparison between our method and the method of [5] at increasing number of input matches (reported on top). Both methods produce smooth, guaranteed bijective solutions; our method requires little computational effort (a few minutes as opposed to $\sim$1 hour for [5]), and yields in comparison more accurate solutions when fed with a very sparse input.

The orthogonal refinement of (15) assumes the underlying map to be area-preserving [31], and is therefore bound to fail in case the two shapes are non-isometric. Rodolà *et al.* [36] proposed to consider the non-rigid counterpart for a given $\mathbf{C}$:

$$\min_{\mathbf{P}\in[0,1]^{n\times n}} D_{\mathrm{KL}}(\mathbf{C}\mathbf{\Phi}^{\mathrm{T}}, \mathbf{\Psi}^{\mathrm{T}}\mathbf{P}) + \lambda\|\mathbf{\Omega}(\mathbf{C}\mathbf{\Phi}^{\mathrm{T}} - \mathbf{\Psi}^{\mathrm{T}}\mathbf{P})\|^2 \quad \text{s.t. } \mathbf{P}^{\mathrm{T}}\mathbf{1} = \mathbf{1}\,. \tag{17}$$

Here D_{KL} denotes the Kullback-Leibler divergence between probability distributions, $\mathbf{\Omega}$ is a low-pass operator promoting smooth velocity vectors, and $\lambda > 0$ controls the regularity of the assignment. The problem is then solved via expectation-maximization by the coherent point drift algorithm (CPD) [30].

We construct the low-rank functional map using the known ground-truth correspondences between the shapes. Since this is supposed to be the ideal input for all the competing methods, we abandon the refinement step in (15). Correspondences returned by the other methods are treated as noisy realizations of the latent bijection and are recovered via PMF with σ^2 set to 2% of the target shape area.

We show quantitative comparisons on 71 pairs from the SCAPE dataset [7] (near isometric, 1K vertices) and 100 pairs from the FAUST dataset [11] (including inter-class pairs, 7K vertices). In Figures 11 and 9 we compare the correspondence accuracy, while in Figure 12 we visualize how lack of smoothness, bijectivity and accuracy affect texture transfer.

The accuracy of all input matchings is increased by applying the product manifold filter. To our knowledge, the matchings obtained by the PMF are the most accurate ones that can be recovered from this type of low-rank approximation. While linear assignment problems are known to be time demanding to solve for larger numbers of variables, the most dramatic increase of run time occurs when applying the coherent point drift algorithm (see Table 1).

Figure 6. Result of our method on two cat shapes from TOSCA [12]. This high resolution shape has 27894 vertices, making it infeasible to store the entire pairwise distance matrix in memory. Our multi-scale approach recovered a smooth matching from only 20 sparse correspondences given as the input using five hierarchical scales as detailed in Section 2.2.

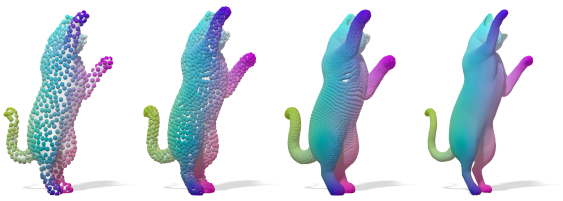

Figure 7. Visualization of the multiscale iterations evaluated in Fig. 8. From left to right: 1K, 2K, 8K, 28K (all) vertices.

3.3. Recovering high-resolution correspondences using multiscale

In this set of experiments we demonstrate how the PMF together with the multiscale method described in 2.2 can recover very accurate matchings on shapes being sampled at high resolution. Figure 6 shows a dense bijective matching between two shapes sampled at $n = 27894$ points each. At each of the six scales $n_i \in \{10^3, 2\times10^3, 4\times10^3, 8\times10^3, 1.6\times10^4, n\}$ the constrained LAP (11) was solved.

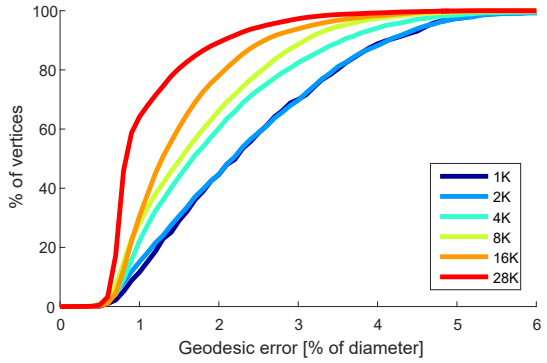

Figure 8. Quantitative analysis of correspondences between the two cats shown in Figure 6, recoverd using the multiscale approach. The geodesic errors are measured with respect to the ground-truth on the finest scale. At coarse scales the minimal expected geodesic error introduced by any matching is in the order of the sampling radius. As expected, the accuracy of the matching increases with each iteration.

n	1000	1000	6890	6890
r	20	50	20	50
Nearest neighbors	0.04	0.06	1.35	2.88
Bijective NN	2.79	2.30	463.66	253.03
CPD	4.79	4.67	1745.06	2085.65
NN + PMF	1.75	1.28	382.86	244.10
Bij. NN + PMF	4.06	3.44	746.00	440.94

Table 1. Average runtimes in seconds. We compare the runtimes of different correspondence recovery methods. Given the rank k of a functional map approximating the correspondence between shapes sampled at n points each, we report the time it takes to obtain a dense matching. See Figures 11,9 and 12 for evaluations of accuracy.

Figure 7 shows the sequence of matchings over the scales. Figure 8 shows the improvement of correspondence accuracy at finer scales. By using the weighting functions we force points to stay close to their nearest neighbor in the coarser sampling and thus can guarantee to approximately keep the accuracy from the coarser scale. Solving the constrained LAP at the finest resolution took less then 9 minutes. Calculating the kernel density matrices (6) for all scales took less than 40 minutes.

Another test was performed on pairs of shapes from the FAUST dataset. As Figure 10 shows, the correspondences obtained using a four-scale scheme are comparable in accuracy to the solution of a single-scale scheme. However, the runtime of the multi-scale approach is significantly lower. Calculating the kernel density matrices (6) for all scales took about 4 minutes, while solving the LAPs at all scales took around 18 seconds.

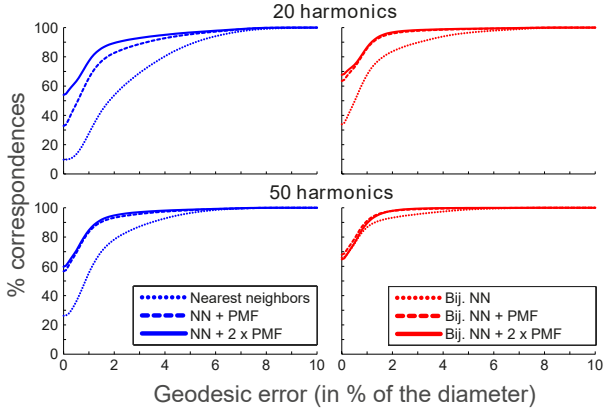

Figure 9. Quantitative comparison of methods for pointwise correspondence on the non-isometric FAUST dataset (about 7K vertices).

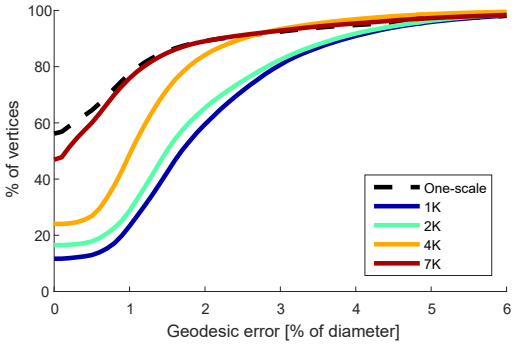

Figure 10. Error curves for a multiscale experiment on the FAUST dataset, showing result for intermediate scales. For comparison, the solution obtained by a single-scale PMF is shown in dashed black.

4. Discussion and conclusion

We considered the problem of bijective correspondence recovery by means of filtering a given set of matches coming from any of the existing algorithms (including those not guaranteeing bijectivity, or producing sparse correspondences). Viewing correspondence computation as a kernel density estimation problem in the product space, we introduced the product manifold filter that leads to smooth correspondences, with the additional constraint of bijectivity embodied through an LAP. We believe that statistical tools that have been heavily used in other domains of science and engineering might be very useful in shape analysis, and invite the community to further explore this direction. Of special interest is the possibility to lift the product space to higher dimensions encoding local similarity of points on the two shapes, for instance by using descriptors. The way the kernel density estimator is constructed does not restrict

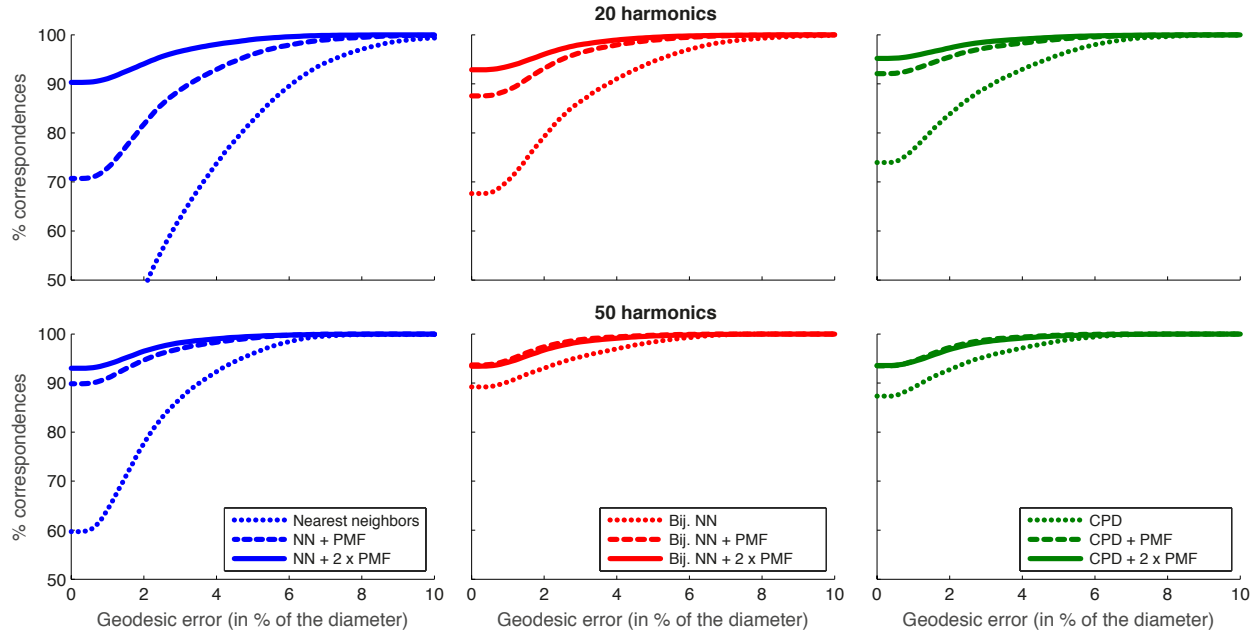

Figure 11. Quantitative comparison of methods for pointwise correspondence recovery from a functional map (20 and 50 eigenfunctions). We matched 70 pairs from the near-isometric SCAPE dataset (1K). Plotted are the histograms of geodesic errors. Filtering the results of nearest neighbors (left) outperforms the state of the art method (right) while having only a fraction of its runtime (Table 1). Even better results are achieved under affordable runtimes when initializing the PMF estimator with the result of bijective NN (center).

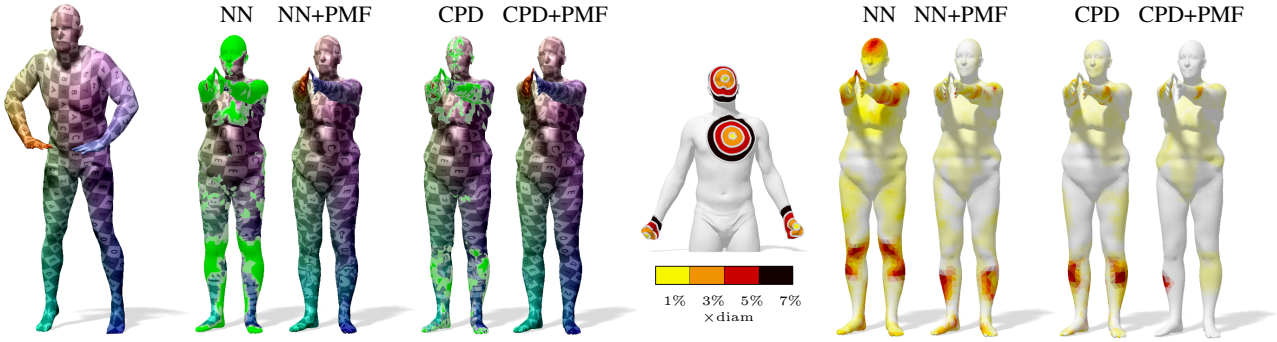

Figure 12. Qualitative comparison of methods for pointwise correspondence recovery from a functional map. Current methods such as nearest neighbors (NN) and coherent point drift (CPD) suffer from bad accuracy and lack of surjectivity. Applying the proposed estimation to either of them gives a guaranteed bijective matching with high accuracy and improved smoothness. Left: We visualize the accuracy of the methods by transferring texture from the source shape $\mathcal{X}$ to the target shape $\mathcal{Y}$. Neither NN nor CPD produce bijective mappings. The lack of surjectivity is visualized by assigning a fixed color (green) to not-hit points. Right: The geodesic error (distance between ground-truth and recovered match, relative to the shape diameter) induced by the matching is visualized on the target shape $\mathcal{Y}$.

the samples per shape to be distinct. Together with the use of weighting factors this alllows to directly work with soft maps as inputs. Finally, we believe that denoising the correspondence manifold in the product space is a useful perspective applicable to different problems in computer vision where smooth correspondences are desired, such as optical flow.

Acknowledgements We thank Zorah Lähner for useful discussions and Florian Bernard for the his mex implementation of the sparse auction algorithm. D.C. and M.V. are supported by the ERC CoG 3D Reloaded. A.B. is supported by the ERC StG RAPID. E.R. is supported by ERC StG No. 307047 (COMET). R.L. is supported by the European Google PhD fellowship in machine learning.

References

[1] Y. Aflalo, H. Brezis, and R. Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160, 2015.

[2] Y. Aflalo, A. Bronstein, and R. Kimmel. On convex relaxation of graph isomorphism. *PNAS*, 112(10):2942–2947, 2015.

[3] Y. Aflalo, A. Dubrovina, and R. Kimmel. Spectral generalized multidimensional scaling. *IJCV*, 2016.

[4] Y. Aflalo and R. Kimmel. Spectral multidimensional scaling. *PNAS*, 110(45):18052–18057, 2013.

[5] N. Aigerman and Y. Lipman. Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics (TOG)*, 35(6):217, 2016.

[6] N. Aigerman, R. Poranne, and Y. Lipman. Lifted bijections for low distortion surface mappings. *ACM Transactions on Graphics (TOG)*, 33(4):69, 2014.

[7] D. Anguelov et al. SCAPE: Shape completion and animation of people. *TOG*, 24(3):408–416, 2005.

[8] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. ICCV*, 2011.

[9] D. P. Bertsekas. *Network optimization: continuous and discrete models*. Citeseer, 1998.

[10] P. J. Besl and N. D. McKay. A method for registration of 3D shapes. *PAMI*, 14(2):239–256, 1992.

[11] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, 2014.

[12] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, 2008.

[13] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006.

[14] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proc. CVPR*, 2010.

[15] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *Proc. ICCV*, 2015.

[16] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proc. Conf. Robotics and Automation*, 1991.

[17] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21):7426–7431, 2005.

[18] A. Elad and R. Kimmel. Bending invariant representations for surfaces. In *Proc. CVPR*, 2001.

[19] I. Kezurer, S. Kovalsky, R. Basri, and Y. Lipman. Tight relaxations of quadratic matching. *Computer Graphics Forum*, 34(5), 2015.

[20] V. G. Kim, Y. Lipman, X. Chen, and T. A. Funkhouser. Möbius transformations for global intrinsic symmetry analysis. *Computer Graphics Forum*, 29(5):1689–1700, 2010.

[21] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *TOG*, 30(4):79, 2011.

[22] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst. Functional correspondence by matrix completion. In *Proc. CVPR*, 2015.

[23] Z. Lähner, E. Rodolà, F. R. Schmidt, M. M. Bronstein, and D. Cremers. Efficient globally optimal 2d-to-3d deformable shape matching. In *Proc. CVPR*, 2016.

[24] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Proc. ICCV*, 2005.

[25] Y. Lipman and I. Daubechies. Conformal Wasserstein distances: Comparing surfaces in polynomial time. *Advances in Mathematics*, 227(3):1047 – 1077, 2011.

[26] R. Litman and A. Bronstein. Spectrometer: Amortized sublinear spectral approximation of distance on graphs. In *2016 International Conference on 3D Vision*, 2016.

[27] D. Mateus, R. P. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *Proc. CVPR*, 2008.

[28] F. Mémoli. Gromov-Wasserstein Distances and the Metric Approach to Object Matching. *Foundations of Computational Mathematics*, pages 1–71, 2011.

[29] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005.

[30] A. Myronenko and X. Song. Point set registration: Coherent point drift. *TPAMI*, 32(12):2262–2275, 2010.

[31] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. on Graphics*, 31(4):30, 2012.

[32] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One point isometric matching with the heat kernel. *Computer Graphics Forum*, 29(5):1555–1564, 2010.

[33] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro. Sparse modeling of intrinsic correspondences. In *Computer Graphics Forum*, volume 32, pages 459–468. Wiley Online Library, 2013.

[34] E. Rodolà, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *Proc. CVPR*, 2012.

[35] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 2016.

[36] E. Rodolà, M. Moeller, and D. Cremers. Point-wise map recovery and refinement from functional correspondence. In *Proceedings Vision, Modeling and Visualization (VMV)*, Aachen, Germany, 2015.

[37] Y. Sahillioğlu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Computer Graphics Forum*, 30(5):1461–1470, 2011.

[38] A. Shtern and R. Kimmel. Matching lbo eigenspace of non-rigid shapes via high order statistics. *arXiv:1310.4459*, 2013.

[39] J. Solomon, A. Nguyen, A. Butscher, M. Ben-Chen, and L. Guibas. Soft maps between surfaces. In *Computer Graphics Forum*, volume 31, pages 1617–1626, 2012.

[40] A. Tevs et al. Intrinsic shape matching by planned landmark sampling. *Computer Graphics Forum*, 30(2):543–552, 2011.

[41] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *Proc. ECCV*, 2008.

[42] S. Umeyama. An eigendecomposition approach to weighted graph matching problems. *PAMI*, 10(5):695–703, 1988.

[43] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 20:1–23, 2010.

[44] T. Windheuser, U. Schlickewei, F. R. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.

[45] A. Zaharescu, E. Boyer, K. Varanasi, and R. Horaud. Surface feature detection and description with applications to mesh matching. In *Proc. CVPR*, 2009.

Chapter 8

Efficient Deformable Shape Correspondence via Kernel Matching

In this paper we extend the previous one on both theoretical and practical aspects. We show how the iterative process relates to solving concave quadratic assignment problems, to an alternating diffusion process and to a low pass filtering of functional maps. In particular the relation to quadratic assignment problems proves convergence of the iterative method and shows that an energy is decreased in each iteration. By introducing slack variables we further allow to match shapes with different numbers of vertices, including partial correspondences and correspondences of shapes with occlusions. We also propose a more elaborate muliscale method to match meshes of high resolution.

Publication: Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodolà, Alex Bronstein, Michael Bronstein, Ron Kimmel and Daniel Cremers. "Efficient deformable shape correspondence via kernel matching". In: *International Conference on 3D Vision (3DV)*, IEEE, 2017. DOI: 10.1109/3DV.2017.00065

Individual contribution

Problem definition	significantly contributed
Literature survey	significantly contributed
Implementation	significantly contributed
Experimental evaluation	significantly contributed
Preparation of the manuscript	significantly contributed

Efficient Deformable Shape Correspondence via Kernel Matching

Matthias Vestner⋆ TU Munich
Zorah Lähner⋆ TU Munich
Amit Boyarski⋆ Technion
Or Litany TAU
Ron Slossberg Technion
Tal Remez TAU
Emanuele Rodolà Sapienza University of Rome / USI Lugano
Alex Bronstein Technion / TAU / Intel
Michael Bronstein USI Lugano / TAU / Intel
Ron Kimmel Technion / Intel
Daniel Cremers TU Munich

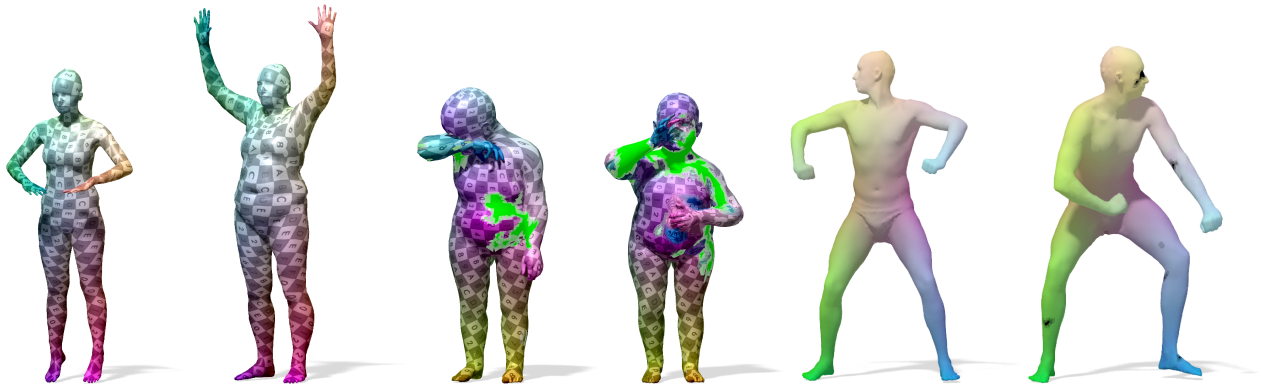

Figure 1: Qualitative examples on FAUST models (left), SHREC'16 (middle) and SCAPE (right). In the SHREC experiment, the green parts mark where no correspondence was found. Notice how those areas are close to the parts that are hidden in the other model. The missing matches (marked in black) in the SCAPE experiment are an artifact due to the multiscale approach.

Abstract

We present a method to match three dimensional shapes under non-isometric deformations, topology changes and partiality. We formulate the problem as matching between a set of pair-wise and point-wise descriptors, imposing a continuity prior on the mapping, and propose a projected descent optimization procedure inspired by difference of convex functions (DC) programming.

1. Introduction

Finding correspondences between non-rigid shapes is a fundamental problem in computer vision, graphics and pattern recognition, with applications including shape comparison, texture transfer, and shape interpolation just to name a few. Given two three-dimensional objects $\mathcal{X}$ and $\mathcal{Y}$, modeled as compact two-dimensional Riemannian manifolds, our task is to find a meaningful correspondence $\varphi : \mathcal{X} \to \mathcal{Y}$. While a rigorous definition of *meaningful* is challenging, one can identify some desirable properties of φ:

1. Bijective.
2. Continuous in both directions, in the sense that nearby points on $\mathcal{X}$ should be mapped to nearby points on $\mathcal{Y}$ (and vice versa).
3. Similar points should be put into correspondence.

For the simplicity of the introduction, we assume the two shapes $\mathcal{X}$ and $\mathcal{Y}$ to be sampled at n points each, and defer the case of different number of samples to the algorithmic part of this paper detailed in Section 3. Assuming a consistent sampling (e.g. via farthest point sampling with a sufficiently large number n of points), the discrete counterpart to the correspondence φ is a mapping $\pi : \{x_1, \dots x_n\} \to \{y_1, \dots y_n\}$, which admits a representation as a permutation matrix $\mathbf{\Pi} \in \{0,1\}^{n \times n}$ satisfying $\mathbf{\Pi}^\top \mathbf{1} = \mathbf{\Pi}\mathbf{1} = \mathbf{1}$ with $\mathbf{1}$ being a column vector of ones. We henceforth denote the space of $n \times n$ permutation matrices by $\mathcal{P}_n$.

The vast majority of shape matching approaches phrase the correspondence problem as an energy minimization problem

$$\mathbf{\Pi}^* = \arg\min_{\mathbf{\Pi} \in \mathcal{P}_n} E(\mathbf{\Pi}) \,, \tag{1}$$

⋆ equal contribution

where $E(\mathbf{\Pi})$ is usually a weighed aggregate of two terms

$$E(\mathbf{\Pi}) = \alpha g(\mathbf{\Pi}) + h(\mathbf{\Pi})\,. \tag{2}$$

The first term $g(\mathbf{\Pi})$ is a fidelity term trying to align a set of *pointwise descriptors* encoding the similarity between points, while the second term $h(\mathbf{\Pi})$ is a regularization term promoting the continuity of the correspondence by aligning a set of *pairwise descriptors* encoding global/local relations between pairs of points. The parameter α governs the trade-off between the two terms.

While the constraint $\mathbf{\Pi} \in \mathcal{P}_n$ guarantees bijectivity of the correspondence, the two terms h and g correspond, respectively, to the second and the third desirable qualities of a meaningful correspondence, and provide a trade-off between complexity, fidelity and regularity. We stress in our work that despite their seemingly unrelated nature, those properties are in fact tightly connected, i.e., choosing a particular set of pairwise descriptors might have a profound effect not only on the regularity of the final solution, but also on the complexity of the resulting optimization problem. We will elaborate on these aspects in more detail.

Related work. Finding correspondences between shapes is a well-studied problem. Traditionally, the solution involves minimization of a distortion criterion which fits into one of the two categories: pointwise descriptor similarity [4, 16, 43, 49, 51], and pairwise relations [17, 18, 34, 52]. In the former case, matches are obtained via nearest neighbor search or, when injectivity is required, by solving a linear assignment problem (LAP). Pairwise methods usually come at a high computational cost, with the most classical formulation taking the form of an NP-hard *quadratic assignment problem* (QAP) [37]. Several heuristics have been proposed to address this issue by using subsampling [50] or coarse-to-fine techniques [44, 54]. Various relaxations have been used to make the QAP problem tractable [1, 13, 17, 21, 27, 39], however they result in approximate solutions. In addition, pairwise geodesics are computationally expensive, and sensitive to noise. In [20] the use of heat kernels was proposed as a noise-tolerant approximation of matching adjacency matrices. In [53] dense bijective correspondences were derived from sparse and possibly noisy input using an iterative filtering scheme, making use of geodesic Gaussian kernels.

A different family of methods look for pointwise matches in a lower-dimensional "canonical" embedding space. Such embedding can be carried out by multidimensional scaling [12, 19] or via the eigenfunctions of the Laplace-Beltrami operator (LBO) [33, 48]. The correspondence is then calculated in the embedding space using a simple rigid alignment technique such as ICP [6]. Functional maps [23, 36] can be seen as a sophisticated way to initialize ICP when using this spectral embedding. Other bases can be used within the functional map framework [24]. In particular, the eigenspaces arising from the spectral decomposition of the geodesic distance matrices have been shown to outperform the LBO basis for the case of isometric shapes [46]. In [55] the matching problem is phrased as an integer linear program, enforcing continuity of the correspondence via a linear constraint. This additional constraint however makes the problem computationally intractable even for modestly-sized shapes, requiring the use of relaxation and post-processing heuristics.

Most recent works attempt to formulate the correspondence problem as a learning problem [42] and design intrinsic deep learning architectures on manifolds and point clouds [9–11, 29, 32, 35]. As of today, these methods hold the record of performance on deformable correspondence benchmarks; however, supervised learning requires a significant annotated training set that is often hard to obtain.

Contribution. The main contribution of this paper is a simple method that works out-of-the-box for finding high quality continuous (regular) correspondence between two not necessarily isometric shapes. The method can be seen as an improved version of [53], and is accompanied by theoretical insights that shed light on its effectiveness. In particular, we contrast the method with other shape matching approaches and elaborate on the computational benefits of using *kernels* rather than distances as pairwise descriptors. The key insight is the realization that high quality regular correspondence can be obtained from a rough irregular one by a sequence of smoothing and projection operations. Remarkably, this process admits an appealing interpretation as an alternating diffusion process [26]. We report drastic runtime and scalability improvements compared to [53], and present an extension to the setting of *partial* shape correspondence and an effective multi-scale approach.

2. Background

2.1. Pointwise descriptors

Similarity of points is often measured with the help of *pointwise descriptors* $f_{\mathcal{X}} : \mathcal{X} \to \mathbb{R}^q$, $f_{\mathcal{Y}} : \mathcal{Y} \to \mathbb{R}^q$ that are constructed in a way such that similar points on the two shapes are assigned closeby (in the Euclidean sense) descriptors, while dissimilar points are assigned distant descriptors. In the discrete case, the descriptors $f_{\mathcal{X}}, f_{\mathcal{Y}}$ can be encoded as matrices $\mathbf{F}_{\mathcal{X}}, \mathbf{F}_{\mathcal{Y}} \in \mathbb{R}^{n\times q}$ giving rise to the optimization problem[1]

$$\operatorname*{arg\,min}_{\mathbf{\Pi}\in\mathcal{P}_n} \|\mathbf{\Pi}\mathbf{F}_{\mathcal{X}} - \mathbf{F}_{\mathcal{Y}}\|^2 = \operatorname*{arg\,max}_{\mathbf{\Pi}\in\mathcal{P}_n} \langle \mathbf{\Pi}, \mathbf{F}_{\mathcal{Y}}\mathbf{F}_{\mathcal{X}}^\top \rangle\,. \tag{3}$$

Problem (3) is linear in $\mathbf{\Pi}$ and is therefore one of the rare examples of combinatorial optimization problems that can be globally optimized in polynomial time; the best known

[1] Throughout this paper we use the Frobenius norm $\|\mathbf{A}\| = \sqrt{\langle \mathbf{A}, \mathbf{A}\rangle}$, where $\langle \mathbf{A}, \mathbf{B}\rangle = \mathrm{tr}(\mathbf{A}^\top \mathbf{B})$ is the Euclidean inner product.

complexity $O(n^2 \log n)$ is achieved by the auction algorithm [5].

Over the last years, *intrinsic* features have extensively been used due to their invariance to isometry. However, they come with two main drawbacks: First, the implicit assumption that the shapes at hand are isometric is not always met in practice. Today's best performing approaches partially tackle this problem using deep learning [9–11, 32, 35]. Secondly, many natural shapes come with at least one intrinsic (e.g., bilateral) symmetry that is impossible to capture by purely intrinsic features, be these handcrafted or learned. Correspondences obtained by (3) may suffer from severe discontinuities due to some points being mapped to the desired destination, and others to the symmetric counterpart.

2.2. Pairwise descriptors

Another family of methods consider *pairwise descriptors* of the form $d_{\mathcal{X}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $d_{\mathcal{Y}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ encoded in the discrete setting as symmetric matrices $\mathbf{D}_{\mathcal{X}}, \mathbf{D}_{\mathcal{Y}} \in \mathbb{R}^{n \times n}$. These methods aim at solving optimization problems of the form

$$\mathbf{\Pi}^* = \operatorname*{arg\,min}_{\mathbf{\Pi} \in \mathcal{P}_n} \|\mathbf{\Pi}\mathbf{D}_{\mathcal{X}} - \mathbf{D}_{\mathcal{Y}}\mathbf{\Pi}\|^2 \tag{4}$$

$$= \operatorname*{argmax}_{\mathbf{\Pi} \in \mathcal{P}_n} \langle \mathbf{\Pi}, \mathbf{D}_{\mathcal{Y}}\mathbf{\Pi}\mathbf{D}_{\mathcal{X}} \rangle \,, \tag{5}$$

known under the names of *graph matching* (GM) or *quadratic assignment problem* (QAP), and are in general not solvable in polynomial time. A typical way to circumvent the complexity issue is to relax the integer constraint $\pi_{ij} \in \{0, 1\}$ and optimize the objectives (4)-(5) over the convex set of *bi-stochastic matrices* $\mathcal{B}_n = \{\mathbf{P} \geq \mathbf{0} : \mathbf{P}^\top \mathbf{1} = \mathbf{P}\mathbf{1} = \mathbf{1}\}$. Note that when viewed as functions over this convex set, the objectives (4)-(5) are no longer equivalent. In particular, (4) will always be convex, while the convexity of (5) depends on the eigenvalues of the matrices $\mathbf{D}_{\mathcal{X}}$ and $\mathbf{D}_{\mathcal{Y}}$, as shown in the following lemma.

Lemma 1. *Let $\mathbf{D}_{\mathcal{X}}, \mathbf{D}_{\mathcal{Y}}$ be symmetric. The function $h(\mathbf{P}) = \langle \mathbf{P}, \mathbf{D}_{\mathcal{Y}}\mathbf{P}\mathbf{D}_{\mathcal{X}} \rangle$ over the set of bi-stochastic matrices $\mathcal{B}_n$ is (strictly) convex iff all eigenvalues of $\mathbf{D}_{\mathcal{X}}$ and $\mathbf{D}_{\mathcal{Y}}$ are (strictly) positive.*

Corollary 1. *If all eigenvalues of $\mathbf{D}_{\mathcal{X}}$ and $\mathbf{D}_{\mathcal{Y}}$ are strictly positive, the optimum of the relaxed problem coincides with that of the original combinatorial problem:*

$$\operatorname*{argmax}_{\mathbf{P} \in \mathcal{B}_n} h(\mathbf{P}) = \operatorname*{argmax}_{\mathbf{\Pi} \in \mathcal{P}_n} h(\mathbf{\Pi}) \,. \tag{6}$$

Notice that we can add a linear term (weighted by a scalar factor α), such as the one in (3), while still keeping this property: $E(\mathbf{P}) = \alpha \langle \mathbf{P}, \mathbf{F}_{\mathcal{Y}}\mathbf{F}_{\mathcal{X}}^\top \rangle + \langle \mathbf{P}, \mathbf{D}_{\mathcal{Y}}\mathbf{P}\mathbf{D}_{\mathcal{X}} \rangle$.

Popular pairwise descriptors include a variety of pairwise distances [13, 15, 38] and kernels [31, 47, 53] tailored for the specific class of deformations. In what follows, we advocate the superiority of using kernels over distances.

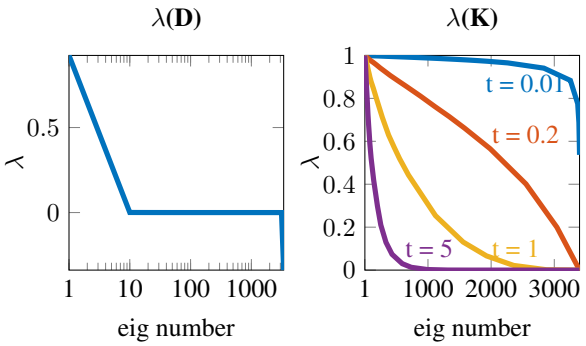

Figure 2: Spectrum of distance matrix (left) vs. spectrum of heat-kernel matrix (right) for several values of $t \in [0.01, 5]$ computed on the cat shape from TOSCA.

Pairwise distances. A common choice for pairwise descriptors are geodesic distances $d_{\mathcal{X}}(x_i, x_j)$, a choice motivated by the fact that, for isometric shapes, these are preserved by the optimal $\mathbf{\Pi}$. Geodesic distances have major drawbacks, both from the modeling and computational point of view. On the modeling side, they introduce a bias towards far away points and are sensitive to topological noise. On the computational side, they are slow to compute and give rise to highly non-convex (and non-differentiable) optimization problems. Note that, although one may employ more robust definitions of distance [15, 17], these do not solve the optimization issues.

Heat kernels. Heat kernels are fundamental solutions to the *heat diffusion equation* on manifold $\mathcal{X}$,

$$\frac{\partial u(t,x)}{\partial t} = \Delta_{\mathcal{X}} u(t,x) \,, \tag{7}$$

with the initial condition $u(0,x) = u_0(x)$ and additional boundary conditions if applicable. Here $u : [0,\infty) \times \mathcal{X} \to \mathbb{R}$ represents the amount of heat at point x at time t. The solution is linear in the initial distribution and is given by

$$u(t,x) = \int_{\mathcal{X}} k(t,x,x') u_0(x') dx' \,, \tag{8}$$

where $k : \mathbb{R}^+ \times \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the *heat kernel* and its values can be interpreted as the amount of heat transported from x' to x in time t. In the Euclidean case, the heat kernel is an isotropic Gaussian kernel with the variance proportional to the diffusion time t.

For a compact manifold $\mathcal{X}$, the heat kernel can be expressed as the exponent of the intrinsic self-adjoint negative semi-definite Laplacian operator $\Delta_{\mathcal{X}}$,

$$k(t,x,x') = \sum_i e^{\lambda_i t} \phi_i(x) \phi_i(x'), \tag{9}$$

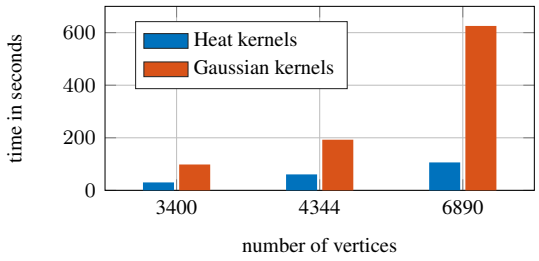

Figure 3: Runtime comparison of matching shapes with varying number of vertices using our algorithm with heat kernels compared to Gaussian kernels [53]. For more info see supp. material.

where $\Delta_{\mathcal{X}}\phi_i(x) = \lambda_i\phi_i(x)$ is the eigendecomposition of the Laplacian with eigenvectors $\phi_1, \phi_2, \ldots$ and corresponding non-positive eigenvalues $0 = \lambda_1 \geq \lambda_2 \geq \ldots$. The null eigenvalue is associated with a constant eigenvector.
In the discrete setting, the heat kernel is given by the positive-definite matrix $\mathbf{K}_{\mathcal{X}} = e^{t\boldsymbol{\Delta}_{\mathcal{X}}} = \boldsymbol{\Phi}e^{t\boldsymbol{\Lambda}_{\mathcal{X}}}\boldsymbol{\Phi}^\top$. The constant eigenvector corresponds to the unit eigenvalue, $\mathbf{K}_{\mathcal{X}}\mathbf{1} = \mathbf{1}$.
An issue that is often overlooked is the relation between the original and relaxed solution of (5), which is tightly connected to the choice of pairwise descriptors. Corollary 1 asserts the sufficient condition under which this relaxation is exact. Whereas heat kernels, being (strictly) positive definite, satisfy this condition, distance matrices never do. A distance matrix, having non-negative entries and trace zero, will always, by the Perron-Frobenius theorem, have one large positive eigenvalue and several low magnitude negative eigenvalues[2] [8]. This distribution of eigenvalues is illustrated in Figure 2.

2.3. Bijective maps and functional maps

The requirement of bijectivity is what makes a problem (1) computationally hard. A variety of relaxation techniques can be applied to alleviate this complexity. Amongst the most popular are relaxing the column or row sum constraints, relaxing the integer constraints, or restricting the matrix to a sphere of constant norm [27]. A bijective mapping can then be recovered by a post processing step, such as projection onto the set of permutation matrices

$$\boldsymbol{\Pi}^* = \operatorname*{arg\,min}_{\boldsymbol{\Pi}\in\mathcal{P}_n} \|\boldsymbol{\Pi} - \mathbf{P}\|^2 = \operatorname*{argmax}_{\boldsymbol{\Pi}\in\mathcal{P}_n} \langle \boldsymbol{\Pi}, \mathbf{P}\rangle \,. \tag{10}$$

One popular technique in recent years replaces the combinatorially hard point-wise map recovery problem with the simpler problem of finding a linear map between functions [36]. A *functional map* is a map between functional spaces $T : L^2(\mathcal{X}) \to L^2(\mathcal{Y})$, which can be discretized (under the previous assumptions of n vertices in each shape) as an $n \times n$ matrix $\mathbf{T}$. Providing a pair of orthonormal bases $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n)$ and $\boldsymbol{\Psi} = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n)$ for $L^2(\mathcal{X})$ and $L^2(\mathcal{Y})$, respectively, one can express $\mathbf{T} = \boldsymbol{\Psi}\mathbf{C}\boldsymbol{\Phi}^\top$, where $\mathbf{C}$ acts as a basis transformation matrix. Two common choices for basis are the Dirac (or hat) basis, in which the functional map attains the form of a permutation matrix, and the Laplacian eigenbasis, which is especially suited when the map is smooth, so it can be approximated using a truncated basis of k first basis functions corresponding to the lowest frequencies. The computation of the functional map thus boils down to solving a linear system $\mathbf{C}\boldsymbol{\Phi}^\top\mathbf{F}_{\mathcal{X}} = \boldsymbol{\Psi}^\top\mathbf{F}_{\mathcal{Y}}$. The recovery of the point-wise map from the functional map can be obtained by ICP-like procedures [36, 41], with the possible introduction of bijectivity constraints [53]. The fact that the map is band-limited is often erroneously referred to as "smoothness" in the literature; however, the bijective map recovered from such a band-limited map is not guaranteed to be continuous let alone smooth (i.e., continuously differentiable).

3. Method

3.1. Optimization

We aim at maximizing $E(\boldsymbol{\Pi})$ over $\mathcal{P}_n$, which by Corollary 1 is equivalent to the relaxed problem

$$\operatorname*{argmax}_{\mathbf{P}\in\mathcal{B}_n} E(\mathbf{P}) = \operatorname*{argmax}_{\mathbf{P}\in\mathcal{B}_n} \langle \mathbf{P}, \alpha\mathbf{F}_{\mathcal{Y}}\mathbf{F}_{\mathcal{X}}^\top + \mathbf{K}_{\mathcal{Y}}\mathbf{P}\mathbf{K}_{\mathcal{X}}\rangle \tag{11}$$

where $\mathbf{F}_{\mathcal{X}}, \mathbf{F}_{\mathcal{Y}}$ are matrices of pointwise descriptors and $\mathbf{K}_{\mathcal{X}}, \mathbf{K}_{\mathcal{Y}}$ are the positive-definite heat kernel matrices on $\mathcal{X}$ and $\mathcal{Y}$, respectively. This maximization problem can be seen as the minimization of the difference of convex functions:

$$\arg\min_{\mathbf{P}\in\mathbb{R}^{n\times n}} B(\mathbf{P}) - E(\mathbf{P}). \tag{12}$$

where B is the (convex) indicator function on the set of bistochastic matrices $\mathcal{B}_n$.
A renowned way to optimize this type of energy is the difference of convex functions (DC) algorithm that starts with some initial $\mathbf{P}^0$ and then iterates the following two steps until convergence:

- Select $\mathbf{Q}^k \in \partial E(\mathbf{P}^k)$.
- Select $\mathbf{P}^{k+1} \in \partial B^*(\mathbf{Q}^k)$.

Here B^* denotes the convex conjugate of B and $\partial E, \partial B^*$ denote the subdifferentials (set of supporting hyperplanes) of E and B^*, respectively.
For a differentiable E, the step of the DC algorithm assumes the form

$$\mathbf{P}^{k+1} = \operatorname*{argmax}_{\mathbf{P}\in\mathcal{B}_n} \langle \mathbf{P}, \nabla E(\mathbf{P}^k)\rangle \,. \tag{13}$$

[2] In the Euclidean case, a distance matrix has exactly one positive eigenvalue and all the rest are negative with small magnitude.

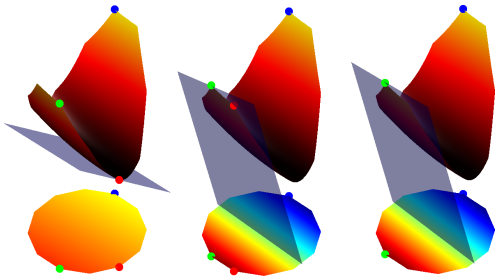

Figure 4: Schematic illustration of the proposed algorithm for maximizing a convex quadratic objective over a convex polytope, by successively maximizing a linear sub-estimate of it. The *hot* color map encodes the function values. The *jet* color map encodes the values of the linear sub-estimate. The point around which the objective is linearized is depicted in red. The global maximum is depicted in blue. The maximum of the linear sub-estimate is depicted in green. Notice that the algorithm travels between extreme but not necessarily adjacent points of the polytope, until it converges to a local maximum.

Moreover, the value of the objective is an increasing sequence, $E(\mathbf{P}^{k+1}) > E(\mathbf{P}^k)$, and each iterate $\mathbf{P}^k$ is a permutation matrix. We provide the proof in the supplementary material. Figure 4 illustrates this iterative process.
Since $\mathbf{P}^k$ is guaranteed to be a permutation matrix, we henceforth use $\mathbf{\Pi}^k$ to denote the iterates. For our choice of E, the gradient is given by

$$\nabla E = \alpha \mathbf{F}_\mathcal{Y} \mathbf{F}_\mathcal{X}^\top + \mathbf{K}_\mathcal{Y} \mathbf{\Pi} \mathbf{K}_\mathcal{X} \tag{14}$$

yielding the step

$$\mathbf{\Pi}^{k+1} = \operatorname*{argmax}_{\mathbf{\Pi} \in \mathcal{B}_n} \langle \mathbf{\Pi}, \alpha \mathbf{F}_\mathcal{Y} \mathbf{F}_\mathcal{X}^\top + \mathbf{K}_\mathcal{Y} \mathbf{\Pi}^k \mathbf{K}_\mathcal{X} \rangle \,. \tag{15}$$

In the experiments presented in this paper, we use the data fidelity term $\langle \mathbf{\Pi}, \mathbf{F}_\mathcal{Y} \mathbf{F}_\mathcal{X}^\top \rangle$ mainly to initialize the process:

$$\mathbf{\Pi}^0 = \operatorname*{argmax}_{\mathbf{\Pi} \in \mathcal{B}_n} \langle \mathbf{\Pi}, \mathbf{F}_\mathcal{Y} \mathbf{F}_\mathcal{X}^\top \rangle. \tag{16}$$

3.2. Partial matching using slack variables

In a general setting, we will be dealing with shapes having different number of vertices. Let us denote by $n_\mathcal{X}$ the number of vertices on $\mathcal{X}$ and by $n_\mathcal{Y}$ the number of vertices on $\mathcal{Y}$, and assume w.l.o.g. $n_\mathcal{X} \geq n_\mathcal{Y}$. We aim at optimizing

$$\arg\max_{\mathbf{\Pi} \in \mathcal{P}_{n_\mathcal{X}}^{n_\mathcal{Y}}} \langle \mathbf{\Pi}, \alpha \mathbf{F}_\mathcal{Y} \mathbf{F}_\mathcal{X}^\top + \mathbf{K}_\mathcal{Y} \mathbf{\Pi} \mathbf{K}_\mathcal{X} \rangle \tag{17}$$

where the space of *rectangular permutation matrices* $\mathcal{P}_{n_\mathcal{X}}^{n_\mathcal{Y}}$ is given by $\mathcal{P}_{n_\mathcal{X}}^{n_\mathcal{Y}} = \{\mathbf{\Pi} \in \{0,1\}^{n_\mathcal{Y} \times n_\mathcal{X}} : \mathbf{\Pi}\mathbf{1} \leq \mathbf{1}, \mathbf{\Pi}^\top \mathbf{1} = \mathbf{1}\}$. Analogously to the previously discussed case in which we had $n_\mathcal{X} = n_\mathcal{Y} = n$, we iteratively solve

$$\mathbf{\Pi}^{k+1} = \arg\max_{\mathbf{\Pi} \in \mathcal{P}_{n_\mathcal{X}}^{n_\mathcal{Y}}} \langle \mathbf{\Pi}, \alpha \mathbf{F}_\mathcal{Y} \mathbf{F}_\mathcal{X}^\top + \mathbf{K}_\mathcal{Y} \mathbf{\Pi}^k \mathbf{K}_\mathcal{X} \rangle \,. \tag{18}$$

In order to solve these optimization problems we pad the rectangular matrix $\alpha \mathbf{F}_\mathcal{Y} \mathbf{F}_\mathcal{X}^\top + \mathbf{K}_\mathcal{Y} \mathbf{\Pi}^k \mathbf{K}_\mathcal{X}$ with constant values c (slack variables) such that it becomes square. After the correspondence is computed, we discard the ones belonging to the introduced slack variables. While such a treatment does not affect the value of the maximum, the constant c has to be chosen appropriately to avoid ambiguity between the slacks and the actual vertices on $\mathcal{X}$. A drawback of this approach is that there are $(n_\mathcal{X} - n_\mathcal{Y})!$ solutions achieving the optimal score, leading to worse runtime in the presence of many slacks. See Fig.5. for a proof of concept of this approach.

3.3. Multiscale acceleration

Solving the LAP (18) at each iteration of the DC algorithm has a super-quadratic complexity. As a consequence, the proposed method is only directly applicable for small n (up to 15×10^3 in our experiments). We therefore propose a multiscale approach that enables us to find correspondences between larger meshes.
We start by resampling both shapes to a number of vertices we can handle and solving for a bijection $\pi_0 : \mathcal{X}_0 \to \mathcal{Y}_0$. This set of initial vertices is called *seeds*. The seeds on $\mathcal{X}$ are clustered into k Voronoi cells and these cells are transfered to $\mathcal{Y}$ via π_0. More points are added iteratively and assigned to the same Voronoi cell as their closest seed. Next, we solve for $\pi_i : \mathcal{X}_i \to \mathcal{Y}_i$ where i refers to the i-th Voronoi cell. This proceeds until all points are sampled (see Figure 6 for a visualization). To keep the correspondence consistent at the boundary of the Voronoi cells, we choose 1000 correspondences from π_0 and use them to orient each Voronoi cell correctly over all iterations. Additional details are provided in the supplementary material.

4. Interpretation

In what follows we provide different, yet complementary interpretations of the proposed method, shedding light on its effectiveness.

4.1. Alternating diffusion

To intuitively understand the efficacy of kernel alignment for the purpose of finding correspondences, consider the k-th iteration (without data term):

$$\max_{\mathbf{\Pi} \in \mathcal{P}_n} \langle \mathbf{\Pi}, \mathbf{K}_\mathcal{Y} \mathbf{\Pi}^k \mathbf{K}_\mathcal{X} \rangle \,. \tag{19}$$

Let us denote by $\boldsymbol{\delta}^j$ the discrete indicator function of vertex j on shape $\mathcal{X}$, representing initial heat distribution concentrated at vertex j. This heat is propagated via the application of the heat kernel $\mathbf{K}_\mathcal{X}$ to the rest of the vertices, resulting in the new heat distribution on $\mathcal{X}$ given by $\mathbf{k}_\mathcal{X}^j = \mathbf{K}_\mathcal{X} \boldsymbol{\delta}^j$. This heat distribution, whose spread depends on the time

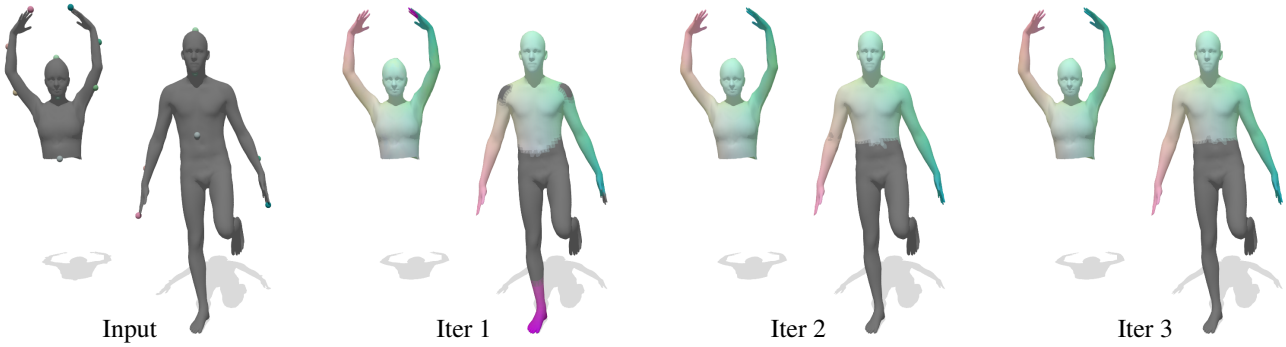

Figure 5: Our approach can tackle the challenging scenario of partial correspondences. As a proof of concept we initialized our method with sparse correspondences, indicated by spheres. We simulated noise by mapping a point on the left hand of the woman to the right foot of the man. At the first iteration all points spread their information, leading to a discontinuity of the mapping at the hand of the woman. After three iterations the method converged to the correct solution. This example was generated with Gaussian kernels. The proper choice of boundary conditions when using heat kernels will be discussed in future work.

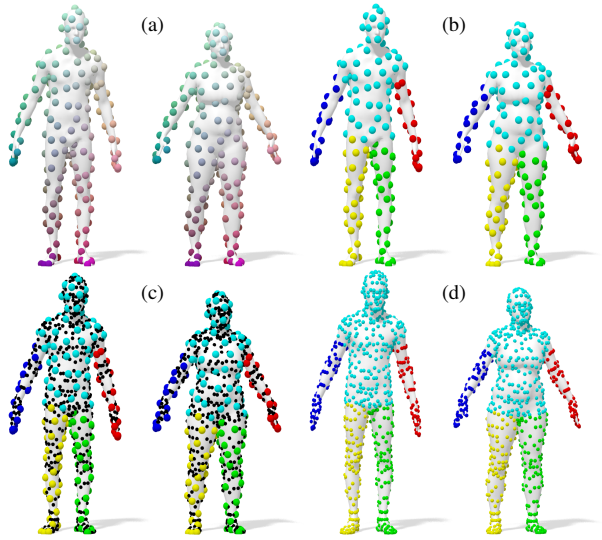

Figure 6: **Conceptual illustration of our multiscale approach:** (a) correspondence at a coarse scale is given; (b) vertices on the source shape are grouped into sets (left), and the known correspondence is used to group vertices on the target shape (right); (c) vertices at a finer scale are added and (d) included in the group they reside in; finally, a correspondence is calculated for each group separately.

parameter t, is mapped via $\mathbf{\Pi}^k$ onto the shape $\mathcal{Y}$, where it is propagated via the heat kernel $\mathbf{K}_\mathcal{Y}$. The ij-th element of the matrix $\mathbf{K}_\mathcal{Y}\mathbf{\Pi}^k\mathbf{K}_\mathcal{X}$,

$$\begin{aligned}(\mathbf{K}_\mathcal{Y}\mathbf{\Pi}^k\mathbf{K}_\mathcal{X})_{ij} &= (\mathbf{k}^i_\mathcal{Y})^\top\mathbf{\Pi}^k\mathbf{k}^j_\mathcal{X} \\ &= \sum_m (\mathbf{K}_\mathcal{Y})_{i,\pi^k(m)}(\mathbf{K}_\mathcal{X})_{jm}, \end{aligned} \tag{20}$$

represents the probability of a point i on $\mathcal{Y}$ being in correspondence with the point j on $\mathcal{X}$. This is affected by both the distance between i and $\pi^k(m)$ on $\mathcal{Y}$ for every m on $\mathcal{X}$, encoded in the entries of $(\mathbf{K}_\mathcal{Y})_{i,\pi^k(m)}$, and by the distance between m and j on $\mathcal{X}$, encoded in the entries of $(\mathbf{K}_\mathcal{X})_{jm}$. This process, as illustrated in Figure 7, resembles the alternating diffusion process described in [26]. Its success in uncovering the latent correspondence is based on the following statistical assumptions on the distribution of correspondences in the initial assignment: we tacitly assume that a sufficiently large number of (uniformly distributed) points are initially mapped correctly while the rest are mapped randomly, such that when averaging over their "votes" they do not bias towards any particular candidate. These concepts will be presented more rigorously in a longer version of this paper.

There is an inherent trade-off between the stability of the process and its accuracy, controlled by the time parameter t. Smaller t enables more accurate correspondence, but limits the ability of far away points to compensate for local inaccuracies in the initial correspondence, while larger t allows information to propagate from farther away, but introduces ambiguity at the fine scale. Examining the extremities, when $t \to 0$ each point is discouraged to change its initial match, while as $t \to \infty$ every point becomes a likely candidate for a match. In practice, we approximately solve a series of problems parametrized by a decreasing sequence of t values, as explained in the experimental section.

4.2. Iterated blurring and sharpening

An alternative point of view is to recall that a diffusion process corresponds to a smoothing operation, or low-pass filtering in the spectral domain. To that end we view each iteration (15) as an application of a series of low-pass filters (smoothing) followed by a projection operation (deblurring/sharpening). To see that, we use the spectral decompo-

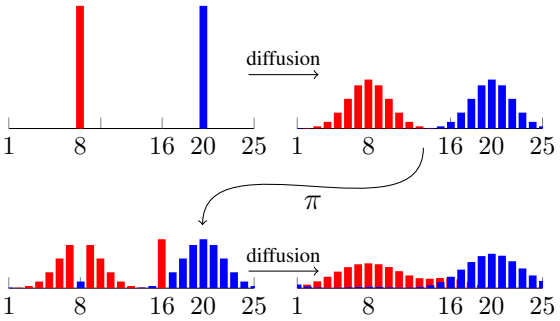

Figure 7: **Illustration of the alternating diffusion process** initialized with a noisy correspondence that wrongly maps $\pi(8) = 16$ and $\pi(16) = 8$ but correctly maps $\pi(x) = x$ elsewhere. Top left: Indicator functions on the source shape, one on a point with a wrong correspondence (red) and one with a correct correspondence (blue). Top right: Both indicator functions are diffused. Bottom left: The diffused functions are transported to the target shape via π. Bottom right: Diffusion on the target shape.

sition of the heat kernels to rewrite the payoff matrix in (15)

$$\begin{aligned}\mathbf{K}_{\mathcal{Y}}\mathbf{\Pi}\mathbf{K}_{\mathcal{X}} &= \mathbf{\Psi}e^{t\mathbf{\Lambda}_{\mathcal{Y}}}\mathbf{\Psi}^{\top}\mathbf{\Pi}\mathbf{\Phi}e^{t\mathbf{\Lambda}_{\mathcal{X}}}\mathbf{\Phi}^{\top} \\ &= \mathbf{\Psi}e^{t\mathbf{\Lambda}_{\mathcal{Y}}}\mathbf{C}e^{t\mathbf{\Lambda}_{\mathcal{X}}}\mathbf{\Phi}^{\top}. \end{aligned} \tag{21}$$

where the functional map $\mathbf{C}$ is seen as a low-pass approximation of the permutation matrix in the truncated Laplacian eigenbasis, $\mathbf{\Pi} \approx \mathbf{\Psi}\mathbf{C}\mathbf{\Phi}^{\top}$. Equation (21) can thus be interpreted as applying a low-pass filter to the functional map matrix $\mathbf{C}$. The second step in (15) can be regarded as a projection of the smoothed correspondence on the set of permutations (10), producing a point-wise bijection.

4.3. Kernel density estimation in the product space

Similar to the interpretation in [53], our approach can be seen as estimating the graph $\Pi = \{(x, \pi(x)) : x \in \mathcal{X}\}$ of the latent correspondence $\pi : \mathcal{X} \to \mathcal{Y}$ on the product manifold $\mathcal{X} \times \mathcal{Y}$. In case of a bijective, continuous π, the graph Π is a submanifold without a boundary of same dimension as $\mathcal{X}$ (2 in the discussed case). In each iteration of the process a probability distribution $P : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ is constructed by placing kernels (geodesic Gaussian kernels in [53], and heat kernels in our case) on the graph of the previous iterate and maximizing

$$\hat{\pi} = \arg \max_{\hat{\pi}:\mathcal{X}\overset{1:1}{\to}\mathcal{Y}} \int_{\mathcal{X}} P(x, \hat{\pi}(x))dx \tag{22}$$

over the set of bijective but not necessarily continuous correspondences.

5. Experiments

We performed an extensive quantitative evaluation of the proposed method on four different benchmarks. All datasets include several classes of (nearly) isometric shapes, with the last one additionally introducing strong topological noise (i.e., mesh 'gluing' in areas of contact). In our experiments we used the SHOT [51] and heat kernel signature (HKS) [49] descriptors with default parameters. For the computation of heat kernels we used 500 Laplacian eigenfunctions. We provide comparisons with complete matching pipelines as well as with learning-based approaches, where we show how using our method as a post-processing step leads to a significant boost in performance. In addition Figure 3 provides runtime comparison against [53] which uses a similar method with geodesic Gaussian kernels. Code of our method is available at `https://github.com/zorah/KernelMatching`.

Error measure. We measure correspondence quality according to the Princeton benchmark protocol [22]. Assume to be given a match $(x, y) \in \mathcal{X} \times \mathcal{Y}$, whereas the ground-truth correspondence is (x, y^*). Then, we measure the geodesic error $\epsilon(x) = d_{\mathcal{Y}}(y, y^*)/\mathrm{diam}(\mathcal{Y})$ normalized by the geodesic diameter of $\mathcal{Y}$. Ideal correspondence should produce $\epsilon = 0$. We plot cumulative curves showing the percentage of matches that have error smaller than a variable threshold.

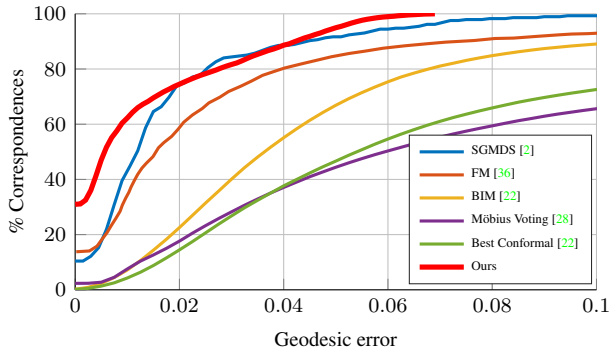

Figure 8: Correspondence accuracy on the SCAPE dataset.

Parameters. The optimal choice of parameters does not only depend on properties of the considered shapes (such as diameterand density of the sampling) but also on the noise of the input correspondence. The exact dependencies in particular on the latter will be investigated in follow up works.

TOSCA. The TOSCA dataset [14] contains 76 shapes divided into 8 classes (humans and animals) of varying resolution (3K to 50K vertices). We match each shape with one instance of the same class. For shapes having more than 10K vertices we use our multiscale acceleration with an initial problem size of 10K and a maximum problem size of 3K for all further iterations. The parameters were set to $\alpha = 10^{-10}$ and $t = [300\ 100\ 50\ 10]$, with 5 iterations per diffusion time. Figure 9 shows a quantitative evaluation.

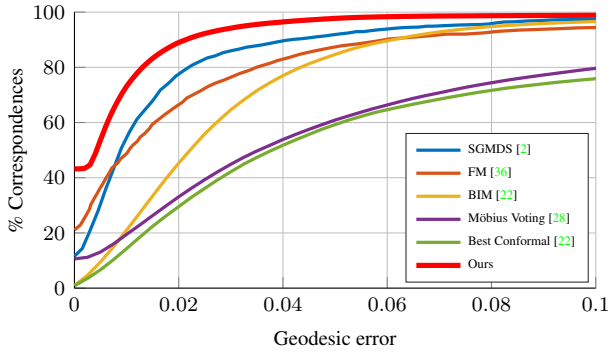

Figure 9: Correspondence accuracy on the TOSCA dataset.

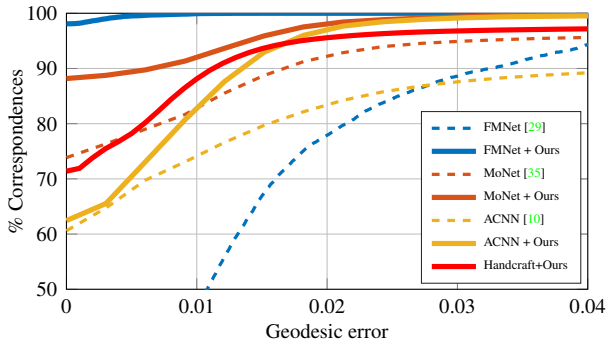

Figure 10: Correspondence accuracy on FAUST. Dashed curves indicate the performance of recent deep learning methods, solid curves are obtained using our method as post-processing. Our method based on handcrafted descriptors (SHOT) is denoted as 'Handcrafted+Ours'.

SCAPE. The SCAPE dataset [3] contains 72 clean shapes of scanned humans in different poses. For this test we set $\alpha = 10^{-7}$, $t = [0.1\ 0.05\ 0.009\ 0.001\ 0.0001]$, and 5 iterations per diffusion time. We used multiscale acceleration with initial size equal to 10K vertices, and equal to 1K for subsequent iterations. Quantitative and qualitative results are given in Figure 8 and 1 (right) respectively.

FAUST. The FAUST dataset [7] contains 100 human scans belonging to 10 different individuals; for these tests we used the template subset of FAUST, consisting of shapes with around 7K vertices each. This allowed us to run our algorithm without multiscale acceleration. We set $\alpha = 10^{-7}$ and $t = [500\ 323\ 209\ 135\ 87\ 36\ 23\ 15\ 10]$. Differently from the previous experiments, here we employ our method as a refinement step for several deep learning-based methods, demonstrating significant improvements (up to 50%) upon the 'raw' output of such approaches. The results are reported in Figure 10. Our results contain a few shapes in which body parts were swapped, preventing us from reaching 100%. An example is presented in the supp. material.

SHREC'16 Topology. This dataset [25] contains 25 shapes of the same class with around 12K vertices, undergoing near-isometric deformations in addition to large topological shortcuts (see Figure 1 middle). Here we use only SHOT as a descriptor, since HKS is not robust against topological changes. We used $\alpha = 10^{-6}$ and $t = [2.7\ 2.44\ 2.1\ 1.95\ 1.7]$, using multiscale with an initial problem of size 12k and the following problems with maximum size 1k. Quantitative results are reported in Figure 11.

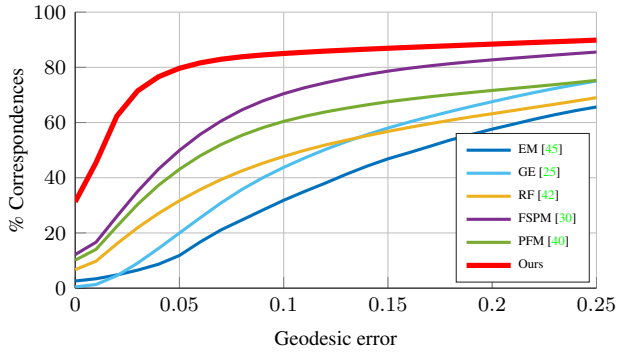

Figure 11: Correspondence accuracy on SHREC'16 Topology.

6. Conclusions

We considered a formulation of the problem of finding a smooth, possibly partial, correspondence between two non-isometric shapes as a quadratic assignment problem matching between point-wise and pair-wise descriptors. We showed that when choosing the pair-wise descriptors to be positive-definite kernel matrices (unlike the traditionally used distance matrices), the NP-hard QAP admits an exact relaxation over the space of bistochastic matrices, which we proposed to solve using a projected descent procedure motivated by the DC algorithm. The resulting iterations take the form of LAPs, which are solved using a multi-scale version of the auction algorithm. We interpreted the proposed algorithm as an alternating diffusion process, as iterated blurring and sharpening, and as a kernel density estimation procedure. The algorithm scales very well to even hundreds of thousands of vertices, and produces surprisingly good results. Experimental evaluation on various datasets shows that our method significantly improves the output obtained by the best existing correspondence methods.

Acknowledgements This work has been supported by the ERC grants 307047 (COMET), 335491 (RAPID), 649323 (3D Reloaded) and 724228 (LEMAN).

References

[1] Y. Aflalo, A. Bronstein, and R. Kimmel. On convex relaxation of graph isomorphism. *PNAS*, 112(10):2942–2947, 2015. 2

[2] Y. Aflalo, A. Dubrovina, and R. Kimmel. Spectral generalized multi-dimensional scaling. *IJCV*, 118(3):380–392, 2016. 7, 8

[3] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: shape completion and animation of people. *Proc. of ACM SIGGRAPH*, 2005. 8

[4] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Proc. 4DMOD*, 2011. 2

[5] D. P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, 1998. 3

[6] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992. 2

[7] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, 2014. 8

[8] E. Bogomolny, O. Bohigas, and C. Schmit. Spectral properties of distance matrices. *J. Physics A*, 36(12):3595, 2003. 4

[9] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum*, 34(5):13–23, 2015. 2, 3

[10] D. Boscaini, J. Masci, E. Rodolà, and M. M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Proc. NIPS*, 2016. 2, 3, 8

[11] D. Boscaini, J. Masci, E. Rodolà, M. M. Bronstein, and D. Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441, 2016. 2, 3

[12] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Efficient computation of isometry-invariant distances between surfaces. *SIAM J. Sci. Comp.*, 28(5):1812–1836, 2006. 2

[13] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006. 2, 3

[14] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer, 2008. 7

[15] A. M. Bronstein, M. M. Bronstein, R. Kimmel, M. Mahmoudi, and G. Sapiro. A Gromov-Hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *IJCV*, 89(2):266–286, 2010. 3

[16] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *Proc. CVPR*, 2010. 2

[17] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *Proc. ICCV*, pages 2039–2047, 2015. 2, 3

[18] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *PNAS*, 102(21):7426–7431, 2005. 2

[19] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *Trans. PAMI*, 25(10):1285–1295, 2003. 2

[20] N. Hu and L. Guibas. Spectral descriptors for graph matching. *arXiv preprint arXiv:1304.1572*, 2013. 2

[21] I. Kezurer, S. Z. Kovalsky, R. Basri, and Y. Lipman. Tight relaxation of quadratic matching. In *Computer Graphics Forum*, volume 34, pages 115–128, 2015. 2

[22] V. G. Kim, Y. Lipman, and T. A. Funkhouser. Blended intrinsic maps. *Trans. Graphics*, 30(4), 2011. 7, 8

[23] A. Kovnatsky, M. M. Bronstein, X. Bresson, and P. Vandergheynst. Functional correspondence by matrix completion. In *Proc. CVPR*, 2015. 2

[24] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. *Computer Graphics Forum*, 32(2):439–448, 2013. 2

[25] Z. Lähner, E. Rodolà, M. M. Bronstein, D. Cremers, O. Burghard, L. Cosmo, A. Dieckmann, R. Klein, and Y. Sahillioglu. SHREC'16: Matching of deformable shapes with topological noise. In *Proc. 3DOR*, 2016. 8

[26] R. R. Lederman and R. Talmon. Learning the geometry of common latent variables using alternating-diffusion. *App. and Comp. Harmonic Analysis*, 2015. 2, 6

[27] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *Proc. ICCV*, 2005. 2, 4

[28] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. In *Trans. Graphics*, volume 28, page 72, 2009. 7, 8

[29] O. Litany, T. Remez, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proc. ICCV*, 2017. 2, 8

[30] O. Litany, E. Rodolà, A. Bronstein, and M. Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2):1681–1707, 2017. 8

[31] X. Liu, A. Donate, M. Jemison, and W. Mio. Kernel functions for robust 3d surface registration with spectral embeddings. In *ICPR*, pages 1–4, 2008. 3

[32] J. Masci, D. Boscaini, M. M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on Riemannian manifolds. In *Proc. 3DRR*, 2015. 2, 3

[33] D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *Proc. CVPR*, 2008. 2

[34] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. *Foundations of Computational Mathematics*, 5(3):313–347, 2005. 2

[35] F. Monti, D. Boscaini, J. Masci, E. Rodolà, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Proc. CVPR*, 2017. 2, 3, 8

[36] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *Trans. Graphics*, 31(4):30, 2012. 2, 4, 7, 8

[37] P. M. Pardalos, H. Wolkowicz, et al. *Quadratic Assignment and Related Problems: DIMACS Workshop, May 20-21, 1993*, volume 16. American Mathematical Soc., 1994. 2

[38] D. Raviv, A. M. Bronstein, M. M. Bronstein, R. Kimmel, and N. Sochen. Affine-invariant geodesic geometry of deformable 3d shapes. *Computers & Graphics*, 35(3):692–697, 2011. 3

[39] E. Rodolà, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *Proc. CVPR*, 2012. 2

[40] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017. 8

[41] E. Rodolà, M. Moeller, and D. Cremers. Point-wise map recovery and refinement from functional correspondence. In *Proc. VMV*, pages 25–32, Aachen, Germany, 2015. Eurographics Association. 4

[42] E. Rodolà, S. Rota Bulò, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *Proc. CVPR*, 2014. 2, 8

[43] R. M. Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP*, 2007. 2

[44] Y. Sahillioglu and Y. Yemez. Coarse-to-fine combinatorial matching for dense isometric shape correspondence. In *Computer Graphics Forum*, volume 30, pages 1461–1470, 2011. 2

[45] Y. Sahillioğlu and Y. Yemez. Minimum-distortion isometric shape correspondence using EM algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2203–2215, 2012. 8

[46] G. Shamai and R. Kimmel. Geodesic distance descriptors. *CoRR*, abs/1611.07360, 2016. 2

[47] A. Shtern and R. Kimmel. Iterative closest spectral kernel maps. In *Proc. 3DV*, volume 1, pages 499–505. IEEE, 2014. 3

[48] A. Shtern and R. Kimmel. Matching the lbo eigenspace of non-rigid shapes via high order statistics. *Axioms*, 3(3):300–319, 2014. 2

[49] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum*, volume 28, pages 1383–1392, 2009. 2, 7

[50] A. Tevs, A. Berner, M. Wand, I. Ihrke, and H.-P. Seidel. Intrinsic shape matching by planned landmark sampling. In *Computer Graphics Forum*, volume 30, pages 543–552, 2011. 2

[51] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *Proc. ECCV*, 2010. 2, 7

[52] L. Torresani, V. Kolmogorov, and C. Rother. Feature correspondence via graph matching: Models and global optimization. In *Proc. ECCV*, 2008. 2

[53] M. Vestner, R. Litman, E. Rodolà, A. Bronstein, and D. Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. *Proc. CVPR*, 2017. 2, 3, 4, 7

[54] C. Wang, M. M. Bronstein, A. M. Bronstein, and N. Paragios. Discrete minimum distortion correspondence problems for non-rigid shape matching. In *Proc. SSVM*, 2011. 2

[55] T. Windheuser, U. Schlickewei, F. R. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *Proc. ICCV*, 2011. 2

Supplementary Material

A. Details about the DC algorithm

In Section 3 of the submission we propose to use the DC algorithm to optimize

$$\arg\min_{\mathbf{P}\in\mathbb{R}^{n\times n}} B(\mathbf{P}) - E(\mathbf{P}). \tag{23}$$

where B is the convex indicator function of the set of bistochastic matrices $\mathcal{B}_n$ and E is strictly convex and differentiable. We will now prove that the two steps

- Select $\mathbf{Q}^k \in \partial E(\mathbf{P}^k)$
- Select $\mathbf{P}^{k+1} \in \partial B^*(\mathbf{Q}^k)$.

of the DC algorithm are equivalent to

$$\mathbf{P}^{k+1} = \operatorname{argmax}_{\mathbf{P}\in\mathcal{B}_n} \langle \mathbf{P}, \nabla E(\mathbf{P}^k)\rangle\,, \tag{24}$$

that each iterate $\mathbf{P}^k$ can be chosen to be a permutation matrix, and that $E(\mathbf{P}^k)$ is a strictly increasing.

We assume that the reader is familiar with the concepts of convex conjugates and sub-gradients and just recall the following Lemma

Lemma 2. *Let X be a Banach space and $f : X \to (-\infty, \infty]$ with $\partial f \neq \emptyset$. Then $f^{**}(x) = f(x)$ and*

$$x^* \in \partial f(x) \Leftrightarrow x \in \partial f^*(x^*) \tag{25}$$

Moreover for convex functions f, $0 \in \partial f(x)$ is equivalent to

$$x = \operatorname{argmin}_x f(x) \tag{26}$$

Let now E be convex differentiable and B the (convex) indicator function of a convex set C. We will derive equivalent expressions for the two steps in the DC algorithm for solving (23). Since E is differentiable, its subdifferential at any point has one element, namely the gradient at that point:

$$\mathbf{Q}^k \in \partial E(\mathbf{P}^k) \Leftrightarrow \mathbf{Q}^k = \nabla E(\mathbf{P}^k) \tag{27}$$

The second step $\mathbf{P}^{k+1} \in \partial B^*(\mathbf{Q}^k)$ can be rewritten using Lemma 2:

$$\begin{aligned}\mathbf{P}^{k+1} \in \partial B^*(\mathbf{Q}^k) &\Leftrightarrow \mathbf{Q}^k \in \partial B(\mathbf{P}^{k+1})\\ &\Leftrightarrow 0 \in -\mathbf{Q}^k + \partial B(\mathbf{P}^{k+1})\\ &\Leftrightarrow \mathbf{P}^{k+1} = \operatorname{argmin}_{\mathbf{P}} -\langle \mathbf{Q}^k, \mathbf{P}\rangle + B(x)\\ &\Leftrightarrow \mathbf{P}^{k+1} = \operatorname{argmax}_{\mathbf{P}\in C} \langle \mathbf{Q}^k, \mathbf{P}\rangle \end{aligned} \tag{28}$$

Thus the DC algorithm in this special case reads

$$\mathbf{P}^{k+1} = \operatorname{argmax}_{\mathbf{P}\in C} \langle \mathbf{P}, \nabla E(\mathbf{P}^k)\rangle\,. \tag{29}$$

In our case the convex set C is the polyhedron $\mathcal{B}_n$ of bistochastic matrices. Since linear functions defined on a polyhedron attain their extrema at the vertices of the polyhedron, we can choose the maximizer to be a permutation matrix.

Due to the strict convexity of E we further see:

$$\begin{aligned} E(\mathbf{P}^{k+1}) &> E(\mathbf{P}^k) + \langle \mathbf{P}^{k+1} - \mathbf{P}^k, \nabla E(\mathbf{P}^k)\rangle\\ &\geq E(\mathbf{P}^k) + \langle \mathbf{P}^k - \mathbf{P}^k, \nabla E(\mathbf{P}^k)\rangle\\ &= E(\mathbf{P}^k) \end{aligned} \tag{30}$$

where the strong inequality holds until convergence and the weak inequality follows directly from (29).

B. Details on multiscale acceleration

The multiscale algorithm begins by solving for an initial sparse bijection $\pi_0 : \mathcal{X}_0 \to \mathcal{Y}_0$ between n_0 samples $s_\mathcal{X}, s_\mathcal{Y}$ (also called *seeds*), obtained with farthest point sampling (Euclidean in our experiments). n_0 can either be the maximum amount of vertices that can be handled (around 15k in our experiments) or smaller if runtime is crucial. Then $s_\mathcal{X}$ is divided into $n_0/(k \cdot maxP)$ Voronoi cells $V_{\mathcal{X},0}$ and these Voronoi cells are transferred to $\mathcal{Y}$ using π_0 to create $V_{\mathcal{Y},0}$. The parameter $maxP$ is the maximum problem size allowed in later iterations and normally much smaller than n_0. k determines how many new samples are added in each iteration. A small $maxP$ makes the method faster but less robust, and a small k slower but more robust. In our ex-

Figure 12: A failure case of our method. Left and right are switched on the upper body, causing a non-continuous correspondence. We observed eight such failure cases in the entire FAUST dataset.

periments, we always choose $maxP = 1500$ and $k = 3$. At the first iteration ($i = 1$) and any following iteration i, $n_i = k \times n_{i-1}$ new points are sampled in a farthest point manner on both shapes to create $\mathcal{X}_i, \mathcal{Y}_i$. Each new point is assigned to the same Voronoi cell as its nearest neighbor in $s_\mathcal{X}, s_\mathcal{Y}$ resulting in the new cells $V_{\mathcal{X},i}, V_{\mathcal{Y},i}$. If any cell has more than $maxP$ vertices, the number of cells is increased until this is not the case anymore. Next we solve for $\pi_i : \mathcal{X}_i \to \mathcal{Y}_i$ by solving for a mapping from the m-th cell of $V_{\mathcal{X},i}$ to the m-th of $V_{\mathcal{X},i}$ using the proposed method from this paper and combining them into a global permutation. Notice that the m-th cells of both shapes correspond to roughly the same areas as long as the previous matching π_{i-1} that was used for its construction is reasonable. Nevertheless, the cells could include a different amount of points due to discretization errors, so we need to apply the partial matching scheme for each cell and some points may stay unmatched (in this iteration). All matched points are added to the sets $s_\mathcal{X}, s_\mathcal{Y}$ for the next iteration. Again, $\mathcal{X}$ is divided into $n_i/(k \cdot maxP)$ Voronoi cells and these are transfered to $\mathcal{Y}$ via π_i. The Voronoi cells of previous iterations are discarded to allow exchange of points between cells. This proceeds until all points have been sampled.

We use Euclidean FPS in all cases and build approximate Voronoi cells on remeshed versions of the shape to keep the runtime small. Each π_i is solved for by using descriptors and initial matches from the previous iteration in the same cell. Additionally, we add 1000 equally distributed matches from π_0 to every problem which aligns the solution along the boundaries of the cells with each other. Notice that even if the shapes have the same number of vertices at the beginning due to the sampling and decoupling of each cell not all vertices might be matched.

If the matched shapes are partial versions of each other, this information needs to be propagated from the first iteration on since all later cells are solved independently and can therefore not see partiality. In this case, $n_{0,\mathcal{X}}, n_{0,\mathcal{Y}}$ can be chosen dependently on the ratio of areas or number of vertices between $\mathcal{X}$ and $\mathcal{Y}$, either assuming the scale or the discretization is comparable. Then certain points of the initial sampling will stay unmatched and be marked *forbidden*. They are handled exactly like any other seed but have their own Voronoi cell and any point that gets a assigned to the forbidden Voronoi cell is also marked forbidden such that the information spreads only to the neighborhood.

C. Run time comparison

The run time experiments, were conducted on a MacBook pro with a 2.5 GHz Intel Core i7 processor and 16 GB RAM running Matlab 2016*b*. The experiments were conducted using 9 pairs of shapes with a varying number of vertices from the TOSCA high and low resolution meshes as well as FAUST registrations set. The complete results are presented in Table 1. We ran all our tests using SHOT descriptors, 10 iterations with $\alpha = 1/10^8$, 400 eigenvectors to construct the heat kernels and a logarithmic scale of time parameters between 400 and 10.

Figure 13: Matching from a horse to an elephant using SHOT and HKS descriptors. The shapes are sampled in a way such that a bijective matching is possible.

D. More results, Failure cases

In this section we show additional results for a pair of dramatically non-isometric shapes (Fig.13), pairs from the Tosca dataset (Fig.14) and failure cases (Figs.12,15).

shapes in experiment	#vertices	runtime with heat kernel in sec	runtime with Gaussian kernel in sec
Tosca: cat0 to cat2	3400	29.25	97.77
Tosca: dog0 to dog2	3400	36	98.43
Tosca: centaur0 to centaur1	3400	25.31	98.91
Tosca: wolf0 to wolf1	4344	60.7	192.72
Faust models: 000 to 098	6890	109.477019	639.9
Faust models: 001 to 031	6890	104.68	609.56
Faust models: 002 to 039	6890	104.5	611.24
Faust models: 003 to 021	6890	106.41	614.23
Faust models: 004 to 033	6890	106.28	652.58

Table 1: Runtime comparison of matching between shapes with different number of vertices using heat kernels and Gaussian kernels.

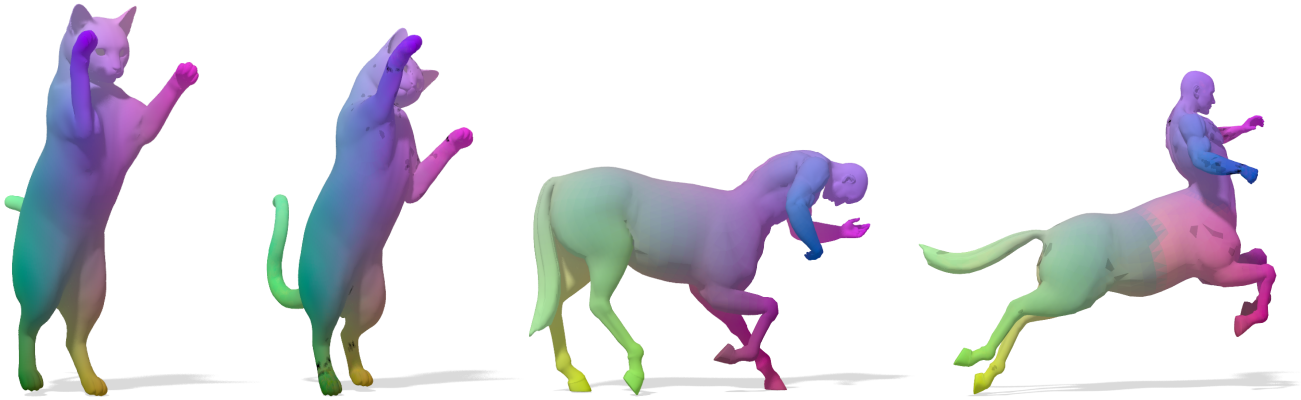

Figure 14: (left) A matching between two cats from the Tosca dataset. The unmatched points resulting from the multiscale (black) are very sparse. (right) A failed matching on the centaurs from Tosca. The front legs are swapped but only few points are unmatched.

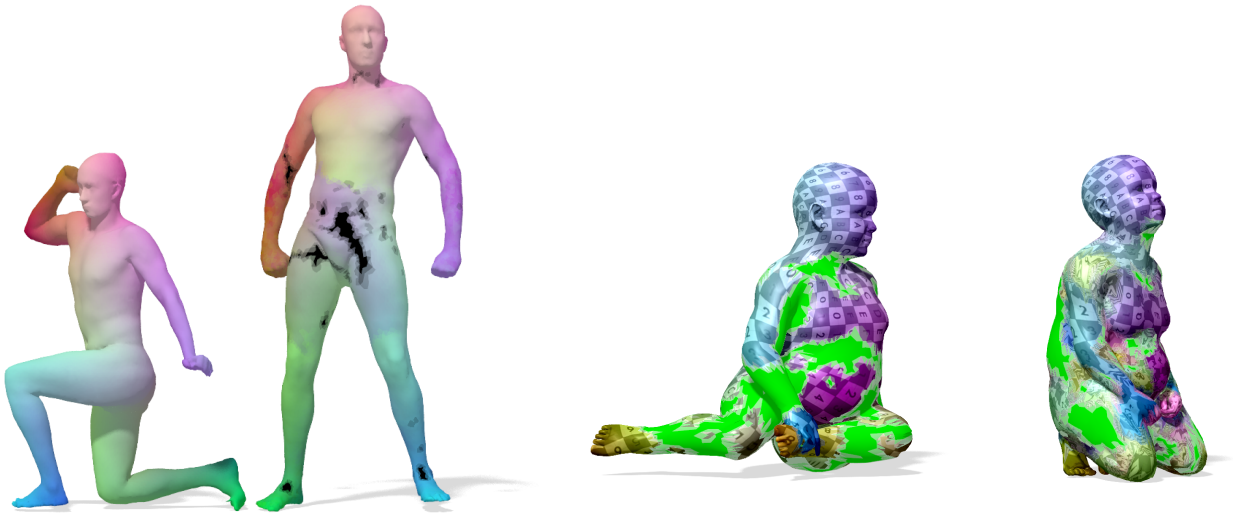

Figure 15: (left) Failure case on the SCAPE dataset.The legs are mapped front to back causing a non continuous correspondence on the torso. Large unmatched areas due to the multiscale also appear there. Over the knees unaligned cell boundaries are visible. (right) Failure case on the SHREC'16 dataset. Many parts are missing or the texture is heavily distorted. These are really challenging shapes to match because the hands and feet are topologically merged to different parts of the body in both cases.

Part III

Conclusion and Outlook

Chapter 9

Summary

In this thesis we provided a comprehensive summary about the theory of non-rigid shape analysis and correspondence. We covered fundamentals of Differential Geometry and Triangle Meshes in Chapter 3. In Chapter 4 the Laplace Beltrami operator was introduced in the continuous and the discrete setting. Chapter 5 was about correspondences between manifolds and triangle meshes. Euclidean and non-Euclidean isometries were discussed and permutation matrices as discrete counterparts to bijections were introduced. We elaborated on pointwise and pairwise descriptors and how they lead to combinatorial optimization problems, so called *linear assignment problems* and *quadratic assignment problems*. We pointed out the computational challenges of combinatorial optimization problems and discussed different types of relaxations, *e.g.* bistochastic matrices and *functional maps*. We further showed how the graph of a correspondence can be seen as a submanifold of the so called product manifold of two shapes and how this point of view can lead to a different type of optimization problems relating the regularity of a correspondence to the regularity of its graph.
In Chapter 6 we introduced a method to match non isometric shapes using a machine learning approach. We treated the correspondence problem as a classification task and exploited *random decision forests* to learn a class of deformations from data and use this knowledge to match shapes undergoing similar types of deformations.
In Chapter 7 we approached the correspondence problem via kernel density estimation in the product manifold. We showed how an iterative process of alternating blurring and sharpening of the graph of the correspondence leads to correspondences with increasing regularity. With a variety of input correspondences (sparse/noisy/fuzzy) this method gave high quality correspondences and with a multiscale approach we were able to apply it to high resolution meshes.
Finally Chapter 8 related the previous approach to quadratic assignment problems based on heat kernels. A variety of different interpretations of the iterative process were presented and theretical gurantees as well as practical extensions were provided.

Chapter 10

Future Work

In the following we would like to point out possible directions for future research as well as pointers to recent literature.

Deep learning based approaches While random decision forests was one of the leading machine learning techniques some years ago, nowadays deep neural networks achieve state of the art results in almost every aspect of Computer Vision and beyond (*e.g.* natural language processing). It is thus evident that also the field of 3D shape analysis and correspondence can benefit from deep learning based approaches. Among the first works that follow that approach are [32] and [36]. Modern approaches include [23], [18] and [49].

Model based approaches This thesis tackled the shape correspondence problem in its most generic form: given two arbitrary triangle meshes (possibly with consistent vertex density) the goal was to find a correspondence without any domain knowledge. However in many scenarios the class of objects of interest are known a priory. In human body tracking, for instance, it is clear that the considered meshes are representations of human bodies. Dedicates parametric models such as SMPL [34] and its extensions [37, 40] can be more suitable in these scenarios. Fitting those models to a given triangle mesh can be phrased as a comparibly simple optimization problem over a relatively small set of parameters. In [39] the authors learn parametric human body models using deep learning techniques. Similar approaches are of course possible for classes of objects other than humans.

Own Publications

[1] E. Rodolà, S. Rota Bulo, T. Windheuser, M. Vestner, and D. Cremers. Dense non-rigid shape correspondence using random forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4184, 2014.

[2] M. Vestner, Z. Lahner, A. Boyarski, O. Litany, R. Slossberg, T. Remez, E. Rodola, A. Bronstein, M. Bronstein, R. Kimmel, and D. Cremers. Efficient deformable shape correspondence via kernel matching. In *2017 International Conference on 3D Vision (3DV)*. IEEE, Oct. 2017. DOI: 10.1109/3Dv.2017.00065 (cited on pp. 6, 8, 59, 65).

[3] M. Vestner, R. Litman, E. Rodola, A. Bronstein, and D. Cremers. Product manifold filter: non-rigid shape correspondence via kernel density estimation in the product space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017. DOI: 10.1109/cvpr.2017.707 (cited on pp. 5, 8).

[4] M. Vestner, E. Rodolà, T. Windheuser, S. R. Bulò, and D. Cremers. Applying random forests to the problem of dense non-rigid shape correspondence. In M. Breuß, A. Bruckstein, P. Maragos, and S. Wuhrer, editors, *Perspectives in Shape Analysis*, pages 231–248, Cham. Springer International Publishing, 2016. ISBN: 978-3-319-24726-7 (cited on pp. 5, 8, 55).

[5] T. Windheuser, M. Vestner, E. Rodola, R. Triebel, and D. Cremers. Optimal intrinsic descriptors for non-rigid shape analysis. In *Proceedings of the British Machine Vision Conference 2014*. British Machine Vision Association, 2014. DOI: 10.5244/c.28.44.

Bibliography

[5] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on pattern analysis and machine intelligence*, (5):698–700, 1987 (cited on p. 42).

[6] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: a quantum mechanical approach to shape analysis. In *2011 IEEE international conference on Computer Vision workshops (ICCV workshops)*, pages 1626–1633. IEEE, 2011 (cited on pp. 5, 8, 55).

[7] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975 (cited on pp. 5, 43, 60).

[8] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981 (cited on pp. 53, 60).

[9] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. DOI: 10.1109/34.121791 (cited on p. 42).

[10] A. I. Bobenko and B. A. Springborn. A discrete laplace–beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4):740–756, 2007 (cited on p. 37).

[11] F. Bogo, J. Romero, M. Loper, and M. J. Black. Faust: dataset and evaluation for 3d mesh registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3794–3801, 2014 (cited on pp. 4, 5, 16, 57, 64).

[12] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon mesh processing*. AK Peters/CRC Press, 2010 (cited on p. 9).

[13] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5):1168–1172, 2006 (cited on p. 47).

[14] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008 (cited on pp. 32, 51, 58).

[15] L. Cosmo, M. Panine, A. Rampini, M. Ovsjanikov, M. M. Bronstein, and E. Rodola. Isospectralization, or how to hear shape, style, and correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7529–7538, 2019 (cited on p. 50).

[16] P. A. M. Dirac. *The principles of quantum mechanics*, number 27. Oxford university press, 1981 (cited on p. 39).

[17] M. P. Do Carmo. *Differential Geometry of Curves and Surfaces: Revised and Updated Second Edition*. Courier Dover Publications, 2016 (cited on pp. 9, 17).

[18] N. Donati, A. Sharma, and M. Ovsjanikov. Deep geometric functional maps: robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8592–8601, 2020 (cited on pp. 41, 127).

[19] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi. The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing*, 6(9):1305–1315, 1997 (cited on p. 51).

[20] L. C. Evans. *Partial Differential Equations* -. American Mathematical Soc., Heidelberg, 2. aufl. Edition, 2010. ISBN: 978-0-821-84974-3 (cited on pp. 24–26, 28).

[21] D. Ezuz and M. Ben-Chen. Deblurring and denoising of maps between shapes. In *Computer Graphics Forum*, volume 36 of number 5, pages 165–174. Wiley Online Library, 2017 (cited on p. 51).

[22] A. Grigor'yan. Heat kernels on weighted manifolds and applications. *Cont. Math*, 398(2006):93–191, 2006 (cited on p. 39).

[23] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019 (cited on p. 127).

[24] M. Kac. Can one hear the shape of a drum? *The american mathematical monthly*, 73(4P2):1–23, 1966 (cited on p. 50).

[25] L. V. Kantorovich. On the translocation of masses. In *Dokl. Akad. Nauk. USSR (NS)*, volume 37, pages 199–201, 1942 (cited on p. 63).

[26] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955 (cited on pp. 53, 60).

[27] Z. Lahner, E. Rodola, F. R. Schmidt, M. M. Bronstein, and D. Cremers. Efficient globally optimal 2d-to-3d deformable shape matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2185–2193, 2016 (cited on p. 69).

[28] H. Lavenant, S. Claici, E. Chien, and J. Solomon. Dynamical optimal transport on discrete surfaces. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018 (cited on p. 63).

[29] E. L. Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963 (cited on p. 59).

[30] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. SIAM, 1998 (cited on p. 36).

[31] B. M. Levitan. On the asymptotic behavior of the spectral function of a self-adjoint differential equation of the second order. *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya*, 16(4):325–352, 1952 (cited on p. 33).

[32] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein. Deep functional maps: structured prediction for dense shape correspondence. In *Proceedings of the IEEE international conference on Computer Vision*, pages 5659–5667, 2017 (cited on pp. 41, 127).

[33] R. Litman and A. M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2013 (cited on p. 55).

[34] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: a skinned multi-person linear model. *ACM transactions on graphics (TOG)*, 34(6):1–16, 2015 (cited on p. 127).

[35] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003 (cited on p. 37).

[36] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017 (cited on p. 127).

[37] A. A. A. Osman, T. Bolkart, and M. J. Black. STAR: a sparse trained articulated human body regressor. In *European Conference on Computer Vision (ECCV)*, pages 598–613, 2020 (cited on p. 127).

[38] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):1–11, 2012 (cited on p. 64).

[39] P. Palafox, A. Božič, J. Thies, M. Nießner, and A. Dai. Npms: neural parametric models for 3d deformable shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12695–12705, 2021 (cited on p. 127).

[40] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. Osman, D. Tzionas, and M. J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE/CVF conference on Computer Vision and pattern recognition*, pages 10975–10985, 2019 (cited on p. 127).

[41] G. Peyre and M. Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019 (cited on p. 63).

[42] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete laplace–beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009 (cited on p. 37).

[43] W. Ritz. Über eine neue methode zur losung gewisser variationsprobleme der mathematischen physik. *Journal fur Mathematik*, 135:s–1, 1909 (cited on p. 33).

[44] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36 of number 1, pages 222–236. Wiley Online Library, 2017 (cited on p. 67).

[45] E. Rodolà, Z. Lähner, A. M. Bronstein, M. M. Bronstein, and J. Solomon. Functional maps representation on product manifolds. In *Computer Graphics Forum*, volume 38 of number 1, pages 678–689. Wiley Online Library, 2019 (cited on p. 65).

[46] R. M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233. Eurographics Association, 2007 (cited on p. 54).

[47] F. R. Schmidt, D. Farin, and D. Cremers. Fast matching of planar shapes in sub-cubic runtime. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–6. IEEE, 2007 (cited on pp. 68, 69).

[48] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998 (cited on p. 69).

[49] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. Diffusionnet: discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022 (cited on p. 127).

[50] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional wasserstein distances: efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015 (cited on p. 63).

[51] G. W. Stewart. A krylov–schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002 (cited on p. 36).

[52] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28 of number 5, pages 1383–1392. Wiley Online Library, 2009 (cited on pp. 5, 8, 54).

[53] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In *European conference on Computer Vision*, pages 356–369. Springer, 2010 (cited on pp. 5, 8, 55).

[54] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun. Discrete laplace operators: no free lunch. In *Symposium on Geometry processing*, pages 33–37. Aire-la-Ville, Switzerland, 2007 (cited on p. 37).

[55] H. Weyl. Über die asymptotische verteilung der eigenwerte. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1911:110–117, 1911 (cited on p. 33).

[56] T. Windheuser and D. Cremers. A convex solution to spatially-regularized correspondence problems. In *European Conference on Computer Vision*, pages 853–868. Springer, 2016 (cited on p. 69).

[57] T. Windheuser, U. Schlickewei, F. R. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3d shapes: a linear programming solution. In *2011 International Conference on Computer Vision*, pages 2134–2141. IEEE, 2011 (cited on p. 5).