University of Verona

Department of Computer Science
Graduate School of Natural Sciences and Engineering
PhD in Computer Science

Merging, extending and learning representations for 3D shape matching.

Riccardo Marin

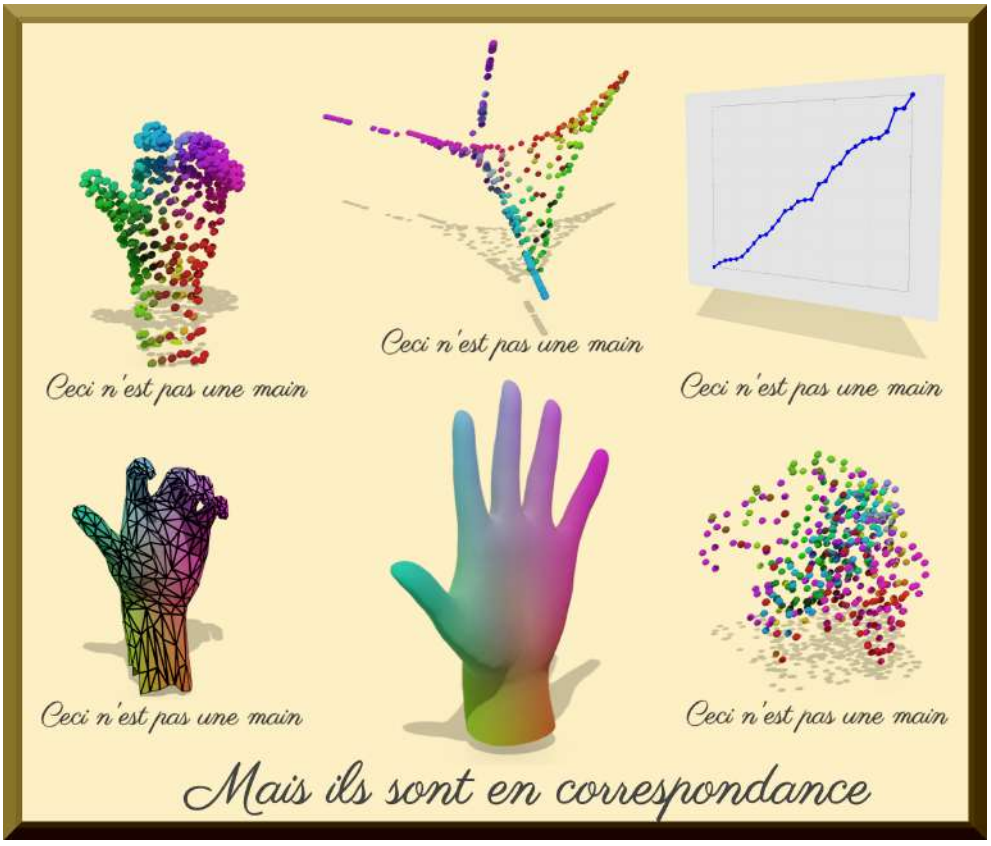
Advisor: Prof. Umberto Castellani

INF/01, XXXIII cycle, 2020

Ph.D. Thesis
Advisor:
prof. U. Castellani

University of Verona
Department of Computer Science
Graduate School of Natural Sciences and Engineering
PhD in Computer Science
Cycle XXXIII
Strada le Grazie 15 Verona
Italy

To those who thought I had a chance


Ceci n'est pas une main
Ceci n'est pas une main
Ceci n'est pas une main
Ceci n'est pas une main
Ceci n'est pas une main
Mais ils sont en correspondance

Abstract

In the last decades, researchers devoted considerable attention to shape matching. Correlating surfaces unlocks otherwise impossible applications and analysis. However, non-rigid objects (like humans) have an enormous range of possibilities to deform their surfaces, making the correspondence challenging to obtain. Computer Graphics and Vision has developed many different representations, each with its peculiarities, conveying different properties and easing different tasks. In this thesis, we exploit, extend, and propose representations to establish correspondences in the non-rigid domain. First, we show how the latent representation of a morphable model can be combined with the spectral embedding, acting as regularization of registration pipelines. We fill the gap in unconstrained problems like occlusion in RGB+D single view or partiality and topological noise for 3D representations. Furthermore, we define a strategy to densify the morphable model discretization and catch variable quantities of details. We also analyze how different discretizations impact correspondence computation. Therefore, we combine intrinsic and extrinsic embeddings, obtaining a robust representation that lets us transfer triangulation among the shapes. Data-driven techniques are particularly relevant to catch complex priors. Hence, we use deep learning techniques to obtain a new high-dimensional embedding for point clouds; in this representation, the objects align with a linear transformation. This approach shows resilience to sparsity and noise. Finally, we connect super-compact latent representations by linking autoencoder latent codes with Laplace-Beltrami operator spectra. This strategy lets us solving a complicated historical problem, enriching the learning framework with geometric properties, and matching objects regardless of their representations. The main contributions of this thesis are the theoretical and practical studies of representations, the advancement in shape matching, and finally, the data and code produced and publicly available.

Acknowledgements

A significant aspect of doing research is telling stories that you are the first to witness. Hence, this thesis contains a story. Just a distilled version, actually, primarily focused on successes rather than the failures behind them. Like any story, what is left outside is the essential part. Among the prominent absences, people are the most important. Any detail of this manuscript would be dramatically different without some providential meets, and I feel fortunate to have such a list of acknowledgement to do. These two pages are an unsatisfactory attempt to give space to them and, indeed, I have forgotten many valuable mentions. However, to those who thought I had a chance: Thank You. Please, keep doing the same with anyone you meet and any time you have the opportunity. Stories are possible only if they have an opening to happen.

First of all, I would thank myself for crafting my chance to arrive here. Every time, it required me to convince myself to make a final attempt - just a last, desperate attempt looking for the truth. By my empirical observations, the final one is the attempt with a higher rate of success.

Then, special thanks go to the whole context of the University of Verona and its Computer Science Department, a second home for ten years. I met many people full of life that love their work, and I cannot express enough appreciation for the enormous and open-minded underlying infrastructure. Some roles (administrators, technicians, representative staff, and many others) will never be acknowledged enough in the history of Science. As part of this system, I owe gratitude to my closest collaborators, starting from my advisor Umberto Castellani. I remember the first time I entered his office for my Master Thesis; he intercepted my passion before anyone else. Also, I would thank Pietro Musoni as the first person who worked with me on digital humans during his Master thesis. He is a true example of intense determination; I am sure it will pay back,

and he will be proud of his story. My deepest gratitude goes to Simone Melzi; I probably have quitted the PhD after few months without his patience and availability. He is perhaps the most influential figure in my education as a researcher. I also thank him for the idea to have this cover book image, and I am sincerely glad we had the chance to keep working together. I would also thank all the friends, colleagues and students I spent time hanging out with, discussing ideas, and organizing events. The breaks with them helped me resting among unlimited work sessions, making my days affordable.

I would thank all my co-authors, especially those who mentored me with their outstanding suggestions and experience: Emanuele Rodolà, Maks Ovsjanikov, and Niloy Mitra. They have been inspirational and reference points in my deadlocks moments. I am sincerely grateful to my thesis reviewers and committee for their time and effort: Stephanie Wuhrer, Alex Bronstein and Tobias Schreck; I am honoured and delighted for their attention to my work. I would also include all the anonymous reviewers and program chairs who met my papers, who probably will never be aware of their massive impact on my work.

Like any story, it starts before the first page and ends after the last one. I would thank those who witnessed my growth more than anyone else: my mother Valeria, my father Alessandro and my brother Matteo. They put the basement of this thesis far beyond the three years which I spent on it. Daily, they constructed something made of imperceptible deeds that rarely need words, and which would be pointless trying to explain here. I extend this gratitude to all my family and our 29 years of mutual tolerance.

Finally, my warmest thanks go to Silvia. She followed these three years closely, with enthusiastic cheering, loving, and understanding. I know how hard it is to relate with me when I am under pressure (even when I am not) and how frustrating it is to wait months and travel hundreds of kilometres to meet just for a weekend. Her caring has been motivating; her brightness has been inspiring; her sense of humour (I feel so lucky to be one of the few beings able to catch it - at least, most of the times) has been relieving. She encouraged me in several key experiences that I did not believe were possible, like spending six months abroad or doing a complete backflip underwater. Her unswerving presence has deeply impacted this work. In particular, without her premium spell-checker account, my English would have been seriously worst. She makes me feel loved every day. She thought I have infinite chances. To her goes all this work.

Contents

1

Introduction

> No subject is terrible if the story is true, if the prose is clean and honest, and if it affirms courage and grace under pressure.
>
> Ernest Hemingway

In this chapter, we first introduce the high-level concepts of representation for 3D objects and the shape matching problem. As conclusion, in Section 1.3 we provide an outline of the thesis, listing the main contributions.

1.1 3D Object Representations: this is not a pipe

The verb "to represent" comes from Latin *re-praesentare*. It is composed of two parts: *praesentare*, which means 'to present, to place before', preceded by the particle *re* which means "again". So: to show something that already exists.

The problem of showing 3D objects is as ancient as humanity: our ancestors depicted pictograms in caves for spiritual and religious purposes. The most ancient known dated cave paint in the World is a red hand stencil from Spain that is considered older then sixty-four thousand years [138] (left of Figure 1.1).

Egyptians, Greeks, and Romans represented their lives and myths in 2D. While their scenes were mainly 2D and lack realism, there was a sophisticated symbolism that conveys properties to the characters and creates stories from a single frame [159]. Only in the fifteenth century, the Italian Renaissance artists developed methodological studies about perspectives, imitating (and fooling) the human view, yielding depth in the images, and reproducing 3D environments. Few centuries later, between *1820* and *1840* the studies of Joseph Nicéphore Niépce, Louis-Jacques-Mandé Daguerre and William Henry Fox Talbot brought the photography invention. When this technology had be-

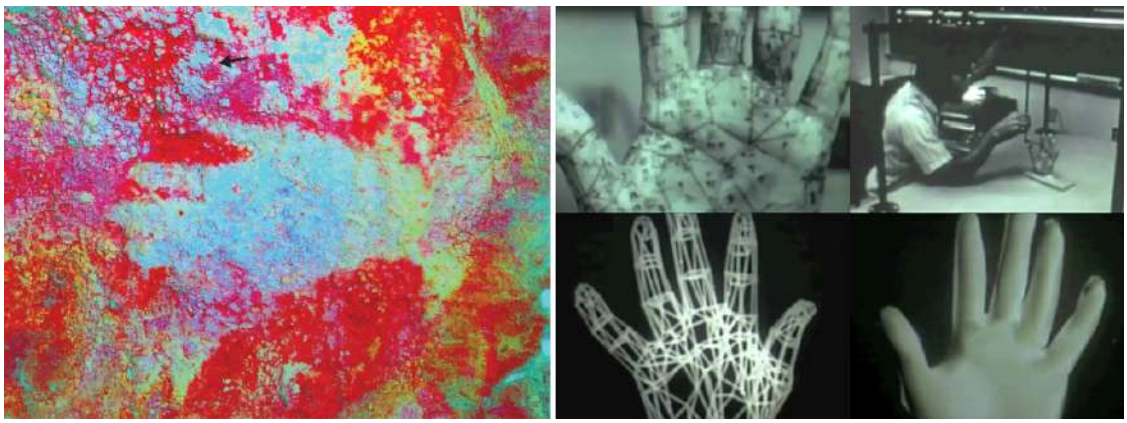


Fig. 1.1: On the left, the oldest dated cave drawn, datated 64'000 years ago [138]; on the right, the first rendered movie of a 3D virtual object [67].

come mature, it provided a significant speedup in reality replication. This representation process is highly independent of the depicted subject, and in some sense, it constitutes an automatic analysis (impressing on a film) and synthesis (developing the film) of the view process. The Lumière first film (exploiting the forgotten León Bouly's cinematograph) in *1895* was a necessary further step to replicate our 3D world experience, that is, in general, a continuous process, instead of an instantaneous snapshot. This continuous representation was also prone to be edited, and the first 'special effects' (i.e., manipulations of the representation) was made by cutting and composing the film frames, art that was pushed to its limits by Georges MÍiès, the "Cinemagician", director of *Le voyage dans la Lune* [221]. The invention of the Cathode-ray tube and the research in electronics produced an increasing interest in representing objects (like missiles) on monitors, giving us the first two blinks of Computer Generated Images (CGI) in *1958*: *Vertigo* of Alfred Hitchcock, which is the first film using computer made effects, and *Tennis for Two* of William Higinbotham, the first videogame. Both of them involved analogical technologies. In *1971* at the University of Utah, the two students Edwin Catmull (recently awarded with the Turing Prize) and Fred Parke modeled and rendered the animation of Catmull's hand (Figure 1.1, right). The four minutes short called *Hand*, required sixty thousand minutes of work [67]. It is recognized as the first 3D rendered movie, and some years later, it was included in the film *Futureworld* (*1976*). This pipeline for acquiring, synthesizing, and modeling a 3D digital object is a milestone in CGI history. Before this moment, analytical processes and physical phenomenons of the acquisition guide the objects' obtained representation. Moving to the digital domain, this is no longer true: between the real object and its 2D image on the monitor, there are intermediate representations handled by the computer in logical structures.

They have to encode the geometry, be handier for us, and be computationally tractable for the computer. Which is the best paradigm to achieve these points? We have several options: a continuous surface modeled by a mathematical equation, a discrete set of tiny 2D polygonal patches glued together (as done by Catmull for his hand), a composition of actions that univocally build the desired object, a volume and its surface, possibly specifying also the material qualities like its density or elasticity. The advantages and drawbacks of each one make them suitable for different applications. We will see in this thesis that the right choice (or craft) has a dramatic impact.

Before introducing a technical discussion on the possible different representation (it will be presented in Chapter 2.1), we would consider the relationship between the representation of an object and the object itself. A philosophical discussion of this comes from "C'est n'est pas une pipe" (This is not a pipe) [111], the book from Michel Foucault that comments on the paint *The Treachery of Images*, of the surrealist René Magritte (Figure 1.2).

Fig. 1.2: The Treachery of Images, René Magritte

This paint provokes two divergent reactions: the sentence and the paint are in evident contradiction because the drawing is trivially the one of a pipe. Nevertheless, the two are also tautological: it is obvious that it is not a *real* pipe, but only an illustration of it. Foucault approaches the paint with the same naive approach, comparing the two divergent perspectives. He derives the existence of implicit bias in our language habits: the honest answer to the question "What is this?" is: "It is a pipe". However, this pipe lacks several crucial elements that push it away from our experience of a *real* pipe: it is floating without coordinates, it is illuminated but without any shadow, and it is extraordinarily smooth and shorts of details. Furthermore, the text itself is a description, and so represents the object that could refer or not to a pipe. The text and the figure live in the same non-coordinate space, replicating the interaction between the text and the image, and we have to believe in their coherence. The representation of an object differs from the object itself, and even if we can interchange these concepts in our communication, in our research, we should wonder how e certain representation interacts with its object. For us, the *representations* will be the computational expression of the object's *geometry*, where this term is the most abstract and heterogeneous possible, relying on our intuition of the 3D World. Digital representations for 3D objects should informatively convey the geometry *and* efficiently support computations on them.

The primary discipline that investigates how to handle such objects and their digital representation is called Geometry Processing. It aims to study (analysis), replicate (visualization), and deform (modeling) our experience of geometry (shapes) through its

representations. However, such representations are not the geometry itself, and this is a harrowing problem at the base of this discipline: we would work with geometry, but we can only do it on its footprints.

1.2 Non-rigid shape Matching

We can see from Figure 1.1 that Catmull carefully marked his hand with dots and lines, constructing an appropriate triangulation for the applications he has in mind. He aimed to acquire the geometry and identify some precise locations of interests, designing a proper discretization of the object. However, once Catmull concluded this tremendous effort to model his hand animation, it would be undesirable to redo everything for each new hand. Furthermore, it would be particularly annoying to spread such an acquisition process among his colleagues: if they annotate their hand in a slightly different way, the animation system's effect would be unpredictable. To keep the original properties, they should annotate all the hands in the same way such that they are *in correspondence*. The problem of establishing a correspondence given two objects is called *shape matching*. Solving this by manual annotation requires time and produces uncertainty, and nowadays, it is common to separate the process of geometry acquisition from defying dense correspondence between objects. The problem moves from the acquisition and annotation to the digital representation of the object. In this thesis, we will mainly consider 2D surfaces embedded in the 3D space and discretized by polygons (while in some cases, we also target different representations, like depth views and point clouds); in this case, the shape matching problem requests to pair each point from one surface to one point of the other.

This problem is tough when the shapes to match exhibit *non-rigid deformations*. We will call non-rigid deformations all transformations of a surface that are more than a composition of rotations, translations, or reflections of the whole shape. These deformations arise in many interesting domains like humans, animals, and clothing, to mention a few. An example of non-rigid deformation is the fingers bending in Figure 1.3; also, the two hands are not from the same source (this is evident from the border of the wrist), and this makes the paths on their surfaces (so called *metric*) different. However, non-rigid deformations

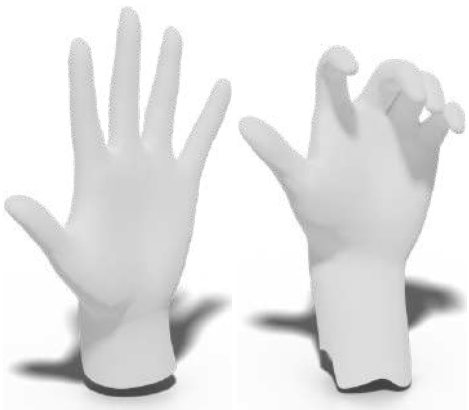


Fig. 1.3: Hands can be non-rigidly deformed.

can be much more complex than this: humans have several different behaviors depending on muscles tone, skin elasticity, and fat quantity. Also, surfaces can show missing parts due to partial views, varying resolutions and details, presenting clutter, and other unrelated objects.

We have a 'good' matching when we can use it to transfer the color defined on the first to the second, and the two colorizations look coherent, as shown in Figure 1.4. This kind of visualization will be proposed several times during the manuscript. Considering the hands, we can imagine it as fitting the same glove on both of them. This action will be problematic if they have significant differences in finger lengths, palm width, or, even worse if one of the two is grabbing an object.

Fig. 1.4: A visualization of a point-to-point correspondence.

The representations of objects are fundamental; combining different perspectives and tools is crucial to ease the matching task. In this thesis, we mix, extend, and learn representations to tackle 3D non-rigid shape matching, providing theoretical and practical advancements.

1.3 Outline

We organize the rest of the thesis as follows: in Chapter 2 we introduce the main background and tools used in this thesis, giving a proper contextualization to the different common representations and the problem of shape matching with a particular focus on Functional Maps framework. Its purpose is purely introductory to our contribution, presented in the following chapters divided into three Parts.

Chapter 3 opens the first Part, presenting a pipeline for human body recovery from RGB+D single view, developing a strategy to deal with occlusions using a data-driven prior as regularization. Chapter 4 presents *FARM*, an automatic registration pipeline to fit a morphable model to a broad set of heterogeneous inputs exploiting the Functional Maps framework. We will also show how, after two years from its first appearance, FARM is still state of the art in non-rigid matching tasks. Chapter 5 extends this pipeline by exploiting recent advancements of functional-maps matching methods and proposing a High-Resolution augmentation technique to catch high-frequencies details of the target mesh.

The second Part of the thesis examines axiomatic matching methods on triangular meshes. First of all, Chapter 6 proposes a benchmark to match human bodies with dif-

ferent connectivities, showing that identical triangulations inject biases in the matching intrinsic pipelines. Chapter 7 extends the standard Laplace-Beltrami Operator eigenfunctions set to include three extrinsic bases (called Coordinate Manifold Harmonics), enhancing the geometry representation of the points and then the matching provided by functional maps. We use this new basis to provide a remeshing pipeline between objects in a similar pose.

Finally, the third Part explores innovative deep learning techniques. Chapter 8 proposes to learn a new basis set dedicated to point clouds. This new representation is entirely extrinsic, showing robustness to noise, outliers, and clutter. The presented framework is also the first entirely differentiable for Functional Maps on point clouds which also consider the learning of the basis. We conclude the thesis with Chapter 9, where we learn a linkage between two super-compact representations: an AutoEncoder latent space and the eigenvalues of Laplace-Beltrami Operator. The latter can be computed starting from several different representations, and therefore our network can instantaneously recover geometries regardless of their implementations, bringing them in a common discretization.

Summarizing, our contributions are:

1. proposing innovative automatic template-based matching pipelines, exploring the regularization benefits of the data-driven generative models on several challenging settings. We combine different representations and we propose a strategy to obtain an arbitrary level of detail in the final result;
2. providing the first analysis on the impact of tessellations of triangular mesh for non-rigid 3D shape matching. We highlight the bias induced by matching two shapes with the same connectivity, realizing a new test benchmark for matching methods;
3. extending the Functional Maps framework in two directions: firstly enriching the intrinsic information with an extrinsic one, and secondly introducing its first version designed for point clouds matching. These extensions provide results in challenging settings and open to new applications;
4. proposing the first framework that joins a super-compact axiomatic representation with a learned one, instantly recovering a geometry from its spectrum regardless of its original representation or discretization. We not only propose a practical solution to a historical problem; we also enrich an autoencoder with geometric tools and let us directly match shapes.

Finally, we also provide code and data produced by our studies freely available for research purposes.

Publications linked to this thesis

This thesis is mainly based on the publications listed below:

- R. MARIN, S. MELZI, N.J. MITRA AND U. CASTELLANI; *POP: Full parametric model estimation for occluded people*, in proceedings of Eurographics Workshop on 3D Object Retrieval, 2019 (pp. 1-8). [202]

- R. MARIN, S. MELZI, , E. RODOLÀ AND U. CASTELLANI; *FARM: Functional automatic registration method for 3d human bodies*, in Computer Graphics Forum 2020 (Vol. 39, No. 1, pp. 160-173). [203]

- R. MARIN , S. MELZI , E. RODOLÀ AND U. CASTELLANI; *High-Resolution Augmentation for Automatic Template-Based Matching of Human Models*, in proceedings of the International Conference on 3D Vision (3DV), 2019 (pp. 230-239). [204]

- S. MELZI, R. MARIN, E. RODOLÀ, U. CASTELLANI, J. REN, A. POULENARD, P. WONKA, M. OVSJANIKOV; *Matching Humans with Different Connectivity*, in proceedings of Eurographics Workshop on 3D Object Retrieval, 2019 (pp. 121-128). [212]

- S. MELZI, R. MARIN, P. MUSONI, F. BARDON, M. TARINI AND U. CASTELLANI; *Intrinsic/extrinsic embedding for functional remeshing of 3D shapes* in Computers & Graphics, 2020 (Vol. 88, pp.1-12). [211]

- R. MARIN, M.J. RAKOTOSAONA, S. MELZI and M. OVSJANIKOV; *Correspondence learning via linearly-invariant embedding*, in proceedings of Thirty-fourth Conference on Neural Information Processing Systems (NeurIPS), 2020. [205]

- R. MARIN, A. RAMPINI, U. CASTELLANI, E. RODOLÀ, M. OVSJANIKOV, S. MELZI; *Instant recovery of shape from spectrum via latent space connections*, in proceedings of the International Conference on 3D Vision (3DV), 2020. [206]

2

Background

In this chapter, we revise the common representations for 3D objects and the state of the art of shape matching, formalizing the contents useful for the rest of the thesis.

2.1 Common representations in Computer Graphics

As we anticipated in the introduction, the "representation" term explodes in many different interpretations. We focus our interest in a computational perspective, as the digital simulacrum of real world geometry. We represent objects for different purposes:

- *Acquisition*: different technologies and processes are used to acquire 3D objects. Representations for different purposes are required depending on applicative requirements regarding geometry reliability, timing, and resource availability.
- *Depicting*: visualizing objects lies at the heart of Computer Graphics. Representations that are computationally efficient to render are essential in many scenarios.
- *Analyzing and query*: analyzing an object and investigating its properties are crucial aspects. Computing geodesic distances, surface area, topological genus, detect disconnected components are just few of them. In some representations, these computations are immediate, while convoluted in others.
- *Manipulating*: editing 3D objects in their digital representations is important to obtain new geometries without acquiring them. Modifications can interest its space occupancy and appearance characteristics (e.g., colors and textures, triangulations, normals).

These purposes motivated the vast amount of different representations proposed over the decades. Listing all of the published variants would not be practical for our discussion. Thus, we offer our categorization, and we link to different sources that give more in-depth and rigorous taxonomies [50, 55, 133, 328].

2.1.1 Images

A 2D view (or more than one) is a natural encoding for a 3D object. This kind of representation is natural since it is similar to observing the world with our eyes and the standard output of computer monitors. Images of 3D objects also have several advantages from an analysis perspective: they are a structured grid of pixels, easy to analyze thanks to all the techniques developed in signal processing. There are several techniques to obtain 3D information from the 2D, like use multiple RGB views or acquiring an extra depth channel (we refer to it as 2.5D). Images are easy to acquire, cheap, and 2.5D has got much attention thanks to consumer-level depth sensors (Kinect, RealSense). Several methods use synthetic images renderings to replicate real-life scenarios in a controlled setup and learn in conditions otherwise impossible [310] [270] [169]. On the other hand, images present several challenges. They require a careful parameters setup (camera calibration) to assure the coherence of acquisitions. Camera calibration is a non-trivial task, especially when it involves multiple cameras and different technologies. Also, the projection of a 3D object into an image plane causes a loss of information. In a scene where only the object appears, it is caused by self-occlusion that sometimes even a multi-view perspective cannot fix. Finally, the resolution of the acquisition bound the representation of the geometry, which can also include external clutter (e.g., floor, additional objects, background)

2.1.2 Implicit representations

An object can be expressed by a function that relates it with its surrounding space by defining a scalar-valued function $F : \mathbb{R}^3 \rightarrow \mathbb{R}$, in which the object's surface is given by the zero-level isosurface $S = \{x \in \mathbb{R}^3 | F(x) = 0\}$. This representation is particularly convenient to query points' locations and decide if they lie inside, outside, or on the object surface. We can synthetize an explicit surface in a second moment, potentially at an arbitrary resolution, for example by using marching cubes algorithm [196]. Among them, *voxels* offer a discretized version, as a 3D grid of volume particles (in general as cubes). For each grid element, its value is 1 if the object occupies it, and 0 otherwise. In some applications (e.g., neuro-imaging), it is meaningful to store non-binary values in the voxels to specify the object's density at that point. Techniques developed for 2D images (e.g., convolutions) straightforward extend to voxels thanks to their gird-organized nature. They are also useful to model changes of topology without caring how this reflects on the underlying surface. However, due to the curse of dimensionality, the exponential growth of the grid cells limits the representation's resolution. This issue can be tackled by using efficient data structures like octrees and promoting sparsity in the occupancy grid [189] [64].

While the voxels are discrete, the same paradigm can be extended to work with analytical functions. For example, the *Occupancy Functions* express the voxels in a continuous way with a function o:

$$o : \mathbb{R}^3 \rightarrow \{0,1\}. \tag{2.1}$$

The continuity of the representation permits to obtain an arbitrary resolution when it is converted in an explicit surface. It is also convenient to query points to know if they are inside or outside the shape and catch object collisions. This representation can be extended to *Signed Distance Functions*, where the function o does not express only presence or absence, but also the distance from the surface.

$$o : \mathbb{R}^3 \rightarrow \begin{cases} \min\limits_x ||d(x,a)|| & \text{if } a \text{ is outside the shape,} \\ 0 & \text{if } a \text{ is on the shape,} \\ \min\limits_x -||d(x,a)|| & \text{if } a \text{ is inside the shape.} \end{cases} \tag{2.2}$$

While it seems an unpractical representations, recently several works rediscovered them [27, 73, 74, 125]. They have been used for object acquisition [151], rendering [240], and recently also for generation [290] and template registration [39].

Finally, implicit representation can be easily merged via set operation, producing the *Constructive Solid Geometry* paradigm, representing a 3D object by a set of boolean operations (like union, intersection, difference) between primitives. This kind of representation is useful to model complex analytical objects [169].

2.1.3 Explicit surfaces

Having an explicit definition of the surface is particularly popular in Computer Vision and Computer Graphics; it provides an intuitive way to work with surfaces and exhausts the object's appearance property (i.e., in many applications, we can forget what there is inside a shape).

We will consider them as two-dimensional Riemannian manifolds $\mathbf{S}$ embedded into $\mathbb{R}^3$, and equipped with the standard metric induced by the volume form. The surface can be then modeled by a *parametrization* function f that maps $\Omega \subset \mathbb{R}^2$ to the surface $\mathscr{S} = f(\Omega) \subset \mathbb{R}^3$. In general, this precise and analytical description of the surface is not feasible for complex objects if we limit it to a single continuous function. Thus, the natural idea is to represent a 3D object's surface as a discrete sampling. In a 1D case, a function can be approximated by lines connecting the points of the sampling. For a 2D surface case, we can analogously connect little tiles glued continuously together. As in the 1D dimensional case, we can approximate the surface using linear (i.e., planar) or more complex (i.e., curved surfaces, like polynomials) piecewise surfaces. We know

that a $g \in C^\infty$ function with bounded derivatives can be approximated in a limited interval h by a polynomial of degree p with an approximation error $O(h^{p+1})$. However, increasing the polynomial degree produces complex surfaces that are arduous to glue continuously. For these reasons, the most popular choice is using *polygonal* surfaces. The standard is to use the simplest polygon possible, that are triangles, and this representation takes the name of *triangle meshes*. A triangle mesh can be then formalized as $\mathscr{M}$ [50]:

$$\mathscr{V} = \{v_1, \ldots, v_n\}, v_i \in \mathrm{R}^3 \tag{2.3}$$

$$\mathscr{F} = \{f_1, \ldots, f_m\}, f_i \in \mathscr{V} \times \mathscr{V} \times \mathscr{V}. \tag{2.4}$$

Note that this can be seen as a specialization of graphs (called planar graph). The approximation power for the area is $O(h^2)$ with h as the maximum edge length. However, it worth mentioning that a discretization should also approximate other properties than space's occupancy. For example, the Schwarz Lantern [282] shows that we can approximate a cylindrical surface with arbitrary precision (in the sense of Hausdorff distance) with a triangular mesh, but it can diverge in the surface area and produce incoherent surface normals. These representations are simple and equipped with several theoretical results that permit depicting and analyzing them as a proper surface. However, they necessitate attention in their manipulation to keep the properties of a valid 2D manifold: for example, it is desirable avoiding self-intersections of the polygons, and change the object topology is far from trivial. If we consider the case in which $\mathscr{F} = \emptyset$, then we obtain *point clouds*. Point clouds are collections of points that witness an underlying surface in those locations. We consider them an intermediate representation mainly derived by acquisition pipelines (e.g., the output of depth sensors). The surface can be reconstructed by inferring the local surface behavior and properties (e.g., its normals directions) and using algorithms like the Poisson method. We refer to [36] for a survey on surface recovery from point clouds. Finally, we would highlight that the geometry expressed by a points collection is invariant to their order; each sorting describes the same surface (for meshes, we also need to update the triangulation indexing). Unordered point cloud obtained increasing attention in recent years, thanks to modern deep learning architectures [252] [253] [142].

2.1.4 Pointwise embeddings

All previous representations aim to represent a 3D model in $\mathbb{R}^3$ and strictly describe the space's object occupancy. However, especially for analysis purposes, there could be better representations. For example, previously we already introduced the idea of parametrization of a 3D surface in a 2D plane. This idea is at the base of cartography; in

fact, we are much better at working on a planar surface where we compute distances as straight lines, rather than on spheres where distances are paths on a curved surface. The intrinsic geometry (and in particular the topology) of the object impacts the mapping; for example, for genus 0 surfaces, it is possible to find a map that preserves the area (equiareal) or the angles (conformal), but it is not possible to achieve both (isometric). However, we will discuss the interesting inverse direction, in which we seek to map our 3D object in an $\mathbb{R}^m$ space (so our 3D object is the preimage of the mapping instead of the image) with m arbitrarily large. We refer to the realization of our object in the $\mathbb{R}^m$ space as a *pointwise embedding*. These kinds of representations gained popularity to study the intrinsic properties of a shape, like its metric. Given a shape M equipped with its metric d_M, an *Isometric embedding* problem requires to find a map $f : (M, d_M) \rightarrow (\mathbb{R}^m, d_{\mathbb{R}^m})$ such that:

$$d_M(x, x') = d_{\mathbb{R}^m}(f(x), f(x')), \tag{2.5}$$

where $d_{\mathbb{R}^m}$ is the natural metric induced by the euclidean space [55]. A result in this direction is the *Nash embedding theroem*, which states that it is possible to embed any R^3 surface into an isometric R^17 surface [228]. However, *the Linial's example* shows that if we impose that the final surface metric is the *restricted* metric $d_{\mathbb{R}^m}$, this is in general not possible [182]. Thus, several methods have been proposed to find an approximation in terms of the *minimum-distortion* embedding. This approximation leads to non-convex optimization, historically called multidimensional scaling (MDS). An elegant and efficient way to solve this problem using linear algebra is called *classic MDS* [304]. If an isometric embedding into $\mathbb{R}^m$ exists, the distances in that space lead to a Gram matrix, that is positive and semi-definite. Then, it can be shown that the final embedding can be retrieved using the spectral decomposition of this matrix. This method is particularly appealing: it bases on some basic linear algebra that we can compute efficiently. These approaches point to the so called *spectral embeddings*, in which the coordinates of the embedded object are the eigenvectors of some matrix. Motivated by these, we could wonder if there exist other matrices of interest. In particular, looking for a minimum-distortion embedding for the whole shape enforces a global property on the embedding of arrival. However, since we know that this problem does not have a solution in general, we can look for some *local* criterion, i.e., for an embedding that achieves this property in the neighborhood of each point. With this spirit, a particular relevance has been devoted to *differential* representations [292], where each point obtain δ-coordinates; the difference between a point coordinate and the mass center given by its immediate neighbors:

$$\delta_i = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in N(i)} \mathbf{v}_j. \tag{2.6}$$

This can also be expressed as a matrix, that is well known in the graph theory. *Laplacian* matrix elements are:

$$l_{ij} = \begin{cases} -1 & (i,j) \in E \\ d_i & i = j \\ 0 & otherwise, \end{cases} \tag{2.7}$$

where d_i is the degree of the vertex (number of incident edges). Then to the equality $L\mathbf{x} = D\delta$ show how this matrix is linked with δ-coordinates. The matrix L is also called topological Laplacian [110]. This operator is only related to the connectivity of the points without considering the underlying surface. The differential coordinates have an important role in expressing local relations, and, in particular, can be shown that their direction approximate surface normals, while the local mean curvature is related to their magnitude [300]. Also, from δ-coordinates is possible to recover the original vertices positions if at least one point position is known [292]. To generalize this formulation, we can rephrase our formulation to admit general weights on the edges:

$$l_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k \neq i} w_{ik} & i = j \\ 0 & otherwise, \end{cases} \tag{2.8}$$

for some proper weights w_ij. Their modifications lead to several kinds of Laplacians, producing different embeddings and enforcing different idea of neighboorod. Laplacian eigendecomposition has shown several utilities in studies of graphs connectivty [110] [219] [224]. The analogue for surfaces is the *Laplace-Beltrami Operator* (LBO). The LBO is discretized as a matrix of dimension $n_{\mathcal{M}} \times n_{\mathcal{M}}$. This matrix is defined as $\mathbf{\Delta}_{\mathcal{M}} = (\boldsymbol{A}^{\mathcal{M}})^{-1}\boldsymbol{W}^{\mathcal{M}}$, where $\boldsymbol{A}^{\mathcal{M}}$ is the *mass matrix* and $\boldsymbol{W}^{\mathcal{M}}$ is the *stiffness matrix*. The mass matrix is a diagonal matrix whose entries are equal to the area element associated to each vertex. The stiffness matrix represents the local geometry. In the cotangent scheme the weights are defined as:

$$w_{ij} = \begin{cases} (\cot\alpha_{ij} + \cot\beta_{ij})/2 & ij \in \mathscr{E}_i \subset \mathscr{E}; \\ (\cot\alpha_{ij})/2 & ij \in \mathscr{E}_{\partial\mathcal{M}} \subset \mathscr{E}; \\ -\sum_{k \neq i} w_{ik} & i = j; \\ 0 & \text{otherwise;} \end{cases} \tag{2.9}$$

where α_{ij}, β_{ij} are the angles $\widehat{ivj}, \widehat{jwi}$ of the triangles that have ij as edge, $\mathscr{E}_i$ are the edges connected to the vertex i and $\mathscr{E}_{\partial\mathcal{M}}$ are the edges on the boundary. We will see that the LBO matrix's spectral embedding is relevant for solving the surface matching

problem, but it also has several applications in function analysis, watermarking, multiresolution, and many others. We refer to [246] for the LBO discretization with cotangent weights and to [292] for a full discussion on classical applications of this.

2.1.5 Latent embeddings

Police departments have drawers that sketch the suspects following the witness indications. This process is lead by several questions that are progressively tuned by witness feedback. A union of directives models each characteristic of the face. Ideally, a set of instructions should produce a univocal result. In the same spirit, we can assume that a set of hidden (i.e., *latent*) decisions generate a certain geometry. Hence, We assume that there exists a function $f, \mathbb{R}^d \rightarrow \mathbb{R}^{n\times 3}$ that associate each point of a d-dimensional vector space to a specific 3D model. Each dimension of our vector represents one of these decisions as a continuous variable. This representation is a model-wise encoding, or more popularly called, a *latent embedding* of our shape. This representation is super-compact (it is a single n-dimensional point), and the function has to "un-zip" it to a complete geometry in one of the previous representations. One possibility is to use axiomatic super-compact representations like the Laplacian Spectrum, which has an intimate relationship with the object's geometry (in this case, we can imagine the latent directives purely geometrical ones). While we know that it is not theoretical possible recovering the shape from its spectrum [124], Chapter 8 will discuss its feasibility in practice [85] [255] [206]. Learning the latent representation and the function f from data is another option that gets increasing popularity in the last decades, thanks to the availability of datasets, computational power, and advancement in the machine learning field. In 3D geometry we identify two different branches for this representation: *Morphable Models* and *Deep Generative Models*.

Morphable Models Modeling techniques need to be powerful enough to represent a large variety of deformations. However, they should permit only valid deformations, which is even more difficult if for "valid" we imply a semantical coherence that belongs to a particular object's population. In general, the properties of a surface that are subject to modifications are:

- *Identity*: that concerns all the attributes that identify and object from similar ones. For example, in human body models different people present different body structures, proportions, and traits.
- *Pose*: that is the location in the space of the whole shape and displacement of its articulated parts. In the human body case, this corresponds to the subject's global rotation, posture, and limb positioning.

– *Appearance*: that is related to color and albedo of the surface. For a human, this corresponds to skin tone, clothes material, but also ambient light impact.

These properties touch different aspects of an object, but they are also strictly correlated, and they interact in the resulting depiction. From our human body example, arms width depends on the subject's physicality and by muscle contraction of the specific pose; a tone of skin depends on the identity, environment illumination, and light occlusion induced by deformations. This task is far from trivial, and it is the main motivation of more than thirty years of research in the field. The idea that a template modification can belong to a certain distribution of possible deformations gained popularity at the end of the '80s in the 2D domain. The pioneering Active Contour Model [163] was presented at the first ICCV conference and was then extended by [283], where some 2D contour families have been deformed using axiomatic rules. From these, [82] proposed Active Shape Models where the deformation has been trained from examples. This data-driven approach was crucial to overcoming the limits of axiomatic methods to handle such complex deformation rules. While this work mainly focuses on the identity of simple objects (i.e., transistors), [81] extends it with Active Appearance Models to model the gray-scale of a 2D image. In this work, the modeling relies on Principal Component Analysis (PCA), a technique extensively used by many other works thanks to its optimality among the linear possibilities. As one of the first works in the field, the joint modeling of shape and appearance is remarkable. [316] introduced also pose deformations, and finally [156] used a dense correspondence on pixels instead to use sparse points of facial features. The same work mentioned the *Morphable Model* term in the 2D domain for the first time. All the previous works act in modeling 2D images, probably due to the availability of data and computational power of that time. The first work that translates the idea in 3D was [40] that also addresses faces as the main applicative domain; faces are simple objects with a high interest in several fields (e.g., medical, security, entertainment). After that one, many other faces and heads datasets appeared, jointly with learned models [61] [179]. For an exhaustive history of 3D Morphable Models for faces, we refer to [96].

Moving to human bodies, they are a more complex domain than faces; their acquisition is difficult, and harvesting information from images is difficult. Also, their variabilities in terms of shape, pose, and colors are wider than faces. Thus, we had to wait more than a decade from [40] to see the first attempts in this direction. [20] was the first to use PCA to learn a deformation space for humans identities. Two years later, the SCAPE model [24] handles both the identity and the pose of the subject. The SCAPE's primary purpose was to solve computer vision problems (e.g., real-scan registrations), and the triangles of the template deform through a complex optimization problem. By itself, this model does not provide a straightforward way to synthetize shapes. Several

following up works tried to simplify the animation modeling of SCAPE, proposing other versions like BlendScape [135], and S-SCAPE [152] [107] where the pose deformation framework is coherent with modern animation pipelines. In particular, they use the Linear Blend Skinning (LBS) paradigm: given a skeleton (as a collection of joints and their hierarchy) and the skinning weights for each vertex (that weight the impact of the joints rotations), three scalars control the deformations for each body part. These scalars encode rotations and provide an interpretable way to deform the shape. Finally, it has been proposed the groundbreaking SMPL [195]. SMPL is a parametric model learned on a large dataset of real human body scans [262]. Two sets of parameters control its template T_{smpl}: the one of 10 values β to modify the identity using PCA, and one of 72 values θ to control the pose. The latter set encodes the rotations of the 24 joints J, which deform the initial T-Pose. The initial positions for joints of different subjects are obtained by applying a linear regressor $\mathscr{J}$ to the vertices of the subject's T-pose. SMPL can exploit different animation frameworks, but LBS (with a corrective factor) is a common choice; all the SMPL properties (the template, the PCA, the skinning weights $\mathscr{W}$, the joints regressor $\mathscr{J}$, the corrective factor for LBS) are all learned jointly from data. SMPL is interesting from a representation perspective: it generates a distribution of plausible human bodies as triangular meshes, and this is done through the space of its parameters (that act as a latent set of decision, discussed above); it also has a hierarchical structure of its part thanks to the skeleton, that is obtained by applying a function to the vertices. The steps from parameters to obtain the triangular output mesh are differentiable, making the model particularly suitable for solving computer vision problems. SMPL was then extended also to model hands [269], expressions [241] and (using deep learning) also clothes [198]. Similar models have been proposed to model soft tissue dynamic [250] and different parts compositions [157].

To conclude, we would mention that several works extended the Morphable Model approach to other domains, like dorsal spine [218], animals [348], ears [88], dolphins [65], kids [134] and flowers [344] among the others.

2.1.6 Deep generative Models

The breakthrough of deep learning affected several Computer Science domains, thanks to IA researchers' determination in pushing the neural networks studies. This advancement was also possible thanks to the increasing computational power availability, particularly regarding the single-instruction-multiple-data paradigm efficiently implemented by GPUs. From its dawn to *2013*, [281] provide a complete story of the field. We will quickly revise few milestones, with particular attention to deep learning interaction with Computer Graphics and Computer Vision. The first trace of effort to inject vision into multilayer perceptrons was made in *1982* by [112], while backpropagation was not

already involved. In *1989* [318] and [172] contemporary proposed to introduce convolutions to learn from images, while the standard reference for convolution is usually considered [173] due to its maturity. For these works, the researches Yan LeCunn, Geoffrey Hinton, and Yoshua Bengio was credited with a Turing Award in *2018* for their contribution to the development of the field. Convolutional layers gained particular popularity for their effectiveness in learning multi-scaling features, possible because all images share a standard structure: they are a grid of pixels where querying a neighborhood is natural, the operations are efficiently computed, and the convolution operation has several theoretical properties (e.g., shift-invariance) in planar domains. However, 3D surfaces representations are mostly far from this trivial setup, and the only natural extension for applying it to learn from objects is using a voxels representation [325] [210], or images from multiple views [294] [143]. In the non-euclidean domain, [280] introduced the Graph Neural Networks, and [57] bring convolution paradigms on them, exploiting clustering and Laplacian Spectrum. From these works, a large amount of subsequent extensions has been proposed opening the field of *Geometric Deep Learning*; we point to [56] for an excellent theoretical overview, and also [62] as the most recent survey on the topic. Another approach to learn from 3D data is to treat them as points, accepting a faint underlying structure. Recently, extensive attention has been devoted to point cloud representation, where the underlying structure is faint. However, the groundbreaking [252] introduced the first generalistic network directly applicable to point cloud, that is flexible concerning numerosity and order of the input points. Some subsequent works extend this approach by inferring local structures in neighborhoods of the points [253]. A complete survey on classification, detection, and segmentation for point clouds in deep learning can be found here [129]. Finally, another way to learn from non-euclidean domains is to represent the input as descriptors and feed them into the network [105] [128].

While these deep learning works are mainly devoted to analyzing existing objects, several architectures were also proposed to generate new ones: Autoencoders, Generative Adversarial Networks, and Regularizing Flows are some of the most popular choices. They mainly aim to infer an underlying (implicit or explicit) data-distribution and then sample the latent space to generate realistic models. The data-driven approach provides knowledge that helps to solve also unconstrained problems, like 3D objects generation from a single image [77] [103]. Contemporary, [323] proposed to solve the problem with voxels representation. On point cloud data, several alternatives have been investigated by [14] exploiting PointNet architecture [252]. All previous methods act in rigid domains (e.g., chairs, tables, sofas). The use of deep learning for non-rigid objects is investigated only recently, like in [179], that also exploit convolution to learn from surfaces of triangular meshes. Some works addressed body registration of

point clouds starting from deformation of parametric models [154] [175] or also from images [42]. Only recently a complete end-to-end learning model for the entire human body has been proposed [329].

2.2 Non-rigid correspondence

Establishing a correspondence between two Non-Rigid objects is a broad topic, and a full discussion would take (at least) an entire thesis by itself. For an extensive overview, we suggest [307] and the recent [274] that covers research advancements year by year from *2011* to *2019*. Here we are going to discuss only the principles that will be useful for the thesis.

For us, correspondence will be a map $T_{\mathcal{M}\mathcal{N}} : \mathcal{M} \rightarrow \mathcal{N}$; an association for each point that belongs to the $\mathcal{M}$ shape, one point of the $\mathcal{N}$. In triangle mesh and point clouds cases, this association is generally between their sets of vertices V. Looking for pairs (or matching) for non-rigid objects requires to face a broad set of deformations. In real case scenarios, this kind of correspondence is an ill-posed problem: usually, a perfect matching between two different discretizations is impossible, especially at a vertex level. However, there is also a *semantic* problem; for example, let us say that we would retrieve a correspondence between two human bodies. While there are points with a precise solution (e.g., face features, prominent bones, endpoints of the fingers), other places lack sharp-features to solve it (e.g., stomach, chest, and different musculature on arms or legs). Furthermore, there could be holes and noise due to corrupted data and clutter due to differences in their origins (e.g., the differences between a synthetic body without clothes and a real scan with garments). For this reason, this field received much attention in the last decades, and several challenging cases are far from being solved. In general, there some properties underlying a good correspondence: intuitively, we desire that nearby point on $\mathcal{M}$ arrives on nearby points on $\mathcal{N}$, and so the metric is not distorted by the correspondence; we desire that for each point on $\mathcal{M}$ as a distinct image on $\mathcal{N}$, and so that each point of $\mathcal{N}$ as distinct counter-image on $\mathcal{M}$ (i.e., the correspondence should be bijective); in the presence of symmetries (e.g., humans are left-right symmetric) we want to disambiguate them. Given an object and its rigid modified version, there exists an optimal solution that satisfies all these properties. Also, in the presence of a global scale factor, perfect matching still exists. A more challenging case occurs when deformations happen locally, like bending and stretching. These deformations are typical of soft-tissues and organic objects, like faces, humans, animals, to mention a few. As we saw in the previous section, an exciting embedding to deal with locality is the LBO operator, and we anticipated how its eigendecomposition

could serve to efficiently solve for the matching, shifting the point-to-point problem to a function-to-function one.

2.2.1 Laplace-Beltrami Operator Spectrum

We would recall that LBO directly comes from the necessity to have a local representation of our mesh. LBO on the discrete meshes has a linkage with the Laplacian in the continuous case, up to the theoretical limitations involved in the chosen implementation (it is well known that it cannot be equivalent to the continuous case, and there is no free lunch [320]). Another interesting way to re-discover it is from a physical perspective [56]. In particular, given the Dirichlet energy:

$$E_{Dir} = \int_M f(x)\Delta f(x)dx, \tag{2.10}$$

that measures the smoothness of a function, we look for a set of orthonormal basis that minimizes it:

$$\min_{\phi_0} E_{Dir}(\phi_0) \qquad \text{s.t. } \|\phi_0\| = 1 \tag{2.11}$$

$$\min_{\phi_i} E_{Dir}(\phi_i) \qquad \text{s.t. } \|\phi_i\| = 1, i = 1,2,\dots k-1 \tag{2.12}$$

$$\perp span\{\phi_0,\dots,\phi_{i-1}\}. \tag{2.13}$$

That solution is the smoothest orthonormal basis for functional space, that in the discrete setting leads to:

$$\min_{\Phi\in\mathbb{R}^{n\times k}} trace(\Phi_k^T\Delta\Phi_k) \text{s.t. } \Phi_k^T\Phi_k = \mathbf{I} \tag{2.14}$$

$$\Delta\Phi_k = \Phi_k\Lambda_k, \tag{2.15}$$

where Φ_k are the set $\{\phi_0,\phi_1,\dots,\phi_{k-1}\}$. Equation (2.15) says that Laplacian eigenfunction are exactly the solution of our request, that can be ordered by the eigenvalues λ_k (that represent the frequencies, so by the lowest and smoothest one, to the highest). Since the LBO is a positive semidefinite operator $\mathbf{\Delta}_{\mathcal{M}}\colon L^2(\mathcal{M}) \to L^2(\mathcal{M})$, it always admits an eigendecomposition $\mathbf{\Delta}_{\mathcal{M}}\boldsymbol{\phi}_l = \lambda_l\boldsymbol{\phi}_l$. The set of LBO eigenfunctions $\mathbf{\Phi}$ defines an orthonormal basis for $L^2(\mathcal{M})$, the space of square-integrable functions on $\mathcal{M}$. Functions in $\mathbf{\Phi}$ are usually referred to as *Manifold Harmonics* (MH) [306] and correspond to the Fourier basis on $\mathcal{M}$.

The analysis and the synthesis of a given function $\boldsymbol{f}$ are respectively defined as $\hat{f}_l = \langle \boldsymbol{f}, \boldsymbol{\phi}_l\rangle_{\mathcal{M}}$ and $\boldsymbol{f} = \sum_l \hat{f}_l\boldsymbol{\phi}_l$, were $\hat{f}_l$ is the l-th Fourier coefficient of $\boldsymbol{f}$. We refer to [174,

301, 306] for more details. This Fourier basis on $\mathcal{M}$ is composed of smooth functions, and it is optimal for the representation of functions with bounded variation defined on $\mathcal{M}$ as shown in [16]. Commonly, an efficient approximation of $L^2(\mathcal{M})$ is given by the k eigenfunctions corresponding to the k smallest eigenvalues of the LBO. This set of functions is usually referred to as a truncated basis for $L^2(\mathcal{M})$ or a basis for $L^2(\mathcal{M})$.

2.2.2 Functional Maps

As previously highlighted, computing a correspondence $T_{\mathcal{MN}} : \mathcal{M} \to \mathcal{N}$ between two surfaces could be complex and ill-posed problem. Also, it is hard to impose constraints or semantic principles to it. To simplify the problem, it is possible to change the perspective (or the representation of the correspondence) to a functional domain [234] [235]. A function over a mesh is defined as $f : \mathcal{M} \to \mathbb{R}$, that for every point of the surface assign a scalar value. Functions can model physical processes, colors, segmentations and even locations of specific points; their semanticity has no limit. Assuming we have $T_{\mathcal{MN}}$, it can be used to transfer a function from $\mathcal{M}$ to $\mathcal{N}$ via composition $g(p) = f(T^{-1}_{\mathcal{MN}}(p))$. This implies that a correspondence $T_{\mathcal{MN}}$ induces a map for functions (so called *functional map*) in the opposite direction $T^F_{\mathcal{NM}} : (F(\mathcal{N})) \to (F(\mathcal{M}))$ (via pull-back). Given a map that perfectly transfer functions between two surfaces , we are also able to solve for the correspondence by transferring delta functions.

These two objects (the correspondence and the induced functional map) are intimately related, and retrieving one of the two permits to recover also the second one (while not any functional map is associated with a point-to-point bijective map). At first sight, the reader could wonder why add such a structure to solve for the correspondence. However, from this new functional perspective, we can assume that our functional spaces $\mathscr{F}(\mathcal{M})$ and $\mathscr{F}(\mathcal{N})$ are equipped with some basis set of functions $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$ respectively. A function can then be expressed by a linear combination of such basis: $f = \sum_i a_i \phi_i^{\mathcal{M}}$. This let us to rewrite $T^F_{\mathcal{MN}}$ as:

$$T^F_{\mathcal{MN}}(f) = T^F_{\mathcal{MN}}(\sum_i a_i \phi_i^{\mathcal{M}}) = \sum_i a_i T^F_{\mathcal{MN}}(\phi_i^{\mathcal{M}}) = \sum_i a_i \sum_j c_{ji} \phi_j^{\mathcal{N}} = \sum_j \sum_i a_i c_{ji} \phi_j^{\mathcal{N}} \quad (2.16)$$

for some $\{c_{ji}\}$. This equation tell us that a function can be transferred by its coefficents, and reconstructed in the functional space of $\mathcal{N}$. $\{c_{ji}\}$ act as a transfer over the coefficents a_i and they are indipendent from function f; given the basis and the map T they are determinated. We can then say that $T^F_{\mathcal{MN}}$ can be represented as a matrix $C_{\mathcal{MN}}$ (when possible, we will do abuse of notation by omitting the pedix), and it is applied to transfer the coefficients:

$$\mathbf{b} = \mathbf{C}_{\mathcal{MN}} \mathbf{a}, \quad (2.17)$$

where $\mathbf{a}$ are the coefficient in the basis $\Phi^{\mathcal{M}}$ and $\mathbf{b}$ the proper coefficient to reconstruct the function with basis $\Phi^{\mathcal{N}}$. Note that in presence of an orthonormal basis, the matrix $C_{\mathcal{MN}}$ has a specific expression:

$$\mathbf{C}_{\mathcal{MN}} = \langle T^F_{\mathcal{MN}}(\phi^{\mathcal{N}}_j), \phi^{\mathcal{M}}_i \rangle = \Phi_{\mathcal{N}} \mathbf{A}_N T_{\mathcal{MN}} \Phi_{\mathcal{M}} \tag{2.18}$$

where $\langle , \rangle$ denotes the functional inner product. Equation (2.18) give us a closed form computation if the correspondence is known. In general this is exactly what we would retrieve from this process, so instead we can exploit Equation (2.17):

$$\mathbf{b} = \mathbf{C}_{\mathcal{MN}} \mathbf{a} \tag{2.19}$$

$$\mathbf{b^T} = \mathbf{a^T} \mathbf{C^T_{\mathcal{MN}}} \tag{2.20}$$

$$\mathbf{C^T_{\mathcal{MN}}} = (\mathbf{a^T})^{\dagger} \mathbf{b^T}, \tag{2.21}$$

Where the † symbol denotes the Moore Penrose pseudo-inverse. Given the two sets of coefficients, we can efficiently retrieve a matrix just with matricial computations. We would remark the generality of the framework: we have just assumed two sets of orthonormal basis and two sets of related functions, without giving any specific requirements to what these two object encodes. For basis the most common choice is the LBO eigenfunctions, that we have already introduced before. The most common *probe* functions are geometrical descriptors, landmarks or segment correspondences.

Conversion to pointwise map. Before to dive into more details of FMAPS, we would state the common pipeline to convert a C into a point-to-point correspondence. In fact, the indicator function method while it is really intuitive, it requires a complexity of $O(V_{\mathcal{M}} V_{\mathcal{N}})$ where V_M and V_N are the number of vertices of the two meshes. We can obtain a more efficient method if we notice that a delta function δ_x around a point $x \in \mathcal{M}$ has the coefficients $a_i = \phi^{\mathcal{M}}(x)$. Then, $\mathbf{C}\Phi_{\mathcal{M}}$ returns all the delta functions of $\mathcal{M}$. Given the Plancherel's Theorem, difference between coefficient vector is equal to L^2 distance of functions: $\sum_i (b_1 i - b_2 i)^2 = \int_{\mathcal{N}} (g_1(y) - g_2(y))^2 \mu(y)$ where $\mu(y)$ are volume elements of $\mathcal{N}$. Given a functional map matrix $\mathbf{C}$, the underlying pointwise map $\mathbf{\Pi} \in \{0,1\}^{n_{\mathcal{N}} \times n_{\mathcal{M}}}$ is recovered by solving the projection problem [235]

$$\min_{\mathbf{\Pi}} \|\mathbf{C}\mathbf{\Phi}^{\top} - \mathbf{\Psi}^{\top}\mathbf{\Pi}\|^2_F \quad \text{s.t. } \mathbf{\Pi}^{\top}\mathbf{1} = \mathbf{1}\,. \tag{2.22}$$

If the underlying map is bijective, we would expect the matrix $\mathbf{\Pi}$ to be a permutation; however, in order to allow addressing partiality, we relax bijectivity to left-stochasticity. We then solve the problem above *globally* by a nearest-neighbor approach akin to [234]. Note that while more sophisticated approaches exist for this step, they either tend to

be very slow [267, 314] or do not demonstrate any result with partial shapes [100, 263]. Conversely, problem (2.22) is both scalable and works under missing geometry. Therefore an efficient way to find a correspondence is considering for each point $\mathbf{C}\Phi_{\mathcal{M}}$ its nearest-neighbor $\Phi_{\mathcal{N}}$. It is fascinating from a representation perspective: considering the nearest neighbor in such space establishes a linkage between functions and high-dimensional spectral embeddings. Matrices $\mathbf{C}$ are not other than a transformation for point clouds that aims to align them in a convenient space. It is not surprising that a given C could be further refined in postprocessing applying standard registration algorithm in that space, like ICP.

Optimization problem

If the rare case that probe functions are enough and reliable, Equation 2.21 can be directly solved in a closed-form. Otherwise, we can cast it as an optimization problem:

$$\mathbf{C}_{\mathcal{MN}} = \arg\min_{\mathbf{C}} \|\mathbf{CA} - \mathbf{B}\|^2. \tag{2.23}$$

This optimization admits more than one solution, and some of them act poorly on unseen descriptors. We would choose one that guarantees some properties among all the possible C that minimize the map for the given probe functions. To this purpose, the optimization can be extended, including several regularization energies. For example, we could enforce commutativity with operators:

$$E_{comm}(C) = \|S_F^{\mathcal{N}} C - C S_F^{\mathcal{M}}\|^2, \tag{2.24}$$

where $S_F^{\mathcal{N}}$ and $S_F^{\mathcal{M}}$ are arbitrary linear operators. The LBO operator can be plugged in this equation exploiting its spectral representation (i.e., its eigenvalues' diagonal matrix). Another constraint is obtained by observing that if the underlying point-to-point map is locally volume-preserving, then the C is orthonormal: $\mathbf{C}^{\mathbf{T}}\mathbf{C} = \mathbf{I}$, that can also be included in the optimization.

Among many possible regularizations [99, 117, 266, 337], a particular effective formulation is [233]:

$$\min_{\mathbf{C}} \|\mathbf{CF} - \mathbf{GC}\|_F^2 + \lambda_1 \|\mathbf{CF} - \mathbf{G}\|_F^2 + \lambda_2 \|\mathbf{C}\boldsymbol{\Lambda}_{\mathcal{M}} - \boldsymbol{\Lambda}_{\mathcal{N}}\mathbf{C}\|_F^2 \tag{2.25}$$

where $\mathbf{C} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}}$ is the functional map expressed in the Laplacian eigenbases $\boldsymbol{\Phi} \in \mathbb{R}^{n_{\mathcal{M}} \times k_{\mathcal{M}}}$, $\boldsymbol{\Psi} \in \mathbb{R}^{n_{\mathcal{N}} \times k_{\mathcal{N}}}$, and $\boldsymbol{\Lambda}_{\mathcal{M}} \in \mathbb{R}^{k_{\mathcal{M}} \times k_{\mathcal{M}}}, \boldsymbol{\Lambda}_{\mathcal{N}} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{N}}}$ are diagonal matrices of the Laplacian eigenvalues. Matrices $\mathbf{F} \in \mathbb{R}^{k_{\mathcal{M}} \times q}, \mathbf{G} \in \mathbb{R}^{k_{\mathcal{N}} \times q}$ contain the Fourier expansion

coefficients of q probe functions $f_i : \mathcal{M} \to \mathbb{R}, g_i : \mathcal{N} \to \mathbb{R}, i = 1,\dots,q$, i.e., $(a_{ij}) = \langle \phi_i, f_j \rangle_{\mathcal{M}}, (b_{ij}) = \langle \psi_i, g_j \rangle_{\mathcal{N}}$.

Problem (2.25) allows to estimate functional maps in a considerably more accurate way than the baseline approach of [234]. The main rationale being that the commutativity penalty $\|\mathbf{CF} - \mathbf{GC}\|$ promotes solutions that more closely resemble pointwise maps.

The functional map representation has been successfully used in recent years to estimate dense correspondence between deformable 3D shapes [234, 249], in the presence of missing parts or clutter [86, 187, 188, 264], as well as in machine learning pipelines [83, 186]. The research evolved trying different descriptors [116, 209, 213, 297] and regularizations [233, 258], and extending the framework to look for a correspondence also inside the triangles [100], addressing partial data [265], hierarchical structures of subdivision surfaces [285], and efficiently compute large functional maps [214]. Despite this, applying the framework of Functional Map to point clouds is still problematic due to the lack of reliable basis for functions on this representation [35, 79].

2.2.3 Deep Learning

While early works on the Functional Maps framework were purely axiomatic [168, 232, 249, 265], this framework has also recently been adapted to the learning setting. Specifically, starting with the seminal work of Deep Functional Maps [186], several methods have been proposed to learn convenient descriptors for the Functional Maps framework [131, 271]. More recently, it was demonstrated in [91] that useful probe functions could be learned directly from the shape geometry (i.e., from the 3D coordinates of the points). This work has also shown that a functional map layer helps to regularize shape correspondence learning, achieving better results with less training data than state-of-the-art purely point-based methods [127]. Nevertheless, the approach of [91] is still tied to the choice of the Laplace-Beltrami eigenbasis and therefore lacks robustness and cannot be applied to point clouds. Apart from Functional Maps framework extensions, other methods learn correspondences directly from 3D coordinates of point clouds via a template shape [127] or using convolution operations on the surface [89, 120, 207, 321]. Other methods look for a canonical embedding of the input; they involve 2D images [76, 302] or 3D data [343].

Part I

Morphable Models as regularizations

In this Part, we exploit data-driven latent representations as strong regularization to solve the matching in two different scenarios. In the first Chapter, we consider the unconstrained problem of retrieving human bodies from a single-view RGB+D. In this setting, we propose a pipeline to recover body geometry and plausible limbs positioning by matching a Morphable Model to the depth information [202]. In the second Chapter, we combine the same Morphable Model with the Functional Maps matching framework, facing human body surfaces registration. The proposed pipeline [203] is entirely automatic, regardless subject's identity or pose, and working in many
challenging scenarios (e.g., point clouds, partiality, noise, topological changes). Finally, in the third Chapter, we enhance the latter method with a high-resolution augmentation strategy [204], including recent high-frequencies refinements of Functional Maps and subdivision surfaces, improving the details caught by the registration pipeline, and thus the matching quality.

3

POP: Full Parametric model Estimation for Occluded People

We propose POP, a novel and efficient paradigm for estimation and completion *of human shape to produce a full parametric 3D model directly from single RGBD images, even under severe occlusion. Our method's heart is a novel human body pose retrieval formulation that explicitly models and handles occlusion. A robust optimization then refines the retrieved result to yield a full representation of the human shape. We demonstrate our method on a range of challenging real world scenarios and produce high-quality results not possible by competing alternatives. The method opens up exciting AR/VR application possibilities by working on 'in-the-wild' human motion measurements.*

3.1 Introduction

Analysis and modeling of human shape from images and video is an widely topic across several research domains including robotics for human-robot interaction [123, 279], in pattern recognition for video surveillance and action recognition [166], in biometry for person (re-)identification and gait recognition [52, 342], and in computer graphics for authoring digital content creation [195, 305, 309].

In early efforts of human motion analysis, the overall aim was to accurately estimate 2D and, to a limited extent, 3D skeleton joint-locations as a proxy for recovering *human pose* (i.e., human skeleton) [278].

A particularly challenging scenario consists of estimating *both* human pose and shape 'in-the-wild,' i.e. when one or more people move in a very generic environment and are oblivious of the acquisition goals [278]. In this scenario, since the subjects move uninhibited, occlusions are commonly arising due to the presence of other objects or from self-occlusion (see Figure 3.1).

We investigate the above problem relying on RGBD sensors for input snapshots. The available depth information, albeit noisy, effectively avoids the scale-ambiguity prob-

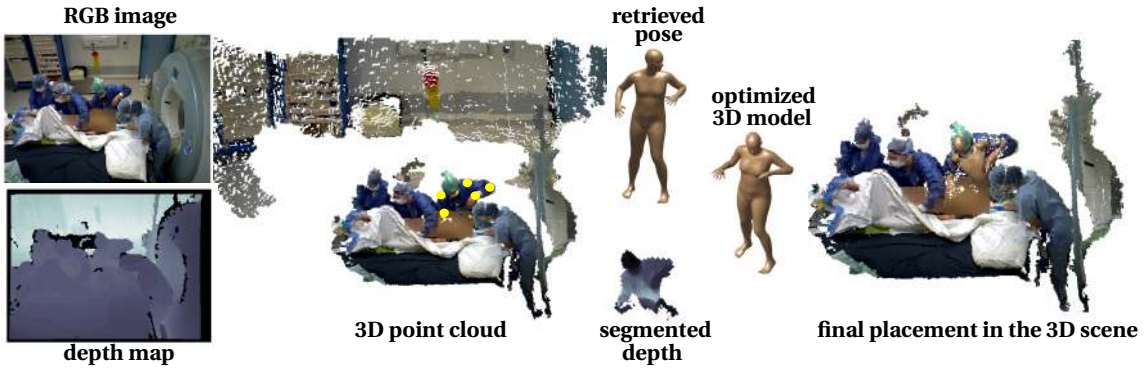


Fig. 3.1: Our estimation pipeline tested on a challenging example from the MVOR dataset [293]. From left to right: RGBD input, 2D image (top) and depth map (bottom); point cloud generated from the input and the camera parameters (top) and 3D joints of the estimated skeleton are depicted as yellow disks on the point cloud (bottom); data-driven pose initialization (top), and estimated segmentation of the depth map (bottom); model optimized on the input data; and the final model placed in the 3D space. The result is compelling for the quality of the estimation and the placement of the 3D shape, even in the presence of several challenging properties of the input.

lem encountered using single RGB images instead [345]. Further, depth helps determine the relative position between the human body and occluding objects (e.g., furniture). With this motivation, we investigate the following problem: *Given a single RGBD image of human(s) in a natural environment, obtain a full parametric 3D estimation of human shape(s), even under occlusion.*

The above problem is challenging due to three main reasons: (i) the raw input does not come with any object/human segmentation; (ii) information about which parts of human subjects are occluded and what objects cause the occlusion is unknown; and (iii) the raw RGBD scans are noisy and suffer from heterogeneous point cloud density based on camera location. We propose POP, a fully automatic pipeline that produces accurate human pose, shape and placement in the 3D space from single RGBD images, even in the presence of very significant occlusion.

Our main contributions are:

- proposing a first method explicitly designed for the analysis and modeling of human occlusion and self occlusion in single RGBD images;
- introducing a complete and fully automatic pipeline for 3D human pose and accurate full shape estimation that can deal with occlusions;

– developing an occlusion-aware shape retrieval strategy that recovers plausible information on the missing body parts provides a reliable model parameter initialization for joints location and shape, and imposes a new constraint that avoids degenerate shape on the unseen part; (iv) segmenting the human subject(s) from the rest of the scene without requiring an explicit learning procedure or involving green screens;
– hallucinating the shape of the occluded part by exploiting the data-driven prior via a novel idea akin to *null-space* that constraints the optimization procedure to reliably estimations.

We will show that the availability of parametric statistical representation is useful to formalize a prior knowledge for the domain. The geometric constraint given by this representation will overcome the under-constrained formulation of a single RGBD view. Also, the modeling of a null-space can be seen as a representation of the *absence* of geometry.

3.2 Related Works

Human body modeling is a widely studied issue over the last two decades [66, 149, 278]. In most of the proposed methods, the main objective is 3D pose estimation, i.e., location of 3D joints of the body according to a given skeleton [278]. Usually, a two-steps procedure is employed: first, joints locations are estimated on the 2D image domain, and then, 3D joints are computed using a regression approach or a model-based reprojection strategy [42, 171]. Recently, instead of relying on 2D joints estimation, direct methods have been proposed to esteem 3D pose directly from the entire image by exploiting additional information enclosed in the pixels [161].

An emerging trend is to estimate the 3D pose and the full body shape within the same framework, namely, *end-to-end modeling* methods [161, 223, 305]. The main idea consists of adopting a template-based approach estimating the shape and pose parameters of a given morphable model properly designed for human-shapes [158, 195]. Methods differ between those that use only 2D image and those that employ RGBD data [41, 66, 149, 345]. In the RGBD domain, the main effort is devoted to 3D pose estimation in real-time [345], by heavily harnessing the temporal constraint that can be introduced for video sequences [41, 44]. Other methods use multiple devices to enlarge the acquisition view and reduce the effect of occlusions (see survey [345]). In contrast, we focus on the case of recovering full human body shape from a *single* RGBD scan with background clutter (i.e., without the human body being pre-segmented) and in the presence of medium-strong *occlusion*.

Methodologically, the estimation of shape and pose is usually obtained by formulating an optimization model [42, 192]. Recently, deep neural network methods are the widest used technique [92, 161, 278, 305, 309]. This has led to very impressive results even from single 2D image at the cost of a very accurate manual annotation of 2D and 3D joint positions, foreground-background segmentation, 2D silhouettes and so on [149, 305, 310].

However, modeling occlusion directly from RGBD inputs remains a significant open challenge in this domain.

Dealing with occlusions. Although widely appreciated that human modeling can be drastically affected in the presence of occlusions and missing parts, very few works have treated this topic [279]. Some methods address this issue implicitly by imposing a pose-prior [19], by allowing only plausible poses. Similarly, learning-based approaches regularize the pose and shapes according to the examples observed during the training phase [118, 144]. These strategies can reduce the conditioning of occlusions, but they are not designed for this purpose. In [254], a method for explicitly estimating the 3D pose of occluded parts from RGBD data was introduced. The invisible joint position is predicted through a classification of the semantic label of the occluded object. An alternative for human pose estimation from partially occluded RGBD data was proposed in [13], which relies on a probabilistic occupancy grid that is exploited to identify hidden body parts. Recently, the first systematic study [279] of various types of occlusions in 3D human pose estimation has also shown that employing data augmentation with new occluded scenes improves the overall pose estimation. Finally, last year two works [132, 341] aim to analyze the human interactions with the environment, introducing spatial and semantical constraints of the given scene, that while do not explicitly address occlusions, show promising results in limiting unrealistic intersections.

Our method. To the best of our knowledge, POP is the first method that proposes an explicit strategy to estimate the full body-shape, 3D pose, and the 3D placement of the human body in the presence of strong occlusions and missing parts. These three estimations are provided consistently and at the same time. These are complete novels in literature. Our method is focused on RGBD data trying to achieve the best results from both appearance (2D) and geometric (3D) data. We propose a two-step procedure where 2D pose is estimated from RGB image while 3D pose and the full body shape are estimated from the depth map. Our 2D pose estimation is used for the initialization procedure, and in the following optimization the estimated model is free to move avoiding the conditioning of a bad starting pose. Moreover, since we evaluate the confidence of the 2D estimate, only the most reliable joints are considered. Our method fosters an optimization approach using a Convolutional Neural Network for only the

2D pose phase. We adopt a model based approach using the very popular SMPL morphable model [195]. Our strategy is data-driven since we rely on the assumption that alike occluded shape has been already observed on a dataset that is recovered through a 3D shape retrieval procedure. Similar idea was exploited in [31, 150, 220] for pose estimation only.

3.3 Method

We first provide an overview of the tools and steps of our strategy. Then, we present our data-driven initialization and how the optimization is formulated.

3.3.1 Overview

Tools. OpenPose is a fully automatic method for detecting the 2D pose of multiple people in an RGB image [63, 288, 322] wherein a non parametric pose representation, referred as *Part Affinity Fields* [63], has been proposed. This representation consists of a set of 2D vector fields, each of which encodes the orientation and the location of a limb's image. A learning strategy is adopted on the whole image with high accuracy and real-time performance. For each of these joints, a confidence value is also provided. The final full body pose corresponds to a set of labeled 2D key points as ordered joints of a human skeleton.

The SMPL model [195], already presented in Chapter 2, is a skinned vertex-based parametric model for the full human body. We recall that two different sets of parameters control pose and shape: $\theta \in \mathbb{R}^{72}$ are the pose ones defined as the relative rotation of each of 24 joints with respect to its parent in a hierarchical kinematic tree; $\boldsymbol{\beta} \in \mathbb{R}^{10}$ are the shape parameters. SMPL provides a skeleton composed of 24 joints. Of these, 15 joints can be matched with 15 joints in the OpenPose skeleton. Figure 3.2 shows the 24 joints from SMPL, the 25 joints from OpenPose, and the shared 15 joints directly used in our optimization.

The SURREAL dataset [310] is a large-scale synthetically-generated dataset of more than 6 million frames. This dataset contains realistic scenes of people that are rendered using the SMPL model with real motion capture information. For each frame, a ground truth pose, a depth map, and a segmentation mask are provided.

OpenDR [193] is an approximate and differentiable renderer (DR) that explicitly connects the relationship between the SMPL parameters and the projection of the corresponding 3D shape to a 2D image. OpenDR is publicly-available and well suited to work with SMPL model and SURREAL dataset. Starting from a shape generated by SMPL in the 3D space, with OpenDR, we associate to the shape a 2D image and a 2D depth

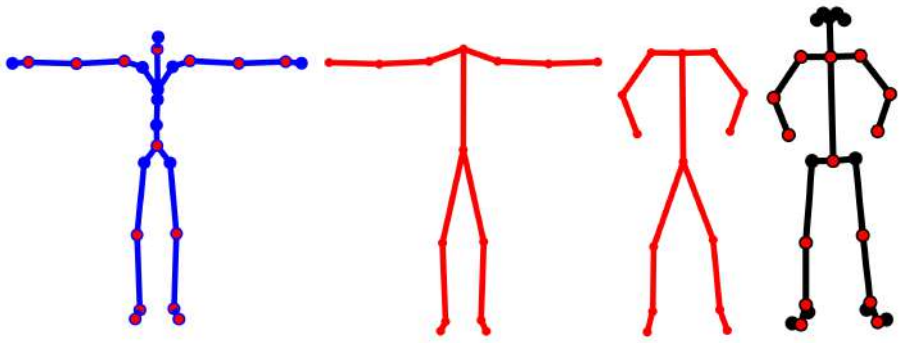


Fig. 3.2: On the left the SMPL skeleton, in the middle the shared joints and the OpenPose skeleton on the right.

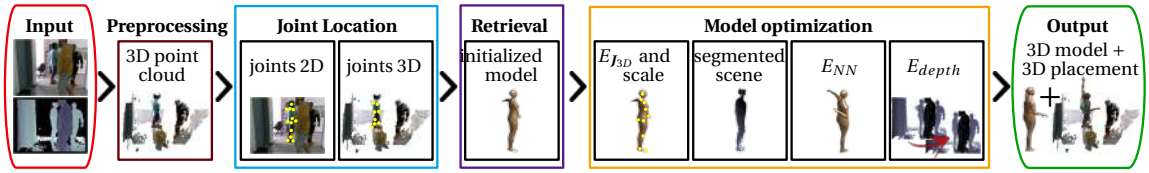


Fig. 3.3: POP pipeline. From left to right: Input (red), 3D point cloud construction (dark red), Coarse joints location and occlusion detection (light blue), Retrieval-based model initialization(purple), Model optimization (yellow) and the Output (green).

map representation of the scene. As already highlighted, the relation between the SMPL shape (i.e. its parameters) and this 2D representation is differentiable, and so can be used in an optimization pipeline.

Pipeline in brief. The entire pipeline, depicted in Figure 3.3, can be outlined as follows:

INPUT: Single RGBD image with internal camera parameters.
STEP 1: From the input depth map D_{in} and camera parameters, we estimate the point cloud PC of the scene.
STEP 2: $\boldsymbol{J}_{2D}$ a standard skeleton on the 2D image is obtained using OpenPose [63].
STEP 3: A subset of the 2D OpenPose joints are then lifted on the 3D space obtaining $\boldsymbol{J}_{3D}$.
STEP 4: We retrieve the most similar 3D skeleton with respect to $\boldsymbol{J}_{3D}$ in a subset of the SURREAL dataset and select the correspondent SMPL pose parameters $\widetilde{\boldsymbol{\theta}}$.
STEP 5: The joints of SMPL are aligned to the $\boldsymbol{J}_{3D}$ optimizing for the scale of SMPL.
STEP 6: Based on the retrieval, we segment the human body input depth $\widetilde{D}_{in}$ and the human body point cloud $H \subset PC$.

STEP 7: We iteratively optimize the SMPL parameters in order to fit the J_{3D} and the nearest neighbor energy E_{NN} between the points in H and the SMPL surface.
STEP 8: We deform the SMPL minimizing the E_{depth}.
OUTPUT: The optimized 3D model placed in the 3D scene.

We now describe each step of our method. For each choice, we explicitly clarify the respective strategy for handling occlusions.

3.3.2 Initialization

Our input is a single RGBD image with the internal camera parameters of the acquisition sensor. We use both the image representation and the 3D information in term of 3D cloud of points. We refer to D_{in} for the input depth map and PC for the point cloud. Although we now describe a single human's handling, the method can be easily iterated to deal with multi-person scenarios (see Section 3.4).

Coarse joints location and occlusion detection. We apply the OpenPose framework to the input RGB image to obtain the 2D joints of the skeleton of a human body. We use version 1.4, relying on the BODY_25 skeleton model. An example of the skeleton provided by OpenPose is shown in Figure 3.2. OpenPose returns only visible joints, which in our case are at most 25. After a re-targeting procedure between the OpenPose and the SMPL skeletons we define $\boldsymbol{J}_{2D}$ as the subset of the 15 joints of SMPL that are shared with OpenPose and visible (see Figure 3.2 where overlapping joints are marked in red). The remaining joints are classified as occluded.

Using the camera parameters we can project $\boldsymbol{J}_{2D}$ to the 3D space on the point cloud PC. However, these 3D points can be wrongly estimated due to noise and located in some inconsistent region far in the background. We compute basic statistics to detect and remove such unreliable points as outliers automatically. Indeed we obtain the set of 3D joints $\boldsymbol{J}_{3D}$ after a position refinement to accommodate a consistent skeleton.

Retrieval-based model initialization. From the SURREAL dataset, we select 1.6 millions frames from all the *run1* training set folder. We apply the same steps explained above on the input RGBD data for each of such frames, providing a coherent representation for the input data and the frames from SURREAL. We explore all these frames to find the best match for which exist a transformation in the 3D space that minimize the average of the distance between all the joints $\boldsymbol{J}_{3D}$ of the input and the 3D joints estimated on the SURREAL frame. We consider only frames that have the same visible part and therefore the same occlusion. For each considered instance i in the retrieval dataset we look for a global homogeneous transformation $\boldsymbol{T}$ composed by scale, rotation, reflection and translation given by the solution of:

$$\arg\min_{i}\Big(\arg\min_{\boldsymbol{T}}(\|\boldsymbol{T}(\boldsymbol{J}_i)-\boldsymbol{J}_{3D}\|_F)\Big), \tag{3.1}$$

where $\|\cdot\|_F$ is the Frobenius norm and $\boldsymbol{J}_i$ is the set of joints of the frame i. Note that restricting this search to the frames that share the same visible part $\boldsymbol{J}_i$) and $\boldsymbol{J}_{3D}$ are composed by the same joints thus Equation (3.1) is well defined. The solution is the index i of a frame that best matches the $\boldsymbol{J}_{3D}$ skeleton. Every frame in the SURREAL dataset is associated with a SMPL set of parameters to generate the corresponding body instance. We take those related to the solution frame retrieved by Equation (3.1) and use them to set SMPL pose parameters $\widetilde{\boldsymbol{\theta}}$.

Initialization of the SMPL parameters. From the Equation (3.1) we obtain the transformation $\boldsymbol{T}$. Applying the translation and scale components to the SMPL model, we have a good initialization in the 3D space placement. Note that the initialization $\widetilde{\boldsymbol{J}}$ obtained from the retrieval step also provides a good initialization for the occluded part. Thanks to this data-driven prior, we both avoid an implausible initialization of SMPL (that direct parameters optimization can provide) and we improve efficiency starting closer to the correct pose.

3.3.3 Model Optimization

We optimize the SMPL model in order to fit the input data. We refer to SMPL shape as $\mathscr{M}$ and to its vertices $\boldsymbol{V}_{\mathscr{M}} \in \mathbb{R}^{6890\times 3}$ represented as the collection of the 3D coordinates of its embedding.

Joints and scale optimization. Our SMPL model is initialized with the retrieved pose $\boldsymbol{\theta}$ and is placed coherently in the 3D space with respect to the $\boldsymbol{J}_{3D}$. The $\boldsymbol{J_{3D}}$ can also be involved in the optimization as a stability penalty; we force the joints of the SMPL that correspond to the joints in $\boldsymbol{J}_{3D}$ (denoted as $\widetilde{\boldsymbol{J_{SMPL}}} \subseteq \boldsymbol{J_{SMPL}}$) to remain near to $\boldsymbol{J}_{3D}$. This is expressed by the penalty term:

$$E_{\boldsymbol{J_{3D}}} = \|\boldsymbol{J_{3D}} - \widetilde{\boldsymbol{J_{SMPL}}}\|_F. \tag{3.2}$$

A first optimization is thus performed on the SMPL joints placement and on the scale of SMPL with respect to the energy $E_{\boldsymbol{J_{3D}}}$.

Constraints on the parameters. We start the optimization with strong constraints over $\boldsymbol{\theta}$ parameters because we would avoid extremely unreliable rotations. Subsequently we weaken them, increasing adherence with the seen joints.

Scene segmentation. Applying the OpenDR we obtain a synthetic depth map $D_{\boldsymbol{\beta},\boldsymbol{\theta}}$, which directly depends on the SMPL parameters. $D_{\boldsymbol{\beta},\boldsymbol{\theta}}$ and D_{in} differ for the presence

in the D_{in} of all object outside our target; while in $D_{\boldsymbol{\beta},\boldsymbol{\theta}}$ all the points that do not belong to SMPL are on the far plane, in D_{in} other objects participate. $D_{\boldsymbol{\beta},\boldsymbol{\theta}}$ can be considered as a mask of the subject, and we can apply it to D_{in}, cutting out an approximated segment for the human. To improve the approximation of this segment we analyze the neighbor of the points that belong to the human segment. Let p be one such point. We consider a 2D neighbor defined on the 2D image B_p. For all points $q \in B_p$ we have two possibilities: q belongs to the human body segments or q belongs to the background. In the first case, we assign to q its value in D_{in}. In the second case, we classify q with respect to the inequality $|D_{in}(p) - D_{in}(q)| < \gamma$ for a fixed threshold $\gamma > 0$. If this inequality holds, we assign to q the value $D_{in}(q)$, otherwise we set its value to the background. Through this procedure, we define a *clean* input depth map $\widetilde{D}_{in}$ that contains the values of the original D_{in} for all the points that are expected to belong to the human body, and the background value for the others. $\widetilde{D}_{in}$ is comparable to the artificial depth map $D_{\boldsymbol{\beta},\boldsymbol{\theta}}$ as they only describe the depth of the human body points in the scene. We refer to the human body segment in the point cloud as $H \subset PC$.

Fitting to the visible part. We compute $\pi_{\text{NN}}(V_{\mathcal{M}})$, the list of the vertices $V_{\mathcal{M}}$ obtained as the ordered euclidean nearest neighbor with respect to the points in H. Relying on $\pi_{\text{NN}}(V_{\mathcal{M}})$, we optimize first for the pose parameters $\boldsymbol{\theta}$, and then jointly for the pose and the shape ($\boldsymbol{\theta}$ and $\boldsymbol{\beta}$) minimizing $E_{\text{NN}} = \|H - \pi_{\text{NN}}(V_{\mathcal{M}}(\boldsymbol{\theta},\boldsymbol{\beta}))\|_F$.

Consistency with the depth map. To optimize the occluded body part directly in the closest plausible place, we define a *null-space*, where human body parts are *not allowed*. To do this we rely over the information from the depth map of D_{in} that is not represented in $\widetilde{D}_{in}$. It includes all objects in the environment that are possible causes of occlusions, thus it specifies all the places where the human body should not appear.

We want to exploit these elements to hide parts if this is a reliable solution. We generate the depth map $\hat{D}_{in}$ as: $\hat{D}_{in} = \text{far}$, if $D_{in}(u,v) \in \widetilde{D}_{in}$ otherwise $\hat{D}_{in} = D_{in}(u,v)$, where u and v are the image plane coordinates and *far* is the value of the far plane. Then, we minimize $E_{depth} = \|min(D_{\boldsymbol{\beta},\boldsymbol{\theta}}, \hat{D}_{in}) - D_{in}\|_F$ to have $D_{\boldsymbol{\beta},\boldsymbol{\theta}}$ approximating D_{in} by hiding part behind objects present in the scene that are nearer to the camera or exploiting the body itself. Figure 3.9 shows an example where the left arm is moved to be self-occluded by the body, and the right one is hidden behind the other person in foreground.

3.4 Results

We provide evaluations on different datasets and challenging cases highlighting the robustness to the occlusions. We omit comparison with other methods; it would be am-

biguous because POP is the first method that provides at the same time an estimation of the shape, the pose and the 3D placement of the human body shape, it relies on depth information and also aims to solve occlusions.

Datasets. We evaluate our method on different datasets, that differ for conditions and challenges. F-BODY [286], designed for human body occlusion (self-imposed or generated by people interactions). BIWI RGB-ID dataset [225] offers a variety of human shapes in similar pose and camera view. MVOR [293], a recent dataset with RGBD images in operating room. These scenes are heavily occluded and human elements are hidden from a variety of exacting factors. We select frames from other datasets to analyze different challenges: distant views [60], various occlusions and poses [13] and body shapes [291]. Finally, we test our method on frames from SURREAL providing quantitative measures that permit future comparisons.

Quantitative evaluation on SURREAL. To provide a quantitative evaluation of our method we perform an analysis on the SURREAL dataset. We select 18 frames with self occlusions from 18 different videos not used in the retrieval. We evaluate the shape and pose parameters for each frame and surface difference between the ground truth provided by SURREAL and the estimated one. The errors are computed as follows.

$$\textbf{Shape error (w.r.t. } \boldsymbol{\beta}\textbf{)} = \mathbf{err}_{\boldsymbol{\beta}} = \frac{\|\boldsymbol{\beta}_{gt} - \boldsymbol{\beta}\|_F}{\|\boldsymbol{\beta}_{gt}\|_F}. \tag{3.3}$$

$$\mathbf{err}_{\boldsymbol{J}_{SMPL}} = \sum_{j=1}^{23} \frac{\|\boldsymbol{J}_{gt}^{SMPL}(j) - \boldsymbol{J}_{\boldsymbol{\beta},\boldsymbol{\theta}}^{SMPL}(j)\|_F}{23}. \tag{3.4}$$

$$\mathbf{err}_{\boldsymbol{pose}} = \sum_{j=1}^{14} \frac{\|\boldsymbol{J}_{gt}^{3D}(j) - \boldsymbol{J}_{\boldsymbol{\beta},\boldsymbol{\theta}}^{3D}(j)\|_F}{14}. \tag{3.5}$$

$$\mathbf{err}_{\boldsymbol{pose}}^{visible} = \sum_{j \in visible} \frac{\|\boldsymbol{J}_{gt}^{visible}(j) - \boldsymbol{J}_{\boldsymbol{\beta},\boldsymbol{\theta}}^{visible}(j)\|_F}{\sharp(visible)}. \tag{3.6}$$

$$\mathbf{err}_{\boldsymbol{pose}}^{occluded} = \sum_{j \in occluded} \frac{\|\boldsymbol{J}_{gt}^{occluded}(j) - \boldsymbol{J}_{\boldsymbol{\beta},\boldsymbol{\theta}}^{occluded}(j)\|_F}{\sharp(occluded)}. \tag{3.7}$$

$\mathbf{err}_{\boldsymbol{J}_{SMPL}}$ evaluates the difference between the 24 ground truth SMPL joints and the one obtained from our optimization. $\mathbf{err}_{\boldsymbol{pose}}$ is the same restricted to the 15 joints shared by SMPL and OpenPose. $\mathbf{err}_{\boldsymbol{pose}}^{visible}$ is limited to the joints (≤ 15) that are considered as visible by our pipeline. $\mathbf{err}_{\boldsymbol{pose}}^{occluded}$ consider the joints (≤ 15) that were not found by our

pipeline. All these errors are computes excluding the root joint that only represents the placement in the 3D space. Together with these shape and pose measures we compute the normalized registration error:

$$\mathbf{err_{p2p}} = \sum_{p \in H} \frac{\| H(p) - \pi_{\mathrm{NN}}(V_{\mathscr{M}}(\boldsymbol{\theta}, \boldsymbol{\beta}))(p) \|_F}{\sharp(H)}. \tag{3.8}$$

defined through the point-to-point distances between H and registered SMPL surface. The mean and the standard deviation of these errors are reported in the Table of Figure 3.4. Except for the $err_{\boldsymbol{\beta}}$ all the others errors are reported in meters. On the right of Figure 3.4, a quantitative evaluation of the point-to-point distance between our output and H is depicted. These curves represent cumulative frequencies of the above error for each of the considered frames. For most subjects, our method stays for 90% under the threshold of 6cm of error. Although a fair comparison with other methods is not possible, we can note that our error is coherent with the declared surface error for the state-of-the-art method in [309] on the entire T1 Surreal middle frame, i.e., a less challenging scenario. In Figure 3.4, we visualize the error encoded by the heatmap; white is 0 while black represents a large error saturated to $3cm$.

	mean	std
$\mathbf{err}_{\beta}$	1.0008	0.013
$\mathbf{err}_{J_{SMPL}}$	0.1157	0.0741
$\mathbf{err}_{\boldsymbol{pose}}$	0.1515	0.0950
$\mathbf{err}_{\boldsymbol{pose}}^{visible}$	0.1335	0.0788
$\mathbf{err}_{\boldsymbol{pose}}^{occluded}$	0.2724	0.1994
$\mathbf{err}_{\boldsymbol{p2p}}$	0.0228	0.0064

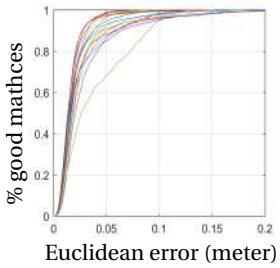


Fig. 3.4: Evaluation of our method on a set of SURREAL shapes; mean and standard deviations on the left, cumulative frequencies in the middle, a qualitative example on the right.

Qualitative pose estimation on the other Datasets. The retrieval step already provides good approximations of the $3D$ human pose, as shown in Figure 3.5, highlight the power of the data driven approach. For all the examples in Figure 3.5 we provide the final registration in Figures 3.6,3.7,3.8, showing how much the rest of the pipeline improves the results. Figure 3.9 shows the contribution of the consistency in the depth map.

Full pipeline results. We show results in a large variety of cluttering, occlusions and noisy conditions. Results in Figure 3.7 are obtained on dataset [286]. We would like to

Fig. 3.5: Some 3D pose approximations obtained from the only retrieval step. These are the SMPL initializations in our pipeline.

Fig. 3.6: An example from SBM dataset [60]. Our method offers a good solution for reconstruct group of people without ambiguity.

underline that the child in Figure 3.8 is an extreme case for the shape estimation. Finally, in Figure 3.6 we show that our method is robust also to the presence of many people and on the right of Figure 3.8 a case of a far and occluded subject.

Implementation and Timing. Both the SMPL model and the OpenDR tool are built upon a Python based autodifferentiation framework. For OpenPose, we use the free online version with the suggested parameter setting. The solution of (3.1) is solved using the *procustes* MATLAB function. Our pipeline needs around 5 minutes to produces the final 3D pose and shape estimation for a human body. We perform our experiments on an Intel 3.6 GHz Core i7-7700 cpu with 16GB RAM.

3.5 Conclusion and future work

We presented **POP**, a fully automatic pipeline for *end-to-end* modeling of human shape where RGBD data are exploited to estimate the pose and the accurate shape of a real

Fig. 3.7: Experiments from [286] dataset show different occlusions caused by external agents. Multi-person does not introduce confusion. In the middle, the arm has been placed to a different solution for the occluded part, but consistent with acquired view.

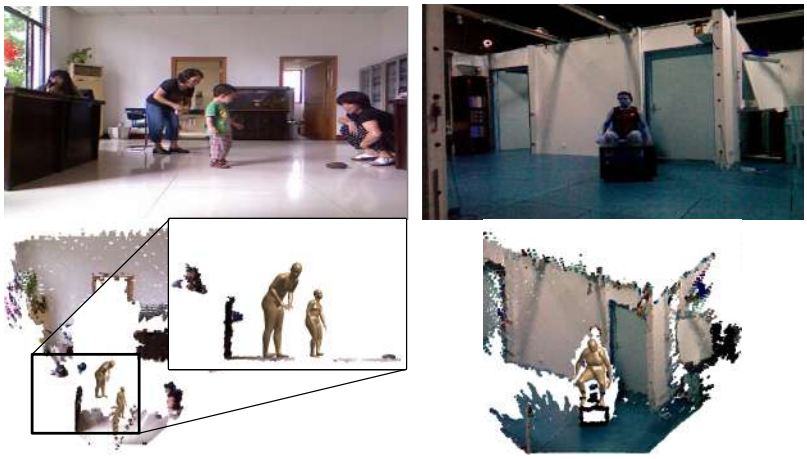


Fig. 3.8: Two results from [291] and [13] respectively. A child is an extreme case of human body shape due to his proportions. Despite this, we have a good approximation. On the right, a challenging case of a man sat far from cam and occluded by a table.

person observed on very generic scenarios (i.e., in the wild). We propose for the first time a *modeling from reality* method that is properly designed for handling occlusions. We have shown that ingredients and suggestions for modeling occlusions can be effectively employed in the proposed pipeline, from 2D joint estimation to model initialization and missing parts completion. Although the proposed method is based on the SMPL template, our approach can be naturally extended to other parametric models.

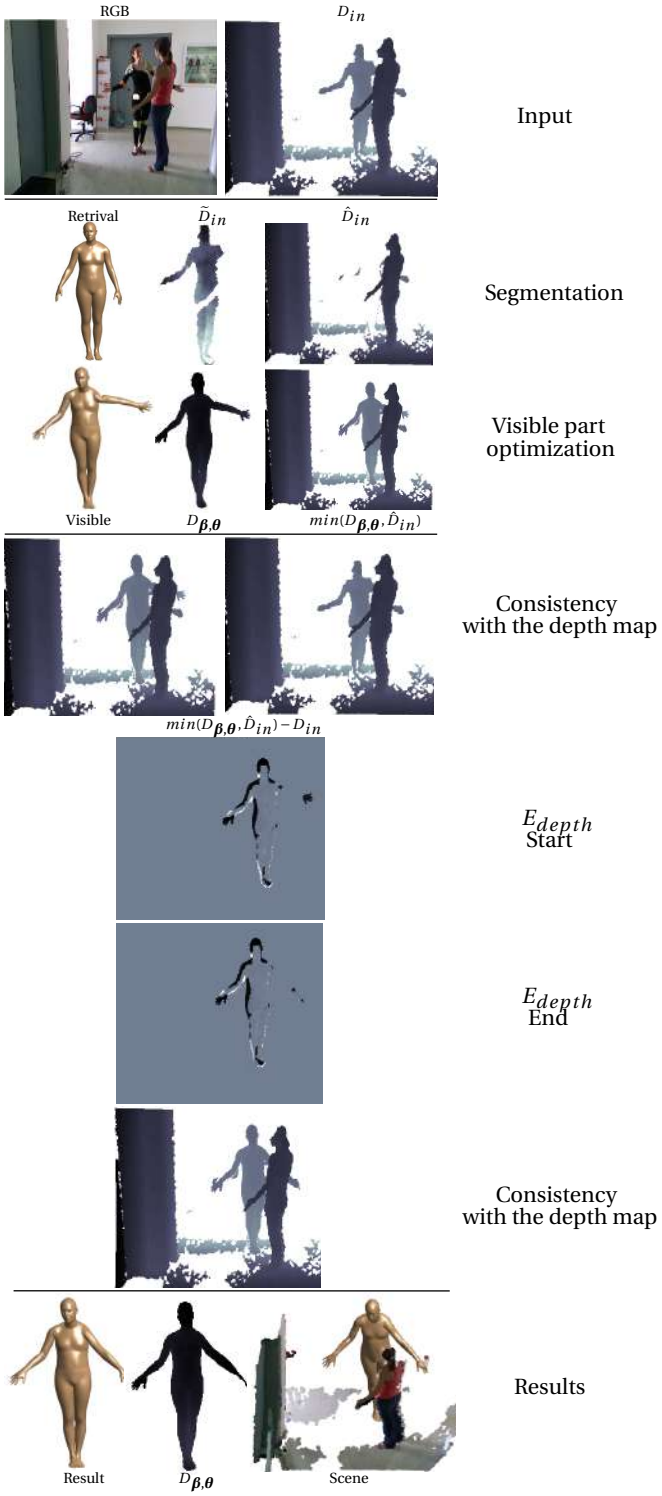


Fig. 3.9: A scheme to present our null-space formulation and its contribution in the optimization.

4

FARM: Functional Automatic Registration Method for 3D Human Bodies

We introduce a new method for non-rigid registration of 3D human shapes. Our proposed pipeline builds upon a given parametric model of the human, and makes use of the functional map representation for encoding and inferring shape maps throughout the registration process. This combination endows our method with robustness to a large variety of nuisances observed in practical settings, including non-isometric transformations, downsampling, topological noise, and occlusions; further, the pipeline can be applied invariably across different shape representations (e.g. meshes and point clouds), and in the presence of (even dramatic) missing parts such as those arising in real-world depth sensing applications. We showcase our method on a selection of challenging tasks, demonstrating results in line with, or even surpassing, state-of-the-art methods in the respective areas.

4.1 Introduction

Non-rigid 3D shape registration is a crucial problem in computer vision and geometry processing, meeting with increasing attention due to the ever growing amounts of 3D data at our disposal. It is often the case that such data derive from a sensing process, requiring an alignment step to exploit their informativeness fully. The main goal of *non-rigid registration* is therefore to determine the correct non-rigid alignment between two or more data observations. Despite much research being devoted to this issue, this problem is far from being solved.

We remark here that the problem of *registering* two shapes is slightly different than estimating a point-to-point correspondence between them (which can be seen, in fact, as a side-product of registration). Specifically, registration methods attempt to explicitly *deform* the source shape to align well with the target. Perhaps the most prominent setting in which non-rigid registration plays a key role is 3D reconstruction of deformable

objects. In this context, several partial scans must be aligned non-rigidly to obtain a single object in some canonical pose. This apparently simple task is frustratingly complex due to several reasons; first and foremost, the *partial overlap* among the scans as well as the wide variety of noise factors make this problem particularly challenging. Typical applications include semantic segmentation, motion tracking, recognition, and animation among several others [121, 229, 308].

The main focus of this Chapter is non-rigid registration of *human* shapes. Despite the less generic setting, we are here confronted with several issues: Human bodies can take countless different poses, there exists a large variety of inter-subject variations (different individuals), and humans interact with the environment giving rise to *occlusions, missing parts*, and *topological artifacts*. In order to address these issues, in this work we make use of a parametric model to which we register the observed data. Our registration method is realized as a full pipeline whose individual steps are carefully designed to maximize accuracy, consistency and robustness, and to avoid any user input. A crucial step of our pipeline relies on *functional correspondence*, which enables addressing several challenging forms of artifacts in a unified and consistent language. Importantly, our proposed pipeline is completely automatic, and performs reliably well on a range of challenging cases where other state-of-the-art approaches typically fail. We make use of a large set of different representations: spectral embedding to solve the correspondence, triangular meshes to describe the surfaces we would align, a morphable model parameters space to modify our template, and skeletons as a hierarchical hint to solve symmetries by constructing a common reference frame.

We summarize our main contributions as follows:

- Our key contribution is a novel *fully automatic* pipeline for non-rigid registration of human shapes. To our knowledge, previous approaches either require user input, or impose strong assumptions on the data initialization (e.g., prior alignment);
- we propose for the first time a *unified solution* to address missing parts, topology artifacts, different sampling, occlusions, surface noise, non-isometric transformations, which can be applied invariably to different shape representations including meshes and point clouds;
- we define a way to identify a set of consistently labeled body landmarks, which is demonstrably robust to the aforementioned types of noise. Additionally, the left/right ambiguity typically found in intrinsically symmetric shapes is completely resolved in the process.

Finally, we showcase our method on a number of emerging applications in computer vision and geometry processing, demonstrating results that outweigh the state of

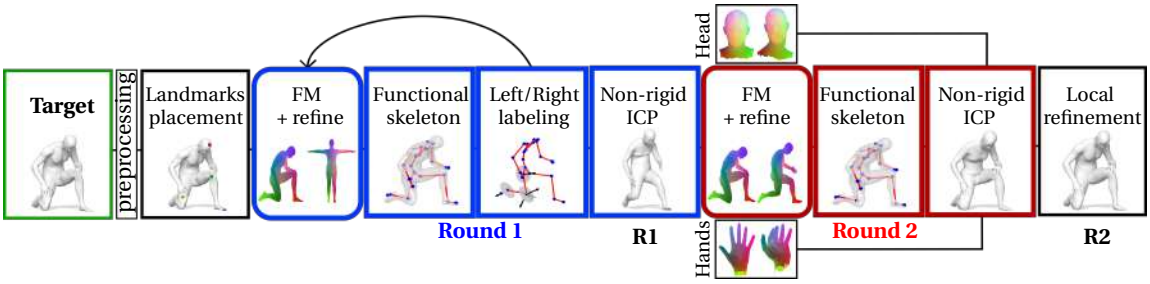


Fig. 4.1: Our registration pipeline. We refer to the main text for details on the individual steps. To get a sense of the results, compare the ***Target*** shape with the shapes in boxes ***R1*** and ***R2***. See also Figure 4.2.

Fig. 4.2: Registration results after Round 1, Round 2 and local refinement. The heatmap encodes point-to-surface registration error (expressed in cm, saturated at a maximum value of 1).

the art in several challenging settings. This Chapter is closed by novel experiments, not presented in the original paper.

In this Chapter, we leverage such flexibility to address several challenging registration scenarios in a unified manner.

We present the steps of our method as separate modules, which are then composed in a full registration pipeline. We emphasize here that our approach is completely automatic as *it requires no human supervision*; this is in contrast with many existing state-of-the-art approaches, whose initialization either relies on a set of sparse hand-picked matches, or on the assumption that the given human shapes are placed in approximate rigid alignment. A direct comparison with such approaches, with and without human supervision, will be provided in Section 4.4. The overall pipeline is illustrated in Figure 4.1.

The complete code for our method is publicly available [3].

Remark. By embracing the functional map representation, we shift the difficulty of accounting for geometry and partiality artifacts from the embedding to the functional

space, which has a vector space structure, thus allowing us to operate completely within the realm of linear algebra.

4.2 Related work

Non-rigid surface registration has attracted the attention of several researchers in the last few decades. To remain within the scope of our work, we provide here an overview of the methods that are more closely related to our approach.

Non-rigid correspondence. As already introduced in Section 2.2, the literature abounds with fully automatic or semi-automatic methods dealing with sparse or dense correspondence estimation. In [137] the authors proposed a method for registering human bodies under the assumption that the given subjects start with a similar pose; the method exploits face and ankle detection to drive the correspondence process.

In [339,347], a registration method is applied that requires a manual alignment of the human torso; similarly, [71] proposed an optimization procedure based on Markov random fields that assumes the given shapes to be pre-aligned. The method demonstrated high accuracy on a correspondence benchmark of real human shapes (comparisons with this method will be shown in the experimental section).

A data-driven approach for anthropometric landmarking was proposed in [327] by learning over a large dataset of human shapes in the same pose. Differently, in our work we extract stable landmarks over human bodies without the need for data collection, training, or human interaction, since we rely exclusively upon geometric properties in the spectral domain. Other purely geometric methods [340] that work well for human shapes assume the complete absence of topological or geometric errors, limiting their applicability to real-world data.

Body landmark detection was explored in the SHREC'14 challenge [119], showing unreliable results under strong changes in pose.

Human body registration. Various model-based techniques have been proposed in the literature. Usually high resolution templates [20] or morphable models [24,135,195] are used to register the target shape. These methods usually start by defining a pose prior under some regularization constraint and sparse correspondence; model and template are then aligned, and shape details are estimated by local non-rigid methods [148].

Such approaches, however, usually employ accurate hand-placed landmarks.

Wührer et al. [326] do template fitting based on a dataset of similar shapes; Anguelov et al. [25] enforce the preservation of a constraint over geodesic distances that fails in the presence of topological error and strong isometric distortion. A stochastic approach

Fig. 4.3: Landmarks stability under mesh perturbation. We compare our approach with heat diffusion [296], a geodesics-based approach [275], and different kernel choices for the score function (4.3). Our solution (biharmonic DEP) returns stable head/hands/feet landmarks under topological gluing, missing parts, surface noise, point cloud representation, and clean meshes (left-to-right within each mesh sequence).

is given in [347], which is based on a random particle system over a segmented template. This method represents the state of the art in the FAUST challenge [43], but it requires an initialization of the torso to fix the correct body orientation. Finally, automatic rigging methods like [33, 107, 108] are also related to our approach in that they can be seen as an *application* of the registration pipeline. As we will show in the experimental evaluation, automatic rigging for animation is but one of the many tasks that one can address with an automatic registration method at hand.

4.3 Method

4.3.1 Parametric model

Our registration pipeline employs a parametric model for the human body [20, 24], which is to be fitted to a given, possibly very noisy and deformed input observation. In this pipeline we adopt SMPL [195], already presented in Section 2.1.5. Our choice is mainly motivated by its relatively small number of parameters; together with the functional map representation, this choice endows our approach with desirable efficiency and representation compactness. To demonstrate the flexibility of our pipeline, in the experimental section we additionally show results with an alternative parametric model [247].

4.3.2 Landmarks

This module consists in identifying and labeling a sparse set of body landmarks for a given input 3D model. These landmarks are used to drive the matching process in the

subsequent steps; importantly, since our landmark extraction procedure is resilient to noise, partiality, and topological artifacts, it allows addressing several challenging cases that may arise in a practical setting.

Score function. Landmark placement is based upon the construction of a discrete-time evolution process (DEP) [213] on the mesh surface, realized by defining the recursive relations:

$$f_{(t+1)} = A f_{(t)} \tag{4.1}$$

for scalar functions $f_{(t)} : \mathbf{S} \to \mathbb{R}$ and an integral operator defined by the action

$$A f_{(t)} = \int_{\mathbf{S}} d(\cdot, y) f_{(t)}(y) dy, \tag{4.2}$$

where $d : \mathbf{S} \times \mathbf{S} \to \mathbb{R}_+$ is a pairwise potential that depends on the underlying geometry of the surface; if available, one may consider a color-based potential $d : \mathscr{C}(\mathbf{S}) \times \mathscr{C}(\mathbf{S}) \to \mathbb{R}_+$, where $\mathscr{C}(\mathbf{S})$ is a texture map for surface $\mathbf{S}$. Intuitively, the function d encodes the degree of influence that surface points exert on each other, and its selection is crucial for achieving robustness to different types of artifacts.

For a fixed number T of time steps, we consider the score:

$$s(x) = f_0(x) + \sum_{t=1}^{T} A^t f_0(x), \tag{4.3}$$

summing up the contributions of the evolution process (4.1) across all discrete times $t = 1, \ldots, T$. Here A^t denotes repeated application $A(A(\cdots(A)))$ of the operator t times. A DEP descriptor is obtained by letting $T \to \infty$ and using a multiscale approach to choose the pairwise potential, as shown below.

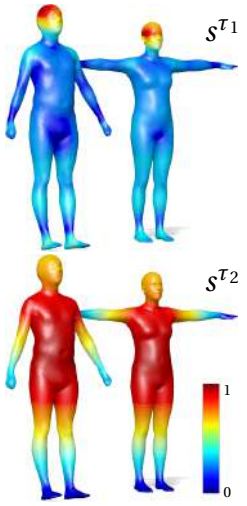
Pairwise potential. In this Chapter, we advocate the adoption of biharmonic distances [183], due to their efficiency and robustness to missing parts and resampling. When used in the definition of the score, they lead to observed resilience to inter- and intra-subject variation, partiality, surface noise and topological gluing. Our complete pairwise potential is defined as:

$$[0,1] \ni d(x,y) = 1 - \frac{d_B^\tau(x,y)}{\mathrm{diam}_B(\mathbf{S})}, \tag{4.4}$$

$$\text{where } d_B^\tau(x,y) = \begin{cases} d_B(x,y) & d_B(x,y) \le \tau \\ 1 & \text{otherwise} \end{cases}$$

and $\mathrm{diam}_B(\mathbf{S}) \equiv \max_{x,y \in \mathbf{S}} d_B(x,y)$ is the biharmonic diameter of surface $\mathbf{S}$. The thresholding operation makes d_B^τ more local, thus bringing increased resilience to partiality and topological noise.

Landmark extraction. We use a constant initial state $f_{(0)}(x) = 1 \ \forall x \in \mathbf{S}$ and distance thresholds $\tau_1 = 0.05, \tau_2 = 1$, resulting in two score functions s^{τ_1}, s^{τ_2} (depicted in the inset).



We mark the tip of the head by seeking for a local (within the region identified by s^{τ_1}) extremum of the first five non-constant Laplacian eigenfunctions; s^{τ_1} is observed to reliably correspond to the head region, while the eigenfunction extrema tend to concentrate around shape protrusions. The remaining landmarks are identified by considering the 4 clusters of points having a value of s^{τ_2} below 0.9. For each cluster, we keep the point that is farther from the head, resulting in 4 unlabeled landmarks. The hand/foot labels are assigned according to the distance to the head landmark.

We remark that at this point, although we are able to determine the correct hand/foot pairings according to the side of the body they reside in, we are not yet able to attach a *semantic* left/right labeling to them. Instead, we tentatively assign the left/right labels to the two hand/foot pairs, and we fix or invert these labels in a successive step as described in the following. See Figure 4.3 for an evaluation of landmark placement.

4.3.3 Map inference

Registering deformable surfaces entails the computation of dense maps as an intermediate step in the alignment process. We adopt the functional map representation in the Laplacian eigenbasis (Section 2.2.2), due to the guaranteed invariance to isometric transformations (changes in pose), resilience to mesh downsampling, applicability to different representations (e.g., meshes vs. point clouds), surface noise, and compactness of the resulting map representation. Further, functional maps can be robustly estimated in the presence of missing parts, clutter, and alterations of the mesh topology (e.g., "gluing" of the discrete surface around areas of self-contact). To our knowledge, there are no other methods allowing to address this variety of issues in a unified language.

Estimating a functional map. Let $\mathcal{M}$ be a fixed template (with $n_{\mathcal{M}}$ vertices) in a canonical pose, and let $\mathcal{N}$ be the observed, possibly noisy and incomplete data (with $n_{\mathcal{N}}$ vertices). We estimate a functional map $\mathbf{C}$ between $L^2(\mathcal{M})$ and $L^2(\mathcal{N})$ as the solution to non-convex problem proposed in Equation (2.25) of Section 2.2.2. A local optimum to it is obtained via conjugate gradient, and further refined with the spectral ICP-like method of [234]. In all our tests we used $k_{\mathcal{M}} = 50, k_{\mathcal{N}} = 30$, and $\lambda_1 = 0.1, \lambda_2 = 0.001$ (default values used in [233]). As probe functions f_i, g_i, for the first step we use 20-

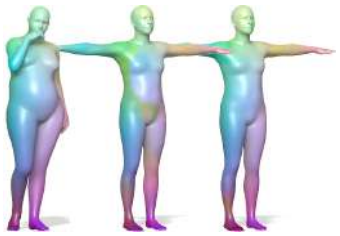
dimensional WKS descriptors [209] concatenated with 20-dimensional wave kernel maps [234] around each body landmark.

The conversion of the map to a point-to-point correspondence will be performed as described in the Equation (2.25) of Section 2.2.2.

Refinement. The goal of this module is to improve the quality of an input map by filtering out gross mismatches. We do so by considering a sequence of *convex* problems:

$$\mathbf{C}^{(t+1)} = \arg\min_{\mathbf{C}} \|\mathbf{C}^{(t)}\mathbf{F}^{(t)} - \mathbf{G}^{(t)}\|_{2,1} + \mu\|\mathbf{C}^{(t)} \circ \mathbf{W}\|_F^2, \tag{4.5}$$

with $t = 0, \dots, T$ and $\mathbf{C}^{(0)}$ being the input map to refine. If the input map has a pointwise representation $\mathbf{\Pi}^{(0)}$, it is first converted to a spectral representation by the change of basis $\mathbf{C}^{(0)} = \mathbf{\Psi}^\top \mathbf{A}\mathbf{\Pi}^{(0)}\mathbf{\Phi}$.


target noisy refined

The μ-term enforces a diagonal structure on matrix $\mathbf{C}$, where the shape of the diagonal is encoded in the "mask" matrix $\mathbf{W}$; this allows to address *partiality* by simply setting the diagonal angle of $\mathbf{W}$ according to the area ratio $\frac{\text{area}(\mathcal{N})}{\text{area}(\mathcal{M})}$ [264]. An example of map refinement is shown in the inset (corresponding points between target and model have the same color).

Remark. Map refinement works *as-is* under missing geometry and topological noise, as we will demonstrate in the experiments.

Here, as probe functions $(f_i, g_i)_{i=1}^q$ we use pairs of deltas $(\delta_{x_i}^{\mathcal{M}}(x), \delta_{\pi^{(0)}(x_i)}^{\mathcal{N}}(y))_{i=1}^q$ supported at corresponding points $(x_i, \pi^{(0)}(x_i))_{i=1}^q$ where the map $\pi^{(0)}$ is the one given as input. Input functional maps $\mathbf{C}^{(0)}$ are converted to $\mathbf{\Pi}^{(0)}$ by solving (2.22).

A crucial element of this refinement step is the adoption of the $\ell_{2,1}$ norm in the data term of (4.5). The norm $\|\mathbf{A}\|_{2,1}$ promotes column-wise sparsity for matrix $\mathbf{A}$; in our setting, it is exactly this type of sparsity that allows to filter out mismatches in the input (recall that our probe functions, which are organized as columns of $\mathbf{F}, \mathbf{G}$, are deltas supported at the input matches).

In all our tests, we used $\mu = 0.01$, $T = 5$ iterations, and $q = 1000$ delta functions supported at uniformly distributed points over $\mathcal{M}$.

4.3.4 Left/Right labeling

Resolving the left/right ambiguity typical of intrinsic methods is crucial for a successful registration pipeline.

To this end, the body landmarks are first used to solve for a low-rank functional map $\mathbf{C}$ between the parametric template $\mathcal{M}$ and the input shape $\mathcal{N}$; this is done

by solving problem (2.24). The coordinate functions of $\mathcal{N}$ (i.e., three scalar functions $f_x, f_y, f_z : \mathcal{N} \to \mathbb{R}$ encoding the x, y, z vertex coordinates of $\mathcal{N}$) are then mapped onto $\mathcal{M}$ via $\mathbf{C}$. Note that for the transport of functions a full point-to-point map is not necessary, and indeed a low-rank functional map suffices. A joint regressor is finally used on the mapped coordinates over $\mathcal{M}$, obtaining the skeleton for $\mathcal{N}$ (see Figure 4.4).

Note that, since the body landmarks do *not* at this point contain the correct left/right information, the estimated map might be either the correct one or its symmetrically flipped counterpart. In order to determine which is the case, we detect the front/back symmetry by declaring the tip of the feet (whose landmarks are at our disposal) to be front-facing, and propagate the associated versor up to the rest of the body under torque-penalizing constraints (Figure 4.4 rightmost column; for a detailed algorithm, we refer to Appendix B). The front-facing direction, together with the semantic information attached to the parametric skeleton, can then be used to attribute the correct left/right labels to the landmarks.

The labeled landmarks provide us with the necessary information to disambiguate symmetric flips (which is an *extrinsic* notion) in the estimation of the functional map. In this sense, our map inference step exploits the complementarity of the functional maps framework, which encodes intrinsic information when expressed in the Laplacian eigenbasis, and the SMPL model, which encodes extrinsic pose.

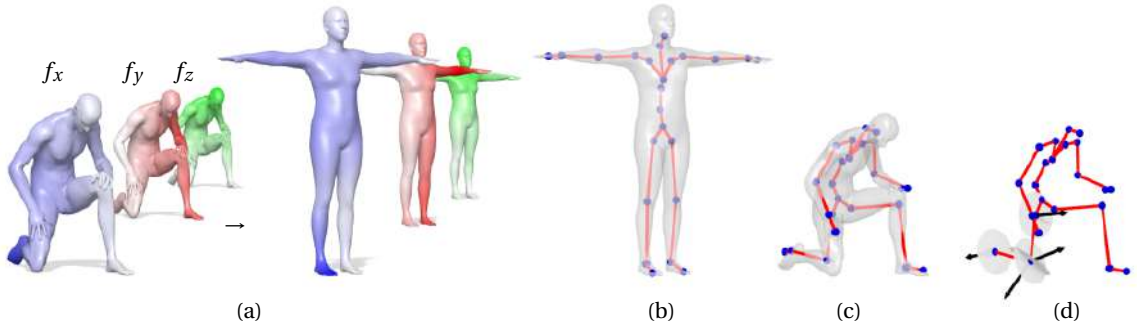


Fig. 4.4: (a) The vertex coordinate functions are mapped from shape to template via an estimated functional map; (b) a joint regressor is defined on the template, and (c) it is applied to the mapped coordinates to obtain a skeleton for the shape; (d) the front-facing direction is given by transporting the foot versor up to the rest of the body. This entire sequence is completely automatic.

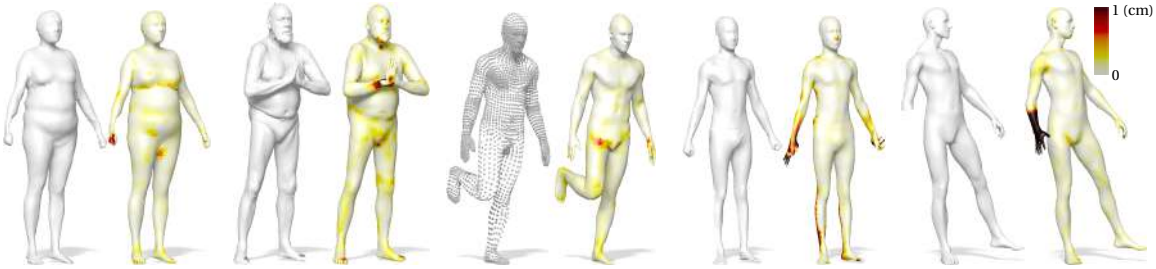


Fig. 4.5: Registration results in different settings. We plot the target surface on the left and the registered parametric model on the right. Here and for the rest of the Chapter, the heatmap encodes point-to-surface registration error (expressed in cm, saturated at 1).

4.3.5 Model fitting

Initialization. As a side-product of skeleton extraction, we have a functional map $T : L^2(\mathcal{M}) \to L^2(\mathcal{N})$ at our disposal. This is converted into a pointwise map $\pi : \mathcal{N} \to \mathcal{M}$, and the resulting point-to-point matches are used, in turn, to estimate a *rigid* alignment among the two shapes.

Shape and pose regression. We now aim to bring the template closer to the model by seeking optimal shape and pose parameters. We do not seek yet a perfect alignment at this stage since this will be refined in follow-up steps. We minimize the following composite energy:

$$\mathscr{E} = w_S E_S + w_L E_L + w_V E_V + w_\beta E_\beta + w_\theta E_\theta \tag{4.6}$$

with respect to shape $\boldsymbol{\beta}$ and pose $\boldsymbol{\theta}$ (see Sec. 4.3.1). Unless otherwise noted, for the rest of this Section we will tacitly assume that all quantities involved are functions of $\boldsymbol{\beta}, \boldsymbol{\theta}$.

The E_S, E_L and E_V terms measure respectively the alignment error (in $\mathbb{R}^3$) of the skeleton joints, body landmarks, and surface vertices of the two shapes:

$$E_S = \|\mathbf{S}_\mathcal{M} - \mathbf{S}_\mathcal{N}\|_F, \tag{4.7}$$

$$E_L = \|\mathbf{L}_\mathcal{M} - \mathbf{L}_\mathcal{N}\|_F, \tag{4.8}$$

$$E_V = \|\mathbf{V}_\mathcal{M} - \pi(\mathbf{V}_\mathcal{N})\|_F \tag{4.9}$$

where $\mathbf{S}_\mathcal{M}, \mathbf{L}_\mathcal{M}$ (resp. $\mathbf{S}_\mathcal{N}, \mathbf{L}_\mathcal{N}$) contain the 3D coordinates of skeleton joints and landmark positions for template and data shapes. Matrices $\mathbf{V}_\mathcal{M}, \mathbf{V}_\mathcal{N}$ contain the vertex coordinates for the two surfaces, and $\pi(\mathbf{V}_\mathcal{N})$ denotes the image of points in $\mathcal{N}$ under the

map π. The terms

$$E_\beta = \|\boldsymbol{\beta}\|^2, \qquad E_\theta = \mathbf{1}^\top \frac{\boldsymbol{\alpha}}{(\pi \mathbf{c}_\theta)^{12}} \tag{4.10}$$

are regularizers for shape and pose (we only care about the rotation angles $\boldsymbol{\alpha} \in \mathbb{R}^{24}$ rather than the full transformations $\boldsymbol{\theta}$), to avoid the occurrence of very large values, and thus unrealistic body shapes and poses. Note that these regularization terms help to alleviate gross registration errors caused by a possibly noisy initial map. Here, division is meant element-wise and $\mathbf{c}_\theta \in \mathbb{R}^{24}$ is a constant vector specifying motion constraints for each of the 24 joints. We use the following values: 2 for joint 0 (max freedom of movement), $\frac{2}{18}$ for hands and feet, $\frac{5}{18}$ for body joints, $\frac{1}{36}$ for head and neck.

In our tests, we set the weights $w_S = 10$, $w_L = 1$, $w_V = 0.1$, $w_\beta = 0.5$. Minimization was performed using the dogleg method [231] as implemented in the Chumpy automatic differentiation library [192].

Head and hands. At the end of the previous stage, the human template $\mathscr{M}$ is deformed in approximate alignment with the data $\mathscr{N}$. We now solve again problem (2.24) to obtain an improved functional map (note that the descriptors $f_i : \mathscr{M} \to \mathbb{R}$ are now computed on the *deformed* $\mathscr{M}$). This new map is used to obtain an improved skeleton for $\mathscr{N}$, and re-initialize the pose/shape regression step to estimate new model parameters for $\mathscr{M}$.

Differently from the previous stage, however, the energy (4.6) is modified with two additional terms that better constrain the alignment of head and hands (detected by growing geodesic balls around the corresponding landmarks). The energy update is simply:

$$\mathscr{E} + \|\mathbf{V}_{\mathscr{M}}^{\text{head}} - \mathbf{V}_{\mathscr{N}}^{\text{head}}\|_F + \|\mathbf{V}_{\mathscr{M}}^{\text{hands}} - \mathbf{V}_{\mathscr{N}}^{\text{hands}}\|_F. \tag{4.11}$$

Non-rigid ICP. Since at this stage the deformed template is expected to align well with the data, we improve the registration further by alternating between the estimation of a point-to-point map π_{NN} via nearest-neighbor search in $\mathbb{R}^3$, and minimization of the bidirectional mean square error:

$$\|\mathbf{V}_{\mathscr{M}} - \pi_{\text{NN}}(\mathbf{V}_{\mathscr{N}})\|_F + \|\pi_{\text{NN}}^{-1}(\mathbf{V}_{\mathscr{M}}) - \mathbf{V}_{\mathscr{N}}\|_F. \tag{4.12}$$

In the estimation of the map π_{NN}, we filter out point-to-point pairings that have a large discrepancy (larger than $\frac{3\pi}{2}$) in the normal directions. Note, once again, that minimization of (4.12) is done over shape and pose parameters $\boldsymbol{\beta}, \boldsymbol{\theta}$.

Local refinement. Since the parametric model can only capture shape and pose within the span of its training set, an additional refinement step is required to reach a final,

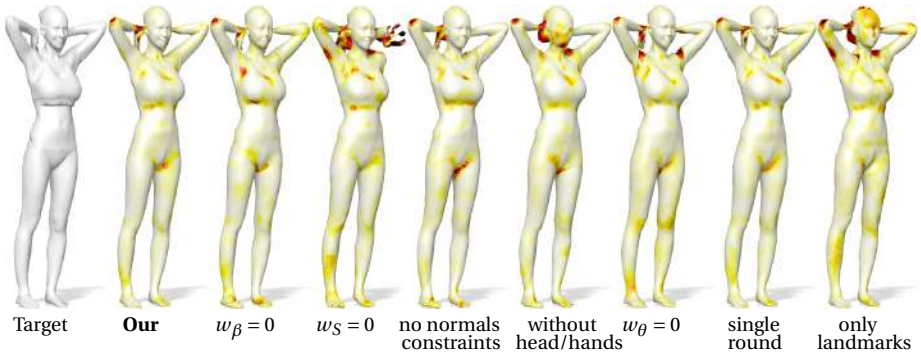


Fig. 4.6: Ablation study. See main text for details.

accurate registration. For example, the SMPL model (which we use in our experiments) does not capture head and hand articulations, while a model incorporating such details may not require refinement at this level. It is also important to note that, while artifacts are present when the hands are far from the default SMPL pose, they do not have a detrimental effect on the rest of the registration.

For the local refinement step, we employ an as-rigid-as-possible [148] in conjunction with the nearest-neighbor energy (4.12).

However, differently from all previous steps, the vertex coordinates appearing in (4.11) are now optimized directly (i.e., they are not functions of $\boldsymbol{\beta}, \boldsymbol{\theta}$).

4.4 Results

Data. In our experiments we use a wide selection of data collected from nine datasets exhibiting a variety of resolutions, sampling, surface artifacts and partiality. Specifically, we used: FAUST [43], Princeton Segmentation Benchmark [72], TOSCA [55], CAESAR [262], KIDS [268], SHREC'11 [53], SHREC'14 [244], SPRING [334] and K3D-hub [330]. All shapes were rescaled and downsampled to a similar density as the parametric model via edge collapse [115], and small artifacts were fixed using MeshFix [26].

Robustness. We first evaluate the robustness of our pipeline under challenging perturbations. In Figure 4.5 we show registration results under low resolution, topological error, point cloud representation, simulated range map, and missing parts respectively. Our pipeline achieves accurate results in all these cases; the registration error is close to zero almost everywhere, and otherwise smaller than 1cm. We refer to Figure 4.14 for additional results, including extreme settings such as clothed people.

Fig. 4.7: Registrations obtained by running our pipeline on top of the S-SCAPE parametric model. The other results in these pages employ the SMPL model.

Ablation study. We conduct an ablation study in which the main terms of our composite energy (4.6) are disabled in turn, thus allowing us to evaluate the effect of each within the registration process. Figure 4.6 shows the results on a challenging case.

Further, we show results as we change the underlying parametric model, namely by substituting SMPL with S-SCAPE [247] without posture normalization.

Within this model, pose is parametrized by 15 joints with the associated linear blend skinning weights. Since no joint regressor is provided, we define one by seeking for the minimizer:

$$\mathbf{R}^* = \arg\min_{\mathbf{R}\in\mathbb{R}^{15\times n_{\mathscr{M}}}} \|(\mathbf{W}\odot\mathbf{R})\mathbf{V}_{\mathscr{M}} - \mathbf{S}_{\mathscr{M}}\|_F^2, \tag{4.13}$$

where $\mathbf{S}_{\mathscr{M}}$ contains the 3D joint coordinates of the S-SCAPE template. The joint regressor is then defined by the element-wise product $\mathbf{W}\odot\mathbf{R}^*$, mapping surface vertices to skeleton joints.

The rest of the pipeline is applied as-is, yielding the results shown in Figure 4.7.

Finally, our pipeline involves a map inference step that can be substituted with other matching approaches. We therefore adopt the matching pipelines [71] and [18] as a plug-in replacement for our correspondence estimation step (the blue "FM+refine" block in Figure 4.1), removing Round 2 while keeping the other steps of the pipeline unchanged, and performing the complete optimization in a single round. In particular, [71] is among the state of the art for shape matching as evaluated on the FAUST challenge [43]; [18] is the only method giving guaranteed continuous bijections, but requires a sparse input correspondence (we use the five landmarks) and does not minimize metric distortion. The results are shown in Figure 4.8, highlighting the effectiveness of our entire pipeline.

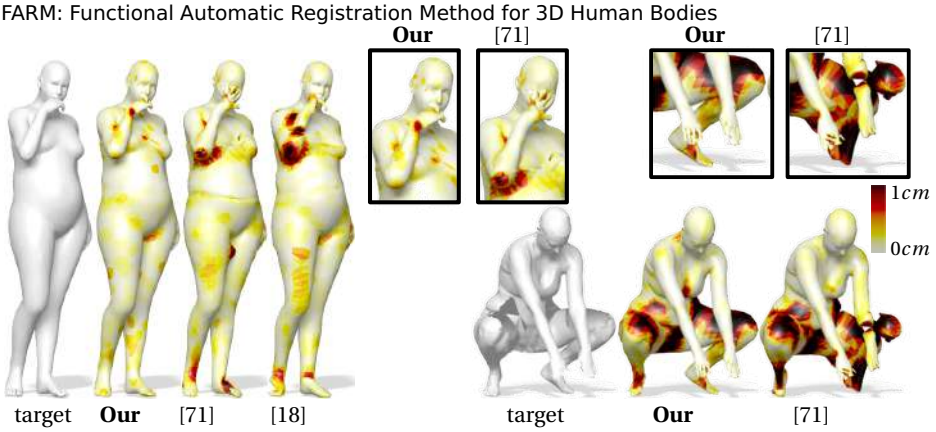


Fig. 4.8: Registration performance when our correspondence step is replaced with the matching pipelines of [71] and [18]. Note that [18] can not be applied on shapes with different genus. The black regions on the legs of the right example are due to the part being missing, as it can be seen from the target (i.e., it is not due to registration error).

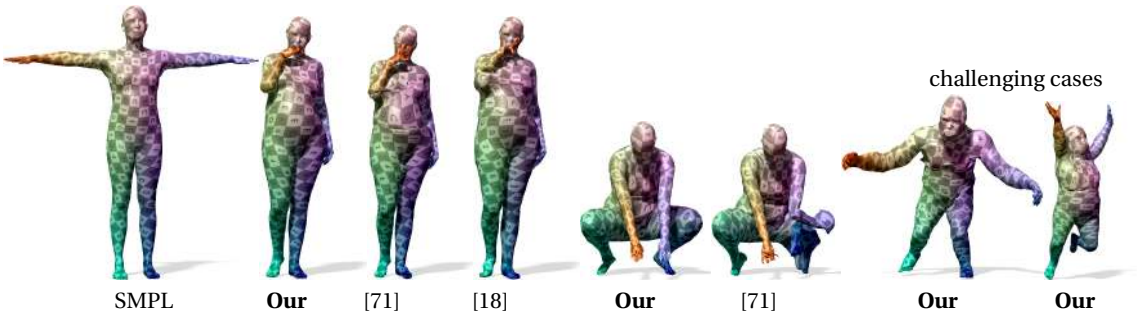


Fig. 4.9: Texture mapping visualization. Each shape is matched to the SMPL template on the left. We compare with the matching pipelines of [71] and [18], showing that our method is suitable for texture transfer. On the right we show results on two challenging cases, where we still observe coherence in the semantics of the estimated mapping.

Texture mapping. In Figure 4.9 we visualize via texture mapping the registrations compared in Figure 4.8, demonstrating comparable if not better results with respect to the competitors. This visualization includes two challenging cases, namely gorilla and kid, where we obtain reasonable results preserving mapping semantics.

4.5 Applications

We finally showcase our registration method in three different applications.

	Inter AE/WE	Intra AE/WE
Zuffi et al. [347]	3.13/6.68	1.57/5.58
Chen et al. [71]	8.30/26.80	4.86/26.57
Litany et al. [186]	4.83/9.56	2.44/26.16
Fan et al. [104]	n.a./n.a.	15.16/57.14
FARM (Ours)	4.12/9.98	2.81/19.42

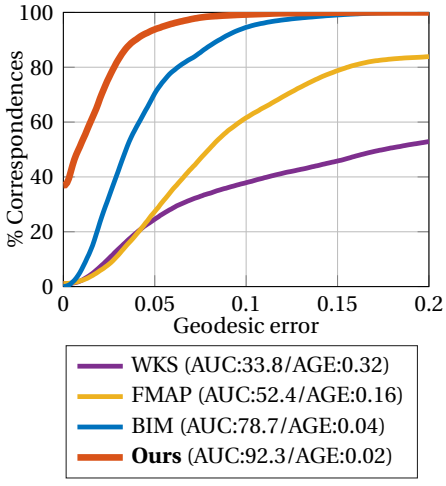
Fig. 4.10: On the left, comparison on the FAUST challenge (real scans with ~350K triangular faces). *AE* and *WE* denote average and worst error respectively. On the right, qualitative results on three pairs. These cases include: pose changes, different subjects, missing geometry, and mesh gluing. Corresponding points are visualized with the same color.

Shape correspondence. Shape registration provides point-to-point correspondences among the involved shapes as a side product. We therefore evaluate our pipeline for this task on the FAUST benchmark [43], consisting of real scans of human subjects acquired using a full-body 3D stereo capture system. These scans exhibit geometric noise, topological errors, and missing parts. The ground-truth correspondences for the challenge are *not* provided, rather an accuracy evaluation is obtained by submitting correspondence results online.

Given a challenge pair, we apply our registration pipeline to each of the two shapes individually. Once the parametric model is registered to the two shapes, we are able to establish point-to-point correspondences via this common domain and then pull them back to the original meshes. Correspondences obtained this way are used to initialize a matching step according to (4.5).

Examples of matching results and a quantitative comparison with the official ranking are shown in Figure 4.10.

Finally, in the inset Figure on the right we quantitatively compare with standard point-to-point matching pipelines between the SMPL template and five different shapes from FAUST. Since these share the same connectivity as SMPL, this provides us with ground truth for the evaluation. We compare our method with WKS [209], FMAP [233], and BIM [165] using the cumulative error protocol of [165]. The legend also reports the area under the curve (AUC) and the average geodesic error (AGE).



Shape completion. As another application, we consider the completion of partial deformable 3D shapes. To illustrate the flexibility of the registration pipeline, we look at both synthetic (artificial cropping of clean meshes) and real-world (incomplete Kinect and D-FAUST [44] scans) data; we stress that the pipeline is applied as-is in all cases, with no further adjustments or tuning to account for the challenging setting.

Results on synthetic data are reported in Figure 4.11, while in Figure 4.12 we compare with the state-of-the-art deformable shape completion method of Litany et al. [184]. Note that the latter method adopts a fully supervised deep learning model (graph convolutional autoencoders), and is limited in mesh resolution. In all these experiments, we let our parametric model assume default parameter values at the joints for the shape parts that do not have a corresponding region in the input data (these are detected automatically during the matching step).

Fig. 4.11: Deformable shape completion results. For each pair, we show the incomplete input (left) and the completed mesh (right).

Shape modeling and animation. Finally, we showcase the application of our registration method in a character animation pipeline. Once the parametric model is registered to the data, the skinning information is transferred to the latter and one can "undo" the data shape to a T-pose. From here, motion parameters can be applied to animate the character or transfer animations across multiple shapes. See Figure 4.13 for examples on full and partial data.

Beyond Humans. Recently, we have submitted the method to the 2020 SHREC challenge on shape correspondence of Physically Based Deformations [95]. The dataset

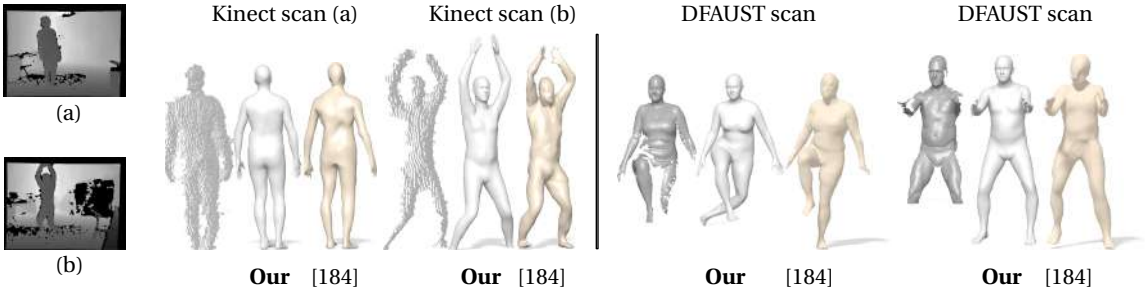


Fig. 4.12: Deformable shape completion with real scans. We compare with the deep learning method of Litany et al. [184], currently the state of the art method for this task.

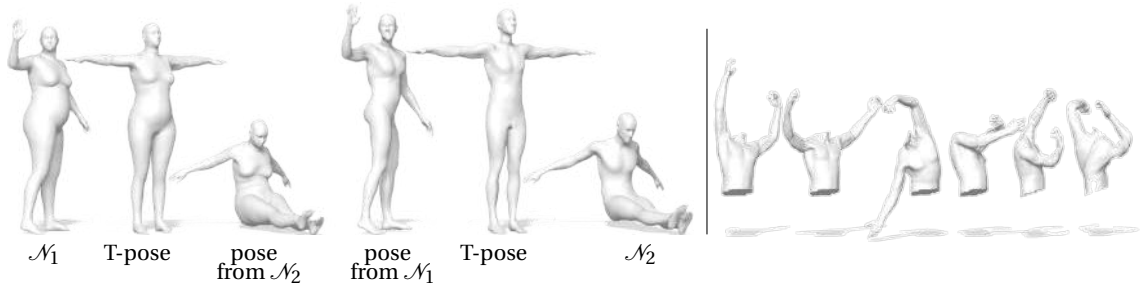


Fig. 4.13: On the left, transferring pose between two full shapes. On the right, transferring skinning information across partial shapes.

consists of a stuffed soft toy rabbit made out of stretchy jersey material with no type of internal skeleton that could otherwise restrict its movement. The purpose was to investigate how different types of physically-based deformations affect non-rigid shape correspondence, so a carefully chosen object with different material fillings is sufficient and makes data capture and analysis more manageable. We adapted FARM pipeline to work in this scenario. Since the provided template is not capable of deformation, it is animated using Mixamo [10], and some deformation basis is defined to inflate or shrink the template along the direction of the surface normals. We used the minimum and maximum of the first Laplacian eigenfunctions to classify six landmarks over ears, arms, and legs. Similarly to [204] we performed a single round and used ZoomOut refinement for the functional map (in the next Chapter we will discuss these differences). We did not include local refinement in this setting since there is no clear need for a higher detail level. All parameters were left unchanged from the original method, tuned for the specific domain of human bodies. The challenge has been performed on several other state of the art methods: Deblurring and denoising functional maps [100], Partial functional maps [265], Continuous and orientation-preserving via functional

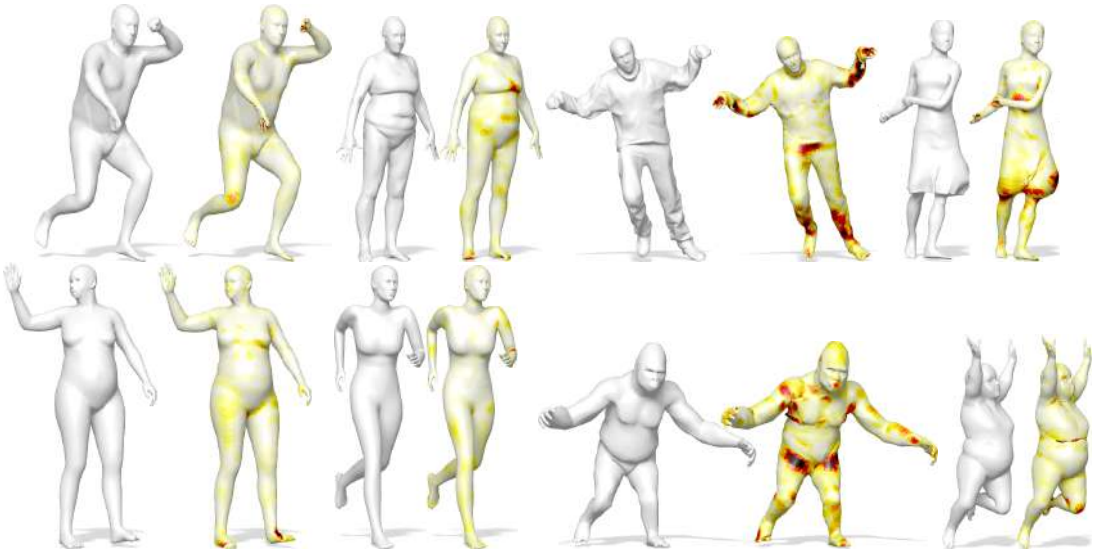
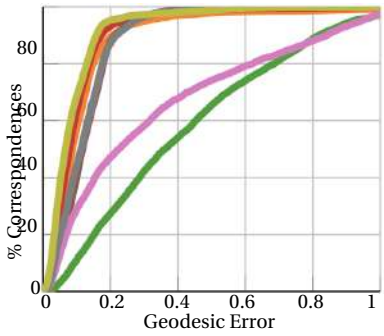


Fig. 4.14: Registration results for 8 shapes from different datasets. Note that gorilla, kid and the two clothed shapes from [317] are particularly challenging cases since they do not fall within the span of the underlying parametric model; the complete pipeline allows to obtain reasonable registrations for these cases as well.

maps [258], Dynamic 2D/3D registration for the Kinect [51], Robust non-rigid registration with reweighted position and transformation sparsity [178], kernel matching [313], Non-rigid registration under anisotropic deformations [93] and the commercial software R3DS Wrap 3. We report the overall results in Figure 4.15. Several methods start from functional map representation but relying on a template deformation provided the best overall results, even without tuning from the original FARM pipeline or design of the template. We report in Figure 4.16 a qualitative examples of FARM results. Target models present highly non-isometric deformation, poses with occlusions and gluing, and partial surface (only the front-view is available). Our results show that the pose is well recovered, while the main limitations are deformations that the template cannot model (e.g., inflating, twisting, and folds). For more details about the data and a deeper analysis of the results, we refer to the challenge report [95].

4.6 Conclusion

We presented a novel approach for the fully automatic registration of non-rigid human shapes. The main **limitations** of our method are to be found in its direct dependence on



	Pose				Internal material			
Method	Twist	Indent	Inflate	Stretch	Couscous	Risotto	Chickpea	Overall
R3DS	**0.9025**	0.8129	0.9005	0.9418	0.9265	0.8071	**0.9277**	0.8837
[100]	0.8730	0.8830	0.8689	0.9195	0.8944	0.8591	0.8979	0.8829
[265]	0.5249	0.6247	0.6136	0.5778	0.5840	0.5570	0.6170	0.5863
[258]	0.8708	0.9004	**0.9070**	0.9420	0.8956	0.8950	0.9122	0.9015
[51]	0.8620	0.8713	0.8646	0.9087	0.8842	0.8579	0.8813	0.8736
[178]	0.8598	0.8719	0.8651	0.9088	0.8822	0.8581	0.8816	0.8733
[313]	0.6369	0.7610	0.6827	0.5521	0.7660	0.5966	0.6695	0.6692
[93]	0.8616	0.8796	0.8712	0.9058	0.8886	0.8604	0.8850	0.8771
FARM	0.8999	**0.9338**	0.8780	**0.9438**	**0.9281**	**0.9111**	0.8992	**0.9113**
Mean	0.7636	0.8023	0.7967	0.7932	0.8126	0.7623	0.8008	
Std.	0.1698	0.1318	0.1307	0.2040	0.1484	0.1581	0.1551	

Fig. 4.15: The total area under the curve of scans grouped by the type of pose exhibited, scans grouped by material, and the overall performance of each method is reported. The method that achieved the best results in each configuration is emphasized in bold. In the final two rows, the mean and standard deviation of each column is reported. The curves represent the *Overall* performance of each method

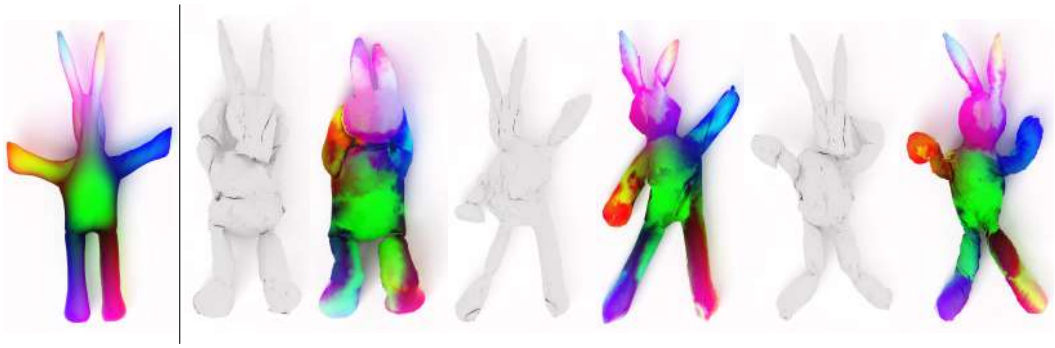


Fig. 4.16: Qualitative examples of our results on 2020 SHREC challenge on shape correspondence of Physically Based Deformations [95]. On the left, the starting template. White models are template deformed using FARM to fit the target. Colored models are the target with colored by the correspondence induced by the registration.

the underlying parametric model, which ultimately determines the quality of the final alignment, as demonstrated in dedicated tests. How the template design impacts the result is a valuable direction for future works. In Figure 4.17 we substitute the original SMPL template with a coarser version with degraded geometry (e.g., collapsed protrusions, change of proportions). The pipeline recovers several details, highlighting the limits of the template geometry. In this experiment, all other properties of SMPL (e.g. rigging system, PCA identity basis) has been preserved. Substituting them with a pure axiomatic framework like proposed in [197] would be a compelling direction, in particular in domains without enough data to obtain a data-driven morphable model. An exciting direction for future work is the introduction of localized manifold harmonics [216] in the map inference steps, which would enable the application of our method in the presence of cluttered scenes [86] without any supervision.

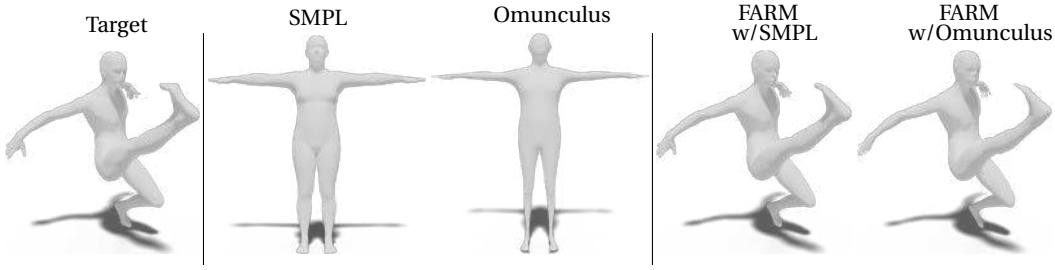


Fig. 4.17: An example of template substitution with a coarser version of SMPL. The results differs where geometry was already collapsed. Several identity and pose features are correctly recovered (e.g. legs inflated).

5

High-Resolution Augmentation for Automatic Template-Based Matching of Human Models

We propose a new approach for 3D shape matching of deformable human shapes. Our approach is based on the joint adoption of three different tools: an intrinsic spectral matching pipeline, a morphable model, and an extrinsic details refinement. By operating in conjunction, these tools allow us to greatly improve the quality of the matching while at the same time resolving the key issues exhibited by each tool individually. In this paper we present an innovative High-Resolution Augmentation *(HRA) strategy that enables highly accurate correspondence even in the presence of significant mesh resolution mismatch between the input shapes. This augmentation provides an effective workaround for the resolution limitations imposed by the adopted morphable model. The HRA in its global and localized versions represents a novel refinement strategy for surface subdivision methods. We demonstrate the accuracy of the proposed pipeline on multiple challenging benchmarks, and showcase its effectiveness in surface registration and texture transfer.*

5.1 Introduction

Accurate shape matching is an essential tool in several applications in 3D vision and graphics, including shape registration [136], pose transfer [174], shape remeshing [211] and shape modelling [295] among others. A powerful direction to solve shape matching is provided by spectral geometry processing techniques. In the spectral setting, the search for correspondences can be formulated as a matching problem in a higher-dimensional embedding space as mentioned in Section 2.2.2. However, due to their band-limited representation, such approaches often suffer from poor point-wise resolution, especially in areas of high geometric detail. To face this limitation, in the previous Chapter we proposed to combine the benefits of a 3D Morphable Model with those of functional maps, thereby unifying the intrinsic description of a surface with a

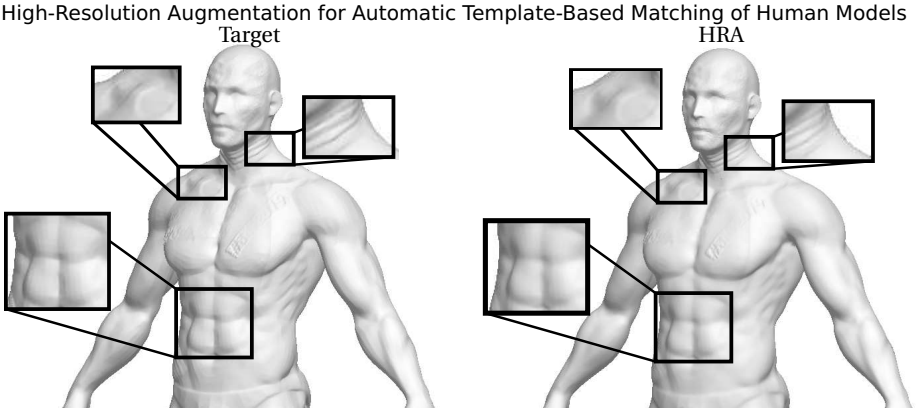


Fig. 5.1: Example of a registration result obtained with our pipeline. We show the original (target from [1]) surface on the left, and our registered result on the right. High-resolution geometric details (such as the wrinkles on the neck) are fully recovered by our pipeline, as put in evidence by the zoom-ins.

robust extrinsic regularization. By doing so, we showed significant improvement over prior work in several challenging cases.

Despite these advantages, that approach can not go beyond the resolution of the underlying parametric model, which can be a significant limiting factor whenever retaining the full resolution of the target shape is a strong requirement.

In this Chapter we propose a method to increase the resolution of the registered template, thus overcoming a key limitation of existing approaches as shown on a real example in Fig. 5.1. We show how our shape matching method can achieve very high-resolution correspondence, and we apply it to the tasks of shape registration and texture transfer. To summarize, the main contributions of our work are the following:

1. We present a novel refinement strategy for shape registration exploiting the advantages in using a surface subdivision scheme. This represents an innovative solution to face limitations imposed by the parametric model-based matching pipelines;
2. we provide insights about the capabilities of different 3D parametric models for the task of shape matching under different resolutions, showing also a dependency between discretization and registration result;
3. we propose a novel, highly accurate matching pipeline for 3D human shapes; our pipeline is especially robust to dramatic changes in *mesh connectivity*, allowing to work with different levels of detail;
4. our pipeline considerably improves performance over the state-of-the-art methods in shape matching;

The complete code for our method is publicly available [5].

5.2 Method

In the previous Chapter, we proposed an automatic pipeline for human body shape registration. Our method is based on three main steps. Given a parametric model for humans, a target body shape and five landmarks (head, hands and feet), the method allows finding an accurate functional correspondence between the two shapes. This is used to deform the template closer to the target surface, and repeat the process to improve the matching quality. The final registration is obtained after this two-stage pipeline.

The pipeline above has two major limitations: (1) The poor runtime efficiency due to the two iterative registration steps; and (2) the reliance of the overall registration quality upon the connectivity of the adopted parametric model (just about ~7K vertices in the case of SMPL). In the sequel we show how to address these limitations, at the same time yielding a significant improvement in standard applications.

One-step dense correspondence. The problem of estimating a high-quality functional correspondence was recently addressed by the elegant refinement method proposed in [214], and named ZoomOut. This method starts from a given initial (functional) matching between the two shapes $\mathcal{N}$ and $\mathcal{M}$, and iteratively performs the following two steps:

1. convert the $k_{\mathcal{N}} \times k_{\mathcal{M}}$ functional map to a pointwise map;
2. convert the pointwise map to a $k_{\mathcal{N}} + 1 \times k_{\mathcal{M}} + 1$ functional map;

The process proceeds while increasing the values of $k_{\mathcal{N}}$ and $k_{\mathcal{M}}$ at each iteration. The initial map completely determines the map obtained through this iterative approach; remarkably, it is a descriptor-free algorithm. It was further observed that, as more and more high frequencies are included in the functional map representation, the level of geometric detail accurately mapped by the estimated correspondence also increases. In our experiments, we show that it is possible to directly optimize the registration of the parametric model by applying this kind of refinement to the matching obtained in the first step of FARM. FARM also adopts a refinement strategy in order to filter out coarse mismatches. We show that, thanks to the matching step described above, this additional refinement can be avoided.

We refer to Figure 5.2 for a qualitative evaluation of ZoomOut on our data. In the figure, we compare the matching provided by the initial functional map of size 50×30, the two rounds of the refinement proposed in FARM, and the ZoomOut correspondence. The matching quality of ZoomOut can be further appreciated in comparison with the previous matching pipeline in Figure 5.6. These results confirm that an intermediate registration is not necessary to get more isometric shapes to obtain a good cor-

Fig. 5.2: A ruined FMAP correspondence in a non isometric case visualized through texture transfer. From left to right: the source texture visualized on the SMPL model; the initialization of the correspondence (Init); the map refined in the first round of FARM (R1); the map refined in the second round of FARM (R2); Our refinement (Our). R1 and Our start on the same Init correspondence.

respondence. For this reason we can avoid the two rounds estimation strategy adopted in FARM. We have empirically seen that this let us to save around 30% of time.

High-Resolution Augmentation. The main novelty of the proposed method is the High-Resolution Augmentation (HRA). To highlight the contribution of this step, we first consider standard results with a parametric model. A parametric model is nothing more than a fixed template, for which a set of parameters govern a group of deformations. Every deformation of the model is obtained over the same template, providing the same connectivity for all the generated shapes. As a direct consequence, the model has a fixed resolution in all its poses.

The recoverable details' quality is thus limited to the ones that can be represented by the parametric model connectivity. As can be seen in Figure 5.3, the registration obtained by FARM is drastically inferior to HRA; SMPL has few vertices (6890) to catch all the details. In contrast, HRA allows us to achieve a very high-quality registration that can finely reproduce all the details encoded by the scan.

HRA is applied to the template once its registration to the target shape is concluded. A subdivision method is then applied to the template (3 times recursively), obtaining a

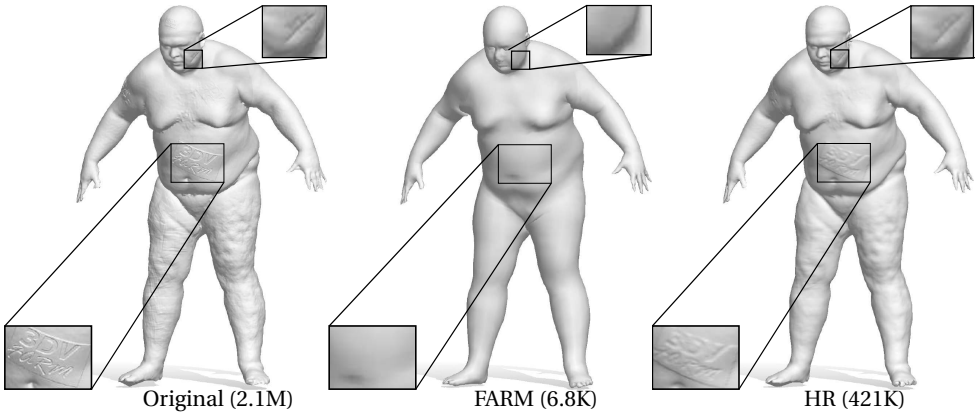
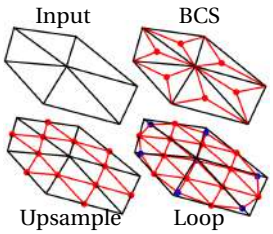


Fig. 5.3: A comparison on registering a highly detailed real scan from [44] adorned with artist details and patterns. On the left the target, (2,1 million vertices), in the middle FARM registration (6,8 thousands vertices) and on the right our result (421 thousands vertices). Using one-fifth of the target resolution, we are already able to acquire the finest details.

mesh with a larger number of vertices. We alternate these iterations with a minimization of the As-Rigid-As-Possible (ARAP) energy [148] in order to fit the extrinsic details of the geometry of the target shape.

In our experiments, we compare three different subdivision methods: the Barycentric Subidvision (BCS), Upsample [346], and the popular Loop [191]. For completeness here we briefly describe the three subdivision methods, highlighting the difference between them. A visualization of the three techniques is provided in the inset Figure, where the initial edges and vertices of the mesh are depicted in black while the newly added edges and vertices are depicted in red and blue respectively.



BCS splits each triangle by adding for each face a vertex in the position of the barycenter. The original triangle is then substituted by the three smaller triangles where each original vertex is connected to the barycentric one. We provide our implementation for BCS, which we consider as a baseline since it is the most straightforward approach among the three. The main drawback of this method is that the triangle aspect ratio increases at each iteration (i.e., the ratio between the shortest and longest edge). For this reason, the ARAP energy becomes quite unstable during our optimization because it relies on a *rigid* preservation of the edge lengths. With BCS, in the current formulation we can ap-

ply the subdivision at most two times. We also noticed that the registration results have artifacts arising from wild connectivity.

The Upsample method [346] requires to add a vertex for each edge of the mesh. In this scheme, each triangle is subdivided into four sub-triangles. If we consider the older vertices as *Even* and the new ones as *Odd*, Upsample adds the *Odds* without modifying the *Evens* positions. In this case we have a more regular mesh compared to BCS. As a drawback, we note that Upsample flattens large areas of the surface. Without smoothing, the triangles become smaller but they get stuck in their rigidity relation. We observed this method is more stable than BCS, and it allows us to perform three iterations of subdivision without energy collapse.

Finally, we tried Loop [191], the more sophisticated and popular method for surface subdivision. This approach adopts the same strategy as Upsample by adding a vertex on each edge and splitting the faces into four triangles. The new vertices are called *odd*, while the original vertices are called *even*. Then, a smoothening step is performed on all vertices, as the weighted means of their neighborhoods. This method yields stable results, does not give rise to evident artifacts, and injects non-rigid changes. Loop subdivision permits the ARAP energy to start over if it gets locked by locally strong deformations.

We select the Loop method mainly due to the latter observation and the quality of the results it can provide (see Figure 5.4 for comparisons). The final mesh produced by three iterations of Loop subdivision has a number of faces that is equal to 4^3 times the initial number of faces of the template (e.g., SMPL grows from 13,776 to 881,664 triangles).

Our approach is therefore an iterative combination of subdivision and optimization. Details are captured progressively, and the smoothness induced by Loop subdivision at each iteration allows to meet the ARAP constraints as the surface gets closer to the target geometry.

Localized High-Resolution Augmentation. As can be seen in Figures 5.1 and 5.3, most of the details presented by a shape are localized in small regions such as the face traits, the sharp abs on the belly or local pattern like cellulite. This suggests to us that a detailed refinement over the whole shape is an overkill, and the problem can be better addressed by refining only proper local regions. Local refinement raises two main problems: firstly, we need to automatically estimate the regions to apply our refinement. Secondly, Loop [191] subdivision method is not naturally applicable locally since it generates vertices over the edges that require linkage with other triangles. We propose a strategy that is aimed to solve both these problems. Our experiments found that the

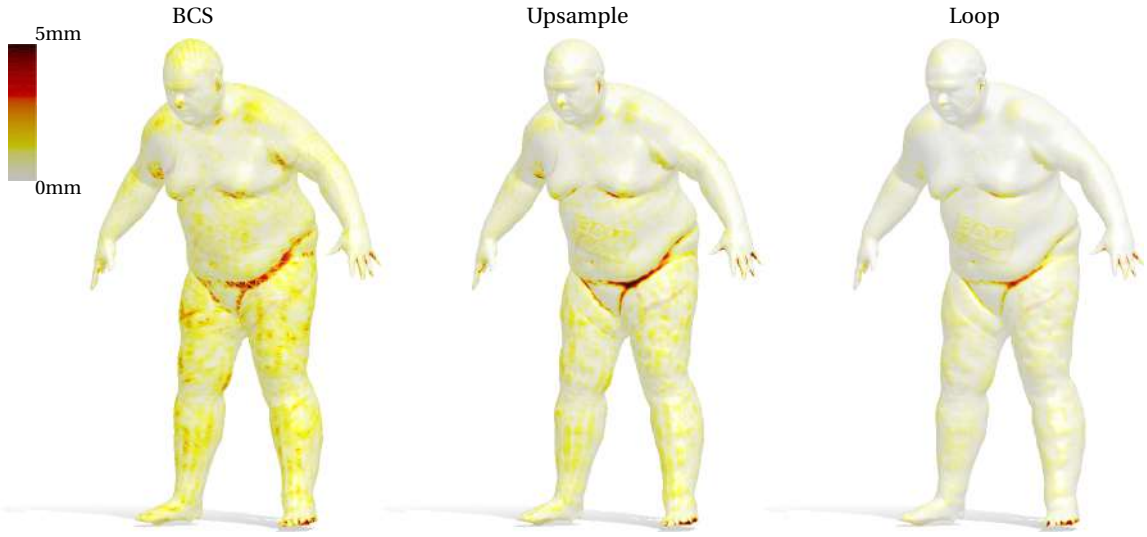


Fig. 5.4: Comparison of different subdivision methods. From left: BCS, Upsample, Loop. Baycentric is strongly penalized by its rigid structure that causes ARAP regularization to become unstable after 2 subdivision. Colors represent distance of template to target surface, with saturation at 5 millimetres.

mean curvature (H) is a good indicator for high detailed regions. We select over the target the regions where $|H| > 0.03$. Then, we project these regions over the template using nearest neighbor search, and we propagate the selection over a surrounding geodesic circumference. At this point, we would subdivide only these local patches, and then reattach them coherently to the unaffected surface. To do this, we identify the Odd vertices on the border of the subdivided patches. These are the new vertices that belong both to one subdivided face and to one that is not subdivided. We link these vertices with the *opposite vertex on the face unaffected by subdivision*, and consequently we split the face into two new faces. We need to handle the only special case when an outer triangle is adjacent to more than one subdivided triangles. When it happens, we include all these outer triangles into the subdivided region. We proceed including outer triangles until this anomaly has been fixed. We want to remark that the new surface is still in correspondence with the older one, and so with all other meshes locally subdivided in this way. This local version of the HRA provides a more efficient fitting to a given target shape's details. Also, it constitutes a new method for the local surface subdivision.

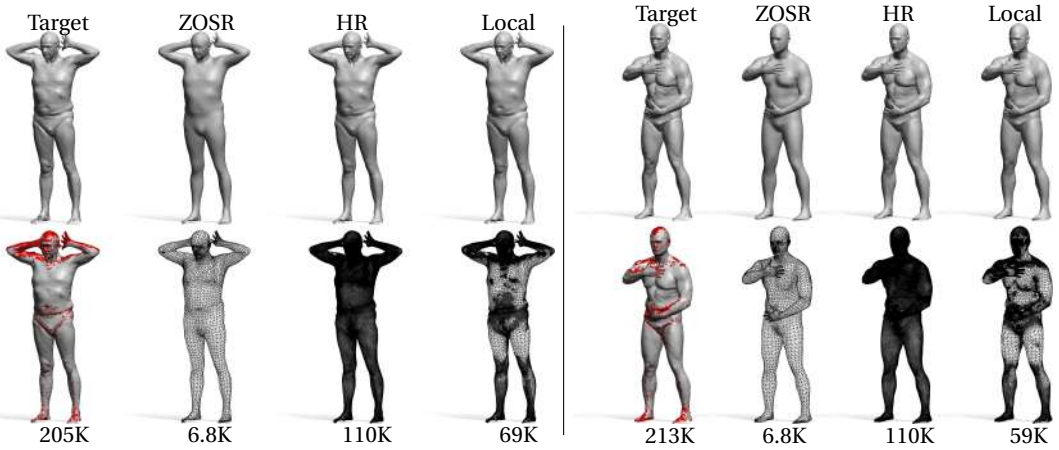


Fig. 5.5: Two examples of localized adaptive refinement. In the first column, the target and the regions with $|H| > 0.03$. Then, we show the FARM output, HR and HRA. Notice how the majority of the details (e.g. face beard) have been caught with half of the vertices.

5.3 Results

In this Section, we collect the experiments and applications of the proposed method. All the experiments are performed on MATLAB 2018, on a machine with 32GB of RAM and an Intel 3,6 GHz Core i7.

Point-to-point matching. We evaluate our method in point-to-point matching task on FAUST [43] and TOSCA [55] datasets. We evaluate both with and without the use of High-Resolution Augmentation strategy (denoted as *HR* and *ZOSR* respectively). We compare our results with 5 different state-of-the-art approaches: *RMH* [102], *PMF* [315], *BCICP* [258], *ZoomOut* [214] and *FARM* [203]. All these methods refine the same initial matching that is added to the evaluations and denoted by *Ini*. The ZoomOut matching is the one exploited by our methods for the parametric model registration. Learning based-approaches are excluded for a fair comparison. We evaluate the matching quality through the cumulative error protocol proposed in [165]. In Figure 5.6, on the left we report the average on ten pairs, each of which is composed of one shape of FAUST and the SMPL template. The considered FAUST shapes are all different subjects in different poses in order to explore the non-isometric cases. SMPL and FAUST shapes share the same connectivity thus it is possible to evaluate the matching quality. As can be seen both ZOSR and HR outperform all the competitors and, in particular, FARM. HR slightly improves ZOSR results, although SMPL owns the same connectivity of FAUST

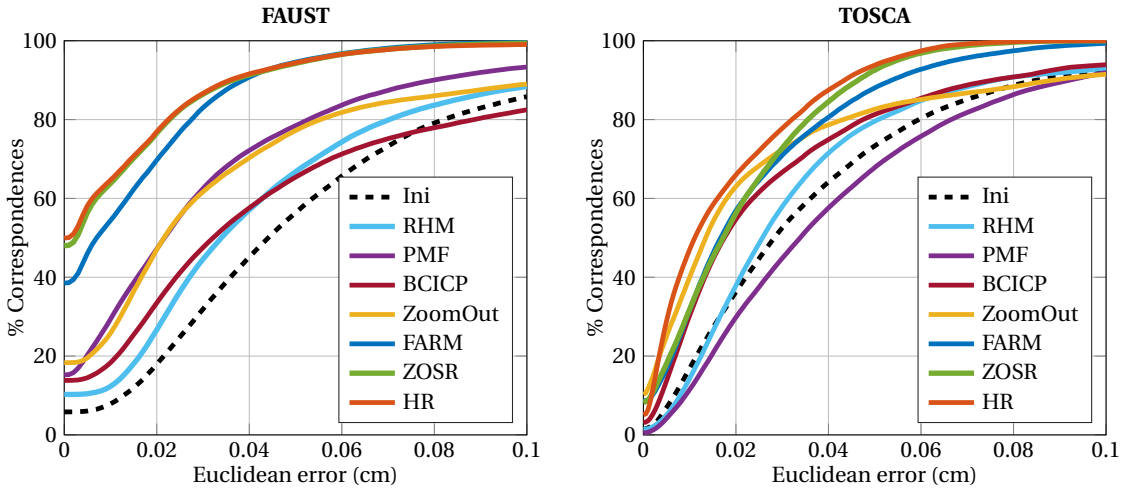


Fig. 5.6: Correspondence comparison curves over FAUST and TOSCA datasets.

and thus the HRA is not necessary. This confirms that Loop subdivision allows us to achieve better minimization of the ARAP energy.

In the same figure, we visualize the average comparison on seven pairs of the David class of the TOSCA dataset on the right. These shapes have the same connectivity and subject but broadly differ for the poses. Also in this case, ZOSR and HR outperform all the competitors. The meshes of David from TOSCA contains around 52K vertices that are many more than the 6890 of the SMPL mesh. For this reason, the improvement achieved by HR is more evident in this case. These results confirm that the HRA improves the performance of the proposed method.

Texture transfer. In Figure 5.7, we visualize three qualitative results of the proposed method in the texture transfer application underlying quantitative results showed in the previous paragraphs. We Consider three pairs of shapes with non-isometric deformations and connectivities from different datasets [24,43,194,335] . The texture transfer quality of our method can be appreciated on the fine details that we are able to transfer as the text *"Approved"* highlighted in the zoom-in of the shapes in the middle. The high resolution obtained allows us to transfer a picture of an artist as done for the pair on the right of Figure 5.7.

Human body registration. We complete the registration of a large number of shapes from various datasets, and also over some ad-hoc modified ones to test our capability in catching details. In Figure 5.3 we have a variety of different local patterns: cellulite, synthetic letters, scars and also dynamic body tissues. All these details are well repre-

sented by our method. Also, in Figure 5.4 we present a quantitative result on the same shape: the colors encode the distance between the registered template and target surface. Few points saturate the error at 5 millimeters in all three subdivision strategies; it is also interesting to notice how BCS and Upsample connectivity affect the geometry fitting (e.g., on the legs). Finally, in Figure 5.5 we show our adaptive strategy to local optimization. Our inference permits us to use just half of the vertices to obtain the same result quality.

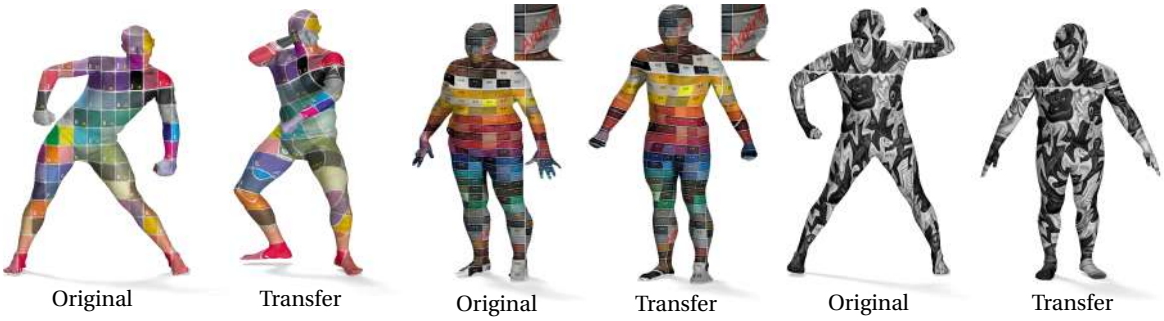


Fig. 5.7: Our qualitative results in the texture transfer application on 3 different pairs from the SHREC'19 Connectivity benchmark [212].

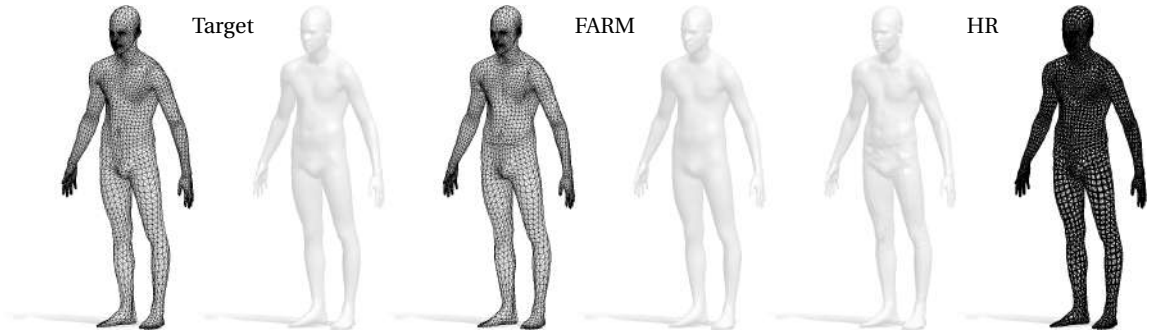


Fig. 5.8: Registration of a low-resolution mesh with our HRA method. From left: target and its wiremesh; FARM result; HR result.

Challenging cases. We also present some experiments on a few challenging cases. Firstly, we emphasize the positive effect of the HRA also in the case of low resolution shapes. In Figure 5.8, we consider a FAUST shape with only 6890 vertices, and

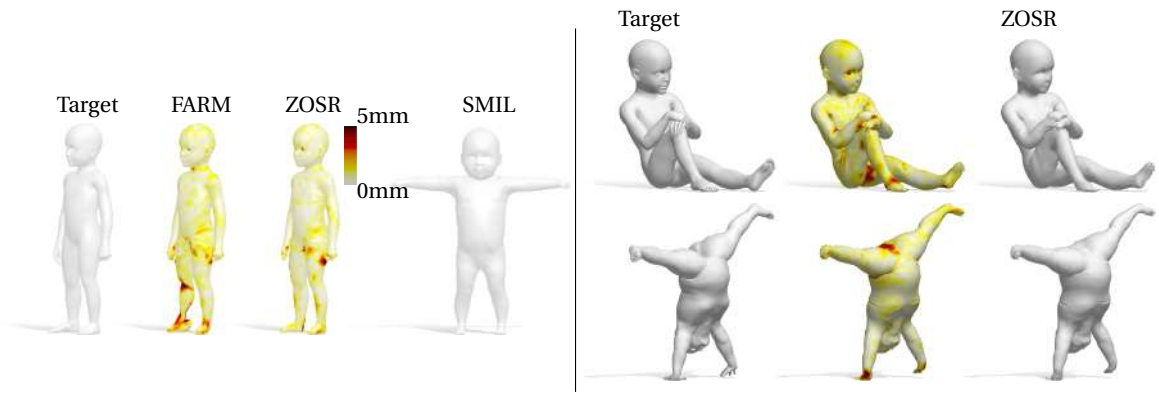


Fig. 5.9: Comparison using the SMIL morphable model. On the left: the target shape, the FARM registration, our method. Finally, the template of SMIL. Colors represent distance of template to target surface, with saturation at 5 millimetres. On the right, two more examples of our method on different poses and identities.

we apply our method together with the proposed HRA. Notice that using the proposed technique, the result is geometry sharpening and some traits become prominent. The High-resolution mesh coherently fits the original connectivity despite the large quantity of added vertices. Moreover, the HRA does not cause errors or collapses. Finally, Figure 5.9 is an original experiment: we test our method not only by changing the morphable model or domain (as shown in FARM with SCAPE and KIDS [268]), but both together. We substitute the SMPL model with the SMIL morphable model [134], a parametric model similar to SMPL where the template respects more the proportions of infant shapes instead of adult ones. The proportions of the kid shape (on the left of Figure 5.9) and SMIL template are different, thus also in this setting strong non-isometries are addressed. We correct the landmark detection heuristic in both FARM and ZOSR pipelines to perform correctly on different body proportions. All other settings are left unchanged. As can be seen, FARM fails dramatically on the right leg; it cannot be used without parameters redefinition. ZOSR performs robustly thanks to ZoomOut refinement. Furthermore, we prove that the proposed method can adopt different morphable models without additional effort.

5.4 Conclusions

In this Chapter we presented a new approach for 3D shape matching of deformable human shapes. Our approach jointly exploits a spectral matching method, a parametric model, and an extrinsic high-resolution refinement strategy. The proposed *High-*

Resolution Augmentation, in its global and innovative localized version, can fill the gap between the parametric model resolution and a general target geometry, also in the case of large mesh resolution differences. The quantitative evaluation shows that our approach outperforms the competitors on standard benchmarks. The HRA constitutes a promising solution to overcome the parametric models resolution limitations, giving rise to future directions in the high definition modeling.

We would conclude this Chapter by remarking how increasing the triangulation resolution is crucial. It not only permits to represent more details but also provide better matching. For many entertainment applications, it is usual to left high-details to textures. However, in several pure geometrical studies, this is not enough. For example, in the idea of *object replication*, being able to re-print also trademarks on objects is not negligible. Also, the experiment in Figure 5.4 highlights that not all the triangulations are equal in describing underlying geometry, and they play a fundamental role. In the next part of this thesis, we will emphasize the triangulation role, how it affects matching (particularly in the intrinsic domain), and how we can transfer triangulation to different geometries. This latter is an essential step in the process of disentangling a geometry from its representation.

Part II

Discretizations impact on Geometry

In this Part, we analyze the impact of the discretization in the geometry representation, particularly dealing with triangular meshes. In the previous Chapter, we already introduced how different connectivities impacts geometry expression. In the next one, we present a new benchmark to stress $3D$ *shape matching methods on different connectivities [212]. Our challenge shows that all the methods are favorably biased if the correspondence is established between objects that share the same connectivity, while the performance significantly degrades when this does not hold. Then, we present enrichment of the intrinsic spectral embedding with extrinsic information [211]. We use this representation enhancement to develop a mesh transfer algorithm, moving the correspondence from a discretization level to a geometric one.*

6

Matching Humans with Different Connectivity

The research community spent a lot of effort to address object matching problem, and has we have already review in Section 1.2, increased set of innovative methods has been proposed for its solution. In order to provide a fair comparison among these methods, different benchmarks have been proposed. However, all these benchmarks are domain specific, e.g., real scans coming from the same acquisition pipeline, or synthetic watertight meshes with the same triangulation. To the best of our knowledge, no cross-dataset comparisons have been proposed to date. This chapter provides the first matching evaluation in terms of large connectivity changes between models that come from totally different modeling methods. We provide a dataset of 44 shapes with dense correspondence as obtained by an accurate shape registration method (FARM). Our evaluation proves that connectivity changes lead to Objects Matching difficulties and we hope this will promote further research in matching shapes with wildly different connectivity.

6.1 Introduction

Recent technological advances provide new modeling techniques, enlarging the set of applications and involving a broader mass of consumers [242]. Modeling software enables artists to deform shapes easily, perform surface remeshing and make models ready for real-time animation [141].

Moreover, off-the-shelf sensing devices put 3D body scanning technology at the disposal of everyone [345]. These facts have led to a wide production of 3D models with different resolution (as shown in Figure 6.1), sampling density, distribution of details, noise artifacts, and so forth [43, 55, 72, 194].

This Chapter evaluates different matching pipelines and descriptors over a collection of shapes originating from a diverse set of datasets. This entails dealing with different surface discretizations, as well as other types of nuisance such as the presence of

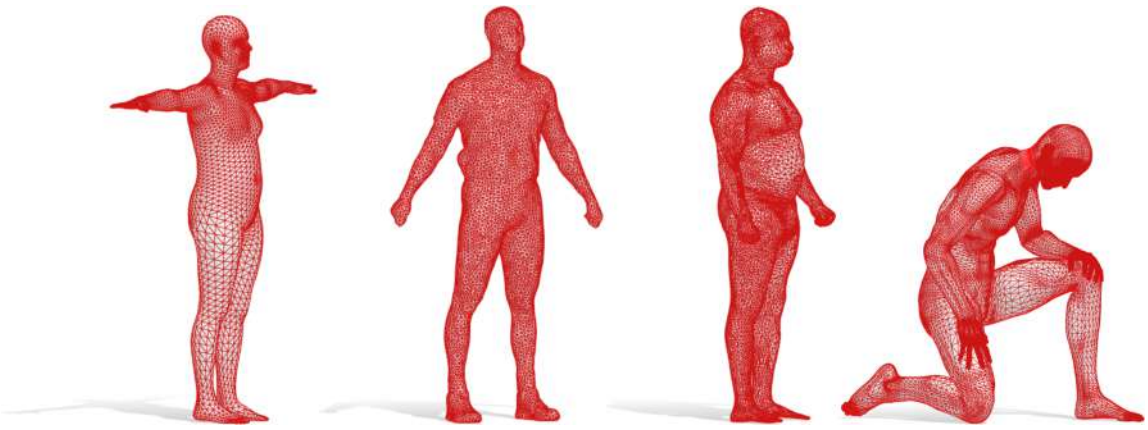


Fig. 6.1: Sample meshes involved in this track. The shapes differ in terms of mesh resolution as well as for triangles and vertex distribution. Several additional variations are also considered.

disconnected components, noisy or weakly cluttered surfaces (e.g., due to clothes and accessories), and a wide range of topology and geometry dissimilarities. The evaluated methods are compared in terms of correspondence coherence, runtime, and other implementation details. Furthermore, we test the stability and properties of different point descriptors in challenging setups, in an attempt to identify the most suitable for real-world applications. From our results, we observe that shape correspondence remains an open and challenging problem whenever connectivity changes are present in the data, and conclude that a concerted effort is required in this direction. This report is accompanied by dense ground-truth correspondences and evaluation code to foster further research [9].

6.2 Data

The dataset we provide consists of 430 shape pairs. The shapes themselves come from different sources, and dense *cross-dataset* correspondences are obtained with FARM registration method. We remark that correspondence between human bodies is ill-posed and a unanimous definition is hard to achieve; indeed, even when experts are involved, only sparse key points are provided. By using FARM, our dense correspondences adhere to the semantics of the SMPL human template [195]. The source datasets are:

SMPL [195] is the vertex-based 3D morphable model already presented in Section 2.1.5. Learned statistically over a wide population of real scan data, the SMPL model provides a simple way to generate realistic bodies that share the same connectivity.

FAUST [43] is a benchmark of real scans of different humans, coming with ground-truth correspondence for a subset. Each shape is obtained by annotating a human subject by anthropometry experts, using 17 bones as sparse key points. The SMPL template is aligned to the scans, providing a dense map. Both the aligned template and the real scans are available. This dataset provides interesting challenges, including acquisition noise, holes, and self-contact.

SCAPE [24] is a milestone in human body registration. This pipeline fits a template to real data by solving a complex optimization problem over triangle faces. The method is able to capture different body shapes, and is capable to work with different data representations (e.g., range maps and mocap markers).

TOSCA [55] high-resolution is a synthetic dataset with non-rigid deformations of shapes from different classes. All shapes have around $50K$ vertices, and models of the same class are in correspondence. TOSCA shapes are a good example of handcrafted objects.

SPRING [335] is a dataset generated by modifying a template using parameters carrying anthropometric semantics (e.g., height, calf circumference, etc.). The body shape space is learned by registering a template in SCAPE fashion to over $3K$ body models. Then, PCA is used to find directions with a clear and useful meaning. The dataset consits of a high variety of shapes in full correspondence.

MoSh Mocap [194] is a dataset produced from motion capture data acquisition, with soft tissue information yielding comparable quality with full-fledged 3D body scanners. The dataset provides clean real data with highly regular tessellation, reflecting nowadays' expectations of real acquisitions.

BadKing [1] is a website collecting contributions from professional artists which are made freely available. To our knowledge, correspondence methods have never been compared with this sort of data. The meshes present high levels of detail, disconnected components, holes, and an enormous amount of different styles.

CAESAR [262] is a rich real body scans set. It has been used widely in data-driven works, and is nowadays the baseline to learn generative human body models. Unfortunately, it is not freely available and redistribution is limited. For this reason we rely over [247]

that provides registrations of this dataset to the research community. All shapes are in correspondence and in a neutral pose.

Princeton [72] is a segmentation benchmark built on top of the SHREC 2007 Watertight Models track. Its shapes come from different sources and include synthetic human bodies with robot-like proportions as well as noisy real scans. They span both low ($\sim 4.7K$ vertices) and moderately high resolutions ($15K$ and more).

SHREC14 [244] Shape Retrieval of Non-Rigid 3D Human Models track has two subsets. A realistic one, with CAESAR shapes registered using SCAPE and remeshed to ~15K vertices; and a synthetic one with plastic poses, a smooth surface, many details and hand articulations (~60K) vertices.

K3D-Hub [330] provides a method to register a high-quality template to a low-quality Kinect scan. The low-resolution setup provides some interesting challenges: subjects may be clothed, with few details but with dense ($\sim 10K$ vertices) and regular connectivity. Matching these low-res shapes to more detailed ones may have interesting applications in entertainment.

Every data source comes with unique characteristics: different purposes require different modeling principles, affecting connectivity. With this analysis, we want to encourage the community to consider this type of tricky variations:

- *Different orientation:* the shapes in our composite dataset are *not* pre-aligned into a coherent orientation. Thus, it is no possible relying upon apriori knowledge on the position in ambient space,
- *Connectivity artifacts:* there are shapes with broken or missing connectivity (e.g., outlier points belonging to no triangle). We propose to take into consideration these scenarios.
- *Different density:* all these shapes have different discretizations. This is particularly challenging for methods that rely upon similar discretization. Artists create models with a clean and optimized meshing, with few degenerate triangles and different densities depending on the surface region. On the other hand, real scans may result from a complex surface reconstruction pipeline giving rise to degenerate triangles and non-manifold artifacts. Fitted templates either assume uniform density (e.g., SCAPE triangles are equally distributed over the surface), or provide more detail around salient points (e.g., SMPL is denser on the human face).
- *Additional variations:* we also consider variations of **identity** and **pose**, and include **different surface artifacts** such as **topological noise**, clothes, hair or accessories. We analyze both watertight meshes and meshes with disconnect components.

6.3 Descriptors

Point descriptors characterize the neighborhood of each point on a discrete surface, and are expected to be (1) discriminative (different points should have different descriptors); (2) repeatable under noise and deformation; (3) fast to compute; and (4) compact.

Given two descriptor fields $DESC_{\mathcal{M}}$ and $DESC_{\mathcal{N}}$ on shapes $\mathcal{M}$ and $\mathcal{N}$ respectively, a point correspondence for each $x \in \mathcal{M}$ can be obtained by a nearest-neighbor search in descriptor space:

$$y^* = \underset{y \in \mathcal{N}}{\arg\min} \|DESC_{\mathcal{M}}(x) - DESC_{\mathcal{N}}(y)\|_F . \tag{6.1}$$

We use the equation above in our tests. All meshes are rescaled to a similar surface area to eliminate differences caused by different scales.

GPS. The *Global Point Signature* (GPS) [272] is a point descriptor defined as the q-dimensional vector:

$$GPS(x) = [\lambda_2^{-\frac{1}{2}} \boldsymbol{\phi}_2(x), \ldots, \lambda_{q+1}^{-\frac{1}{2}} \boldsymbol{\phi}_{q+1}(x)] , \tag{6.2}$$

in our experiments, we set $q = 100$.

HKS. The *Heat Kernel Signature* (HKS) [296] is built upon the heat kernel between a point and itself, expressed in a (truncated) spectral decomposition as $k_t(x,x) = \sum_{l=1}^{K} e^{-t\lambda_l} \boldsymbol{\phi}_l(x)^2$. This can be interpreted as the amount of heat that remains at point x after a delta distribution is diffused for time t. Given a fixed set of time scales $\{t_1, \ldots, t_q\} \in \mathbb{R}^q$, the HKS at a point x is defined as:

$$HKS(x) = [k_{t_1}(x,x), \ldots, k_{t_q}(x,x)]. \tag{6.3}$$

We consider $q = 100$ as suggested in [296] and $K = 200$.

WKS. The *Wave Kernel Signature* (WKS) [209] extends the ideas above by modeling a quantum particle on the surface with a given initial energy E. The descriptor for point $x \in \mathcal{M}$ is defined as the average probability over time to find the particle at position x, and is computed as $wks_E(x) = \sum_{l=1}^{K} f_E(\lambda_l)^2 \boldsymbol{\phi}_l(x)^2$, where $f_E(\lambda_l)^2$ is log-normal energy probability distribution. Given a set of energy levels $\{E_1, \ldots, E_q\}$, the WKS is defined as:

$$WKS(x) = [wks_{E_1}(x), \ldots, wks_{E_q}(x)] . \tag{6.4}$$

We use $K = 200$ basis functions and $q = 100$ energy levels.

AWFT [215] is based on the definition of Anisotropic Windowed Fourier transform on non-Euclidean domains, and uses the Anisotropic LBO [23]. A family of such operators is defined depending on two parameters, anisotropy α and orientation θ. A Gaussian window $g^{\tau}_{x,\alpha,\theta}$ is expressed for given α,θ in the Anisotropic LBO basis, with variance $\tau > 0$, translated to each vertex and modulated with respect to the K smallest Laplacian eigenvalues. Given a scalar function $f : \mathscr{M} \to \mathbb{R}$, the coefficients of its windowed Fourier transform $(Sf)^{\tau}_{x,l.\alpha,\theta}$ are given by the inner product between f and the atoms $g^{\tau}_{x,\lambda_l,\alpha,\theta}$. Dependence on parameter l is removed in [215] via application of the total weighted power, aggregating in a single value $(S_{TWP}f)^{\tau}_{x}$ all coefficients with different modulation. For a given point $x \in \mathscr{M}$, its AWFT is:

$$AWFT(x) = [(S_{TWP}f)^{\tau_1}_{x,\alpha_1,\theta_1}, \ldots, (S_{TWP}f)^{\tau_T}_{x,\alpha_A,\theta_O}]. \tag{6.5}$$

We use $K = 200$ eigenfunctions and the parameters τ, α, θ are fixed as suggested in [215], obtaining a 100-dimensional descriptor. For increased efficiency, we remesh via edge collapse [115] all shapes with $> 60K$ vertices and then extend the matches to full resolution via nearest-neighbors in $\mathbb{R}^3$.

DEP. The *discrete-time evolution process* (DEP) [213] encodes the action of an integral operator on the surface. This action is defined on top of a pairwise potential $d : \mathscr{M} \times \mathscr{M} \to \mathbb{R}_+$ that depends on the geometry of the surface and encodes the degree of influence that surface points exert on each other:

$$Af_{(t)} = \int_{\mathscr{M}} d(\cdot, y) f_{(t)}(y) dy, \tag{6.6}$$

for scalar functions $f_{(t)} : \mathscr{M} \to \mathbb{R}$. The action of d is realized by the following recursive relations:

$$f_{(t+1)} = Af_{(t)}. \tag{6.7}$$

A score is defined for a fixed number T of time steps as $s(x) = f_0(x) + \sum_{t=1}^{T} A^t f_0(x)$, summing up the contributions of the evolution process (6.7) across all discrete times $t = 1, \ldots, T$. A DEP descriptor is obtained by letting $T \to \infty$ and using a multiscale approach on the choice of the pairwise potential d, giving $DEP(x) = [S_{d_1}(x), \ldots, S_{d_K}(x)]$. In our experiments we use the biharmonic distance [183] for the definition of d, approximated with $K = 200$ eigenfunctions and the parameters of [213], resulting in a 100-dimensional descriptor. For meshes with $> 7K$ vertices, we only consider $7K$ farthest point samples and map the matches back to full resolution through nearest-neighbors in Euclidean space.

SHOT. The *SHOT* descriptor [303] encodes histograms of normals, which are more representative of the local structure of the surface than plain 3D coordinates. This descrip-

tor is built on top of a stable *Local Reference Frame* (LRF) defined as the principal eigenvector of a modified covariance matrix around each point. An isotropic spherical grid with 32 partitions is aligned to the computed LRF, and the 3D distribution of the normals is represented as a local histogram per partition. The ordered concatenation of these histograms defines the descriptor at each point. We use the standard parameters of [303], yielding 320-dimensional descriptors.

GFrames SHOT is a variant of SHOT constructed on top of a novel, more stable LRF as proposed in [217]. The GFrames LRF is based on the computation of the gradient of a scalar function defined on the surface. By varying the scalar function, it is possible to produce several LRFs depending on the desired stability properties. Following [217], we adopt the square of the first non-constant Laplacian eigenfunction as a scalar function. We refer to the resulting 320-dimensional descriptor as GSHOT.

6.4 Matching pipelines

Functional Maps [234, 235] are based on the idea that seeking functional (as opposed to point-to-point) correspondences makes the problem independent of the shape discretization and easier to optimize. In this analysis, Following [233], we estimate a functional map $\mathbf{C}$ by solving the non-convex problem of (2.25). We use 20-dimensional WKS descriptors concatenated with 20-dimensional wave kernel maps [234] around body landmarks detected as in [203]. We set $k_{\mathscr{M}} = 60$ and $k_{\mathscr{N}} = 60$.

Iteratively Refined Functional Maps (bFMAP) follows the map refinement and estimation method of [186, 203] already presented in (4.5). As probe functions we use pairs of corresponding deltas $(\delta^{\mathscr{M}}_{x_i}(x), \delta^{\mathscr{N}}_{\pi^{(0)}(x_i)}(y))_{i=1}^{q}$, where $\pi^{(0)}$ is the point-wise conversion of $\boldsymbol{C}^{(0)}$ via (2.22). The $\ell_{2,1}$ norm promotes column-wise sparsity, allowing downweigh mismatches during the refinement process. As done in [203], we set $\mu = 0.01$, $T = 20$ iterations, and $q = 1000$ uniformly distributed delta functions over $\mathscr{M}$.

BCICP is a recent algorithm for functional map estimation, employing an *orientation-preserving* regularizer and a new refinement procedure named Bijective and Continuous ICP (BCICP).

Orientation-preserving regularizer. Given two shapes $\mathscr{M}$ and $\mathscr{N}$, and a set of q pairs of probe functions $\{(f_i, g_i)\}_{i=1}^{q}$, a data term is setup as in (2.24). Then, the following regularizer is introduced:

$$E_{\text{orient}} = \sum_{i=1}^{k} \left\| \boldsymbol{C} \circ \Omega_{f_i} - \Omega_{g_i} \circ \boldsymbol{C} \right\|_F^2, \tag{6.8}$$

where Ω is an operator that extracts the *orientation* of a local frame at each point, as encoded by the surface normal and the gradients of the given descriptors. Equation (6.8) attempts to preserve the orientation of every corresponding local frame induced by the descriptors.

BCICP refinement. Similar to ICP, the refinement alternatively solve for a point-wise map and a functional map. However, this happens *both* in the spectral domain and the spatial domain by making use of several heuristics as follows:

Continuity of the point-wise map is improved by smoothing out the displacement vector field induced by the map and filtering out the outlier regions. Assume $\pi : \mathbf{x}_i \mapsto \mathbf{y}_{\pi(i)}$ from source to target shape. To smoothen the correspondence at a vertex, we smooth out the associated displacement vector $\mathbf{t}_i = \mathbf{y}_{\pi(i)} - \mathbf{x}_i$ using the neighboring ones. Edges are classified as 'outliers' if the mapped endpoints have a large distance since they are likely to be the boundary of outlier regions; such edges are removed from the mesh adjacency matrix, and points that do not belong to the largest connected component of the modified connectivity will be regarded as outliers.

Bijectivity is improved by considering extra energies defined on the compound point-wise maps from both sides. Specifically, the original ICP uses the energy:

$$E(C_{\mathcal{MN}}, \pi_{\mathcal{NM}}) = \left\| \mathbf{\Psi} C_{\mathcal{MN}} - \pi_{\mathcal{NM}} \mathbf{\Phi} \right\|^2, \tag{6.9}$$

where $C_{\mathcal{MN}}$ is the functional map from $\mathcal{M}$ to $\mathcal{N}$, and $\pi_{\mathcal{NM}}$ is the associated point-wise map from $\mathcal{N}$ to $\mathcal{M}$. To promote bijectivity, the modified energy

$$\hat{E}(C_{\mathcal{MM}}, \pi_{\mathcal{MN}}, \pi_{\mathcal{NM}}) = \left\| \Phi_1 C_{\mathcal{MM}} - \pi_{\mathcal{MN}} \pi_{\mathcal{NM}} \mathbf{\Phi} \right\|_F^2 \tag{6.10}$$

is used, where the auxiliary variable $C_{\mathcal{MM}}$ is a functional map from shape $\mathcal{M}$ to itself. This energy helps to regularize the compound map $\pi_{\mathcal{MN}} \pi_{\mathcal{NM}}$ to be identity. A similar term for $\pi_{\mathcal{NM}} \pi_{\mathcal{NM}}$ is also added to the total energy.

Finally, map *coverage* is improved by spreading out the correspondences of vertices with a large pre-image. A vertex on the target shape is "covered" if it is the image of at least one vertex on the source shape. More discussion can be found in [258].

For the application of BCICP, meshes are downsampled to $\approx 5K$ vertices using [331]. BCICP is then executed with the default parameters and 10 iterations per pair. The estimated maps are propagated back to the original shapes by simple nearest-neighbor search; therefore, the final maps may have low coverage.

6.5 Evaluation

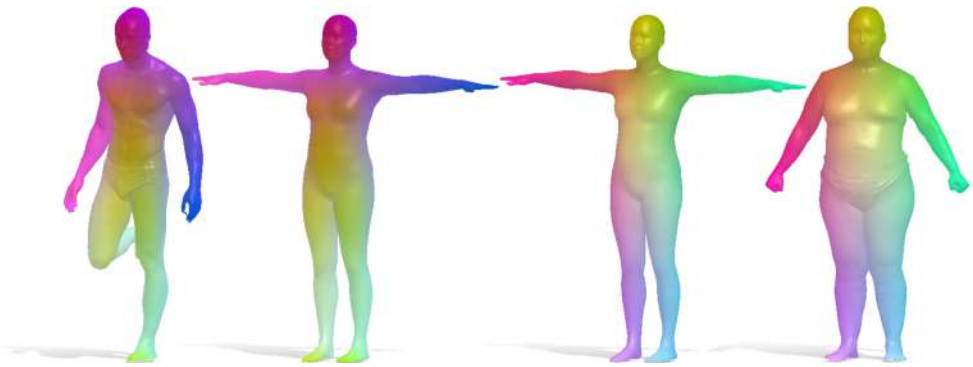
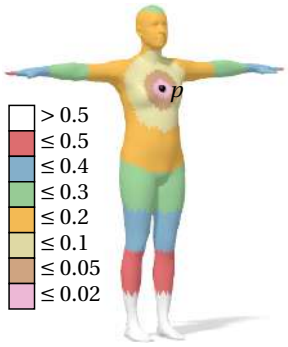


Fig. 6.2: Two examples of dense point-to-point correspondence provided by the FARM registration pipeline. The model in T-pose is the SMPL template. Correspondence is encoded by colors.

We adopt the standard error measure defined in [165]. Each method is represented as a curve denoting the percentage of correspondences (y-axis) with (normalized) geodesic error below a varying threshold (x-axis). We only plot the interval $[0,0.5]$, while the *average geodesic error* (AGE) considers the interval $[0,1]$. The normalized distances from a point p are shown in the inset figure.



Ground truth. The ground-truth is given by the state-of-the-art registration method FARM [203]. We use it to register all the shapes to SMPL, obtaining as a side-product a meaningful dense map in both directions. A map between two given shapes $\mathscr{M}$ and $\mathscr{N}$ is then obtained by composing the map from $\mathscr{M}$ to SMPL with the one from SMPL to $\mathscr{N}$ (see two examples in Figure 6.2).

Approximate geodesic error. Since our dataset includes several meshes with hundreds of thousands of vertices, we approximate geodesic error. Given two shapes $\mathscr{M},\mathscr{N}$ and a correspondence:

- We only consider the subsets of 6890 vertices of the SMPL model registered via FARM, denoted by $\mathscr{M}_v$ and $\mathscr{N}_v$.
- We denote by $\widetilde{\mathscr{N}_v}$ the estimated matches for the points in $\mathscr{M}_v$.
- We compute distances between the 6890 vertices of $\mathscr{N}_v$ and the 6890 vertices of $\widetilde{\mathscr{N}_v}$ using Dijkstra's algorithm.

– For points located on disconnected components (e.g. due to partiality or accessories) we use Euclidean distances.
– Distances are normalized to within $[0,1]$.

6.6 Results

The dataset of 430 pairs is partitioned into separate (and possibly overlapping) subsets described in the following.

Others vs. SMPL. We test the capability to map different human bodies to a common template. Here we measure the stability to noise over the *source* shape (43 pairs).

SMPL vs. Others. This measures how noise over the *target* shape affects the method's performance (43 pairs).

with SMPL. The data here is a combination of pairs from the two previous sets. This simulates a more realistic setting in which source and target do not have any a-priori role (86 pairs).

Others vs. Others. The SMPL template never appears; hence one cannot rely on any mesh regularity expectation (344 pairs).

Different Connectivity. This category is the core of our challenge. Differently from the *Others vs. Others* experiment, we do not allow pairs from the same dataset. Therefore, *all* shape pairs have different connectivity (415 pairs).

Different Connectivity plus Symmetry. Same as above, but for each point, we consider correct both the ground-truth correspondence and its symmetric counterpart (415 pairs).

Same Connectivity. With this reduced set, we evaluate how much methods improve by exploiting this assumption (15 pairs).

Same Connectivity plus Symmetry. same pairs evaluated for the Same Connectivity case, but also considering the symmetric points of the ground-truth correspondence as correct (15 pairs).

All pairs. The complete dataset, unifying all previous experiments. In Figure 6.5, we povide a visual summary of this set (430 pairs).

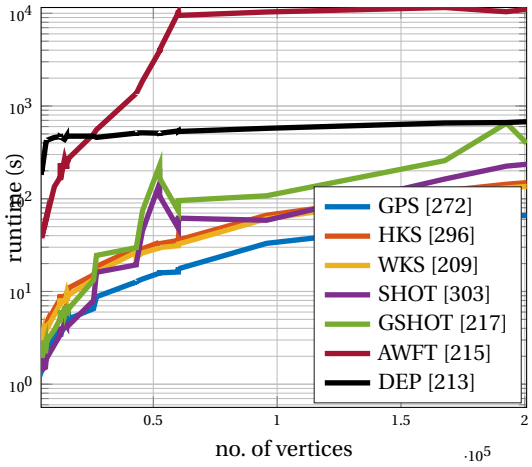


Fig. 6.3: Runtime comparisons for descriptor computation.

6.6.1 Comparisons

All descriptors were computed on an Intel 3.6GHz i7 CPU with 32GB RAM. In Figure 6.3, we measure time from shape loading to descriptor storage for each shape and descriptor. FMAP has an average runtime of 28s, with a large standard deviation of 22s (worst case 151s). The average runtime for bFMAP is 427s. BCICP requires 150s of preprocessing, and 100 – 300s for map estimation.

Descriptors. In Figure 6.4 we compare the descriptors of Section 6.3 in all the settings detailed above. Overall, the worst results are obtained by GPS, while WKS is consistently the best except for the *Same Connectivity* case. HKS is second best, followed by AWFT. We found that the latter seems quite sensitive to the specific setting, with variations in quality even among *SMPL vs Other* and *Other vs SMPL*. **All methods perform significantly better in the case of same connectivity**. DEP seems the most sensitive overall, with a dramatic drop in accuracy according to the symmetric evaluation. Table 6.1 (left top) reports a summary quantitative evaluation, largely confirming the remarks above. The largest error is observed for pairs that involve mesh n. 40 (depicted in the bottom corner of the same Table), as also confirmed by the analysis on the 5 pairs with the largest AGE (right top).

In Figure 6.5 we further plot the complete set of curves (one per shape pair) for each method. We find this visualization informative, as the curves for the SHOT, GSHOT, HKS, and WKS descriptors exhibit less spread and are more concentrated around their mean, while for AWFT, GPS, and DEP the curves are less repeatable.

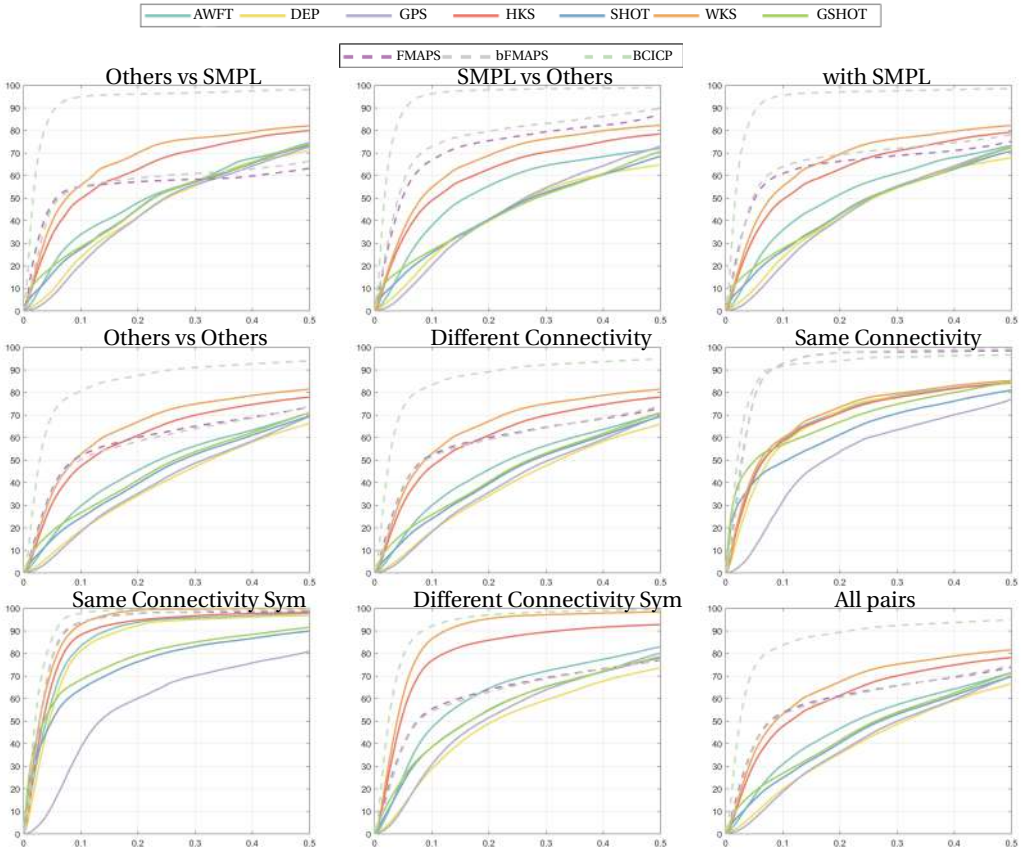


Fig. 6.4: Descriptor and matching pipelines comparisons. Overall, we observe that shapes with the same mesh connectivity tend to induce better correspondences.

Matching pipelines.

In Figure 6.4 we also report comparisons for the matching pipelines, where BCICP comes out as the best performing method. We attribute this gap in performance to its special regularizers, and further note that the accuracy of BCICP does not directly depend on mesh resolution, since it mainly relies on geodesic distances that are not affected too much by changes in mesh connectivity. bFMAP outperforms standard FMAP on the pairs involving the SMPL shape, while it does not improve the other cases. This is mainly due to its optimized parameters for the registration performed by FARM, which operates toward the SMPL template. A quantitative comparison between the three pipelines is better summarized in Table 6.1 (left bottom). In addition, in the same Table (right bottom) we show comparisons on the 5 pairs with the worst AGE. Differently from the case of descriptors, however, here we do not observe any consistently dif-

Descriptor	Avg AGE	Max AGE	Max AGE Pair
AWFT	0.33	0.50	27_40
DEP100	0.38	0.73	10_16
GPS100	0.35	0.63	22_40
HKS100	0.25	0.49	37_40
SHOT	0.35	0.47	18_40
WKS100	0.21	0.54	1_40
GSHOT	0.34	0.49	39_18
Corr Methods			
FMAP	0.27	0.75	39_10
bFMAP	0.27	0.75	34_10
BCICP	0.08	0.57	12_40

Pairs	Mean AGE	AWFT	DEP100	GPS100	HKS100	SHOT	WKS100	GSHOT
35_40	0.45	0.50	0.39	0.56	0.37	0.42	0.52	0.40
27_40	0.44	0.50	0.43	0.57	0.35	0.41	0.44	0.40
24_40	0.44	0.44	0.37	0.56	0.39	0.43	0.52	0.40
22_40	0.44	0.47	0.34	0.63	0.39	0.43	0.44	0.39
31_40	0.43	0.43	0.41	0.57	0.35	0.45	0.39	0.42

Pairs	Mean AGE	FMAP	bFMAP	BCICP
38_3	0.62	0.74	0.74	0.39
38_9	0.60	0.71	0.72	0.37
34_13	0.59	0.69	0.69	0.40
32_26	0.57	0.74	0.74	0.21
21_1	0.56	0.73	0.72	0.24

Table 6.1: On the left, descriptor (top) and matching pipelines (bottom) results, reporting the shape pair achieving the max AGE in the right column. On the right, two tabels reporting descriptors and matching pipelines on the 5 pairs with largest AGE. In the image, a comparison between the high-resolution, non-uniform and partial mesh 40 and the SMPL template mesh.

ficult shape. Finally, in Figure 6.5 we plot curves for all the shape pairs for each matching pipeline. BCICP exhibits significantly less variance, confirming the good quality of the correspondences. We further note how the bFMAP curves are more concentrated toward the top of the graph than the standard FMAP pipeline, confirming its better behavior.

In Figure 6.5 (bottom right) we also plot the distribution of shape pairs (430 in total, x axis) at increasing average geodesic error (AGE, y axis). The mean AGE over all methods is shown in blue while the minimum and maximum AGE are depicted respectively as green and red shaded areas. The vertical blue lines identify shape pairs with the same connectivity: these are mainly located on the lower end of the graph, meaning that estimating point-to-point correspondences for such cases is more manageable than for cases with different connectivity.

6.7 Conclusion

With this Chapter, we compared point-to-point matching algorithms for human shapes represented as triangular meshes *with different connectivity*. We demonstrate that the

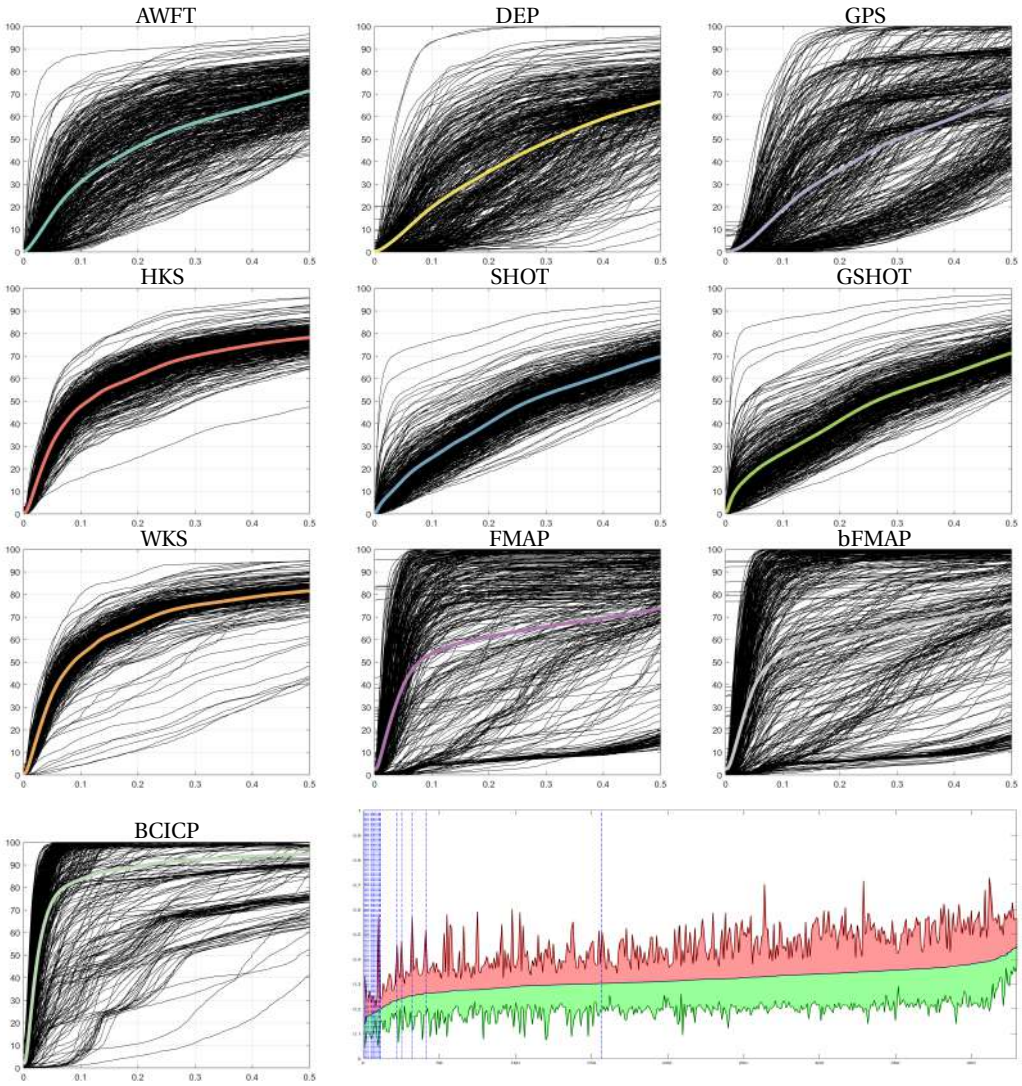


Fig. 6.5: Full comparisons on the entire dataset. Every black curve corresponds to a shape pair (430 per subplot); the colored curves represent each method's mean. In the bottom right, we show the AGE (y axis) over all pairs (x axis); the blue curve is the mean AGE across all methods, while the red and green areas denote min and max AGE. Vertical dashed lines identify the pairs with shared connectivity.

recent BCICP pipeline and standard descriptors such as HKS and WKS are more stable to connectivity variations, which still pose a strong challenge to the shape analysis community. We conclude that, differently from standard practice, past and future

matching methods should be conceived and evaluated for their robustness to connectivity changes. Finally, while we only considered human shapes (as a consequence of using an accurate, although model-specific registration pipeline [203]), we conjecture that our remarks on the relevance of connectivity in matching tasks may still hold for more generic shape classes.

7

Intrinsic/extrinsic embedding for functional remeshing of 3D shapes

3D acquisition pipeline delivers 3D digital models accurately representing real-world objects, improving the geometric accuracy and realism of virtual reconstructions. However, even after intensive clean-up, the captured models fall short of many of the requirements imposed by the downstream application, such as video-games, virtual reality, digital movies, etc. Often, the captured 3D model can only serve as a starting point for a cascade of subsequent phases, by either digital artists or geometry processing algorithms, such as a complete remeshing (or retopology), surface parameterization, skinning for animation, and so on. In contrast, we propose a novel remeshing-by-matching *approach, where we automatically combine the accurate* 3D geometry *of the captured model with the tessellation of a target pre-existing template which already satisfies all the professional requirements. At the core of this process, there is a matching strategy based on the functional mapping framework. To this end, we introduce a new set of basis functions designed for this context: termed* Coordinates Manifold Harmonics *(CMH). We evaluate this strategy (quantitatively and qualitatively) over models of different classes, obtaining a favourable comparison with existing methods.*

7.1 Introduction

Digital 3D models can often be captured from reality using acquisition systems such as range-scanning, shape-from-motion, and others. The high geometric accuracy offered by modern 3D acquisition technologies improves the coherence between the real-world objects and their virtual counterparts. The acquired 3D data can be automatically post-processed to remove defects (such as holes or other inconsistencies) resulting, in the ideal case, in a "clean" polygonal mesh. Even then, this mesh is unsuitable for direct use in downstream applications, such as real-time rendering, animation, simulation, or to undergo re-editing operations.

Captured models are inherently different from those crafted for movies or video-game production pipelines. Their meshing is triangle based and irregular; they present a low-quality UV-map or no UV-map at all; they have no other attribute apart from pre-shaded colors; the vertex density is usually both roughly constant and too dense; the mesh does not match the shape symmetries; and so on. Crafted 3D models, conversely, are semi-regular quad meshes, with irregular vertices carefully placed in appropriate locations; their edge orientation is optimized for shape representation and animations; they present edge-flows favoring re-editing; symmetries are exploited, when available; they present carefully optimized and customized UV-maps; they feature adaptive resolution (vertex sampling is denser in semantically important regions); they are enriched with a variety of useful per-vertex attributes, e.g. links to bones which make them ready for animation, and so on.

A variety of automatic Geometry Processing algorithm can fill this gap. Primarily, *surface remeshing* to improve the quality of the polygonization, but also, among others: *mesh simplification* to reduce and control polygonal complexity, *mesh parametrization* to improve UV maps. Automating these tasks is challenging due to the variety and subtleties of the objectives to be fulfilled. Despite the progress in all these areas (see Section 7.2), the quality obtained by skilled artists is unmatched. Consequently, in the industry, captured objects routinely undergo intensive manual labor before they are usable, starting with a complete redefinition of the connectivity (a process termed "re-topology").

As an alternative, we propose a new *remeshing-by-matching* approach. We assume that a manually crafted model is available featuring a reasonably similar overall shape, and all the desired characteristics, except accurate geometric adherence to the real-world. This assumption is reasonable in many contexts; for example, all humans characters share a roughly similar shape and so do all quadrupeds (such as cats or dogs). Our idea consists of automatically combining in one output mesh both the accurate geometry of a captured model and the high-quality meshing and other desirable structural characteristics of a crafted model. Per-vertex attributes defined on the latter (such as UV-map or skinning) are also inherited when they are reusable. Our output is a morph of the crafted mesh, which assumes the geometry of the captured shape.

To implement this idea, we need an automatic and accurate estimation of point-to-surface correspondence. We propose a workflow to meet this challenge (Section 7.3) where an initial estimation is refined in a subsequent phase. The initial estimation (Section 7.3.2) is based on the functional map framework, which is state of the art for non-rigid shape matching (see Section 7.2). We improve over it by introducing a set of orthonormal basis functions, which we term *Coordinates Manifold Harmonics* (CMH) (Section 7.3.1); they ameliorate the efficiency and reliability of the functional map esti-

mation by integrating extrinsic geometric information with the standard intrinsic spectral shape processing. In the refinement phase (Section 7.3.3), we maximize the local rigidity of the mapping with a local-global approach, thus inducing better preservation of local geometric features. The code of our method is available online [7].

7.2 Related works

Different basis for Functional Maps. Hamiltonian operators have been proposed as an alternative to the Laplace-Beltrami basis [75,216]. In [232], the Fourier basis is extended with all the set element-wise products, without requiring any additional optimization.

Regularized Principal Component Analysis (RPCA) [17] obtains a function basis which is well suited for shape processing by leveraging PCA and regularizing it according to the Laplace-Beltrami operator. RPCA is similar to our CMH in that it combines both intrinsic and extrinsic properties in one hybrid basis. However, RPCA depends on a statistical analysis of a set of input shapes (for which the one-to-one correspondences are sought). Conversely, our CMH definition relies only on a single shape, making it better suited for our targeted scenario. Moreover, RPCA is designed for shape reconstruction, and its adaptation to shape matching is not necessarily trivial.

The functional map has been successfully employed in tangent-vector-field transfer [29], which is useful for generating a consistent quadrangulation of shape pairs [30]. The latter task resembles our work in that an existing tessellation is recreated over an input shape, but our premises are more general (for example, we are not limited to quad meshes), and our approach differs in many crucial elements, including the strategy to define the functional map.

Surface remeshing. Surface remeshing aims at constructing a good meshing for a surface that is initially given in terms of (most frequently) an irregular mesh. The literature on surface remeshing spans decades and presents a large variety of approaches [22], based on slightly different problem statements and pursuing a variety of specific characteristics on the produced mesh.

Existing approaches range from connecting existing vertices of a point cloud with new triangles [106], parametrizing the surface and then lifting regular tri or quad grid in the parametric space [46], coarsening the original mesh and then regularly subdividing the resulting low-poly mesh [245], following a tangent direction-field to guide edges of the output mesh [257], aligning edges to user strokes [298], performing a sequence of local operations on the input mesh [139], or restricting a Voronoi diagram on the surface and leverage Lloyd relaxation to drive vertex placement [332,333]. Some geom-

etry processing tasks are a prerequisite for surface re-meshing, and are often studied together: surface parametrization [140], tangent field definition [312], and coarse quad layouts partitioning [58].

This variety of approaches reflect the variety of sought objectives, which are ultimately imposed by different downstream applications. Objectives often include regularity of the meshing, tolerating only few irregular vertices (*semi-regular* remeshing), or none (*fully regular* remeshing). The location of irregular vertices can also be carefully optimized. Irregular vertices can be required to be connected by short sequences of edges, implying the implicit partition of **S** into a few large, fully-regular patches (a *coarse quad-domain*) [59, 299], and thus easing texture mapping or shape editing. Targeted *polygons* can also vary: in traditional scenarios, triangles are used; other times, quadrilaterals are sought (*quad-remeshing* [45]), sometimes tolerating a few exceptions (*quad-dominant* remeshing); other cases, such as hexagonal meshes, are also occasionally studied [230, 311]. The *shape of the faces* is often a concern; the default ideal shape are, implicitly, equilateral triangles and squares; alternatively, controlled anisotropy can be sought, e.g. [21, 239], where elements are elongated along prescribed directions (*anisotropic remeshing*). Constant face *size* is also required, implying that vertices constitute a regularly distributed sampling (i.e., isotropic, or *uniform remeshing*); alternatively, *adaptive resolution* is sought, concentrating vertices in more geometrically complex or more semantically meaningful areas. *Edge orientation* can also also be important. Reproducing creases of the surface as mesh edges is crucial for geometrical fidelity in, e.g., CAD models; more in general, edges orientation must adhere to curvature directions or to arbitrary prescribed directions (also interactively so, e.g., [153]). For *animated meshes*, edge orientation can be optimized according to the intended deformations [200]. Finally, specialized techniques strive to explicitly identify *symmetries* and reproduce them in the meshing [238, 248].

Despite a long history of advancement and breakthroughs, and the number of existing specialized solutions, automatic surface remeshing is not yet capable of entirely replacing the manual design of meshes by digital artists trained to fulfill a variety of objectives. This motivates our approach, which can be seen as a way to sidestep, rather than solve, the task of surface remeshing. In our approach, most characteristics of carefully crafted models are automatically reproduced in the output.

The class of remeshing approaches which most closely resemble our work is *example based remeshing*, where good polygonal configurations are automatically learnt from existing examples [201]. Like in our case, good meshes, e.g. manually crafted by artists, are assumed to exist and are leveraged to drive the construction of new meshes with similar characteristics.

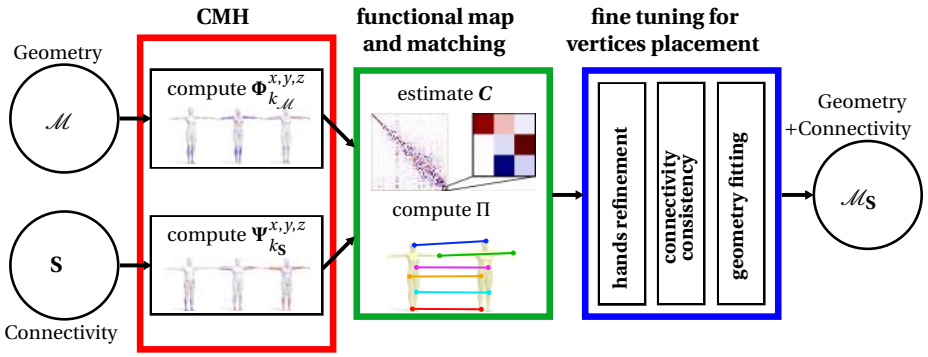


Fig. 7.1: A visualization of the proposed method. The scheme highlights the modules that compose the proposed pipeline.

7.3 Method

Given a triangular target mesh $\mathbf{S}$ and a source shape $\mathcal{M}$, our goal is to automatically transfer the connectivity of $\mathbf{S}$ on the geometry $\mathcal{M}$. For this purpose, we define a new basis: the Coordinate Manifold Harmonics (CMH). We exploit this basis in the Functional Maps framework and we propose a proper refinement strategy to improve the transfer on local regions. In Figure 7.1, we visualize the proposed pipeline. In this Section, we describe step by step all the modules involved. Each of the following subsections corresponds to a colored box in Figure 7.1: the Coordinate manifold Harmonics (red box), the functional map estimation (green box), the fine-tuning for vertices placement (blue box).

7.3.1 Coordinates Manifold Harmonics (CMH)

The functional map framework is an efficient and effective solution for the point-to-point matching of non-rigid shapes that is independent of the adopted discretization. However, the functional map $\boldsymbol{C} \in \mathbb{R}^{k_{\mathbf{S}} \times k_{\mathcal{M}}}$ is estimated by solving the optimization problem in Equation (2.23), where the number of unknowns is given by the product of $k_{\mathbf{S}}$ and $k_{\mathcal{M}}$. These two numbers define the number of frequencies involved, namely, the larger $k_{\mathbf{S}}$ and $k_{\mathcal{M}}$ are, the wider the low-band filter applied is. This relation introduces a trade-off: to represent the higher frequencies correctly, Functional Maps requires a higher number of unknowns, and therefore a more complex optimization should be solved.

The standard choices for $k_{\mathbf{S}}$ and $k_{\mathcal{M}}$ are $30, 60, 100$ while 200 or 300 are already considered too large values. However, as can be seen in the center of Figure 7.2 with $k_{\mathbf{S}} = k_{\mathcal{M}} = 106$ the representation of the geometry is very poor and many details are

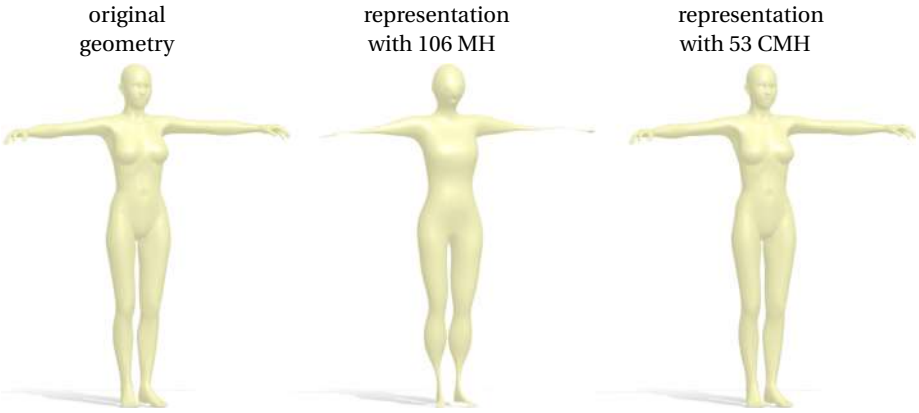


Fig. 7.2: Comparison between the geometry representation provided by 106 standard MH and the proposed 53 CMH. From left to right: the original shape, the low-pass representation provided by 106 MH and our geometry representation. Using a small number of basis, we drastically improve the quality of the represented geometry.

lost. This lack of quality motivates our definition of the *CMH*. This basis perfectly represents the 3D embedding of the given shapes just by adding three functions to a fixed set of standard manifold harmonics. On the right of Figure 7.2, the improvement provided by the CMH can be appreciated.

Given a shape $\mathcal{M}$ and the set of its first $k_{\mathcal{M}}$ LBO eigenvectors $\mathbf{\Phi}_{k_{\mathcal{M}}} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{k_{\mathcal{M}}}]$. The coordinates Manifold Harmonics (CMH) are a set of $k_{\mathcal{M}}+3$ orthonormal functions composed by the $k_{\mathcal{M}}$ first eigenfunctions of the LBO plus three new functions: $\boldsymbol{\phi}_x, \boldsymbol{\phi}_y$ and $\boldsymbol{\phi}_z$. We introduce $\boldsymbol{\phi}_x$, then sequentially $\boldsymbol{\phi}_y$ and $\boldsymbol{\phi}_z$.

Let $\boldsymbol{X}_{\mathcal{M}}$ be the x-coordinates of the vertices of $\mathcal{M}$. We compute $\widetilde{\boldsymbol{X}}_{\mathcal{M}}$ as the low-pass filter representation of $\boldsymbol{X}_{\mathcal{M}}$ provided by $\mathbf{\Phi}_{k_{\mathcal{M}}}$:

$$\widetilde{\boldsymbol{X}}_{\mathcal{M}} = \mathbf{\Phi}_{k_{\mathcal{M}}} \mathbf{\Phi}_{k_{\mathcal{M}}}^{\top} \boldsymbol{A}_{\mathcal{M}} \boldsymbol{X}_{\mathcal{M}}, \tag{7.1}$$

where $\boldsymbol{A}_{\mathcal{M}}$ is the mass matrix of $\mathcal{M}$. The function $\boldsymbol{\phi}_x$ is defined as the representation error of $\boldsymbol{X}_{\mathcal{M}}$ provided by the first $k_{\mathcal{M}}$ eigenfunctions of the LBO:

$$\boldsymbol{\phi}_x = \widetilde{\boldsymbol{X}}_{\mathcal{M}} - \boldsymbol{X}_{\mathcal{M}}. \tag{7.2}$$

We emphasize that $\boldsymbol{\phi}_x$ is orthogonal to the space spanned by $\mathbf{\Phi}_{k_{\mathcal{M}}}$ for construction.

Then we normalize $\boldsymbol{\phi}_x$:

$$\boldsymbol{\phi}_x = \frac{\boldsymbol{\phi}_x}{\|\boldsymbol{\phi}_x\|_{\mathcal{M}}}, \tag{7.3}$$

where $\|\boldsymbol{\phi}_x\|_{\mathcal{M}} = \sqrt{\langle \boldsymbol{\phi}_x, \boldsymbol{\phi}_x \rangle_{\mathcal{M}}}$.

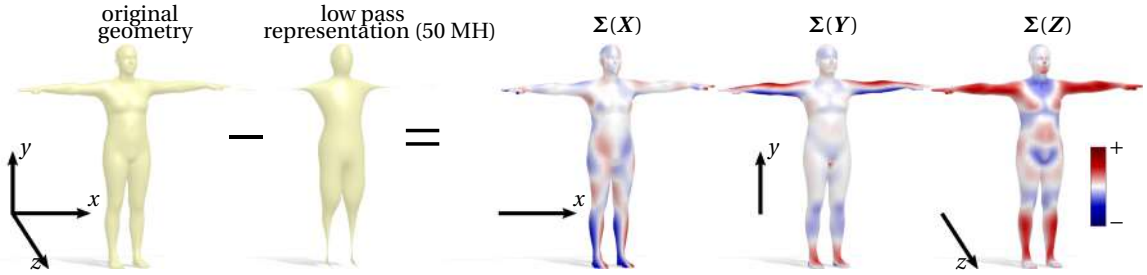


Fig. 7.3: CMH construction. With respect to the 3D embedding, we compute the difference in each coordinate between the original geometry and its low pass representation provided by the first MH (in our case 50 MH). On the right we visualize the three different functions, each of which will generate a CMH function. Positive values are represented in red, negatives in blue and 0 in white.

We update $\mathbf{\Phi}_{k_{\mathcal{M}}}$ adding $\boldsymbol{\phi}_x$ obtaining a new set of $k_{\mathcal{M}}+1$ orthonormal functions $\mathbf{\Phi}^{x}_{k_{\mathcal{M}}} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{k_{\mathcal{M}}}, \boldsymbol{\phi}_x]$. At this point, $\boldsymbol{\phi}_y$ can be computed applying the steps in Equations (7.1), (7.2) and (7.3), substituting $\boldsymbol{X}_{\mathcal{M}}$ with $\boldsymbol{Y}_{\mathcal{M}}$ and $\mathbf{\Phi}_{k_{\mathcal{M}}}$ with $\mathbf{\Phi}^{x}_{k_{\mathcal{M}}}$. The same can be done for $\boldsymbol{\phi}^{\mathcal{M}}_{Z}$ substituting $\boldsymbol{Y}_{\mathcal{M}}$ with $\boldsymbol{Z}_{\mathcal{M}}$ and $\mathbf{\Phi}^{x}_{k_{\mathcal{M}}}$ with $\mathbf{\Phi}^{x,y}_{k_{\mathcal{M}}} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{k_{\mathcal{M}}}, \boldsymbol{\phi}_x, \boldsymbol{\phi}_y]$. At the end of this process we obtain $\mathbf{\Phi}^{x,y,z}_{k_{\mathcal{M}}} = [\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_{k_{\mathcal{M}}}, \boldsymbol{\phi}_x, \boldsymbol{\phi}_y, \boldsymbol{\phi}_z]$, namely the *Coordinates Manifold Harmonics* (CMH) on $\mathcal{M}$.

These new functions $\boldsymbol{\phi}_x, \boldsymbol{\phi}_y$, and $\boldsymbol{\phi}_z$ encode, by definition, the essential extrinsic information to fully reconstruct the original shape (see Figure 7.2) recovering the details lost by the few LBO eigenfunctions (low-frequency). Therefore, the CMH exploits the benefit to integrate intrinsic and extrinsic geometry of the surface $\mathcal{M}$. On the other hand, our dependence on extrinsic information limits our CMH basis to work with shapes with the same (or very similar) poses. In Figure 7.4 we visualize an example of the CMH computed on two shapes.

7.3.2 Functional Map Estimation.

We equip $\mathcal{M}$ and $\mathbf{S}$ with the CMH, respectively $\mathbf{\Phi}^{x,y,z}_{k_{\mathcal{M}}}$ and $\mathbf{\Psi}^{x,y,z}_{k_{\mathbf{S}}}$. We exploit these bases in the functional map framework estimating a map $\boldsymbol{C} \in \mathbb{R}^{(k_{\mathbf{S}}+3)\times(k_{\mathcal{M}}+3)}$ between $\mathcal{M}$ and $\mathbf{S}$.

We rely on the formulation already presented in Section 2.2.2, that we report for the sake of clarity:

$$\min_{\boldsymbol{C}} \sum_{p} \|\boldsymbol{C}\hat{\mathbf{X}}^{(p)} - \hat{\mathbf{Y}}^{(p)}\boldsymbol{C}\|_F^2 + \gamma_1 \|\boldsymbol{C}\hat{\mathbf{F}} - \hat{\mathbf{G}}\|_F^2 + \gamma_2 \|\mathbf{C}\mathbf{\Lambda}_{\mathcal{M}} - \mathbf{\Lambda}_{\mathbf{S}}\boldsymbol{C}\|_F^2. \tag{7.4}$$

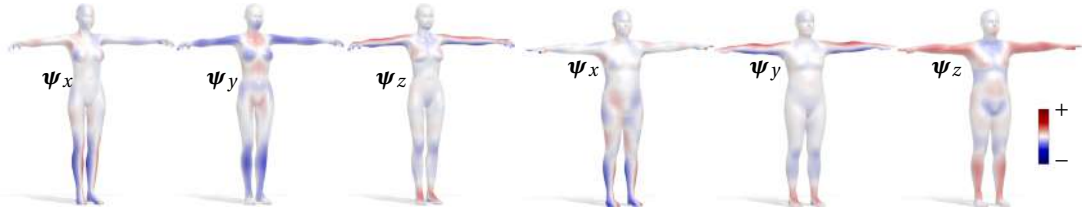


Fig. 7.4: CMH computed on two different shapes (a female shape from TOSCA [55] and SMPL model [195]). As can be seen, the order of the CMH is not shared by the two shapes. This is due to a different embedding in the 3D shapes; in other words, the two shapes are not aligned and differ for a rigid transformation.

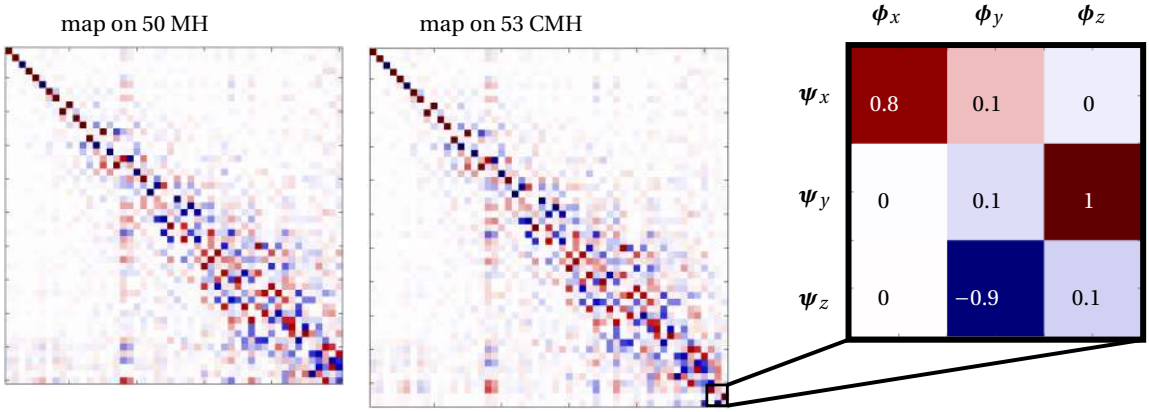


Fig. 7.5: A comparison between functional maps. From left to right: the maps computed for the first 50 eigenfunctions, the map computed for the proposed CMH, a detail of the coefficients representing how the three new bases are mapped from one shape to the other. These maps are estimated for the pair visualized in Figure 7.4. As can be seen, the map computed on the CMH solves for the switches of the CMH functions. Red encodes values close to 1, blue close to -1 while white is equal to 0.

For all our experiments we set $\gamma_1 = 0.1$ and $\gamma_2 = 0.001$.
The selection of the probe functions is fundamental since they lead the energy, and so the optimization. Following [233], we adopt the WKS descriptors [209] (20 time scales) and five landmarks functions (delta functions to which is applied a diffusion operator for 20 different scales). Usually, these landmarks require supervised selection, but they can be automatically selected for the human body through the method proposed in FARM (See Section 4.3.2). In Figure 7.5, we compare the 50×50 map in the standard

MH (left) and the 53×53 map exploiting the CMH (right) estimated on the same set of probe functions.

Given the functional map $\boldsymbol{C}$, as proposed in [234] we convert it in a point-to-point map $\boldsymbol{\Pi} : \mathbf{S} \rightarrow \mathscr{M}$ solving $\forall s \in \mathbf{S}$ the following nearest neighbor assignment in the spectral domain:

$$\boldsymbol{\Pi}(s) = \underset{m \in \mathscr{M}}{\arg\min} \| \boldsymbol{C}\boldsymbol{\Phi}^{x,y,z}_{k_{\mathscr{M}}}(m) - \boldsymbol{\Psi}^{x,y,z}_{k_{\mathbf{S}}}(s) \|_F \tag{7.5}$$

7.3.3 Fine tuning for vertices placement

Once we have this point mapping $\boldsymbol{\Pi}$, we need to refine it for the mesh transfer application. We introduce a point-to-surface matching strategy to allow the target mesh to "slide" along the source object to enclose its surface coherently.

Hands refinement through local correspondence. On the hands, denoted as $H(\mathbf{S})$, where usually the error is large, we apply a matching refinement. Similarly to FARM, we define a local geodesic ball around hands and, we register them using Coherent Point Drift (CPD) approach [226], which provides a local correspondence on the hands $\boldsymbol{\Pi}_{cpd}$. Merging global and local correspondences we obtain $\boldsymbol{\Pi}_{final}$:

$$\boldsymbol{\Pi}_{final}(x) = \begin{cases} \Pi_{cpd}(x) & x \in H(\mathbf{S}) \\ \Pi(x) & \text{otherwise} \end{cases} \tag{7.6}$$

which we adopt to transfer the mesh.

Note that there are no guarantees that meshes have the same number of vertices. Furthermore, a vertex-to-vertex correspondence is a constrained solution and can lead to undesired collapsing, flipping and other artifacts. For this reason, we would provide to our pipeline the possibility to generate solutions different from the vertex-to-vertex ones.

Connectivity consistency. In this step, we regularize the transferred connectivity with an as-rigid-as-possible formulation as proposed in [68]. From now on we refer to $\mathscr{M}_{\mathbf{S}}$ as the shape with the geometry of $\mathscr{M}$ and the connectivity of $\mathbf{S}$ that is produced by our method. We optimize the position of the vertices of $\mathscr{M}_{\mathbf{S}}$, minimizing the following energy:

$$E_{arap} = \frac{1}{4} \sum_{e_{ij} \in \mathscr{E}} \cot \alpha_{ij} |q_{ij} - R p_{ij}|^2 \tag{7.7}$$

where $p_{ij} = i - j$, q_{ij} is the value of p_{ij} in the new configuration and R_{ij} is the rotation that best approximate the transformation occurred between the two configurations. In [68] the authors provide a gradient derivation for the energy (7.7). This energy

measures how far we are from a rigid transformation and encourage the vertices neighborhoods to find an elastic equilibrium. In our case, we apply this energy between the mesh $\mathbf{S}$ and the new vertex positions obtained from $\mathscr{M}_{\mathbf{S}}$. This approach has two main advantages: firstly, the as-rigid-as-possible approach fosters local rigidity and helps solve inconsistent situations derived from vertex transfer. Secondly, it can be efficiently optimized by gradient techniques. The optimization tends to fall in a local minimum without destroying the global correspondence achieved before.

Geometry fitting. Finally, note that for now, we do not require to fit the geometry of $\mathscr{M}$. It is reasonable in some cases because changing the connectivity of a model also means a different discretization of the underlying geometry, but the geometry that we obtain is close to the one we desire. On the other hand, some artifacts can still arise: they may be caused by pose misalignment between two models (e.g. different pose traits between two subjects), noise in correspondence and also by stitching the $\mathbf{\Pi}$ with $\mathbf{\Pi}_{cpd}$. All these issues are transformed in a coherent continuous surface by previous ARAP step and we do not expect huge artifacts (e.g. intersections, triangles flipping). In the worst cases, the surface locally has been collapsed in some points and need to be inflated, or the optimization for connectivity consistency has sensibly modified local details of geometry.
For these reasons, we perform a final geometry fitting:

$$E_{reg} = w_a E_{arap} + w_d E_{dato} \tag{7.8}$$

where E_{arap} is defined as in Equation (7.7) (we still require the coherence with the original mesh). E_{dato} is defined as:

$$E_{dato} = \sum_{i \in \mathscr{V}_{\mathscr{M}_{\mathbf{S}}}} \min_{x \in \mathscr{M}} ||i - x||^2 \tag{7.9}$$

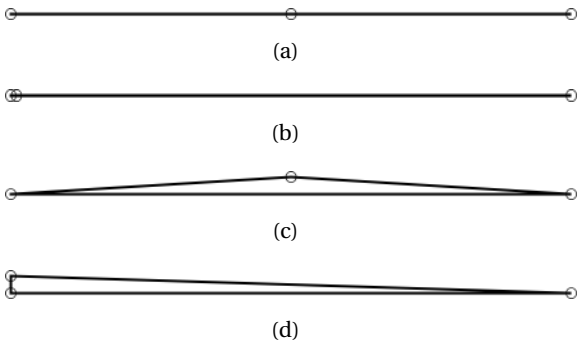
where x is a point on the surface of $\mathscr{M}$. This energy is needed to well approximate the geometry of $\mathscr{M}$ and it encodes the distances between the vertices of $\mathscr{M}_{\mathbf{S}}$ and the surface of $\mathscr{M}$. The weights w_a and w_d are chosen empirically; in our experiments, we found that 0.3 and 1 respectively provide a stable setup. We optimize E_{reg} using a gradient technique. The optimal solution leads to the final $\mathscr{M}_{\mathbf{S}}$; a model that shares the connectivity of $\mathbf{S}$ and approximates the geometry of $\mathscr{M}$.

7.4 Evaluation measures

We introduce the set of measures adopted in the quantitative evaluation of the proposed method. We divide the evaluation into two categories: mesh quality and transfer quality.

7.4.1 Mesh quality

These measures are used to evaluate our method in terms of preservation of the original connectivity.



Degenerate triangles. These artifacts are generated when a vertex get aligned on an edge or on another vertex incident on the same triangle (inset Figure). This measure is important to ensure that mesh is adapt. We classify a triangle as degenerated if:

- at least one of the angles is between 5° and 175°, as done in [49];
- the longest edge length is major or equal to the sum of the lengths of the other two;
- the area is close to 0.

We exclude the degenerate triangles from the calculation of the other measures.

Aspect ratio. A common method to evaluate the quality of a mesh is the Aspect ratio (AR) of the triangles. This measure has several definitions in literature. As done in [199], we adopt the following one:

$$AR = 4\frac{\sin\alpha\sin\beta\sin\gamma}{\sin\alpha+\sin\beta+\sin\gamma} \tag{7.10}$$

where α, β and γ are the angles of a given triangle. In figure 7.6 we show the differences between different AR average values on planar meshes and on a given human model.

Small angles. As done in [164], we consider θ_{min}, $\overline{\theta}_{min}$ and $\theta_{<30^\circ}$ which are respectively: the minimum angle of the mesh, the average of the minimum angles of all the triangles and the percentage of angles $< 30°$. Large values for θ_{min} and $\overline{\theta}_{min}$ denote higher quality meshes while meshes with small $\theta_{<30^\circ}$ value are preferred to the ones with larger $\theta_{<30^\circ}$.

7.4.2 Transfer Quality

These measures highlights the quality of the transfer in terms of distortion introduced by the re-meshing concerning the surface area, angles of the mesh and error fitting to the original data.

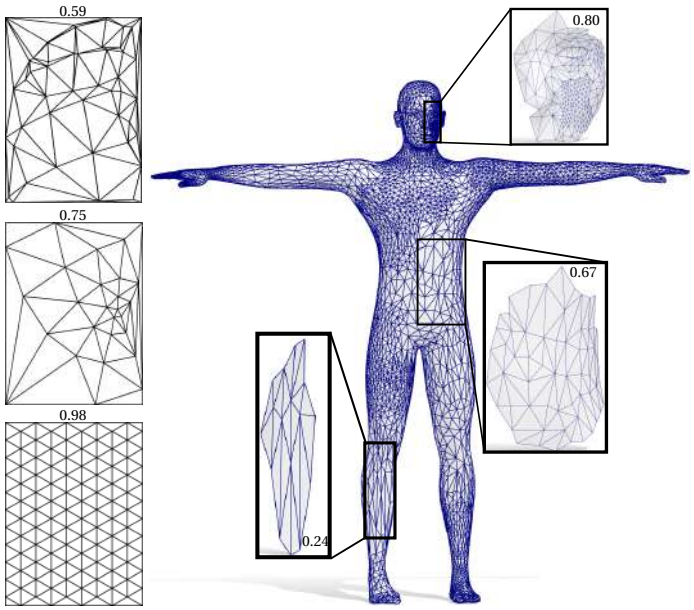


Fig. 7.6: On the left, three examples with different values of average Aspect Ratio. Low values attest irregularities in connectivity, while for values close to 1, the triangles are almost equilateral. On the right, a model that hosts various connectivities. The three different patches highlight differences in the Aspect Ratio.

Area distortion. This error considers the variation of the area of each triangle in the transfer. The area distortion (ArD) is evaluated on $\mathbf{S}$ and $\mathscr{M}_{\mathbf{S}}$. Large values for this distortion indicate that the area of the triangles is not preserved in the transfer. As in [338], the area distortion definition is:

$$ArD = \sum_t \left| \frac{A(T_t)}{A(\mathbf{S})} - \frac{A(U_t)}{A(\mathscr{M}_{\mathbf{S}})} \right| \tag{7.11}$$

where $A(T_t)$, $A(U_t)$ are respectively the areas of the t^{th} triangles of $\mathbf{S}$ and $\mathscr{M}_{\mathbf{S}}$ while $A(\mathscr{M})$ and $A(\mathscr{M}_{\mathbf{S}})$ are their total area. The summation is on the set of triangles of the transferred mesh. The value that represents each triangle is the ratio between the triangle area and the total area of the shape. This measure is thus scale-invariant. Small values of this distortion correspond to good transfer quality.

Angle distortion. Similarly, we consider the variation of the angles of each triangle in the tessellation transfer. Given $\mathbf{S}$ and $\mathscr{M}_{\mathbf{S}}$ we can define the angle distortion (AnD) as:

$$AnD = \frac{1}{3F} \sum_t \sum_{l=1}^{3} |\theta_{t,l} - \omega_{t,l}|. \tag{7.12}$$

$\theta_{t,l}$ and $\omega_{t,l}$ are the l^{th} angle of the t^{th} triangle of $\mathbf{S}$ and $\mathscr{M}_{\mathbf{S}}$ respectively. Also in this case the first summation is on the set of triangles of the transferred mesh and small values of this measure correspond to good transfer quality.

Error fitting. The fitting error is defined as the point-to-surface distance in Equation (7.9). With this measure, we evaluate how much the output fits the original shape's geometry M. When it is not specified, the roles of $\mathscr{M}$ and $\mathscr{M}_{\mathbf{S}}$ are the same as in Equation (7.9) measuring how each point of $\mathscr{M}_{\mathbf{S}}$ is close to the surface of $\mathscr{M}$. In some cases can be useful to evaluate the opposite direction, how well each point of $\mathscr{M}$ is approximated by the surface of $\mathscr{M}_{\mathbf{S}}$. When we consider the last evaluation, we explicit it in the related text.

7.5 Results

This Section evaluates our method through several experiments and applications showing results both on humans and animals among heterogeneous datasets. We show the connectivity transfer between different models through qualitative and quantitative results. Then, we investigate the contribution of the different components of our pipeline, performing an ablation study. Finally, we also show our performances on matching and property (e.g. texture) transfer.

Data. Here, we briefly list the data involved in our experiments. **SMPL model** [195] a widely used parametric model of the human body represented as a triangular mesh with 6890 vertices. With this model, we can generate several different human shapes that share a common pose and connectivity. We will refer to these shapes as **SMPL dataset**.In particular, we generate three subsets of shapes, namely SMPL 10K, SMPL 6K, and SMPL 3K, where the original SMPL models have been remeshed to reach 10K, 6K, and 3K vertices respectively. **FAUST** [43] is another dataset that shares the same connectivity of SMPL. It is composed of 100 shapes from ten different subjects in the same ten different poses. For all these shapes the ground truth correspondence is provided. **TOSCA** [55] a synthetic dataset that contains different classes including three human shapes (two male and one female). All the shapes in the same class share the same connectivity. **MakeHumans** [34] is an open-source software for human body generation. Varying the parameters, it is possible to obtain different shapes and details of the human shapes. We refer to shapes generated by this tool as **MakeHumans**.

Finally, for animal experiments we rely upon **SMAL** [348], which provides a parametric model for a wide class of animals. We also test a real-world texture transfer scenario using two freely available artist-made meshes of dogs [2, 4].

	degTri	AR^{ave}	θ_{min}	θ^{ave}_{min}	$\theta_{<30°}$ (%)
SMPL 10K	81.80	0.67	0.11	39.14	17.54
Tosca	4	0.70	2.46	39.14	15.49
S	8	0.83	1.88	32.42	4.47
TOSCA_S	32	0.78	1.18	36.55	9.72
SMPL 10K_S	22.80	0.80	1.17	37.68	7.09

M	S	ArD	AnD	E_{dato}	Time
SMPL 3K	SMPL_{model}	0.08	6.79	0.047	342.83
SMPL 6K	SMPL_{model}	0.08	5.53	0.037	389.56
SMPL 10K	SMPL_{model}	0.08	5.60	0.044	502.46
Tosca	SMPL_{model}	0.11	8.55	0.063	437.66
MakeH 8k	SMPL_{model}	0.12	6.59	0.087	539.83
MakeH 13k	SMPL_{model}	0.12	6.70	0.090	774.07

Table 7.1: **On the left**, mesh quality evaluation of the examples shown in the first two columns of Figure. 7.7. $Tosca$ and $SMPL10K$ rows highlight the features of the connectivity on the target Geometry. **S** is the SMPL model that provides the connectivity. The last two rows evaluate the quality provided by our method. The measures adopted are (from left to right): the number of degenerate triangles, average Aspect Ratio, minimum angle (in degrees), the average of the minimum angle for each triangle in the shape and the percentage of angles under 30°. **On the right**, the transfer quality evaluation using our method to transfer the SMPL model connectivity to shapes from 6 different datasets. The values in the columns represent in order: the area distortion, the angle distortion, the fitting error (cm) and the execution time (s).

Connectivity transfer. To assess the performance in connectivity transfer, we conduct various experiments and collect quantitative evaluations. Firstly, we transfer the SMPL template mesh over two different data: a SMPL model remeshed at 10K and a subject from TOSCA. Table 7.1 reports the measures described in Section 7.4.1. Even in the presence of substantially geometry differences, our method produces a satisfactory output, as qualitatively shown in the first two columns of Figure 7.7. In another experiment, we test on a large collection of shapes from six different datasets. Table 7.1 (right) reports measures on connectivity quality, geometry error and computation time. The results show a good performance for several different numbers of vertices and tessellations. In Figure 7.7 we report further experiments on an articulated pose (third column) and on animals domain (fourth and fifth column). The last row reports the point-to-surface distance between our output and the target geometry.

Ablation. We perform an ablation study to highlight the different contributions of each step in our pipeline. We selected two models from the FAUST dataset in A-pose. Then, we transferred the original mesh of the first over the remeshed geometry of the second. The two subjects have non-isometric deformations induced by the different identities and performed remeshing (e.g. sharper fingers compared to the ground truth). The various pose traits not precisely aligned introduce a further challenge. The result

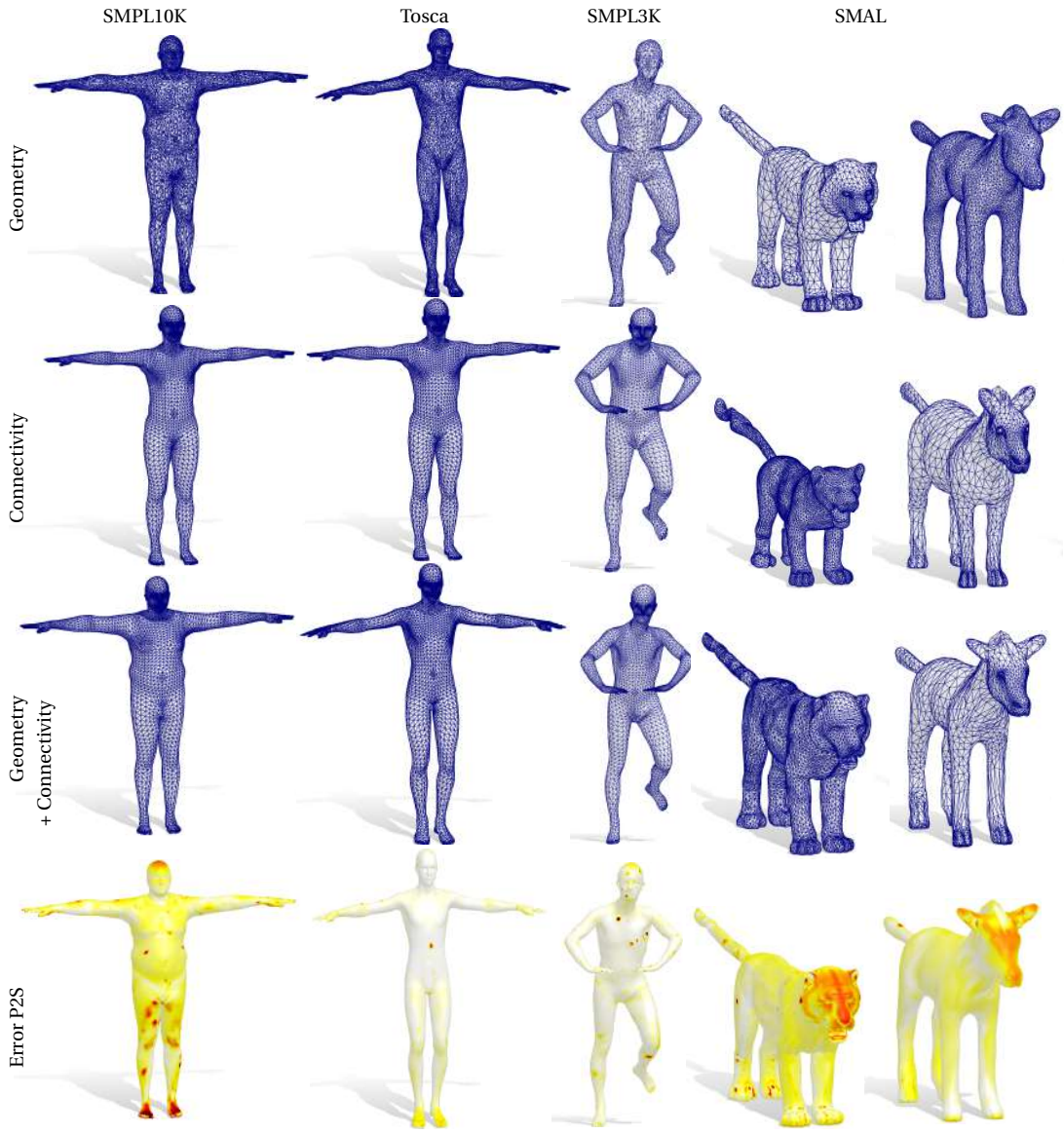


Fig. 7.7: Some qualitative results for connectivity transfer. In the last row we show the point-to-surface error of our output from the target geometry. This error is encoded by the heatmap (expressed in cm and saturated at 1cm).

of the ablation is reported quantitatively in Table 7.2 and it is presented qualitatively in Figure 7.8. As a first remark, the worst result is obtained by removing CMH basis. This underlines the key role of our novel basis set in catching geometry representation,

in particularly on peripheral regions. In the same spirit, removing the ARAP energy for geometry fitting causes dramatic misrepresentations. This is evident in the highly non-isometric differences among the shapes. Another critical aspect is the representation of the thinnest details; while they are overall reproduced correctly, hands requires an ad-hoc strategy. Without our local CPD correspondence, several collapses occur. Finally, the ARAP energy for connectivity consistency is the only element in our pipeline specialized in the original mesh features preservation. While it is hard to appreciate qualitatively, we refer to Table 7.2 to notice the regularization role of this step. This optimization avoids several artifacts like triangles collapsing and angles distortions that would make the result unusable for several applications. We would conclude this Section by highlighting that the most misrepresented part is the face due to the presence of the beard in the target geometry. However, in all cases, it is correctly mapped in the front of the head and the local connectivity is coherent with the global one.

	complete	w/o CMH	w/o CPD	w/o CC	w/o GF
mean $\mathscr{M}_\mathbf{S}$ to $\mathscr{M}$	0.057	0.105	0.063	0.057	0.169
max $\mathscr{M}_\mathbf{S}$ to $\mathscr{M}$	1.938	3.026	1.938	1.749	4.649
mean $\mathscr{M}$ to $\mathscr{M}_\mathbf{S}$	0.141	0.605	0.156	0.149	0.300
max $\mathscr{M}$ to $\mathscr{M}_\mathbf{S}$	1.396	7.569	1.987	1.733	5.282
ArD	0.114	0.229	0.119	0.577	0.146
AnD	7.80	11.13	9.11	NaN	9.41

Table 7.2: Ablation Study. In order to assess the contribution of each individual step, we compare the results obtained by the complete pipeline (second column) with the results obtained omitting one step (subsequent columns). We report the error in terms of: point-to-surface distances (cm), evaluated in both direction, Area distortion, and Angle distortion. CC and GF denote, respectively, ARAP for connectivity consistency and ARAP for geometry fitting.

Point-to-point matching. Firstly, we show how CMH can improve the spectral representation of the models in the same pose. Adopting the point-to-point matching approach based on the functional map framework [233], we estimate a functional map on a set of given probe functions, then assign the matching through the nearest neighbor in the spectral domain. In Figure 7.9, we quantitatively compare the results using 53 MH against 50 MH plus the 3 CMH. Although this experiment is performed only for shapes with the same pose, we appreciate that the CMH improves the point-to-point matching. In particular, as we will confirm in the next experiments, the improvements of CMH are concentrated on the detailed parts such as the hands and arms, where even small errors introduce unpleasant visual effects.

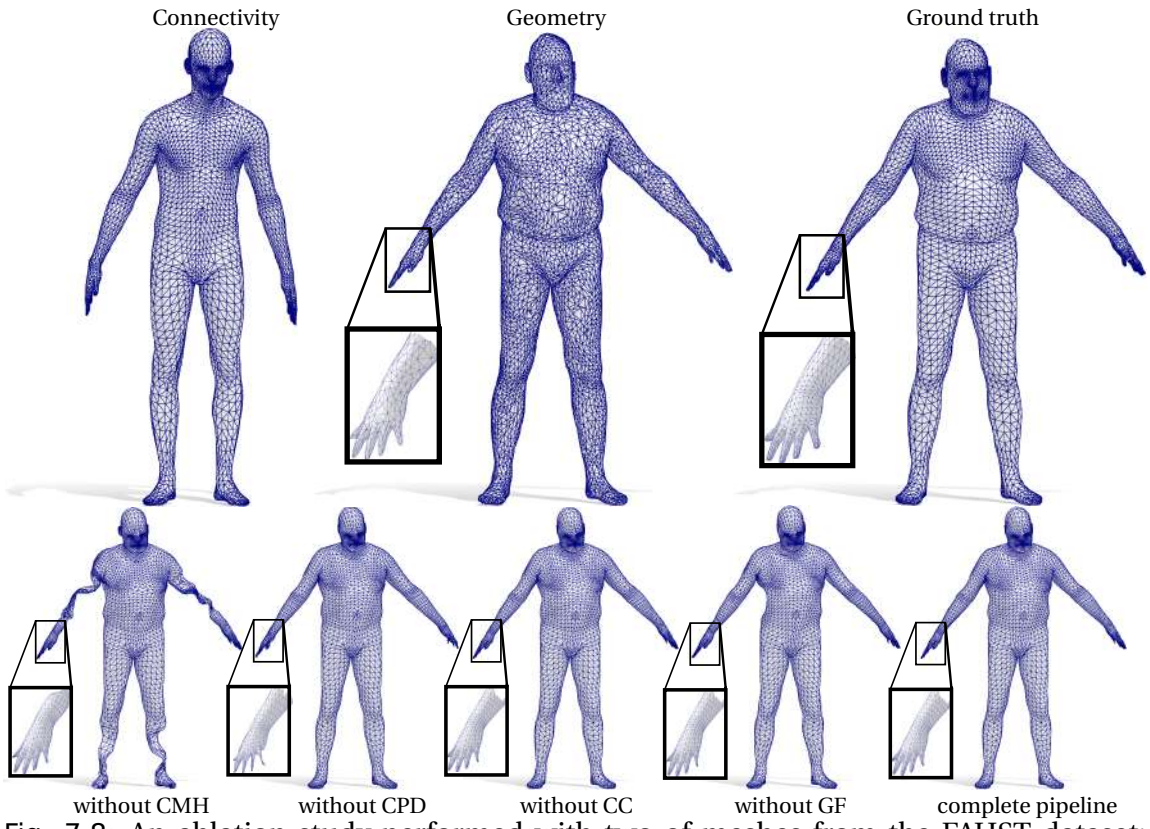


Fig. 7.8: An ablation study performed with two of meshes from the FAUST dataset: "Connectivity" and "Ground truth". Our system takes in input the former, and an irregular arbitrary remeshing of the latter (labelled "Geometry"). Below: results obtained if the specified part of our pipeline is omitted.

7.5.1 Comparison with other methods

The proposed matching pipeline is quantitatively compared to the following approaches:

MH: this method is defined by our pipeline applied on a standard functional map of size 53×53 using the classic manifold harmonics. This is a baseline to test improvement induced by our CMH basis.

LMH: this method involves another strategy to exploit local information on the spectral domain using the localized manifold harmonic recently proposed in [216]. The functional map is estimated from 50 standard plus 3 localized manifold harmonics.

D&D: this method employs the refinement procedure proposed in [100] starting from a functional map of size 53×53. This refinement strategy represents the state of the

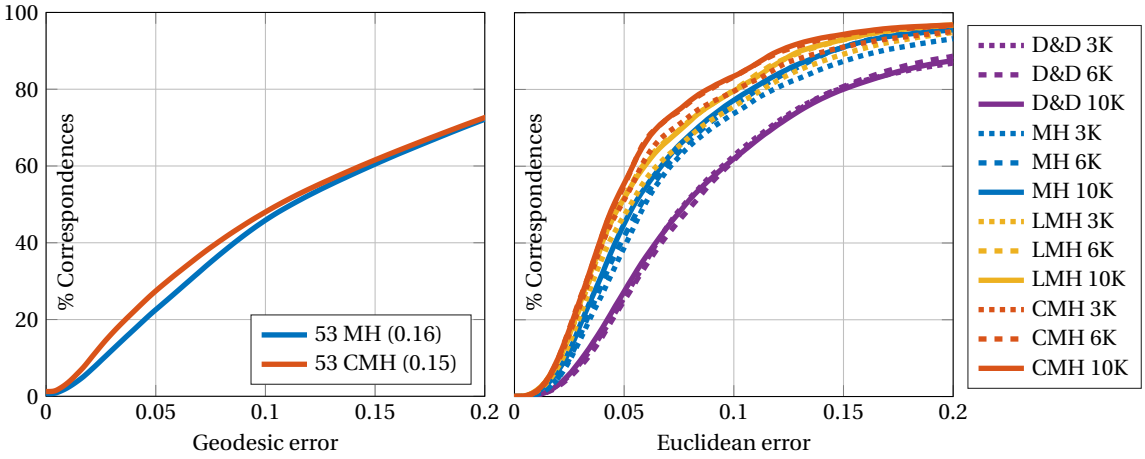


Fig. 7.9: **On the left**, point-to-point matching evaluation on all the possible (450) pairs with the same pose selected at random from the FAUST dataset. The evaluation is performed accordingly to [165]. The y-value corresponds to the percentage of matched points with an error smaller or equal to the x-value. In the legend we report the average geodesic error in centimeters. **On the right**, comparison on the SMPL dataset. The results are on average on ten shapes remesh with the SMPL connectivity. Solid lines represent results on 10K vertices shapes, dashed lines results on meshes with 6K vertices and dotted ones on meshes with 3K vertices.

art for point-to-surface matching based on the functional map framework, which is very similar to our use of the ARAP constraint to deal with fine details.

All the functional maps used in these experiments are estimated through the method proposed in [233] using the same probe functions and landmarks. The datasets involved have three different resolutions (3K, 6K, and 10K) and each one consists of ten shapes generated by SMPL. The results depicted in Figure 7.9 show that the proposed CMH are better suited for this task with respect to both MH and LMH. Furthermore our pipeline, applied on all the three bases, provides results that outperform the state of the art refinement methods.

Texture transfer. A compelling application of a remeshing pipeline is texture transfer, for example when one texture is to be shared among different models; texture transfer requires skills and artists direction, according to criteria that are hard do encapsulate as general rules. In Figure 7.10 we show a mesh equipped with a texture and a target geometry for which would be challenging apply the same. By transferring the connectivity,

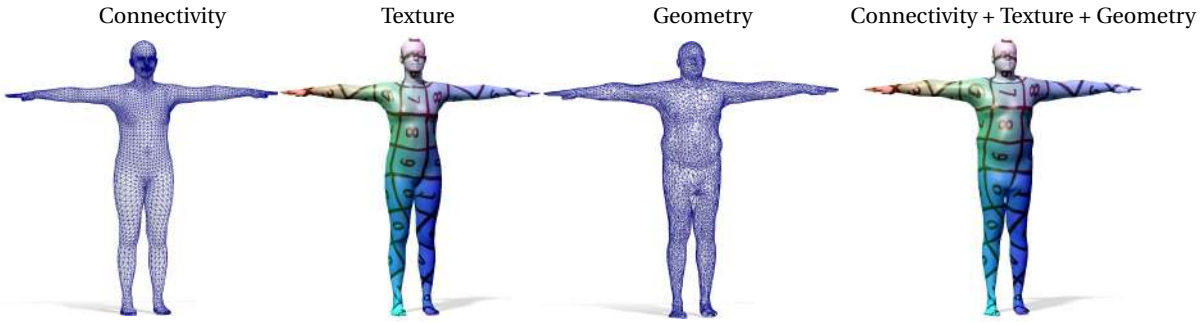


Fig. 7.10: An example of texture transfer between models. From the left: a regular connectivity, a defined texture and an irregular geometry we desire to texturize. On the right the result. We would like to underlie that transferring good connectivity let the geometry to inherit some desirable properties (e.g. face and feet details).

the resulting model can be textured directly with the inherited UV-map. Figure 7.11 reports two real-world examples: on the left, the texture is obtained Autodeskś Character Generator and transferred over a MakeHumans model in a similar pose. On the right, we transfer a texture between two models of dogs of different breeds. Texture coherence is obtained in spite the different sources, morphology, proportions and traits of the models.

Fig. 7.11: Two artist made texture transfer example. On the left the original texture on S, on the right the texture transferred using our method on a different geometry.

Full model transfer from multiple targets. In modeling and animation frameworks, it is often required to effectively define not only the tessellation or the texture as seen

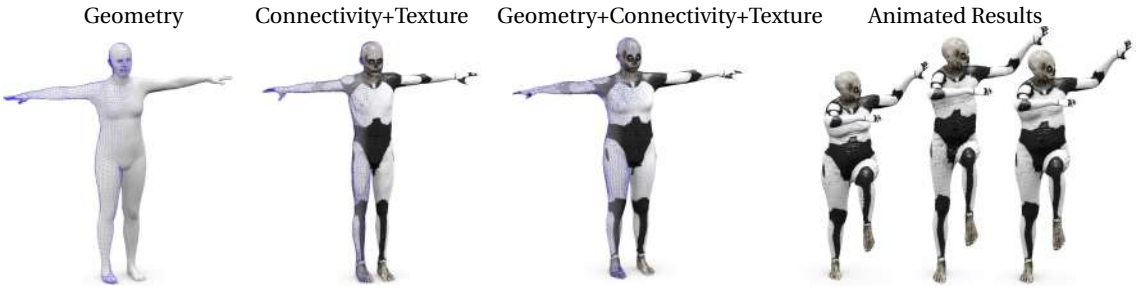


Fig. 7.12: The full model transfer experiment. From left to right: the SMPL model provides the geometry, an Autodesk Character Generator model (around 10K vertices) that provides the connectivity and the texture; the result of our transfer and some model generated by including our output inside SMPL framework. We can *automatic* substitute the SMPL template with an arbitrary one, generating many texturized and shaped models, and move them using inherited skinning information.

above but also an additional set of attributes, e.g., the rigging and skinning properties. Such amount of properties may not be designed directly on a single target shape. We show that using our method it is possible to equip the source shape with the required properties from a collection of targets that disjointly have such features. For instance, using SMPL we can obtain rigging and skinning information ready-to-use. However, no texture is provided, and also its resolution is limited (i.e., 6890 vertices). To overcome these limits, we involve a further target model obtained by Autodesk Character's Generator equipped with a higher resolution and a texture. Given the source shape (as geometry), we employ our pipeline with both the target models. In this fashion, the source shape inherits texture and the high-resolution tessellation from the Autodesk Character, and the rigging and skinning weights from the SMPL model. In particular, the rigging properties are obtained using the devoted regressor (see [195]) on the vertices of the new tessellation from SMPL. The skinning weights are also obtained from the SMPL matching, but then they are extended to the higher resolution mesh by applying a nearest-neighbor procedure between the two (i.e., high and low) tessellations. Results are very promising as illustrated in Figure 7.12 where we can generate more zombie-like models.

7.6 Conclusions

This chapter proposes a new approach for surface remeshing based on a fully automatic shape matching strategy. Our experiments show that this system can produce meshes

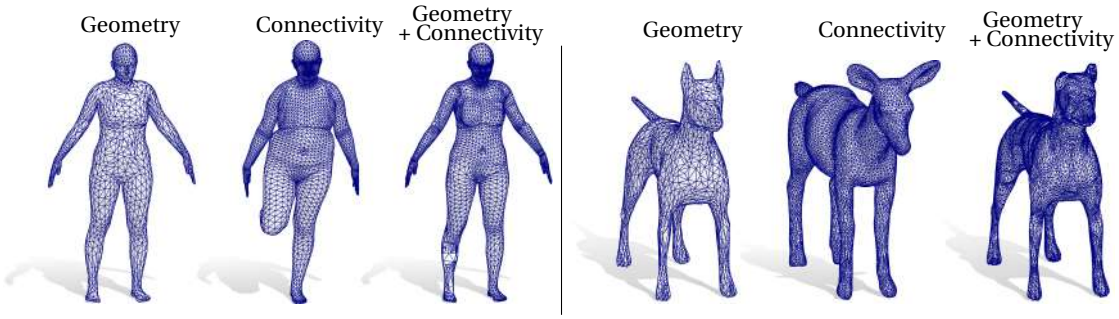


Fig. 7.13: Two visualizations of the main limitation of the proposed method. In the first row, the input shape and the target mesh have different poses. In the second row, the two quadruped animals (a dog and a fawn) are not isometric. The mesh transfer error is larger where the differences in the poses (the right leg) and from isometry (on the ears) are more evident.

combining the geometrical shape of a captured mesh shape with the tessellation and associated attributes of another similar shape. To this end, we introduced a new set of basis functions, the *Coordinates Manifold Harmonics* (CMH), that accounts for both intrinsic and extrinsic information within one unified spectral matching framework. We leverage on the capabilities of CMH in a pipeline that extracts the point-to-point correspondences from the functional map and refines the results by optimizing for the preservation of local isometries. Our experimental results show the improved ability of our method of transferring a desired connectivity on previously unseen geometry for different application domains. The method has been recently applied in SHREC 2020 challenge [94], showing state-of-the-art results in matching non-isometric animals.

7.6.1 Limitations

The main limitation of our method it is that it needs input shapes in a relatively similar geometry, including the pose. In Figure 7.13 we exemplifies this limitation.

7.6.2 Future work

A shape retrieval strategy can be introduced to improve the automatic identification of the most appropriate target shape with the desired tessellation. In future work, we plan to make our method able to work also for stronger non-isometric pairs. Recently,

other refinement methods for correspondence have been proposed such as, among others, [101, 102, 214, 258]. We plan to inject each of these (and possibly others) refinement methods and compare them in the surface remeshing task. Furthermore, this work assumes that polygonal meshes represent shapes. We plan to extend our work exploiting new matching strategies between meshes and point clouds to deal with the cases where the tessellation is not available for the source shape. Finally, thanks to the flexibility of the framework of functional maps, we aim at adopting our CMH tool in the challenging cases of topological error and partiality, exploiting the CMH basis in the partial functional maps framework as proposed in [265].

Concluding this Part, we would offer few comments. We observed that modern tools to match and describe surfaces heavily rely on their discretization; matching non-isometries is more difficult when the triangulations significantly differ. The connectivity impacts the geometry and the linkage of different ones. This fact is as trivial as undesirable; we need to discretize our object for computational reasons, but we would limit information loss. As stated in this thesis introduction, discretization and geometry are two different entities with different purposes. Following this road, we have extended a common intrinsic representation to consider extrinsic information, and we can transfer connectivity between different geometries in the same pose. The enhancement given by the extrinsic information has an exact reason: disentangling the object geometry from its connectivity is easier by knowing something about its spatial position.

Also, we highlight that the mesh transfer task is a particular case of matching. We do not only put them in correspondence, but we describe the two geometry with the same discretization. Someone could reply that a template registration pipeline (like FARM and its resolution augmentation) provides a geometry description with different connectivities. While it is right in some sense, these methods are limited to the availability of a deformable template; our remeshing method is general, providing a way to transfer connectivity between arbitrary shapes. Secondly, registration pipelines require deforming the source template through several intermediate geometries. This optimization can affect final geometry due to local minima, and intermediate results do not define any target geometry property. Our remeshing algorithm skips this process, providing an initial transfer that already represents the target geometry structure. Then, our optimization is only locally, looking for a trade-off between details catching and connectivity preservation.

Finally, working with more faint surface representations (like point clouds) is not explicitly addressed by previous Chapters. We face this challenge in the next Part.

Part III

Learning Representations

In the previous Parts, we mainly focused on correspondences between triangle meshes, combining or extending existing representation. In this Part, we face different representations: firstly, we present a learning pipeline extending our matching capability on point clouds [205]. We learn a high-dimensional pointwise embedding, where a linear transformation is enough to solve for the non-rigid correspondence. Secondly, we propose to link two super-compact representations: the Laplacian spectra of a shapes collection and an AutoEncoder latent space [206]. Our method accepts shapes regardless of their representation and outputs them in a unique discretization, providing natural correspondence between the geometries.

8

Correspondence Learning via Linearly-invariant Embedding

In this chapter, we propose a fully differentiable pipeline for estimating accurate dense correspondences between 3D point clouds. The proposed pipeline is an extension and a generalization of the functional maps framework. *However, instead of using the Laplace-Beltrami eigenfunctions as done in virtually all previous works in this domain, we demonstrate that learning the basis from data can both improve robustness and lead to better accuracy in challenging settings. We interpret the basis as a learned embedding into a higher dimensional space. Following the functional map paradigm the optimal transformation in this embedding space must be linear and we propose a separate architecture aimed at estimating the transformation by learning optimal descriptor functions. This leads to the first end-to-end trainable functional map-based correspondence approach in which both the basis and the descriptors are learned from data. Interestingly, we also observe that learning a* canonical *embedding leads to worse results, suggesting that leaving an extra linear degree of freedom to the embedding network gives it more robustness, thereby also shedding light onto the success of previous methods. Finally, we demonstrate that our approach achieves state-of-the-art results in challenging non-rigid 3D point cloud correspondence applications.*

8.1 Introduction

Computing correspondences between geometric objects is a widely investigated task. Its applications are countless: rigid and non-rigid registration methods are instrumental in engineering, medicine and biology [113, 155, 177] among other fields. Point cloud registration is important for range scan data, e.g., in robotics [122, 276], but the problem can also be generalized to abstract domains like graphs [109, 319].

The *non-rigid* correspondence problem is particularly challenging as a successful solution must deal with large variability in shape deformations and be robust to noise in

the input data. To address this problem, in recent years, several data-driven approaches have been proposed to learn the optimal transformation model from data rather than imposing it *a priori*, including [48, 127, 321] among others. In this domain, a prominent direction is based on the functional map representation [234], which has been adapted to the learning-based setting [91, 131, 186, 271]. These methods have shown that optimal feature or descriptor functions (also known as "probe" functions) can be learned from data and then used successfully within the functional map pipeline to obtain accurate dense correspondences. Unfortunately, the reduced functional basis, which forms the key ingredient in this approach, has so far been tied to the Laplace-Belrtami eigen-basis, specified and fixed *a priori*. While this choice might be reasonable for near-isometric 3D shapes represented as triangle meshes, it does not allow to handle more diverse deformations classes of or significant noise in the data. The main limitations of this pipeline are two-fold: first, the quality of the map is strongly tied to the choice of probe functions, and second, the choice of the basis plays a fundamental role both for the expressive power and the accuracy of the final results. Several approaches have been proposed to learn the probe functions from data [91, 131, 186, 271]. However, as mentioned above, no existing methods have attempted to learn the basis. This is particularly problematic since, as we show below, the Laplacian eigen-basis is not only tied to near-isometric deformations; even more fundamentally, it can only be reliably computed on shapes represented as triangle meshes. While some attempts (e.g., in [214, 265]) have been made to compute eigenfunctions using existing discretizations of Laplace-Beltrami operators on point clouds, e.g., [35, 180]. Nevertheless, in part due to the *differential nature* of the Laplacian, such discretizations cannot handle even mild noise levels in practice. Inspired by the success and robustness of these techniques, we propose the first fully-differentiable functional maps pipeline, in which both the probe functions and the functional basis are learned from the data. Our key observation is that basis learning can be phrased as computing an embedding into a higher-dimensional space in which a non-rigid deformation becomes a *linear transformation*. This follows the functional map paradigm in which functional maps arising from pointwise correspondences must always be linear [234] and computing such a linear transformation is equivalent to solving the non-rigid correspondence problem. In the process, we also observe that training a network that aims to compute a *canonical* embedding, in which optimal correspondences are simple nearest neighbors, leads to a drop in performance. As we discuss below, this suggests that the additional degree of freedom, by learning a linearly-invariant embedding, helps to regularize the learning process and avoid overfitting in challenging cases.

Finally, we demonstrate that our simple (but effective) formulation leads to accurate dense maps. The code, datasets and our pre-trained networks can be found online [8].

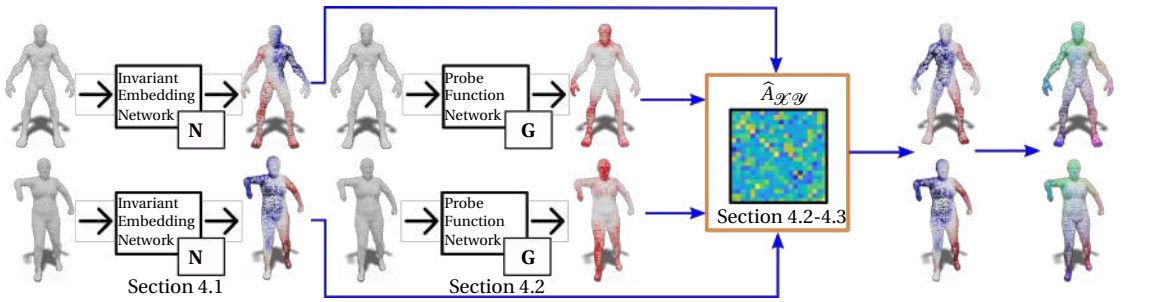


Fig. 8.1: Pipeline overview: starting from point cloud coordinates we obtain a set of linearly-invariant basis functions via the Invariant Embedding Network **N** (a), and descriptors using the Probe Function Network **G** (b). The learned basis and probe functions are used to compute the optimal linear transformation $\widehat{A}_{\mathcal{XY}}$ (c). This transformation is used to align the two sets of bases (d). The correspondence between point clouds is then estimated using nearest neighbors between the aligned basis sets (e). Note that the underlying meshes are depicted only for the sake of clarity of visualization.

8.2 Motivation and notation

Our main goal is to learn an optimal basis for the functional map pipeline on point cloud data. One possibility would be to use triangle meshes and learn a discretization of the Laplacian that would approximate the low-frequency basis functions on point clouds. However, this requires differentiating through a sparse eigen-decomposition, which can be expensive and unstable.

Instead, we propose an end-to-end learnable pipeline that uses a dual point of view. We summarize our overall pipeline in Figure 8.1. Our first remark is that entries of the basis functions can be interpreted as an embedding of the original 3D shape into a higher k-dimensional space. Namely, each point $x \in \mathcal{X}$ gets associated with a k-dimensional vector $[\phi_1^{\mathcal{X}}(x), \phi_2^{\mathcal{X}}(x), \ldots, \phi_k^{\mathcal{X}}(x)]$. This is called the "spectral" embedding and it is well-known (see e.g., [272]) that when using the Laplacian basis on smooth surfaces, as $k \to \infty$ this embedding becomes injective so that no two points can have the same associated vectors.

The spectral embedding plays a role in the conversion between functional and pointwise maps. The standard approach for this conversion [234] is by mapping Dirac δ_x functions associated with each point x on the source shape and finding the nearest Dirac δ function on the target. Interestingly, δ_x is *not* a real-valued function but is rather a *distribution*, which acts on real-valued functions through inner products: $< \delta_x, f >= f(x)$. As functional maps are operators that map real-valued functions, in

principle they *cannot* be used to transport Dirac δ's. To transport such distributions, a more sound approach is to use the *adjoint* operator of a functional map [146]. Surprisingly, although the notion of the adjoint has been studied , both its role and the limitations of functional maps in transferring δ functions seems to have been ignored in the functional maps literature so far. The adjoint operator is defined implicitly as follows: given a functional map $C_{\mathcal{Y}\mathcal{X}}$, its adjoint $A_{\mathcal{X}\mathcal{Y}}$ is defined so for any pair of real-valued functions $f \in \mathcal{F}(\mathcal{X})$ and $g \in \mathcal{F}(\mathcal{Y})$: $< C_{\mathcal{Y}\mathcal{X}} g, f >=< g, A_{\mathcal{X}\mathcal{Y}} f >$. Note that the adjoint operator: 1) associates functions in the opposite direction to that of the functional map, and 2) is defined using the L_2 inner products, and can thus be used to transport distributions. It is easy to see that the adjoint of the pull-back of a point-to-point map $T_{\mathcal{X}\mathcal{Y}}$ has the following nice property: $A_{\mathcal{X}\mathcal{Y}}\delta_x = \delta_{T_{\mathcal{X}\mathcal{Y}}(x)}$. We refer to the appendix C for a more complete treatment of the adjoint operator.

Finally, we note that the coefficients of Dirac δ function δ_x are precisely the vector of values $[\phi_1^{\mathcal{X}}(x), \phi_2^{\mathcal{X}}(x), \ldots, \phi_k^{\mathcal{X}}(x)]$. Moreover, the adjoint is a linear operator that associates δ functions with δ functions. As such, the adjoint can be seen as a linear transformation that aligns the spectral embeddings of $\mathcal{X}$ and $\mathcal{Y}$. We emphasize that the same *does not hold* for a functional map, in general.

This discussion implies that in the functional map framework, the basis can be interpreted as an embedding, and moreover the corresponding embeddings are related by a linear transformation, which is precisely the adjoint of the functional map.

Strategy Our overall strategy is to mimic this construction using a learning-based approach. We propose to train a network that computes for each shape an embedding into some k dimensional space, such that the embeddings of two shapes are related by a linear transformation. We then train a separate network that computes probe functions to establish the optimal linear transformation at test time. Remarkably, this decomposition of the problem consistently outperforms a baseline approach that aims to compute a canonical embedding, in which correspondences can be obtained through nearest neighbor search directly. As described below, we attribute this primarily to the fact that learning a canonical embedding is a difficult problem, and splitting it into two parts (invariant embedding + transformation) helps to regularize the problem in challenging practical settings. Note that we use the term "basis" only by analogy with the Laplace-Beltrami eigenfunctions, and do not formally impose a basis structure on our learned set of functions.

8.3 Linearly-invariant embedding

This section proposes a novel learning strategy to generalize the Functional Maps framework to noisy and incomplete data.

We discretize a shape $\mathcal{X}$ as a collection of 3D points $x_i \in \mathbb{R}^3$ where $i \in \{1,\dots,n_{\mathcal{X}}\}$. We collect these $n_{\mathcal{X}}$ points in a matrix $P_{\mathcal{X}} \in \mathbb{R}^{n_{\mathcal{X}}\times 3}$ such that the i-th row of $P_{\mathcal{X}}$ captures the 3D coordinates of x_i. We refer to the matrix $P_{\mathcal{X}}$ as the *natural* embedding of $\mathcal{X}$.

Given a pair of shapes $\mathcal{X}$ and $\mathcal{Y}$ our goal is to find a correspondence between them. This correspondence is a mapping between the points of $\mathcal{X}$ and the points of $\mathcal{Y}$. We denote a correspondence as a map $T_{\mathcal{X}\mathcal{Y}} : \mathcal{X} \to \mathcal{Y}$ such that $T_{\mathcal{X}\mathcal{Y}}(x_i) = y_j$, $\forall i \in \{1,\dots,n_{\mathcal{X}}\}$ and some $j \in \{1,\dots,n_{\mathcal{Y}}\}$. This map has a natural matrix representation $\Pi_{\mathcal{X}\mathcal{Y}} \in \mathbb{R}^{n_{\mathcal{X}}\times n_{\mathcal{Y}}}$ such that $\Pi_{\mathcal{X}\mathcal{Y}}(i,j) = 1$ if $T_{\mathcal{X}\mathcal{Y}}(x_i) = y_j$ and 0 otherwise.

Let $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{Y}}$ denote the matrices, whose rows can be interpreted as embeddings of the points of $\mathcal{X}$ and $\mathcal{Y}$ as described in Section 2.1.4. Below we do not assume that $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{Y}}$ represent the Laplacian eigenbasis, but consider general embeddings into some fixed k dimensional space. Recall that in the formalism of Functional Maps, there must exist a linear transformation $A_{\mathcal{X}\mathcal{Y}}$ that aligns the corresponding embeddings. This can be written as: $A_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{X}}^T = (\Pi_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{Y}})^T$, where $\Pi_{\mathcal{X}\mathcal{Y}}$ is the binary matrix that encodes the correspondence between $\mathcal{X}$ and $\mathcal{Y}$. In the functional map framework, the linear transformation $A_{\mathcal{X}\mathcal{Y}}$ is precisely the adjoint operator, since $A_{\mathcal{X}\mathcal{Y}} = (\Phi_{\mathcal{X}}^+\Pi_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{Y}})^T = C_{\mathcal{Y}\mathcal{X}}^T$ using the standard definition of a functional map $C_{\mathcal{Y}\mathcal{X}}$ [235].

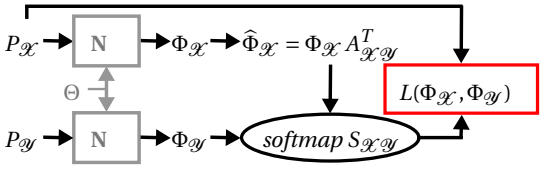
Given $A_{\mathcal{X}\mathcal{Y}}$, we can estimate $\Pi_{\mathcal{X}\mathcal{Y}}$ by solving the following optimization problem:

$$\Pi_{\mathcal{X}\mathcal{Y}} = \arg\min_{x} \min_{\Pi} \|\Phi_{\mathcal{X}} A_{\mathcal{X}\mathcal{Y}}^T - \Pi\Phi_{\mathcal{Y}}\|_2. \tag{8.1}$$

Note that Equation (8.1) can be solved in closed form by finding, for every row of $\Phi_{\mathcal{X}} A_{\mathcal{X}\mathcal{Y}}^T$, the closest row in $\Phi_{\mathcal{Y}}$ in the standard L_2 sense.

Based on Equation (8.1), our general goal is to train a network $\mathbf{N}$ that can produce for any shape $\mathcal{X}$ an embedding $\Phi_{\mathcal{X}}$ into a k-dimensional space, such that embeddings of every pair of shapes $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ are related by a linear transformation. In other words, the network $\mathbf{N}$ transform a shape from the original 3D space, in which complex non-rigid deformations occur, to another space, in which transformations across shapes must always be linear. Interestingly, as we show below, the additional linear degree of freedom helps to regularize the learning procedure, achieving better results than merely learning a canonical embedding in which corresponding points are nearest neighbors.

8.3.1 Learning a linearly-invariant embedding



To learn a linearly-invariant embedding we first observe that for fixed matrices $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ the expression $\|\Phi_{\mathcal{X}} A^T_{\mathcal{X}\mathcal{Y}} - \Pi_{\mathcal{X}\mathcal{Y}} \Phi_{\mathcal{Y}}\|_2$ depends both on $A^T_{\mathcal{X}\mathcal{Y}}$ and $\Pi_{\mathcal{X}\mathcal{Y}}$, which can make training difficult. However, for a fixed correspondence matrix $\Pi_{\mathcal{X}\mathcal{Y}}$ the optimal matrix $A_{\mathcal{X}\mathcal{Y}}$ can be obtained in closed form simply as: $A_{\mathcal{X}\mathcal{Y}} = (\Phi^+_{\mathcal{X}} \Pi_{\mathcal{X}\mathcal{Y}} \Phi_{\mathcal{Y}})^T$, which can be computed by solving a linear system of equations. Importantly, this procedure can be differentiated using the closed-form expression of derivatives of matrix inverses, which we exploit in our approach.

Embedding network training Given a set of training pairs of shapes $\mathcal{X}, \mathcal{Y}$ for which ground truth correspondences $\Pi^{gt}_{\mathcal{X}\mathcal{Y}}$ are known, our embedding network $\mathbf{N}$ computes an embedding $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ for each shape using a Siamese architecture with shared parameters. I.e., $\mathbf{N}_\Theta(P_{\mathcal{X}}) = \Phi_{\mathcal{X}}$ and $\mathbf{N}_\Theta(P_{\mathcal{Y}}) = \Phi_{\mathcal{Y}}$. We use the notation $\mathbf{N}_\Theta$ to highlight that this network has trainable parameters Θ which are shared across shapes. In the following, we refer to this network as simply $\mathbf{N}$.

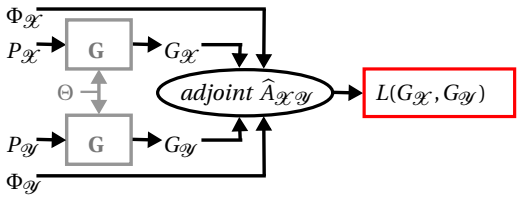
To define our loss we compute $A_{\mathcal{X}\mathcal{Y}} = (\Phi^+_{\mathcal{X}} \Pi^{gt}_{\mathcal{X}\mathcal{Y}} \Phi_{\mathcal{Y}})^T$ as the optimal linear transformation, and use it to obtain a *transformed* embedding $\widehat{\Phi}_{\mathcal{X}} = \Phi_{\mathcal{X}} A^T_{\mathcal{X}\mathcal{Y}}$. We then compare the rows of $\widehat{\Phi}_{\mathcal{X}}$ to those of $\Phi_{\mathcal{Y}}$ to obtain the *soft* permutation matrix $S_{\mathcal{X}\mathcal{Y}}$ that approximates the discrete mapping between the shapes in a differentiable way using the *softmax* operation. Finally, we use the following loss to train the embedding network:

$$L(\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}) = \frac{1}{n_\beta} \sum \| S_{\mathcal{X}\mathcal{Y}} P_{\mathcal{X}} - \Pi^{gt}_{\mathcal{X}\mathcal{Y}} P_{\mathcal{X}} \|^2_2. \tag{8.2}$$

Recall that $P_{\mathcal{X}}$ is the matrix encoding the 3D coordinates of the shape $\mathcal{X}$.

Intuitively, the main goal of the loss in Equation (8.2) is to compare the ground truth correspondence $\Pi^{gt}_{\mathcal{X}\mathcal{Y}}$ to the computed softmap matrix $S_{\mathcal{X}\mathcal{Y}}$. An alternative would to use the geodesic distances as weights as done in [186], but the geodesic distances are expensive and unreliable to compute on point clouds. Other options are the direct Frobenius loss on the permutation matrix or a *multinomial regression loss* as done in e.g. [207, 251]. However, these losses ignore the geometry and penalize incorrect correspondences independently of their proximity to correct ones. Instead, our loss penalizes incorrect correspondences based on the Euclidean distances of associated points. Moreover, Equation (8.2) can be seen as the comparison between the action of the ground-truth functional map in the full basis and the action of the estimated functional map on a specific set of functions that completely describe the geometry of the data. Our loss is efficient, takes the geometry into account, and is related to the Functional Maps formalism.

8.3.2 Learning the optimal transformation



As mentioned above, we train our approach in two stages: first we train an embedding network using the loss described in Section 8.3.1. We then train a separate network that aims to compute an optimal linear transformation between the embeddings, which can be used to compute correspondences at test time. Our observation is that this linear transformation can be obtained given enough constraints, by solving a linear system. Therefore, following the ideas in Deep Functional Maps [186] our second network **G** takes as input the natural embedding of a shape and outputs a set of p "probe" functions via $\mathbf{G}_\Theta(P_{\mathcal{X}}) = G_{\mathcal{X}}$ and $\mathbf{G}_\Theta(P_{\mathcal{Y}}) = G_{\mathcal{Y}}$ using shared trainable parameters Θ. We then minimize the following loss:

$$L(G_{\mathcal{X}}, G_{\mathcal{Y}}) = \| A^{gt}_{\mathcal{X}\mathcal{Y}} - \widehat{A}_{\mathcal{X}\mathcal{Y}} \|_2. \tag{8.3}$$

Here $A^{gt}_{\mathcal{X}\mathcal{Y}}$ is the ground truth linear transformation between the learned embeddings $A^{gt}_{\mathcal{X}\mathcal{Y}} = (\Phi^{+}_{\mathcal{X}} \Pi^{gt}_{\mathcal{X}\mathcal{Y}} \Phi_{\mathcal{Y}})^T$ whereas $\widehat{A}_{\mathcal{X}\mathcal{Y}} = \left((\Phi^{\dagger}_{\mathcal{Y}} G_{\mathcal{Y}})^T\right)^{\dagger} (\Phi^{\dagger}_{\mathcal{X}} G_{\mathcal{X}})^T$. This equation arises from the fact that if $A_{\mathcal{X}\mathcal{Y}}$ is the adjoint that aligns the embeddings then $A^T_{\mathcal{X}\mathcal{Y}}$ is a functional map from $\mathcal{Y}$ to $\mathcal{X}$ which implies that $A^T_{\mathcal{X}\mathcal{Y}} \Phi^{\dagger}_{\mathcal{Y}} G_{\mathcal{Y}} = \Phi^{\dagger}_{\mathcal{X}} G_{\mathcal{X}}$ whenever $G_{\mathcal{X}}, G_{\mathcal{Y}}$ are corresponding functions. We report in Appendix C a more detailed discussion

8.3.3 Test phase

Once we train these two networks, we can estimate the correspondence between an arbitrary pair of point clouds $\mathcal{X}$ and $\mathcal{Y}$ in four steps: (1) compute the embeddings $\Phi_{\mathcal{X}}$ and $\Phi_{\mathcal{Y}}$ using the embedding network **N**; (2) compute the set of probe functions, $G_{\mathcal{X}}$ and $G_{\mathcal{Y}}$ using the network **G**; (3) solve for the linear transformation $A_{\mathcal{X}\mathcal{Y}}$ using the expression given for $\widehat{A}_{\mathcal{X}\mathcal{Y}}$ above; (4) estimate for the correspondence $\Pi_{\mathcal{X}\mathcal{Y}}$ via nearest neighbor search as described in Equation (8.1).

Discussion While the basis and probe function networks appear similar as they both output a matrix, they are different in their losses and, consequently, in the task they solve. Our first linearly-invariant embedding (basis) network aims to output a representation in k dimensions so that different shapes share the same structure up to rotation and non-uniform scaling. Further, our loss in Equation (8.2) promotes continuity of the embedding with respect to the original shape coordinates. In contrast, the descriptor network aims to find a small set of reliable descriptors that can establish the linear transformation in the k dimensional space. Our strategy is different from a network which would aim to find an embedding where correspondences are directly

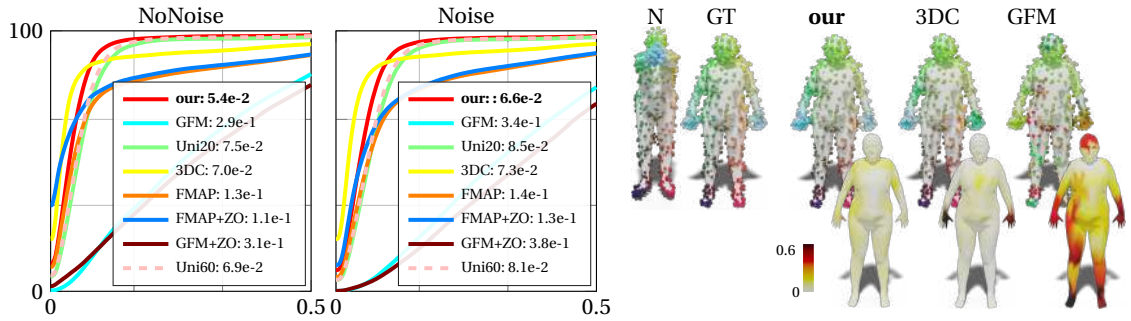


Fig. 8.2: The evaluation of the correspondence for point clouds generated from the FAUST dataset without or with additional noise. On the left, cumulative curves with mean error in the legends. On the right, a qualitative example in Noise setup, with the related hotmap error.

obtained as nearest neighbors (we call this option a "universal embedding"), as such a network would have to disambiguate each point directly. Instead, by first obtaining a smooth embedding and then using a small number of salient feature descriptors (probe functions in our case), our approach allows us to find a dense correspondence even in challenging cases, in which individual points may not be easy to distinguish.

8.4 Experiments

We evaluate our pipeline on the correspondence problem between non-rigid 3D point clouds in the challenging class of human models. We use this class because of the availability of data and baselines for comparison but stress that our method is general and can be applied to any shape category.

Architecture and parameters Both of our networks **N** and **G** are built upon the PointNet architecture [252]. For our experiments we train over 10K shapes from the SURREAL dataset [310], resampled at 1K vertices. We learn a $k = 20$ dimensional embedding (basis) and $p = 40$ probe functions for each point cloud.

8.4.1 Non-isometric pointclouds

We consider a first test set composed by the 100 shapes from the FAUST dataset [43] (10 subjects in 10 poses). We treat each shape as an unorganized point cloud selecting only $1K$ of its vertices and discarding mesh connectivity. We generate a second test set perturbing the first one with Gaussian noise. In both test sets, we deal with non-isometric

	Outlier
our	**2.7e-1**
Uni60	3.5e-1
GFM	3.5e-1

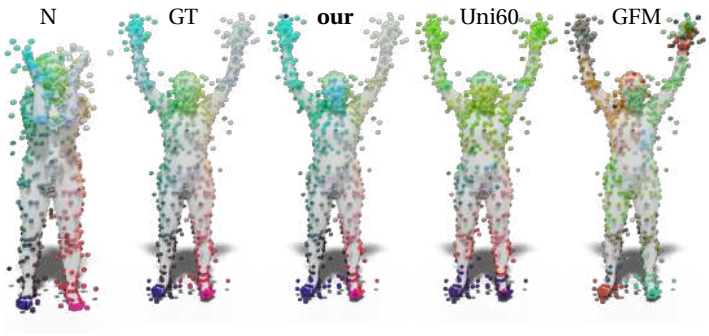


Fig. 8.3: Qantitative results on 100 pairs of the test set with 30% outlier points, compared to the baselines, with a qualitative example.

Fig. 8.4: Despite the presence of clutter, partiality and non-isometry our point cloud-based approach shows resilience.

pairs (different subjects) and strong non-rigid deformations (different poses). The second one is particularly challenging because it ruins the underlying shape structure.

As competitive baselines we consider *universal embeddings* (Uni20 and Uni60) obtained with the same architecture we used for **N** by learning 20 and 60 basis respectively, but enforcing the optimal linear transformation to be identity. We also compare our method with the standard functional maps, with 5 ground-truth landmarks (FMAP), the recent state-of-the-art methods (GFM) [91], and finally against 3D-CODED [127] (3DC). For the GFM and FMAP methods we also compare to a version refined with ZoomOut [214] (FMAP+ZOO, GFM+ZOO). For the methods that require the LBO basis, we adopt the estimation of LBO for point clouds proposed in [80].

As can be seen in Figure 8.2, we outperform the baseline and all the competitors including the state-of-the-art methods GFM and 3DC in both the considered scenarios. We stress that both [127] and [91] are very recent highly complex state-of-the-art methods, with e.g. [127] being directly adapted to point clouds with an expensive test-time post-processing. Our method achieves state-of-the-art results without any additional post-processing. Further robustness of our method is illustrated in Figure 8.3, where we evaluate our networks trained on clean data, on the FAUST test set augmented with outliers points. Our method shows resilience and outperforms competing methods in this challenging setting, despite not being presented with outlier data at training time.

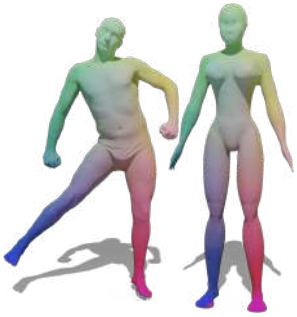


Fig. 8.5: A result of two shapes from different discreti zation and datasets, from the collection presented in Chapter 6.

In Figure 8.4 we also visualize a correspondence, computed using our network, between a pair of real-world scans taken from the *Scan the world* project collection [11]. The presence of significant topological changes, partiality, clutter, non-isometry and self-intersections represents a considerable challenge. Despite this, our method, shows remarkable resilience and provides a reliable result even without retraining or post-processing. Finally, in Figure 8.5 we show a result over two shapes from the collection presented in Chapter 6. The left one is from a real-world scan, while the second is a non-human with different proportion and structure.

8.4.2 Fragmented partiality

Finally, we compare our approach, the universal embedding and the LBO basis (LBO) in an extreme scenario. We compute a correspondence between each of the 100 full shapes from FAUST and a fragmented version that consists of several small disconnected components. This experiment tests how each basis is affected by heavy loss of geometry. Fixing a basis, we evaluate 1) the matching using a ground-truth transformation to retrieve the optimal linear transformation, on the left of Figure 8.6; 2) the correspondence estimated with the best pipeline for the given basis, on the right. The average geodesic errors are reported in the legends. In 2) for LBO we consider partial functional maps (PFM) [265], which extends the functional maps framework to partial cases. In the middle we visualize a qualitative comparison on one of the 100 pairs tested, where the correspondence in encoded by the color transfer. We highlight that it is not always possible to have a transformation that produces a perfect matching. LBO+opt and PFM suffer from the significant sensitivity of the LBO to partiality and topological noise. The universal embedding shows also a significant loss of information. With the

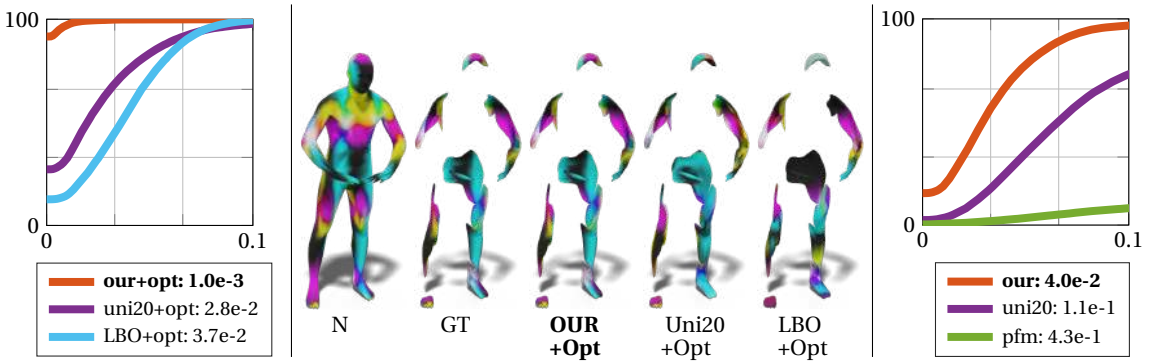


Fig. 8.6: Partial setup. The shape is matched with a fragmented version of itself. We show the amount of information lost by the basis due to surface destruction and compare our method to baseline and partial functional map (pfm) [265]. More details in the main text.

linear invariant embedding it is still possible to retrieve good information and generalize to corrupted data that are completely unseen during training.

8.5 Conclusion

This Chapter presented an extension to the Functional Maps framework by replacing the standard Laplace-Beltrami eigenfunctions with learned functions. We achieve this by learning an optimal linearly-invariant embedding and a separate network that aligns embeddings of different shapes.

While general, our approach still assumes that the input data poses a “natural” embedding in 3D making it yet not applicable to data such as graphs. Moreover, we do not exploit the mesh structure that *might* be available in certain cases. Combining our method with a mesh-aware approach is an exciting direction for future work.

Our preliminary investigation outperforms the competitors in challenging scenarios. We believe that these results only scratch the surface and pave the way for future work on invariant embeddings for shape correspondence and other related problems.

9

Instant recovery of shape from spectrum via latent space connections

We introduce the first learning-based method for recovering shapes from Laplacian spectra. Our model consists of a cycle-consistent module that maps between learned latent vectors of an auto-encoder and sequences of eigenvalues. This module provides an efficient and effective linkage between Laplacian spectrum and geometry. Our data-driven approach replaces the need for ad-hoc regularizers required by prior methods, while providing more accurate results at a fraction of the computational cost. Our learning model applies without modifications across different dimensions (2D and 3D shapes alike), representations (meshes, contours and point clouds), as well as across different shape classes, and admits arbitrary resolution of the input spectrum without affecting complexity. The increased flexibility allows us to address notoriously difficult tasks in 3D vision and geometry processing within a unified framework, including shape generation from spectrum, mesh super-resolution, shape exploration, style transfer, spectrum estimation from point clouds, segmentation transfer and point-to-point matching.

9.1 Introduction

Constructing compact encodings of geometric shapes lies at the heart of 2D and 3D Computer Vision. While earlier approaches have concentrated on handcrafted representations, with the advent of geometric deep learning [56, 208], data-driven *learned* feature encodings have gained prominence. A desirable property in many applications, such as shape exploration and synthesis, is to be able to recover the shape from its (latent) encoding, and various auto-encoder architectures have been designed to solve this problem [14, 114, 185, 222]. Despite significant progress in this area, the structure of the latent vectors is arduous to control. For example, the dimensions of the latent vectors typically lack a canonical ordering, while invariance to various geometric de-

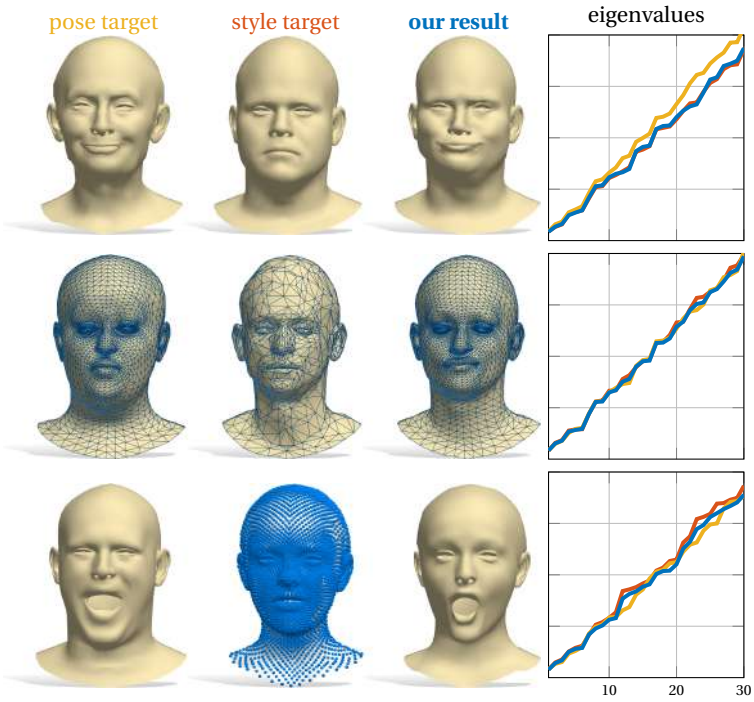


Fig. 9.1: Our spectral reconstruction enables correspondence-free style transfer. Given pose and style "donors" (left and middle columns respectively), we synthesize a new shape with the pose of the former and the style of the latter. The generation is driven by a learning-based eigenvalues alignment (rightmost plots). Our approach handles different resolutions (middle row) and representations (bottom row; the surface underlying the point cloud is only for visualization purposes).

formations is often only learned by data augmentation or complex constraints on the intermediate features.

At the same time, a classical approach in spectral geometry is to encode a shape using the sequence of eigenvalues (spectrum) of its Laplacian operator. This representation is useful since: (1) it does not require any training, (2) it can be computed on various data representations, such as point clouds or meshes, regardless of sampling density, (3) it enjoys well-known theoretical properties such as a natural ordering of its elements and invariance to isometries, and (4) as shown recently [85,255], alignment of eigenvalues often promotes near-isometries, which is useful in multiple tasks such as non-rigid shape retrieval and matching problems.

Unfortunately, although encoding shapes via their Laplacian spectra can be straightforward (at least for meshes), the inverse problem of recovering the shape is very dif-

ficult. Indeed, it is well-known that certain pairs of non-isometric shapes can have the same spectrum, or in other words "one cannot hear the shape of a drum" [124]. At the same time, recent evidence suggests that such cases are pathological and that *in practice* might be possible to recover a shape from its spectrum [85]. Nevertheless, existing approaches [85], while able to deform a shape into another with a given spectrum, can produce highly unrealistic shapes with strong artifacts failing in a large number of cases.

In this Chapter, we combine the strengths of data-driven autoencoders with those of spectral methods. Our key idea is to construct a single architecture capable of synthesizing a shape from a learned latent code and from its Laplacian eigenvalues. We show that by explicitly training networks that aim to translate between the learned latent codes and the spectral encoding, we can recover a shape from its eigenvalues and endow the latent space with certain desirable properties. Remarkably, our shape-from-spectrum solution is extremely efficient since it requires a single pass through a trained network, unlike expensive iterative optimization methods with ad-hoc regularizers [85]. Among the applications, we also propose a new efficient and compact approach for point-to-point matching directly from the Laplacian spectrum. It is used as a bridge to bring different geometries in the same discretization, and so naturally in correspondence. Furthermore, our trainable module acts as a proxy to differentiable eigendecomposition, while encouraging geometric consistency within the network.

Overall, our key **contributions** can be summarized as follows:

- We propose the first learning-based model to robustly recover shape from Laplacian spectra *in a single pass*;
- For the first time, we provide a bidirectional linkage between learned 3D latent space and spectral geometric properties of 3D shapes;
- Our model is *general*, in that it applies with no modifications to different classes even across different geometric representations and dimensions and to data that does not belong to the datasets used at training time;
- We showcase our approach in multiple applications (e.g., Fig. 9.1), and show significant improvement over the state of the art; see Fig. 9.2 for an example.

The code of our method is available online [6].

9.2 Related work

Spectral quantities and in particular the eigenvalues of the Laplace-Beltrami operator provide an informative summary of the intrinsic geometry. For example, closed-form estimates and analytical bounds for surface area, genus and curvature in terms of the Laplacian eigenvalues have been obtained [69]. Given these properties, spectral shape

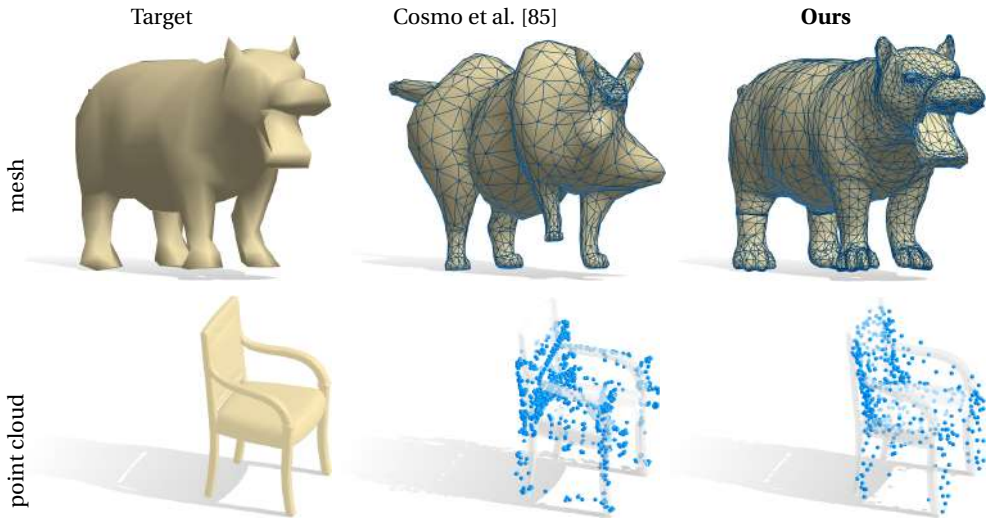


Fig. 9.2: Comparison in estimating a shape from its Laplacian spectrum between the state-of-the-art method [85] (middle) and ours (right) for a mesh and a point cloud. The shapes recovered by our method are significantly closer to the target.

analysis has been exploited in many computer vision and computer graphics tasks such as shape retrieval [260], description and matching [54, 209, 234, 296], mesh segmentation [259], sampling [236] and compression [162] among many others. Typically, the intrinsic properties of the shape are computed from its explicit representation and are used to encode compact geometric features invariant to isometric deformations.

Recently, several works have started to address the inverse problem: namely, recovering an extrinsic embedding from the intrinsic encoding [47, 85]. This is closely related to the fundamental theoretical question of "hearing the shape of the drum" [124, 160]. Although counterexamples have been proposed to show that in certain scenarios multiple shapes might have the same spectrum, there is recent work that proposes effective practical solutions to this problem. In [47] the shape-from-operator method was proposed, aiming at obtaining the extrinsic shape from a Laplacian matrix where the 3D reconstruction was recovered after the estimation of the Riemannian metric in terms of edge lengths. In [84] the intrinsic and extrinsic relations of geometric objects have been extensively defined and evaluated from both theoretical and practical aspects. The authors revised the framework of functional shape differences [273] to account of extrinsic structure extending the reconstruction task to non-isometric shapes and models obtained from physical simulation and animation. Several works have also been proposed to recover shapes purely from Laplacian *eigenvalues* [12, 78, 237] or with mild additional

information such as excitation amplitude in the case of musical key design [38]. Most closely related to ours in this area is the recent *isopectralization* approach introduced in [85], that aims directly to estimate the 3D shape from the spectrum. This approach works well in the vicinity of a good solution but is both computationally expensive and, as we show below, can quickly produce unrealistic instances, failing in a large number of cases in 3D, as shown in Fig. 9.2 for two examples.

In this Chapter we contribute to this line of work, and propose to replace the heuristics used in previous methods such as [85] with a purely data-driven approach for the first time. Our key idea is to design a deep neural network, that both constraints the space of solutions based on the set of shapes given at training, and at the same time, allows us to solve the isospectralization problem with a *single forward pass*, thus avoiding expensive and error-prone optimization.

We note that a related idea has been recently proposed in [147] via the so-called OperatorNet architecture. However, that work is based on shape difference operators [273] and as such requires a fixed source shape and functional maps to each shape in the dataset to properly synthesize a shape. Our approach is based on Laplacian eigenvalues alone and thus is completely correspondence-free.

Our approach also builds upon the recent work on learning generative shape models. A range of techniques have been proposed using the volumetric representations [323], point cloud autoencoders [14, 28], generative models based on meshes and implicit functions [73, 126, 167, 185, 289], and part structures [114, 176, 222, 324], among many others.

Although generative models, and in particular autoencoders, have shown impressive performance, the latent space structure is typically difficult to control or analyze directly. To address this problem, some methods proposed a disentanglement of the latent space [28, 324] to split it into more semantic regions. Perhaps most closely related to ours in this domain, is the work in [28], where the shape spectrum is used to promote disentanglement of the latent space intro intrinsic and extrinsic components, that can be controlled separately. Nevertheless, the resulting network does not allow to synthesize shapes from their spectra.

Extending the studies of these approaches, our work provides the first way to connect the learned latent space to the spectral one, thus inheriting the benefits and providing the versatility of moving across the two representations. This allows our network to synthesize shapes from their spectra, and also to relate shapes with very different input structure (e.g., meshes and point clouds) across a vastness of sampling densities, enabling several novel applications.

9.3 Background

We model shapes as connected 2-dimensional Riemannian manifolds $\mathscr{X}$ embedded in $\mathbb{R}^3$, possibly with boundary $\partial\mathscr{X}$, equipped with the standard metric. On each shape $\mathscr{X}$ we consider its positive semi-definite Laplace-Beltrami operator $\Delta_{\mathscr{X}}$, generalizing the classical notion of Laplacian from the Euclidean setting to curved surfaces.

Laplacian spectrum. $\Delta_{\mathscr{X}}$ admits an eigendecomposition

$$\Delta_{\mathscr{X}}\phi_i(x) = \lambda_i\phi_i(x) \qquad x \in \mathrm{int}(\mathscr{X}) \tag{9.1}$$

$$\langle \nabla\phi_i(x), \hat{n}(x)\rangle = 0 \qquad x \in \partial\mathscr{X} \tag{9.2}$$

into eigenvalues $\{\lambda_i\}$ and associated eigenfunctions $\{\phi_i\}$[1].

The Laplacian eigenvalues of $\mathscr{X}$ (its *spectrum*) form a discrete set, which is canonically ordered into a non-decreasing sequence

$$\mathrm{Spec}(\mathscr{X}) := \{0 = \lambda_0 < \lambda_1 \leq \lambda_2 \leq \cdots\}. \tag{9.3}$$

In the special case where $\mathscr{X}$ is an interval in $\mathbb{R}$, the eigenvalues λ_i correspond to the (squares of) oscillation frequencies of Fourier basis functions ϕ_i. This provides us with a connection to classical Fourier analysis and a natural notion of hierarchy induced by the eigenvalues' ordering. In the light of this analogy, in practice, one is usually interested in a limited bandwidth consisting of the first $k > 1$ eigenvalues; typical values in geometry processing applications range from $k = 30$ to 100.

Furthermore, the spectrum is *isometry-invariant*, i.e., it does not change with deformations of the shape that preserve geodesic distances (e.g., changes in pose).

Discretization. In the discrete setting, we represent shapes as triangle meshes $X = (V, T)$ with n vertices V and m triangular faces T; depending on the application, we will also consider unorganized point clouds. Vertex coordinates in both cases are represented by a matrix $\mathbf{X} \in \mathbb{R}^{n\times 3}$.

The Laplace-Beltrami operator $\Delta_{\mathscr{X}}$ is discretized as a $n \times n$ matrix via the finite element method (FEM) [79]. In the simplest setting (i.e., linear finite elements), this discretization corresponds to the cotangent Laplacian [246]; however, in this Chapter we use *cubic* FEM (see e.g. [259, Section 4.1] for a clear treatment), since it yields a more accurate discretization as shown in Fig. 9.3. Differently from [85, 255], this comes at virtually no additional cost for our pipeline, as we show in the sequel. On point clouds, $\Delta_{\mathscr{X}}$ can be discretized using the approach described in [48, 80].

[1] Similarly to [85] we use homogeneous Neumann boundary conditions; see Equation (9.2), where $\hat{n}(x)$ denotes the outward normal to the boundary.

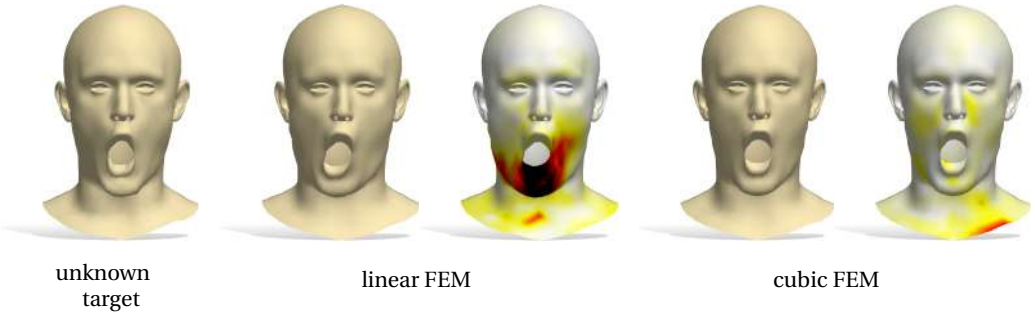
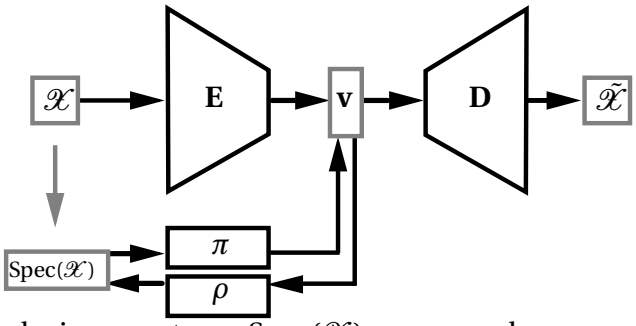


Fig. 9.3: Reconstruction examples of our shape-from-spectrum pipeline. We show the results obtained with two different inputs: the eigenvalues of the Laplacian discretized with linear FEM, and those of the cubic FEM discretization. The heatmap encodes point-wise reconstruction error, growing from white to dark red.

9.4 Method

Our main contribution is a deep learning model for recovering shapes from Laplacian eigenvalues. Our model operates in an end-to-end fashion: given a spectrum as input, it directly yields a shape with a single forward pass, thus avoiding expensive test-time optimization.

Motivation. Our rationale lies in the observation that shape semantics can be learned from the data, rather than by relying upon the definition of ad-hoc regularizers [85], often resulting in unrealistic reconstructions. For example, a sheet of paper can be *isometrically* crumpled or folded into a plane (see inset figure). Since both embeddings have the same eigenvalues, the desirable reconstruction must be imposed as a prior. By taking a data-driven approach, we make our method aware of the "space of realistic shapes", yielding both a dramatic improvement in accuracy and efficiency, and enabling new interactive applications.



Latent space connections. Our key idea is to construct an auto-encoder (AE) neural network architecture, augmented by explicitly modeling the connections between the AE's latent space and the Laplacian spectrum of the input shape; see the inset Figure for an illustration of our learning model. The input shape $\mathscr{X}$ and its Laplacian spectrum Spec($\mathscr{X}$) are passed, respec-

tively, through an AE enforcing $\mathscr{X} \approx \tilde{\mathscr{X}}$, and an invertible module (π, ρ) mapping the eigenvalue sequence to a latent vector $\mathbf{v}$. The two branches are trained simultaneously, forcing $\mathbf{v}$ to be updated accordingly. The trained model allows to recover the shape purely from its eigenvalues via the composition $D(\pi(\mathrm{Spec}(\mathscr{X}))) \approx \mathscr{X}$.

Loosely speaking, our approach can be seen as implementing a coupling between two latent spaces: a learned one that operates on the shape embedding $\mathscr{X}$, and the one provided by the eigenvalues $\mathrm{Spec}(\mathscr{X})$. In the former case, the *encoder* E is trainable, whereas the mapping $\mathscr{X} \to \mathrm{Spec}(\mathscr{X})$ is provided via the eigen-decomposition and fixed a priori. Finally, we introduce the two coupling mappings π, ρ, trained with a bidirectional loss, to both enable communication across the latent spaces and to tune the learned space by endowing it with structure contained in $\mathrm{Spec}(\mathscr{X})$.

We phrase our overall training loss as follows:

$$\ell = \ell_{\mathscr{X}} + \alpha \ell_{\lambda}, \quad \text{with} \tag{9.4}$$

$$\ell_{\mathscr{X}} = \frac{1}{n} \| D(E(\mathbf{X})) - \mathbf{X} \|_F^2 \tag{9.5}$$

$$\ell_{\lambda} = \frac{1}{k} (\| \pi(\boldsymbol{\lambda}) - E(\mathbf{X}) \|_2^2 + \| \rho(E(\mathbf{X})) - \boldsymbol{\lambda} \|_2^2) \tag{9.6}$$

where $\boldsymbol{\lambda}$ is a vector containing the first k eigenvalues in $\mathrm{Spec}(\mathscr{X})$, $\mathbf{X}$ is the matrix of point coordinates, E is the encoder, D is the decoder, $\| \cdot \|_F$ denotes the Frobenius norm, and $\alpha = 10^{-4}$ controls the relative strengths of the reconstruction loss $\ell_{\mathscr{X}}$ and the spectral term ℓ_{λ}. The blocks D, E, π, and ρ are learnable and parametrized by a neural network. Equation (9.6) enforces $\rho \approx \pi^{-1}$; in other words, π and ρ form a translation block between the latent vector and the spectral encoding of the shape.

At test time, we recover a shape from the spectrum Spec simply via the composition $D(\pi(\mathrm{Spec}))$ (Section 9.5). For additional applications we refer to Section 9.6.

Shape representation. We consider two different settings: triangle meshes in point-to-point correspondence *at training time* (typical in graphics and geometry processing), and unorganized point clouds *without* a consistent vertex labeling (typical in 3D computer vision).

Autoencoder architecture. Our model can be built with potentially any autoencoder. In our applications we chose relatively simple ones to deal with meshes and unorganized point clouds, although more powerful generative methods would be equally possible. The latent space dimension is fixed to 30 (the same as k).

Remark. Our architecture takes $\mathrm{Spec}(\mathscr{X})$ as an input, i.e., the eigenvalues are *not* computed at training time. By learning an *invertible* mapping to the latent space, we avoid expensive backpropagation steps through the spectral decomposition of the Laplacian

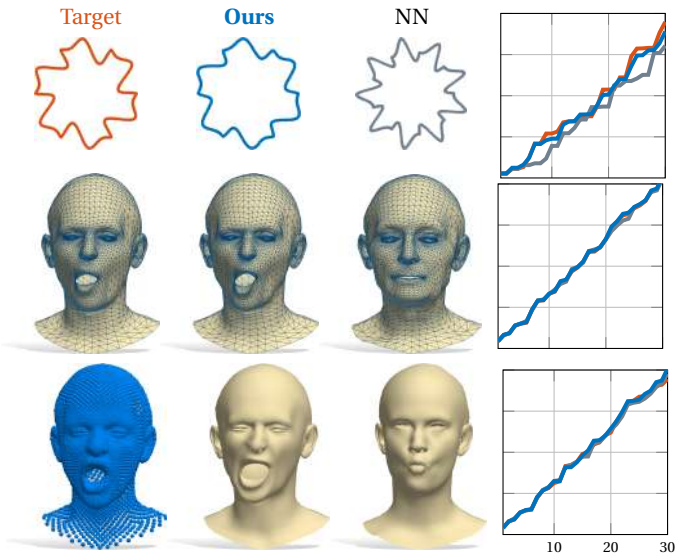


Fig. 9.4: Shape reconstruction from eigenvalues using our approach on different representations (i.e. 2D contours, 3D meshes and point clouds). The eigenvalues of the shapes on the left are given to our network, which outputs the shapes in the middle. For each representation, the eigenvalues are computed on the appropriate Laplacian discretization as per Section 9.3. The NN column shows the nearest-neighbor solution sought in the training set.

$\Delta_{\mathscr{X}}$. In this sense, the mapping ρ acts as an efficient proxy to differentiable eigendecomposition, which we exploit in several applications below.

Since eigenvalue computation is only incurred as an offline cost, it can be performed with arbitrary accuracy (we use cubic FEM, see Fig. 9.3) without sacrificing efficiency.

9.5 Results

In this section we report the results on our core application of shape from spectrum recovery.

To evaluate our method, we trained our model on 1,853 3D shapes from the COMA dataset [256] of human faces; 100 shapes of an unseen subject are used for the test set. We repeated this test at four different mesh resolutions: ~4K (full resolution), 1K, 500 and 200 vertices respectively. For each resolution, we independently compute the Laplacian spectrum and use these spectra to recover the shape.

Comparison. We compared our method in terms of reconstruction accuracy to the state-of-the-art isospectralization method of Cosmo et al. [85], as well as to a nearest-neighbors baseline, consisting in picking the shape of the training set with the closest spectrum to the target one.

In addition, we trained two separate architectures (with and without the ρ block) and compared them.

The test without this network component is an ablation study to validate the importance of the *invertible* module connecting the spectral encoding to the learned latent codes.

	full res	1000	500	200
Ours	**1.61**	**1.62**	**1.71**	**2.13**
Ours without ρ	1.89	1.82	2.06	2.42
NN	4.45	4.63	4.01	2.65
Cosmo et al. [85]	–	16.4	7.11	4.08

Table 9.1: Shape-from-spectrum reconstruction comparisons with nearest neighbors (between spectra) baseline and a state of the art spectral approach; we report average error over 100 shapes of an unseen subject from the COMA dataset [256]. Best results (in bold) are obtained with our full pipeline. '–' denotes out of memory; all errors must be rescaled by 10^{-5}.

The quantitative results are reported in Table 9.1 as the mean squared error between the reconstructed shape and the ground-truth. Figures 9.2 and 9.4 further show qualitative comparisons with the different baselines involving different shape representations. In Fig. 9.4, for the sake of illustration, similarly to [85, 255], we also include 2D contours, discretized as regular cycle graphs.

As the results suggest, the ρ block both contributes to reduce the reconstruction error, and to enable novel applications (see in Section 9.6). Note that our method achieves a significant improvement over nearest neighbors in terms of accuracy, and an order of magnitude improvement over isospectralization. The latter approach also consists of an expensive optimization that requires hours to run, while our method is instantaneous at test time.

Spectral bandwidth has a direct effect on reconstruction accuracy, since increasing this number brings more high-frequency detail into the representation. Following [85, 255, 271], in all our experiments we use $k = 30$.

9.6 Additional applications

Our general model enables several additional applications, by exploiting the connection between spectral properties and shape generation.

Style transfer. As shown in Fig. 5.1, we can use our trained network to transfer the style of a shape $\mathscr{X}_{\text{style}}$ to another shape $\mathscr{X}_{\text{pose}}$ having both a different style and pose. This is

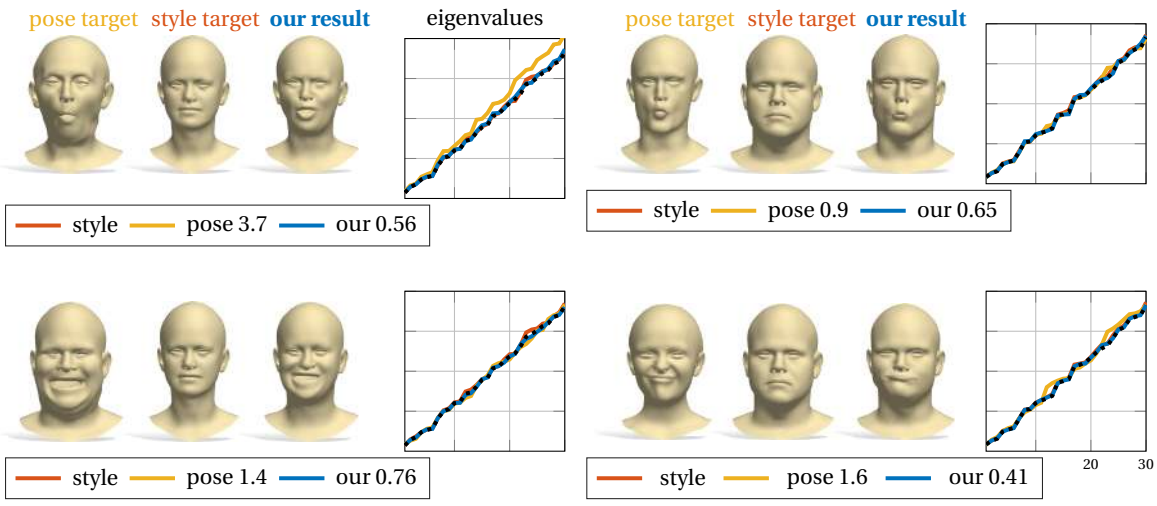


Fig. 9.5: Examples of style transfer. The target style (middle) is applied to the target pose (left) by solving problem (9.7) and then decoding the resulting latent vector (right). For each example we also report the corresponding eigenvalue alignment (rightmost plots). The black dotted line is the image of ρ. The numbers in the legend denote the distance from the target "style" spectrum to the source pose and to our generated shape; a small number suggests near-isometry between the generated shape and the style target.

done by a search in the latent space, phrased as:

$$\min_{\mathbf{v}} \|\mathrm{Spec}(\mathscr{X}_{\mathrm{style}}) - \rho(\mathbf{v})\|_2^2 + w\|\mathbf{v} - E(\mathscr{X}_{\mathrm{pose}})\|_2^2 \tag{9.7}$$

Here, the first term seeks a latent vector whose associated spectrum aligns with the eigenvalues of $\mathscr{X}_{\mathrm{style}}$; in other words, we regard style as an intrinsic property of the shape, and exploit the fact that the Laplacian spectrum is invariant to pose deformations. The second term keeps the latent vector close to that of the input pose (we initialize with $\mathbf{v}_{\mathrm{init}} = E(\mathscr{X}_{\mathrm{pose}})$). We solve the optimization problem by back-propagating the gradient of the cost function of Equation (9.7) with respect to $\mathbf{v}$ through ρ.

The sought shape is then given by a forward pass on the resulting minimizer. In Fig. 9.5, we show four examples.

We emphasize here that the style is purely encoded in the input eigenvalues, therefore it does not rely on the test shapes being in point-to-point correspondence with the training set. This leads to the following:

Property 9.1. Our method can be used in a **correspondence-free** scenario. By taking eigenvalues as input, it enables applications that traditionally require a correspondence, but side-steps this requirement.

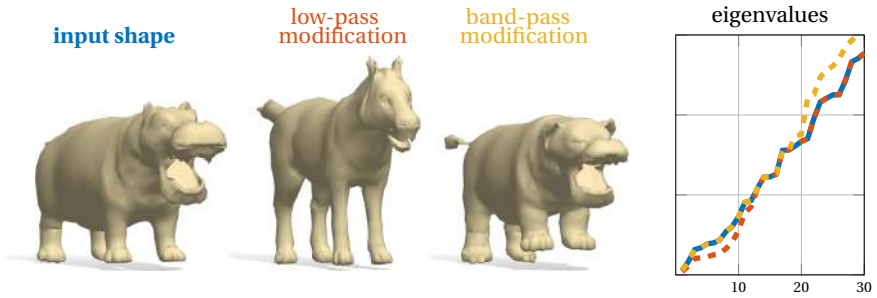


Fig. 9.6: Exploring the space of shapes via manipulation of the spectrum. The low-pass modification (middle) decreases the first 12 eigenvalues of the input shape, leading to more pronounced geometric features (e.g. longer legs and snout); the band-pass modification (right) amplifies the last 12 eigenvalues, affecting the high-frequency details (e.g. the ears and fingers);

This observation was also mentioned in other spectrum-based approaches [85, 255]. However, the data-driven nature of our method makes it more robust, efficient and accurate, therefore greatly improving its practical utility.

Shape exploration. The previous results suggest that eigenvalues can be used to drive the exploration of the AE's latent space toward a desired direction. Another possibility is to regard *the eigenvalues themselves* as a parametric model for isometry classes, and explore the "space of spectra" as is typically done with latent spaces. Our bi-directional coupling between spectra and latent codes makes this exploration feasible, as remarked by the following property:

Property 9.2. Latent space connections provide both a means for **controlling** the latent space, and vice-versa, enable **exploration** of the space of Laplacian spectra.

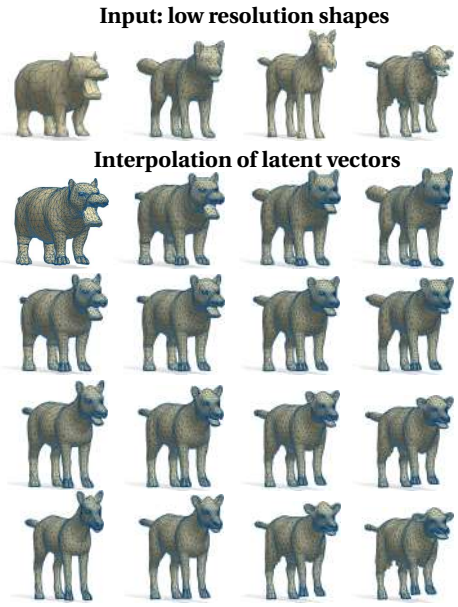


Fig. 9.7: Latent space interpolation of four low-resolution shapes with different connectivity (top row, unseen at training). The spectra of the input shapes are mapped via π to the latent space, where they are bilinearly interpolated and then decoded to $\mathbb{R}^3$. The reconstructions of the input are depicted at the corners of the grid.

Since eigenvalues change continuously with the manifold metric [32], a small variation in the spectrum will give rise to a small change in the geometry. We can visualize such variations in shape directly, by first deforming a given spectrum (e.g., by a simple linear interpolation between two spectra) to obtain the new eigenvalue sequence $\boldsymbol{\mu}$, and then directly computing $D(\pi(\boldsymbol{\mu}))$.

In Fig. 9.7 we show a related experiment. Here we train the network on 4,430 animal meshes generated with the SMAL parametric model following the official protocol [348]. Given four *low-resolution* shapes $\mathscr{X}_i$ as input, we first compute their spectra $\mathrm{Spec}(\mathscr{X}_i)$, map these to the latent space via $\pi(\mathrm{Spec}(\mathscr{X}_i))$, perform a bilinear interpolation of the resulting latent vectors, and finally reconstruct the corresponding shapes. Finally, in Fig. 9.6 we show an example of interactive spectrum-driven shape exploration. Given a shape and its Laplacian eigenvalues as input, we navigate the space of shapes by directly modifying different frequency bands with the aid of a simple user interface. The modified spectra are then decoded by our network in *real time*. The interactive nature of this application is enabled by the efficiency of our shape from spectrum recovery (obtained in a single forward pass) and would not be possible with previous methods [85] that rely on costly test-time optimization.

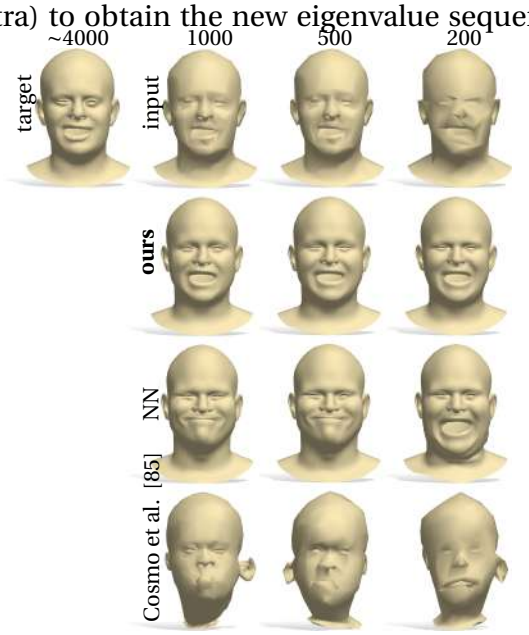


Fig. 9.8: Mesh super-resolution for inputs at decreasing resolution (top row, left to right). Our method fits closely the original input shapes (top left), while other approaches either predict the wrong pose (NN baseline) or generate an unrealistic shape (Cosmo et al.).

Super-resolution. A key feature that emerges from the experiment in Fig. 9.7 is the perfect reconstruction of the low-resolution shapes once their eigenvalues are mapped to the latent space via π. This brings us to a fundamental property of our approach:

Property 9.3. Since eigenvalues are largely **insensitive to mesh resolution and sampling**, so is our trained network.

This fact is especially evident when using cubic FEM discretization, as we do in all our tests, since it more closely approximates the continuous setting and is thus much less affected by the surface discretization.

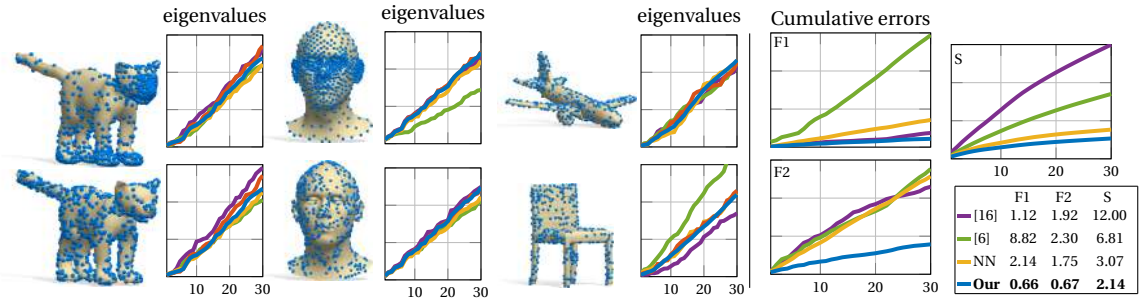


Fig. 9.9: Evaluations of point cloud spectra estimation. **On the left** we show the qualitative comparison for different samplings on three classes (animals, human faces and objects). We show the eigenvalues estimations alongside the input point cloud (depicted as surface samplings), and the ground truth spectrum (in red). **On the last two columns**, we report the average cumulative error curves evaluated on the FLAME dataset for the two different distributions (F1 and F2) and on ShapeNet (S).

Remark. It is worth mentioning that existing methods can employ cubic FEM as well; however, this soon becomes prohibitively expensive due to the differentiation of spectral decomposition required by their optimizations [85, 255].

These properties allow us to use our network for the task of mesh super-resolution. Given a low-resolution mesh as input, we aim to recover a higher resolution counterpart of it. Furthermore, while the input mesh has *arbitrary* resolution and is unknown to the network (and a correspondence with the training models is *not* given), an additional desideratum is for the new shape to be in dense point-to-point correspondence with models from the training set. We do so in a single shot, by predicting the decoded shape as:

$$\mathscr{X}_{\text{hires}} = D(\pi(\text{Spec}(\mathscr{X}_{\text{lowres}}))) . \tag{9.8}$$

This simple approach exploits the resolution-independent geometric information encoded in the spectrum along with the power of a data-driven generative model.

In Fig. 9.8 we show a comparison with nearest-neighbors between eigenvalues (among shapes in the training set), and the isospectralization method of Cosmo et al. [85]. Our solution closely reproduces the high-resolution target. Isospectralization correctly aligns the eigenvalues, but it recovers unrealistic shapes due to ineffective regularization. This phenomenon highlights the following

Property 9.4. Our data-driven approach replaces ad-hoc regularizers, that are difficult to model axiomatically, with **realistic priors** learned from examples.

This is especially important for deformable objects; shapes falling into the same isometry class are often hard to disambiguate without using geometric priors.

Estimating point cloud spectra. As an additional experiment, we show how our network can directly predict Laplacian eigenvalues for unorganized point clouds. This task is particularly challenging due to the lack of a structure in the point set, and existing approaches such as [35, 80] often fail at approximating the eigenvalues of the underlying surface accurately. The difficulty is even more pronounced when the point sets are irregularly sampled, as we empirically show here. In our case, estimation of the spectrum boils down to the single forward pass:

$$\widetilde{\mathrm{Spec}}(\mathscr{X}) = \rho(E(\mathscr{X}))\,. \tag{9.9}$$

To address this task we train our network by feeding unorganized point clouds as input, together with the spectra computed from the corresponding meshes (which are available at training time). For this setting we use a PointNet [252] encoder and a fully connected decoder, and we replace the reconstruction loss of Equation (9.5) with the Chamfer distance. This application highlights the generality of our model, which can accommodate different representations of geometric data.

We consider two types of point clouds: (1) with similar point density and regularity as in the training set, and (2) with randomized non-uniform sampling. We compare the spectrum estimated via $\rho(E(\mathscr{X}))$ to axiomatic methods [35, 80], and to the NN baseline (applied in the latent space); see Fig. 9.9. The qualitative results are obtained by training on SMAL [348] (left), COMA [256] (middle) and ShapeNet watertight [145] (right). To highlight its generalization capability, the network trained on COMA is tested on point clouds from the FLAME dataset, while on ShapeNet we consider 4 different classes (airplanes, boats, screens and chairs). We compute the cumulative error curves of the distance between the eigenvalues from the meshes corresponding to the test point clouds. The mean error across all test sets is also reported in the legend. Our method leads to a significant improvement over the closest state-of-the-art baseline [35].

Matching from spectrum. Finally, we compute dense correspondences between shape pairs using only their spectra. These are fed into our network; since the output points are naturally ordered by the decoder, we exploit this to establish a sparse correspondence. In the case of meshes, we extend it to a dense one by using the functional maps framework [234]. In the case of point clouds, we can propagate a semantic segmentation using nearest neighbors. We perform a quantitative evaluation on SMAL [348], testing on 100 non-isometric pairs of animals from different classes. Two applications that benefit from our approach are texture and segmentation transfer; we tested them respectively on animals and segmented ShapeNet [336]. The comparison baseline consists of 100 iterations of ICP [37] to rigidly align the two shapes followed by nearest-neighbor assignment as correspondence (see Fig. 9.10).

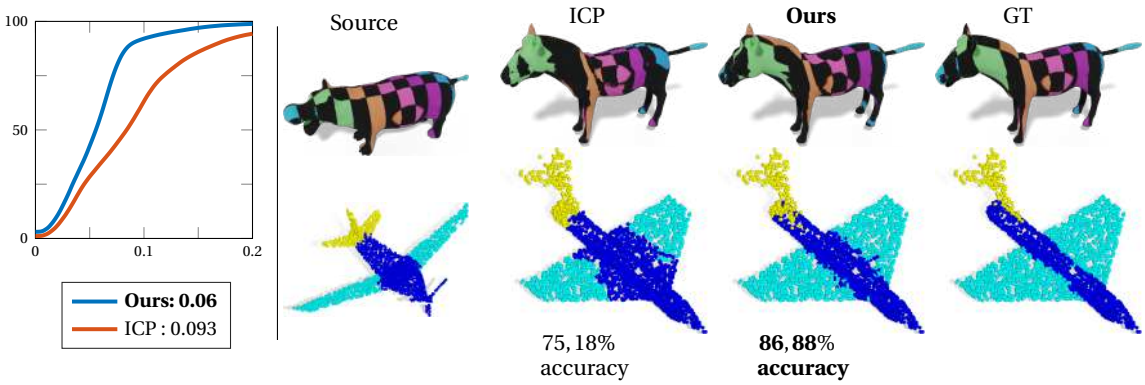


Fig. 9.10: On the left, quantitative evaluation of matching [165] between 100 pairs of animals. On the right, the qualitative comparison on texture and segmentation transfer.

9.7 Conclusions

We introduced the first data-driven method for shape generation from Laplacian spectra. Our approach consists in enriching a standard AE with a pair of cycle-consistent maps, associating ordered sequences of eigenvalues to latent codes and vice-versa. This explicit coupling brings forth key advantages of spectral methods to generative models, enabling novel applications and a significant improvement over existing approaches. Our limitations are shared with other spectral methods in the computation of a robust Laplacian discretization. Adopting the recent approach [284] for such borderline cases is a promising possibility. Further, while the Laplacian is a classical choice due to its Fourier-like properties, spectra of other operators with different properties may lead to other promising applications.

10

Thesis Conclusions

This thesis investigated different representations with their peculiarities, exploring how their properties impact 3D shape matching problem, and proposing innovative solutions to overcome their limitations. We introduced theoretical advancements in the field, moving from the shape matching as a correspondence between representations, to a correspondence between geometries.

Our work opens to many future directions. An emerging trend in functional matching pipelines is including higher frequencies. Recent methods showed exciting results on them [97, 214]. We believe such unstable representation and its detail level will produce much attention to handle them properly. A recent work [98] introduced a learning-based matching pipeline with high-frequencies, replacing the nearest-neighbor in the spectral domain with an optimal transport approximation, and so moving the perspective from a point-to-point pairing to a more global method. This latter work proposes to rediscuss some underlying structure of standard matching approaches. For future, such changes could be extended to many different aspects of the matching pipeline, similarly to our Chapter 8 (in which we propose to replace Laplace-Beltrami eigenfunctions). In particular, basis functions can be expressed as non-linear ones, e.g., other neural networks, using the so called hypernetworks [130]. Similarly, the standard metric of the embedding space and the linear transformation can be replaced by some more sophisticated methods, aiming to *Hyper-Functional Maps* framework. On the same line of work, we just introduced an intrinsic/extrinsic paradigm that could be extended to work at higher frequencies (i.e., catch local extrinsic features and relaxing the same-pose constraint).

Also, many representations are left unexplored. Signed Distance Functions (SDF) have recently shown promising results in shape matching task [39]; find a correspondence in these representations would be continuous and synthesizable at an arbitrary resolution, providing a more grounded representation for the geometry. In the future,

we could extend our registration template-based pipeline to them by designing *morphable SDF*, providing more detailed information for the optimization process, avoiding artifacts like self-intersections, and also providing registration of objects with varying topologies.

In the last years, textual descriptions of shapes gained much popularity [15, 70]. It would be interesting to involve these representations to shape matching, which would be attractive for not-expert people; for example, textual hints could be translated in shape descriptors and plugged in many of our proposed works. Also, recent word embeddings methods [90, 190, 277] would provide some super-compact encodings that could be directly exploited by our latent connections paradigm; shapes descriptions can be linked to geometric entities, learning a sort of translator from human to *geometric languages*. Furthermore, we think that after identity, shape and albedo, also the sound produced by a shape could be part of the morphable model representation [87] and so be included in the registration pipeline.

Finally, we believe that the emerging field of Reinforcement Learning would be tremendously useful for the discrete nature of 3D representatons. Some of the most recent works solve impressive and arduous problems, showing better performance than humans in almost every competitive games [170,243,261,287]. Formulating the matching as a game problem where an agent aims to achieve the highest geometric reward (e.g., bijectivity or minimal distortion) could give insights into the matching process; it could tell us the best places for landmarks in a particular domain, or select descriptors progressively to match increasing frequencies. It can be useful also for modeling, and some preliminary works already started this direction [181,227]. A *remeshing agent* may be trained to modify the representation (e.g., it might choose where to add extrinsic information), improving its coherency among different discretizations and respecting underlying geometry. A 3D model could be represented by a scuplting policy learned from data, open to a new complete set of possibilities for shape representations.

A

Summary of Notation

Here we collect some symbols that appear in the manuscript.

General	
$\mathcal{M}, \mathcal{N}$	Shapes, in general as smooth surfaces or manifold meshes
$V_{\mathcal{M}}$	Vertices set of a mesh $\mathcal{M}$
$F_{\mathcal{M}}$	Faces set of a mesh $\mathcal{M}$
$\Delta_{\mathcal{M}}$	Laplace-Beltrami Operator of mesh $\mathcal{M}$
$\mathbf{A}_{\mathcal{M}}$	Diagonal matrix of area weights on shape $\mathcal{M}$
$\mathbf{W}_{\mathcal{M}}$	Stiffness matrix of shape $\mathcal{M}$
$\Lambda_{\mathcal{M}}$	Diagonal matrix of LBO eigenvalues of shape $\mathcal{M}$
$\boldsymbol{\theta} \in \mathbb{R}^{72}$	SMPL Pose parameters
$\boldsymbol{\beta} \in \mathbb{R}^{10}$	SMPL shape parameters
$T_{\mathcal{M}\mathcal{N}}$	Point-to-point correspondence that associate for each point of $\mathcal{M}$ one point of $\mathcal{N}$
$\Phi_{\mathcal{M}}$	Eigenfunction of $\Delta_{\mathcal{M}}$
$(F(M))$	Functional space defined on $\mathcal{M}$
$T^F_{\mathcal{N}\mathcal{M}}$	Functional Map associated to $T_{\mathcal{M}\mathcal{N}}$ via pull-back
$\mathbf{C}_{\mathcal{M}\mathcal{N}}$	Functional Map in matricial form (pedix omitted if clear from the context
†	Moore Penrose pseudoinverse
$f_{\mathcal{M}}$	A function defined over a surface $\mathcal{M}$ (pedix omitted if clear from the context)
$\mathbf{F}, \mathbf{G}$	Matrices of Fourier expansion coefficients
$\mathbf{\Pi}$	Permutation matrix that econde correspondence

$\odot$	Element-wise product
Chapter 3	
$\boldsymbol{J}_{2D}$	OpenPose skeleton on the 2D image
$\boldsymbol{J}_{3D}$	OpenPose skeleton in 3D
$\boldsymbol{T}$	homogeneous transformation
D_{in}	Input depth map
$D_{\boldsymbol{\beta},\boldsymbol{\theta}}$	OpenDR synthetic depthmap
$\widetilde{D}_{in}$	Depth map part which rapresenting the human
$\hat{D}_{in}$	D_{in} without $\widetilde{D}_{in}$
$H \subset PC$	Part of the pointcloud that belongs to human body
γ	Trashold to separate foreground and background
$\pi_{\text{NN}}(V_{\mathcal{M}})$	List of the vertices $V_{\mathcal{M}}$ obtained as the ordered euclidean nearest neighbor with respect to the points in H
Chapter 7	
$\boldsymbol{\phi}_x, \boldsymbol{\phi}_y, \boldsymbol{\phi}_z$	Three coordinates based orthonormals functions
$\boldsymbol{\Phi}^{x}_{k_{\mathcal{M}}}$	First k LBO basis plus $\boldsymbol{\phi}_x$
$\boldsymbol{\Phi}^{x,y,z}_{k_{\mathcal{M}}}$	First k LBO basis plus $\boldsymbol{\phi}_x, \boldsymbol{\phi}_y, \boldsymbol{\phi}_z$

B

Left/Right Labeling Algorithm

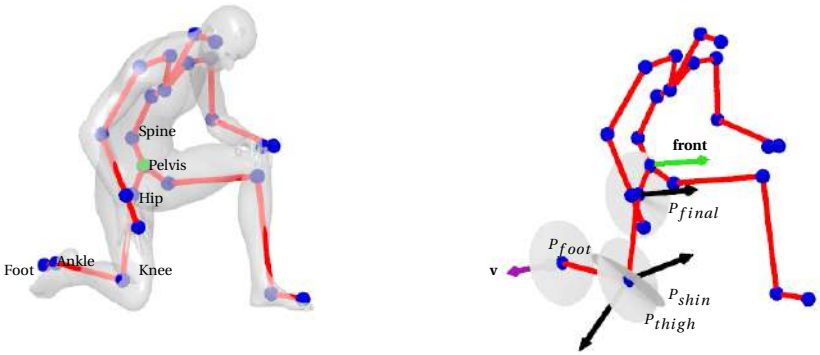


Fig. B.1: Visualization of joints and planes used in the algorithm.

The aim of this step is to detect the front of the target shape in order to discriminate the left side from the right side. Starting from the skeleton as input, we follow the steps:

1. We fix the vector **v** (in purple in Figure B.1, right), defined as the vector that connects the ankle with the foot joint (with direction from ankle to foot), which semantically identifies the front of the shape.
2. We propagate **v** along the leg of the skeleton under torque-penalizing constraints.
3. Once we reach the *pelvis* joint (in green in Figure B.1, left) we identify the front as the direction of the resulting vector (shown in green in Figure B.1, right).
4. Finally we assign the left/right labels to the landmarks as per Algorithm 1.

In the algorithm we adopt the following notation:

– $P = plane(p1,\ p2,\ p3)$ is the plane that contains non-collinear points $p1,\ p2,\ p3$.

- $P = plane(p1,\ p2, \perp Q)$ contains points $p1,\ p2$, and is orthogonal to plane Q.
- $P = plane(p1, \perp \mathbf{u})$ contains $p1$ and is orthogonal to vector $\mathbf{u}$.
- By saying that we "transfer" the normal from one plane to another we mean that we fix the sign of the normal of the second plane such that it is coherent with the normal of the first plane.

ALGORITHM 1: Left/Right labeling

1: **Input:** $S \in Mat_{\mathbb{R}}(24,3)$, *i.e. the 3D coordinates of 24 labeled skeleton joints on* $\mathcal{N}$, $\{pelvis,\ spine,\ foot,\ ankle,\ knee,\ hip\} \in S$.

2: $\mathbf{v}$ = vector($foot - ankle$).

3: P_{foot} = plane($ankle, \perp \mathbf{v}$).

4: $P_{\perp shin}$ = plane($foot, ankle, knee$).

5: P_{shin} = plane($knee, ankle, \perp P_{\perp shin}$).

6: Transport vector $\mathbf{v}$ from the $ankle$ to the $knee$ as the vector applied on the $knee$ joint that is orthogonal to the plane P_{shin}.

7: P_{thigh} = plane($hip, knee, \perp P_{\perp shin}$).

8: Transfer the normal from P_{shin} to P_{thigh}.

9: $P_{\perp thigh}$ = plane($hip, knee, \perp P_{thigh}$).

10: Transport vector $\mathbf{v}$ from the $knee$ to the hip as the vector applied on the hip joint that is orthogonal to the plane P_{thigh}.

11: P_{final} = plane($pelvis$,$hip, \perp P_{\perp thigh}$).

12: Transfer the normal from P_{thigh} to P_{final}.

13: **front** = normal(P_{final}) applied at the $pelvis$ joint.

14: **top** = vector($spine - pelvis$).

15: The **right** versor is obtained as the cross product **top** × **front**.

16: **if** the label of the right hip joint is not consistent with the label of the right hip joint on the template **then** we switch the left/right labels on the landmarks.

17: **else** we leave the left/right labels on the landmarks as they are.

C

Adjoint operator definition and properties

In this section, we provide a concise description of the adjoint operator and its relation to the transfer of Dirac delta functions and functional maps. Note that the adjoint operator of functional maps has been considered, e.g., in [146] although its role in delta function transfer was not explicitly addressed in that work.

Formal definition of the Adjoint operator. Suppose we have a pointwise map $T_{\mathscr{X}\mathscr{Y}} : \mathscr{X} \to \mathscr{Y}$ between two smooth surfaces $\mathscr{X}, \mathscr{Y}$. Then we will denote $T^{\mathscr{F}}_{\mathscr{Y}\mathscr{X}}$ the functional correspondence defined by the pull-back: $T^{\mathscr{F}}_{\mathscr{Y}\mathscr{X}} : f \to f \circ T_{\mathscr{X}\mathscr{Y}}$, where $f : \mathscr{Y} \to \mathbb{R}$ and $f \circ T_{\mathscr{X}\mathscr{Y}} : \mathscr{X} \to \mathbb{R}$ such that $f \circ T_{\mathscr{X}\mathscr{Y}}(x) = f(T_{\mathscr{X}\mathscr{Y}}(x))$ for any $x \in \mathscr{X}$.

The *adjoint functional map operator* $A_{\mathscr{X}\mathscr{Y}}$ is defined implicitly through the following equation:

$$< A_{\mathscr{X}\mathscr{Y}} g, f >_{\mathscr{Y}} = < g, T^{\mathscr{F}}_{\mathscr{Y}\mathscr{X}} f >_{\mathscr{X}} \quad \forall \ f : \mathscr{Y} \to \mathbb{R},\ g : \mathscr{X} \to \mathbb{R}. \tag{C.1}$$

Here we denote with $<,>_{\mathscr{X}}$ and $<,>_{\mathscr{Y}}$ the L^2 inner product for functions respectively on shape $\mathscr{X}$ and $\mathscr{Y}$. The adjoint always exists and is unique by the Riesz representation theorem (see also Theorem 3.1 in [146]).

Adjoint operator and delta functions. As mentioned in the main manuscript, the adjoint can be used to map *distributions* (or generalized functions), which is particularly important for mapping points represented as Dirac delta functions.

Recall that $\forall\, y \in \mathscr{Y}$, a Dirac delta function δ_y is a distribution such that, by definition, for any function f we have $< \delta_y, f >_{\mathscr{Y}} = f(y)$.

Theorem C.1. *If $A_{\mathscr{X}\mathscr{Y}}$ is the adjoint operator associated with a point-to-point mapping $T_{\mathscr{X}\mathscr{Y}}$ as in Eq.* (C.1)*, then $A_{\mathscr{X}\mathscr{Y}}\delta_x = \delta_{T_{\mathscr{X}\mathscr{Y}}(x)}$.*

Proof. Using Eq. (C.1) we get:

$$< A_{\mathcal{X}\mathcal{Y}}\delta_x, f >_{\mathcal{Y}} =< \delta_x, T^{\mathcal{F}}_{\mathcal{Y}\mathcal{X}} f >_{\mathcal{X}} =< \delta_x, f \circ T_{\mathcal{X}\mathcal{Y}} >_{\mathcal{X}} \tag{C.2}$$

$$= f(T_{\mathcal{X}\mathcal{Y}}(x)). \tag{C.3}$$

Therefore, $A_{\mathcal{X}\mathcal{Y}}\delta_x$ equals some distribution d such that $< d, f >_{\mathcal{Y}} = f(T_{\mathcal{X}\mathcal{Y}}(x))$ for any function $f : \mathcal{Y} \to \mathbb{R}$. By uniqueness of distributions this means that: $A_{\mathcal{X}\mathcal{Y}}\delta_x = \delta_{T_{\mathcal{X}\mathcal{Y}}(x)}$.

In other words, the previous derivation proves that, unlike a functional map, *the functional map adjoint always maps delta functions to delta functions.*

Relation between the functional maps and the adjoint operator in the discrete setting. Here we assume that the two shapes are represented in the discrete setting, with two embeddings $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$, and a pointwise map $\Pi_{\mathcal{X}\mathcal{Y}}$. Our goal is to establish the relationship between the functional map matrix and the linear operator, which aligns the two embeddings.

Given two embeddings $\Phi_{\mathcal{X}}, \Phi_{\mathcal{Y}}$ and a pointwise map $\Pi_{\mathcal{X}\mathcal{Y}}$ we would like to find a linear transformation $A_{\mathcal{X}\mathcal{Y}}$ such that:

$$A_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{X}}^T = \left(\Pi_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{Y}}\right)^T, \text{ or equivalently} \tag{C.4}$$

$$\Phi_{\mathcal{X}} A_{\mathcal{X}\mathcal{Y}}^T = \Pi_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{Y}} \tag{C.5}$$

Formulating this as a least squares problem we get:

$$\min_A \|\Phi_{\mathcal{X}} A_{\mathcal{X}\mathcal{Y}}^T - \Pi_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{Y}}\|_2, \tag{C.6}$$

from which the solution is given by:

$$A = \left(\Phi_{\mathcal{X}}^{\dagger}\Pi_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{Y}}\right)^T \tag{C.7}$$

Recall that a functional map induced by $\Pi_{\mathcal{X}\mathcal{Y}}$ is defined as $C_{\mathcal{Y}\mathcal{X}} = \Phi_{\mathcal{X}}^{\dagger}\Pi_{\mathcal{X}\mathcal{Y}}\Phi_{\mathcal{Y}}$. Therefore, we can write: $A_{\mathcal{X}\mathcal{Y}} = C_{\mathcal{Y}\mathcal{X}}^T$. In other words, in the discrete setting the adjoint is nothing but the transpose of the functional map in the opposite direction.

Probe function constraints. Below we derive the relation between the probe function constraints for functional maps and those for the adjoint operator used in our approach, as described in Section 8.3. Here we derive the formula used in the main manuscript directly below Equation (8.3).

In the main manuscript (Equation (2.23) of the main manuscript) we wrote the basic optimization problem for estimating functional maps:

$$C_{\mathcal{X}\mathcal{Y}} = \arg\min_{x \quad C\in\mathbb{R}^{k\times k}} \|C\Phi_{\mathcal{X}}^{\dagger}G_{\mathcal{X}} - \Phi_{\mathcal{Y}}^{\dagger}G_{\mathcal{Y}}\|_2. \tag{C.8}$$

Inverting the role of $\mathscr{X}$ and $\mathscr{Y}$:

$$C_{\mathscr{Y}\mathscr{X}} = \arg\min_{x} \min_{C \in \mathbb{R}^{k \times k}} \|C\Phi_{\mathscr{Y}}^{\dagger} G_{\mathscr{Y}} - \Phi_{\mathscr{X}}^{\dagger} G_{\mathscr{X}}\|_2. \tag{C.9}$$

This implies that the optimal $C_{\mathscr{Y}\mathscr{X}}$ can be found as the solution of $C_{\mathscr{Y}\mathscr{X}}\Phi_{\mathscr{Y}}^{\dagger} G_{\mathscr{Y}} = \Phi_{\mathscr{X}}^{\dagger} G_{\mathscr{X}}$. This is equivalent to $(\Phi_{\mathscr{Y}}^{\dagger} G_{\mathscr{Y}})^T C_{\mathscr{Y}\mathscr{X}}^T = (\Phi_{\mathscr{X}}^{\dagger} G_{\mathscr{X}})^T$ that can be solved as a least squares problem:

$$C_{\mathscr{Y}\mathscr{X}}^T = \left((\Phi_{\mathscr{Y}}^{\dagger} G_{\mathscr{Y}})^T\right)^{\dagger} (\Phi_{\mathscr{X}}^{\dagger} G_{\mathscr{X}})^T. \tag{C.10}$$

From the equation $A_{\mathscr{X}\mathscr{Y}} = C_{\mathscr{Y}\mathscr{X}}^T$ we can conclude that:

$$A_{\mathscr{X}\mathscr{Y}} = \left((\Phi_{\mathscr{Y}}^{\dagger} G_{\mathscr{Y}})^T\right)^{\dagger} (\Phi_{\mathscr{X}}^{\dagger} G_{\mathscr{X}})^T. \tag{C.11}$$

This is precisely the equation used in the main manuscript directly below Equation (8.3).

This provides an explicit connection between the functional map and the linear transformation that we are optimizing for.

To summarize, one advantage of the adjoint is that it can be used to map *distributions* and not just functions. In particular, unlike a functional map, the functional map adjoint always maps delta functions to delta functions. At the same time, similarly to functional maps, it also allows estimation via probe functions and a solution of a linear system. For this reason, despite the strong relation with functional maps, the adjoint is better suited for estimating the correspondence.

References

1. Badking website. http://BadKing.com.au. [Online; accessed: May 18, 2021]. 62, 77
2. CGtrader website. https://www.cgtrader.com/. [Online; accessed: May 18, 2021]. 103
3. FARM code. http://profs.scienze.univr.it/~marin/farm/. [Online; accessed: May 18, 2021]. 43
4. Free3D website. https://free3d.com/. [Online; accessed: May 18, 2021]. 103
5. High resolution augmentation code. https://github.com/riccardomarin/FARM-ZOSR. [Online; accessed: May 18, 2021]. 62
6. Instant recovery code. https://github.com/riccardomarin/InstantRecoveryFromSpectrum. [Online; accessed: May 18, 2021]. 129
7. Intrinsic/extrinsic code. https://github.com/PietroMsn/CMH. [Online; accessed: May 18, 2021]. 93
8. Linearly-invariant embedding code. https://github.com/riccardomarin/Diff-FMaps. [Online; accessed: May 18, 2021]. 116
9. Matching humans with different connectivity code. http://profs.scienze.univr.it/~marin/shrec19. [Online; accessed: May 18, 2021]. 76
10. Mixamo. https://www.mixamo.com. [Online; accessed: May 18, 2021]. 57
11. Scan the world project. https://www.myminifactory.com/scantheworld. [Online; accessed: May 18, 2021]. 124
12. David Aasen, Tejal Bhamre, and Achim Kempf. Shape from sound: toward new tools for quantum gravity. *Physical review letters*, 110(12):121301, 2013. 130
13. François Charpillet Abdallah Dib. Pose estimation for a partially observable human body from rgb-d cameras. 2015. 30, 36, 39
14. Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 18, 127, 131
15. Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8938–8947, 2019. 144
16. Yonathan Aflalo, Haim Brezis, and Ron Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM Journal on Imaging Sciences*, 8(2):1141–1160, 2015. 21
17. Yonathan Aflalo and Ron Kimmel. Regularized principal component analysis. *Chinese Annals of Mathematics, Series B*, 38(1):1–12, Jan 2017. 93
18. Noam Aigerman and Yaron Lipman. Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics*, 35(6):217:1–217:14, November 2016. 53, 54
19. Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3D human pose reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) 2015*, 2015. 30
20. Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: Reconstruction and parameterization from range scans. *ACM Trans. Graph.*, 22(3):587–594, July 2003. 16, 44, 45

21. Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Lévy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM Trans. Graph.*, 22(3):485–493, July 2003. 94
22. Pierre Alliez, Giuliana Ucelli, Craig Gotsman, and Marco Attene. *Recent Advances in Remeshing of Surfaces*, pages 53–82. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. 93
23. Mathieu Andreux, Emanuele Rodolà, Mathieu Aubry, and Daniel Cremers. Anisotropic Laplace-Beltrami operators for shape analysis. In *Computer Vision - ECCV 2014 Workshops*, pages 299–312, Cham, 2015. Springer International Publishing. 80
24. Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics*, 24(3):408–416, July 2005. 16, 44, 45, 69, 77
25. Dragomir Anguelov, Praveen Srinivasan, Hoi-Cheung Pang, Daphne Koller, Sebastian Thrun, and James Davis. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *Proceedings of the 17th International Conference on Neural Information Processing Systems*, NIPS'04, pages 33–40, Cambridge, MA, USA, 2004. MIT Press. 44
26. Marco Attene and Bianca Falcidieno. Remesh: An interactive environment to edit and repair triangle meshes. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 41–41, Washington, DC, USA, June 2006. IEEE Computer Society. 52
27. Matan Atzmon and Yaron Lipman. Sal++: Sign agnostic learning with derivatives. *arXiv preprint arXiv:2006.05400*, 2020. 11
28. Tristan Aumentado-Armstrong, Stavros Tsogkas, Allan Jepson, and Sven Dickinson. Geometric disentanglement for generative latent shape models. In *International Conference on Computer Vision (ICCV)*, 2019. 131
29. Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Maks Ovsjanikov. An operator approach to tangent vector field processing. In *Computer Graphics Forum*, volume 32, pages 73–82, 2013. 93
30. Omri Azencot, Étienne Corman, Mirela Ben-Chen, and Maks Ovsjanikov. Consistent functional cross field design for mesh quadrangulation. *ACM Trans. Graph.*, 36(4):92:1–92:13, July 2017. 93
31. Andreas Baak, Meinard Müller, Gaurav Bharaj, Hans-Peter Seidel, and Christian Theobalt. A data-driven approach for real-time full body pose reconstruction from a depth camera. In *IEEE 13th International Conference on Computer Vision (ICCV), (IEEE 2011)*, pages 1092–1099, 11 2011. 31
32. Shigetoshi Bando and Hajime Urakawa. Generic properties of the eigenvalue of the laplacian for compact riemannian manifolds. *Tohoku Mathematical Journal, Second Series*, 35(2):155–172, 1983. 139
33. Ilya Baran and Jovan Popović. Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics*, 26(3), July 2007. 45
34. Manuel Bastioni, Simone Re, and Shakti Misra. Ideas and methods for modeling 3D human figures: The principal algorithms used by makehuman and their implementation in a new approach to parametric modeling. In *Proceedings of the 1st Bangalore Annual Compute Conference*, COMPUTE '08, pages 10:1–10:6, New York, NY, USA, 2008. ACM. 103
35. Mikhail Belkin, Jian Sun, and Yusu Wang. Constructing Laplace operator from point clouds in rd. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 1031–1040. Society for Industrial and Applied Mathematics, 2009. 24, 116, 141
36. Matthew Berger, Andrea Tagliasacchi, Lee M Seversky, Pierre Alliez, Gael Guennebaud, Joshua A Levine, Andrei Sharf, and Claudio T Silva. A survey of surface reconstruction from point clouds. In *Computer Graphics Forum*, volume 36, pages 301–329. Wiley Online Library, 2017. 12
37. Paul J Besl and Neil D McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 141
38. Gaurav Bharaj, David IW Levin, James Tompkin, Yun Fei, Hanspeter Pfister, Wojciech Matusik, and Changxi Zheng. Computational design of metallophone contact sounds. *ACM Transactions on Graphics (TOG)*, 34(6):223, 2015. 131
39. Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3D human mesh registration. *Advances in Neural Information Processing Systems*, 33, 2020. 11, 143

40. Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 16
41. Federica Bogo, Michael J. Black, Matthew Loper, and Javier Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2300–2308, 2015. 29
42. Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016. 19, 29, 30
43. Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 3794–3801, Washington, DC, USA, 2014. IEEE. 45, 52, 53, 55, 68, 69, 75, 77, 103, 122
44. Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J Black. Dynamic FAUST: Registering human bodies in motion. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '17, pages 5573–5582, Washington, DC, USA, 2017. IEEE. 29, 56, 65
45. David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. Quad-mesh generation and processing: A survey. *Computer Graphics Forum*, 32(6):51–76, 2013. 94
46. David Bommes, Henrik Zimmer, and Leif Kobbelt. Mixed-integer quadrangulation. *ACM Trans. Graph.*, 28(3):77:1–77:10, July 2009. 93
47. Davide Boscaini, Davide Eynard, Drosos Kourounis, and Michael M Bronstein. Shape-from-operator: recovering shapes from intrinsic operators. *Computer Graphics Forum*, 34(2):265–274, 2015. 130
48. Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441. Wiley Online Library, 2016. 116, 132
49. Mario Botsch and Leif Kobbelt. A robust procedure to eliminate degenerate faces from triangle meshes. In *VMV*, pages 283–290, 2001. 101
50. Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon Mesh Processing*. AK Peters / CRC Press, 2010. 9, 12
51. Sofien Bouaziz and Mark Pauly. Dynamic 2D/3D registration for the kinect. In *ACM SIGGRAPH 2013 Courses*, SIGGRAPH '13, New York, NY, USA, 2013. Association for Computing Machinery. 58, 59
52. Nikolaos V Boulgouris and Zhiwei X Chi. Human gait recognition based on matching of body components. *Pattern Recognition*, 40:1763–1770, 2007. 27
53. Edmond Boyer, Alexander M Bronstein, Michael M Bronstein, Benjamin Bustos, Tal Darom, Radu Horaud, Ingrid Hotz, Yosi Keller, Johannes Keustermans, Artiom Kovnatsky, et al. Shrec 2011: Robust feature detection and description benchmark. In *Proceedings of the 4th Eurographics Conference on 3D Object Retrieval*, 3DOR '11, pages 71–78, Aire-la-Ville, Switzerland, Switzerland, 2011. Eurographics Association. 52
54. Alexander M Bronstein, Michael M Bronstein, Leonidas J Guibas, and Maks Ovsjanikov. Shape google: Geometric words and expressions for invariant shape retrieval. *ACM Transactions on Graphics (TOG)*, 30(1):1, 2011. 130
55. Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer Science & Business Media, 2008. 9, 13, 52, 68, 75, 77, 98, 103
56. Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 18, 20, 127
57. Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013. 18
58. Marcel Campen. Partitioning surfaces into quadrilateral patches: A survey. *Computer Graphics Forum*, 36(8):567–588, 2017. 94
59. Marcel Campen, David Bommes, and Leif Kobbelt. Dual loops meshing: Quality quad layouts on manifolds. *ACM Trans. Graph.*, 31(4):110:1–110:11, July 2012. 94
60. Massimo Camplani, Lucia Maddalena, Gabriel Moyá Alcover, Alfredo Petrosino, and Luis Salgado. A benchmarking framework for background subtraction in rgbd videos. In *International Conference on Image Analysis and Processing*, pages 219–229. Springer, 2017. 36, 38

61. Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3D facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):413–425, 2013. 16
62. Wenming Cao, Zhiyue Yan, Zhiquan He, and Zhihai He. A comprehensive survey on geometric deep learning. *IEEE Access*, 8:35929–35949, 2020. 18
63. Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2D pose estimation using part affinity fields. In *CVPR*, 2017. 31, 32
64. Sorana Capalnean, Florin Oniga, and Radu Danescu. Obstacle detection using a voxel octree representation. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 3–9. IEEE, 2019. 10
65. Thomas J Cashman and Andrew W Fitzgibbon. What shape are dolphins? building 3D morphable models from 2D images. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):232–244, 2012. 17
66. Cristian Sminchisescu Catalin Ionescu, Fuxin Li. Latent structured models for human pose estimation. In *International Conference on Computer Vision*, 2011. 29
67. Ed Catmull. *Creativity, Inc.: Overcoming the Unseen Forces That Stand in the Way of True*. 2014. 2
68. Isaac Chao, Ulrich Pinkall, Patrick Sanan, and Peter Schröder. A simple geometric model for elastic deformations. In *ACM transactions on graphics (TOG)*, volume 29, page 38. ACM, 2010. 99
69. Isaac Chavel. *Eigenvalues in Riemannian Geometry*. Academic Press, 1984. 129
70. Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2Shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018. 144
71. Qifeng Chen and Vladlen Koltun. Robust nonrigid registration by convex optimization. In *International Conference on Computer Vision (ICCV)*, pages 2039–2047, Washington, DC, USA, 2015. IEEE. 44, 53, 54, 55
72. Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3):73:1–73:12, aug 2009. 52, 75, 78
73. Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 11, 131
74. Julian Chibane, Aymen Mir, and Gerard Pons-Moll. Neural unsigned distance fields for implicit function learning. In *Neural Information Processing Systems (NeurIPS)*. , December 2020. 11
75. Yoni Choukroun, A. Shtern, A. Bronstein, and R Kimmel. Hamiltonian operator for spectral shape analysis. *arXiv:1611.01990*, 2016. 93
76. Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016. 24
77. Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A unified approach for single and multi-view 3D object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 18
78. Moody Chu, Gene Golub, and Gene H Golub. *Inverse eigenvalue problems: theory, algorithms, and applications*, volume 13. Oxford University Press, 2005. 130
79. Philippe G Ciarlet. *The finite element method for elliptic problems*, volume 40. Siam, 2002. 24, 132
80. Ulrich Clarenz, Martin Rumpf, and Alexandru Telea. Finite elements on point based surfaces. In *Proceedings of the First Eurographics conference on Point-Based Graphics*, pages 201–211. Eurographics Association, 2004. 123, 132, 141
81. Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. In *European conference on computer vision*, pages 484–498. Springer, 1998. 16
82. Timothy F Cootes, Christopher J Taylor, David H Cooper, and Jim Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. 16
83. Étienne Corman, Maks Ovsjanikov, and Antonin Chambolle. Supervised descriptor learning for non-rigid shape matching. In *European Conference on Computer Vision (ECCV) Workshops*, pages 283–298, New York, NY, 2014. Springer. 24
84. Étienne Corman, Justin Solomon, Mirela Ben-Chen, Leonidas Guibas, and Maks Ovsjanikov. Functional Characterization of Intrinsic and Extrinsic Geometry. *ACM Transactions on Graphics*, 17, 2017. 130

85. Luca Cosmo, Mikhail Panine, Arianna Rampini, Maks Ovsjanikov, Michael M Bronstein, and Emanuele Rodolà. Isospectralization, or how to hear shape, style, and correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7529–7538, 2019. 15, 128, 129, 130, 131, 132, 133, 136, 138, 139, 140
86. Luca Cosmo, Emanuele Rodolà, Jonathan Masci, Andrea Torsello, and Michael M Bronstein. Matching deformable objects in clutter. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 1–10, Washington, DC, USA, 2016. IEEE. 24, 60
87. Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael Black. Capture, learning, and synthesis of 3D speaking styles. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 10101–10111, 2019. 144
88. Hang Dai, Nick Pears, and William Smith. A data-augmented 3D morphable model of the ear. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 404–408, 2018. 17
89. Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Learning elementary structures for 3D shape generation and matching. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7435–7445. Curran Associates, Inc., 2019. 24
90. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 144
91. Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 24, 116, 123, 124
92. Mehta Dushyant, Sridhar Srinath, Sotnychenko Oleksandr, Rhodin Helge, Shafiei Mohammad, Seidel Hans-Peter, Xu Weipeng, Casas Dan, and Theobalt Christian. VNect: Real-time 3D human pose estimation with a single RGB camera. *ACM Transactions on Graphics*, 36(4), 2017. 30
93. Roberto M Dyke, Yu-Kun Lai, Paul L Rosin, and Gary KL Tam. Non-rigid registration under anisotropic deformations. *Computer Aided Geometric Design*, 71:142–156, 2019. 58, 59
94. Roberto M Dyke, Yu-Kun Lai, Paul L. Rosin, Stefano Zappalà, Seana Dykes, Daoliang Guo, Kun Li, Riccardo Marin, Simone Melzi, and Jingyu Yang. Shrec'20: Shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28 – 43, 2020. 111
95. Roberto M Dyke, Feng Zhou, Yu-Kun Lai, Paul L. Rosin, Daoliang Guo, Kun Li, Riccardo Marin, and Jingyu Yang. SHREC 2020 track: Non-rigid shape correspondence of physically-based deformations. In Tobias Schreck, Theoharis Theoharis, Ioannis Pratikakis, Michela Spagnuolo, and Remco C Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2020. 56, 58, 59
96. Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models,Äîpast, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020. 16
97. Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12265–12274, 2020. 143
98. Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *Advances in Neural Information Processing Systems*, 33, 2020. 143
99. Davide Eynard, Emanuele Rodola, Klaus Glashoff, and Michael M Bronstein. Coupled functional maps. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 399–407, 2016. 23
100. Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. *Computer Graphics Forum*, 36(5):165–174, August 2017. 23, 24, 57, 59, 107
101. Danielle Ezuz, Behrend Heeren, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. Elastic correspondence between triangle meshes. *Computer Graphics Forum*, 38(2):121–134, 2019. 112
102. Danielle Ezuz, Justin Solomon, and Mirela Ben-Chen. Reversible harmonic maps between discrete surfaces. *ACM Transactions on Graphics*, 38(2):15:1–15:12, March 2019. 68, 112
103. Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3D object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017. 18

104. Jingfan Fan, Jian Yang, Yitian Zhao, Danni Ai, Yonghuai Liu, Ge Wang, and Yongtian Wang. Convex hull aided registration method (CHARM). *IEEE Transactions on Visualization and Computer Graphics*, 23(9):2042–2055, Sept 2017. 55
105. Yi Fang, Jin Xie, Guoxian Dai, Meng Wang, Fan Zhu, Tiantian Xu, and Edward Wong. 3D deep shape descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2319–2328, 2015. 18
106. Holly Rushmeier Cláudio Silva Fausto Bernardini, Joshua Mittleman and Gabriel Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE transactions on visualization and computer graphics*, 5(4):349–359, 1999. 93
107. Andrew Feng, Dan Casas, and Ari Shapiro. Avatar reshaping and automatic rigging using a deformable model. In *Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games*, MIG '15, pages 57–64, New York, NY, USA, 2015. ACM. 17, 45
108. Andrew Feng, Ari Shapiro, Wang Ruizhe, Mark Bolas, Gerard Medioni, and Evan Suma. Rapid avatar capture and simulation using commodity depth sensors. In *ACM SIGGRAPH 2014*, SIGGRAPH '14, pages 16:1–16:1, New York, NY, USA, 2014. ACM. 45
109. Matthias Fey, Jan E Lenssen, Christopher Morris, Jonathan Masci, and Nils M Kriege. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*, 2020. 115
110. Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973. 14
111. Michel Foucault. *Ceci n'est pas une pipe: deux lettres et quatre dessins de René Magritte*. Fata Morgana, 1973. 3
112. Kunihiko Fukushima and Sei Miyake. Neocognitron: a self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982. 17
113. Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020. 115
114. Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. SDM-NET: Deep generative network for structured deformable mesh. *arXiv preprint arXiv:1908.04520*, 2019. 127, 131
115. Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '97, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 52, 80
116. K. Gebal, J. A. Bærentzen, H. Anæs, and R. Larsen. Shape analysis using the auto diffusion function. *Computer Graphics Forum*, 28(5):1405–1413, 2009. 24
117. Anne Gehre, Michael Bronstein, Leif Kobbelt, and Justin Solomon. Interactive curve constrained functional maps. In *Computer Graphics Forum*, volume 37, pages 1–12. Wiley Online Library, 2018. 23
118. Golnaz Ghiasi, Yi Yang, Deva Ramanan, and Charless C Fowlkes. Parsing occluded people. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2401–2408, 2014. 30
119. Andrea Giachetti, Emanuele Mazzi, Francesco Piscitelli, et al. Shrec,Äô14 track: automatic location of landmarks used in manual anthropometry. In *Eurographics Workshop on 3D Object Retrieval*, pages 93–100, 2014. 44
120. Dvir Ginzburg and Dan Raviv. Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. *arXiv preprint arXiv:1912.01249*, 2019. 24
121. Michael Gleicher. Retargetting motion to new characters. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 33–42, New York, NY, 1998. ACM. 42
122. Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3D point cloud matching with smoothed densities. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 115
123. Michael A Goodrich and Alan C Schultz. Human-robot interaction: A survey. *Found. Trends Hum.-Comput. Interact.*, 1(3):203–275, 2007. 27
124. Carolyn Gordon, David L Webb, and Scott Wolpert. One cannot hear the shape of a drum. *Bulletin of the American Mathematical Society*, 27(1):134–138, 1992. 15, 129, 130
125. Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of Machine Learning and Systems 2020*, pages 3569–3579. 2020. 11

126. Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 131
127. Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3D-CODED: 3D correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 230–246, 2018. 24, 116, 123, 124
128. Kan Guo, Dongqing Zou, and Xiaowu Chen. 3D mesh labeling via deep convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 35(1):1–12, 2015. 18
129. Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3D point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 18
130. David Ha, Andrew Dai, and Quoc V Le. Hypernetworks. *arXiv preprint arXiv:1609.09106*, 2016. 143
131. Oshri Halimi, Or Litany, Emanuele Rodola, Alex M. Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 24, 116
132. Mohamed Hassan, Vasileios Choutas, Dimitrios Tzionas, and Michael J Black. Resolving 3D human pose ambiguities with 3D scene constraints. In *Proceedings International Conference on Computer Vision*, pages 2282–2292. IEEE, October 2019. 30
133. Donald Hearn and M Pauline Baker. *Computer graphics with OpenGL*. Upper Saddle River, NJ: Pearson Prentice Hall„ 2004. 9
134. Nikolas Hesse, Sergi Pujades, Javier Romero, Michael J Black, Christoph Bodensteiner, Michael Arens, Ulrich G. Hofmann, Uta Tacke, Mijna Hadders-Algra, Raphael Weinberger, Wolfgang Muller-Felber, and A. Sebastian Schroeder. Learning an infant body model from RGB-D data for accurate full body motion analysis. In *Int. Conf. on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, September 2018. 17, 71
135. David A Hirshberg, Matthew Loper, Eric Rachlin, and Michael J Black. Coregistration: Simultaneous alignment and modeling of articulated 3D shape. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 242–255, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 17, 44
136. David A Hirshberg, Matthew Loper, Eric Rachlin, Aggeliki Tsoli, Alexander Weiss, B Corner, and Michael J Black. Evaluating the automated alignment of 3D human body scans. In *2nd International Conference on 3D Body Scanning Technologies*, pages 76–86, Lugano, Switzerland, October 2011. Hometrica Consulting. 61
137. David A Hirshberg, Matthew Loper, Eric Rachlin, Aggeliki Tsoli, Alexander Weiss, B Corner, and MJ Black. Evaluating the automated alignment of 3D human body scans. In *Proc 2nd Int Conf 3D Body Scanning Technol*, volume 10, pages 76–86, Lugano, Switzerland, 2011. Hometrica Consulting. 44
138. Dirk L Hoffmann, Alistair WG Pike, Marcos García-Diez, Paul B Pettitt, and Jo ao Zilhão. Methods for U-series dating of CaCO3 crusts associated with Palaeolithic cave art and application to Iberian sites. *Quaternary Geochronology*, 36:104 – 119, 2016. 1, 2
139. Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. In *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '93, pages 19–26, New York, NY, USA, 1993. ACM. 93
140. Kai Hormann, Bruno Lévy, and Alla Sheffer. Mesh parameterization: Theory and practice. In *ACM SIGGRAPH 2007 Courses*, SIGGRAPH '07, New York, NY, USA, 2007. ACM. 94
141. Erik Van Horn. *3D Character Development Workshop: Rigging Fundamentals for Artists and Animators*. Mercury Learning and Information, 2018. 75
142. Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 12
143. Haibin Huang, Evangelos Kalogerakis, Siddhartha Chaudhuri, Duygu Ceylan, Vladimir G. Kim, and Ersin Yumer. Learning local shape descriptors from part correspondences with multiview convolutional networks. *ACM Trans. Graph.*, 37(1), November 2017. 18
144. Jia-Binand Yang Huang and Ming-Hsuan. Estimating human pose from occluded images. In *Asian Conference on Computer Vision – ACCV*, pages 48–60, 2010. 30

145. Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018. 141
146. Ruqi Huang and Maks Ovsjanikov. Adjoint map representation for shape analysis and matching. In *Computer Graphics Forum*, volume 36, pages 151–163. Wiley Online Library, 2017. 118, 149
147. Ruqi Huang, Marie-Julie Rakotosaona, Panos Achlioptas, Leonidas Guibas, and Maks Ovsjanikov. OperatorNet: Recovering 3D shapes from difference operators. In *ICCV*, 2019. 131
148. Takeo Igarashi, Tomer Moscovich, and John F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005. 44, 52, 65
149. Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014. 29, 30
150. Umar Iqbal, Andreas Doering, Hashim Yasin, Björn Krüger, Andreas Weber, and Juergen Gall. A dual-source approach for 3D human pose estimation from single images. *Computer Vision and Image Understanding*, 2018. 31
151. Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011. 11
152. Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. Moviereshape: Tracking and reshaping of humans in videos. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2010)*, 29(5), 2010. 17
153. Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. Instant field-aligned meshes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA)*, 34(6), November 2015. 94
154. Haiyong Jiang, Jianfei Cai, and Jianmin Zheng. Skeleton-aware 3D human shape reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5431–5441, 2019. 19
155. Young-Hoon Jin and Won-Hyung Lee. Fast cylinder shape matching using random sample consensus in large scale point cloud. *Applied Sciences*, 9(5):974, 2019. 115
156. Michael J Jones and Tomaso Poggio. Multidimensional morphable models: A framework for representing and matching object classes. *International Journal of Computer Vision*, 29(2):107–131, 1998. 16
157. Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8320–8329, 2018. 17
158. Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3D deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8320–8329, 2018. 29
159. Klaus Junker. *Interpreting the images of Greek myths: an introduction*. Cambridge University Press, 2012. 1
160. Mark Kac. Can one hear the shape of a drum? *The american mathematical monthly*, 73(4P2):1–23, 1966. 130
161. Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Computer Vision and Pattern Regognition (CVPR)*, 2018. 29, 30
162. Zachi Karni and Craig Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 279–286. ACM Press/Addison-Wesley Publishing Co., 2000. 130
163. Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988. 16
164. Dawar Khan, Dong-Ming Yan, Fan Ding, Yixin Zhuang, and Xiaopeng Zhang. Surface remeshing with robust user-guided segmentation. *Computational Visual Media*, 4(2):113–122, Jun 2018. 101
165. Vladimir G Kim, Yaron Lipman, and Thomas Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011. 55, 68, 83, 108, 142
166. Yu Kong and Yun Fu. Human action recognition and prediction: A survey. *arXiv preprint arXiv:1806.11230*, 2018. 27
167. Ilya Kostrikov, Zhongshi Jiang, Daniele Panozzo, Denis Zorin, and Joan Bruna. Surface networks. In *Proc. CVPR*, 2018. 131

168. Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Computer Graphics Forum*, volume 32, pages 439–448. Wiley Online Library, 2013. 24
169. Felix Kuhnke and Jorn Ostermann. Deep head pose estimation using synthetic images and partial adversarial domain adaption for continuous label spaces. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 10, 11
170. Maxim Lapan. *Deep Reinforcement Learning Hands-On: Apply modern RL methods, with deep Q-networks, value iteration, policy gradients, TRPO, AlphaGo Zero and more*. Packt Publishing Ltd, 2018. 144
171. Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3D and 2D human representations. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017*, Piscataway, NJ, USA, July 2017. IEEE. 29
172. Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 18
173. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 18
174. Bruno Levy. Laplace-Beltrami eigenfunctions towards an algorithm that" understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, pages 13–13. IEEE, 2006. 20, 61
175. Chun-Liang Li, Tomas Simon, Jason Saragih, Barnabás Póczos, and Yaser Sheikh. Lbs autoencoder: Self-supervised fitting of articulated meshes to point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11967–11976, 2019. 19
176. Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao Zhang, and Leonidas Guibas. GRASS: generative recursive autoencoders for shape structures. *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, 36(4):52–56, 2017. 131
177. Kedan Li, Min Jin Chong, Jingen Liu, and David Forsyth. Toward accurate and realistic virtual try-on through shape matching and multiple warps. *arXiv preprint arXiv:2003.10817*, 2020. 115
178. Kun Li, Jingyu Yang, Yu-Kun Lai, and Daoliang Guo. Robust non-rigid registration with reweighted position and transformation sparsity. *IEEE transactions on visualization and computer graphics*, 25(6):2255–2269, 2018. 58, 59
179. Tianye Li, Timo Bolkart, Michael. J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 16, 18
180. Jian Liang and Hongkai Zhao. Solving partial differential equations on point clouds. *SIAM Journal on Scientific Computing*, 35(3):A1461–A1486, 2013. 116
181. Cheng Lin, Tingxiang Fan, Wenping Wang, and Matthias Nießner. Modeling 3d shapes by reinforcement learning. *arXiv preprint arXiv:2003.12397*, 2020. 144
182. Nathan Linial. Finite metric spaces–combinatorics, geometry and algorithms. *arXiv preprint math/0304466*, 2003. 13
183. Yaron Lipman, Raif M Rustamov, and Thomas A Funkhouser. Biharmonic distance. *ACM Transactions on Graphics (TOG)*, 29(3):27:1–27:11, 2010. 46, 80
184. Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 1886–1895, Washington, DC, USA, Jun 2018. IEEE. 56, 57
185. Or Litany, Alex Bronstein, Michael Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1886–1895, 2018. 127, 131
186. Or Litany, Tal Remez, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5659–5667, 2017. 24, 55, 81, 116, 120, 121
187. Or Litany, Emanuele Rodolà, Alex M Bronstein, and Michael M Bronstein. Fully spectral partial shape matching. In *Computer Graphics Forum*, volume 36, pages 247–258, Chichester, UK, may 2017. Wiley Online Library, The Eurographics Association and John Wiley & Sons Ltd. 24

188. Or Litany, Emanuele Rodolà, Alexander M Bronstein, Michael M Bronstein, and Daniel Cremers. Non-Rigid Puzzles. *Computer Graphics Forum*, 35(5):135–143, 2016. 24
189. Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020. 10
190. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 144
191. Charles Loop. Smooth subdivision surfaces based on triangles. January 1987. 65, 66
192. Matthew Loper. Chumpy autodifferentiation library. `http://chumpy.org/`, 2014. [Online; accessed: May 18, 2021]. 30, 51
193. Matthew Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *ECCV*, pages 154–169, Cham, 2014. Springer International Publishing. 31
194. Matthew Loper, Naureen Mahmood, and Michael J. Black. Mosh: Motion and shape capture from sparse markers. *ACM Trans. Graph.*, 33(6), November 2014. 69, 75, 77
195. Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015. 17, 27, 29, 31, 44, 45, 76, 77, 98, 103, 110
196. William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 10
197. Marcel Lüthi, Thomas Gerig, Christoph Jud, and Thomas Vetter. Gaussian process morphable models. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1860–1873, 2017. 60
198. Qianli Ma, Jinlong Yang, Anurag Ranjan, Sergi Pujades, Gerard Pons-Moll, Siyu Tang, and Michael J Black. Learning to dress 3D people in generative clothing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6469–6478, 2020. 17
199. Emilie Marchandise, Jean-François Remacle, and Christophe Geuzaine. Optimal parametrizations for surface remeshing. *Engineering with Computers*, 30(3):383–402, 2014. 101
200. Giorgio Marcias, Nico Pietroni, Daniele Panozzo, Enrico Puppo, and Olga Sorkine-Hornung. Animation-aware quadrangulation. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, SGP '13, pages 167–175, Aire-la-Ville, Switzerland, Switzerland, 2013. Eurographics Association. 94
201. Giorgio Marcias, Kenshi Takayama, Nico Pietroni, Daniele Panozzo, Olga Sorkine-Hornung, Enrico Puppo, and Paolo Cignoni. Data-driven interactive quadrangulation. *ACM Trans. Graph.*, 34(4):65:1–65:10, July 2015. 94
202. Riccardo Marin, Simone Melzi, Niloy J. Mitra, and Umberto Castellani. POP: Full Parametric model Estimation for Occluded People. In Silvia Biasotti, Guillaume Lavoué, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. 7, 25
203. Riccardo Marin, Simone Melzi, Emanuele Rodolà, and Umberto Castellani. FARM: Functional automatic registration method for 3D human bodies. *arXiv preprint arXiv:1807.10517*, 2018. 7, 25, 68, 81, 83, 89
204. Riccardo Marin, Simone Melzi, Emanuele Rodolà, and Umberto Castellani. High-resolution augmentation for automatic template-based matching of human models. In *2019 International Conference on 3D Vision (3DV)*, pages 230–239, 2019. 7, 25, 57
205. Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding, 2020. 7, 113
206. Riccardo Marin, Arianna Rampini, Umberto Castellani, Emanuele Rodolà, Maks Ovsjanikov, and Simone Melzi. Instant recovery of shape from spectrum via latent space connections, 2020. 7, 15, 113
207. Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCV)*, ICCV'15, pages 832–840, USA, 2015. IEEE Computer Society. 24, 120
208. Jonathan Masci, Emanuele Rodolà, Davide Boscaini, Michael M Bronstein, and Hao Li. Geometric deep learning. In *SIGGRAPH ASIA 2016 Courses*, page 1. ACM, 2016. 127
209. Ulrich Schlickewei Mathieu Aubry and Daniel 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. 24, 48, 55, 79, 85, 98, 130

210. Daniel Maturana and Sebastian Scherer. Voxnet: A 3D convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 18
211. Simone Melzi, Riccardo Marin, Pietro Musoni, Filippo Bardon, Marco Tarini, and Umberto Castellani. Intrinsic/extrinsic embedding for functional remeshing of 3D shapes. *Computers & Graphics*, 88:1 – 12, 2020. 7, 61, 73
212. Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Matching Humans with Different Connectivity. In Silvia Biasotti, Guillaume Lavoué, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. 7, 70, 73
213. Simone Melzi, Maks Ovsjanikov, Giorgio Roffo, Marco Cristani, and Umberto Castellani. Discrete time evolution process descriptor for shape analysis and matching. *ACM Transactions on Graphics (TOG)*, 37(1):4:1–4:18, January 2018. 24, 46, 80, 85
214. Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155, 2019. 24, 63, 68, 112, 116, 123, 143
215. Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael Bronstein. Shape analysis with anisotropic windowed fourier transform. In *International Conference on 3D Vision (3DV)*, 2016. 80, 85
216. Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael M Bronstein. Localized manifold harmonics for spectral shape analysis. *Computer Graphics Forum*, 37(6):20–34, 2018. 60, 93, 107
217. Simone Melzi, Riccardo Spezialetti, Federico Tombari, Michael M. Bronstein, Luigi Di Stefano, and Emanuele Rodola. GFrames: Gradient-based local reference frame for 3D shape matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4629–4638. IEEE, June 2019. 81, 85
218. Di Meng, Marilyn Keller, Edmond Boyer, Michael Black, and Sergi Pujades. Learning a statistical full spine model from partial observations. In *Shape in Medical Imaging: International Workshop, ShapeMI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4, 2020, Proceedings*, pages 122–133. Springer International Publishing, 2020. 17
219. Russell Merris. A note on laplacian graph eigenvalues. *Linear Algebra and its Applications*, 285(1):33 – 35, 1998. 14
220. Dennis Mitzel, Jasper Diesel, Aljosa Osep, Umer Rafi, and Bastian Leibe. A fixed-dimensional 3D shape representation for matching partially observed objects in street scenes. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1336–1343, 2015. 31
221. Georges MÍiès. Le voyage dans la lune, 1902. 2
222. Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. StructureNet: Hierarchical graph networks for 3D shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 127, 131
223. Omran Mohamed, Lassner Christoph, Pons-Moll Gerard, Gehler Peter V., and Schiele Bernt. Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In *International Conference on 3D Vision (3DV)*, Verona, Italy, 2018. 29
224. Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991. 14
225. Matteo Munaro, Alberto Basso, Andrea Fossati, Luc Van Gool, and Emanuele Menegatti. 3d reconstruction of freely moving persons for re-identification with a depth sensor. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4512–4519. IEEE, 2014. 36
226. Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010. 99
227. Reiichiro Nakano. Neural painters: A learned differentiable constraint for generating brushstroke paintings. *arXiv preprint arXiv:1904.08410*, 2019. 144
228. John Nash. The imbedding problem for riemannian manifolds. *Annals of mathematics*, pages 20–63, 1956. 13
229. Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 343–352, Washington, DC, USA, 2015. IEEE. 42

230. Matthias Nieser, Jonathan Palacios, Konrad Polthier, and Eugene Zhang. Hexagonal global parameterization of arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):865–878, June 2012. 94
231. Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. New York, NY: Springer, New York, NY, USA, 2nd ed. edition, 2006. 51
232. Dorian Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael M Bronstein, and Maks Ovsjanikov. Improved functional mappings via product preservation. *Computer Graphics Forum*, 37(2):179–190, 2018. 24, 93
233. Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, May 2017. 23, 24, 47, 55, 81, 98, 106, 108
234. Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012. 21, 22, 24, 47, 48, 81, 99, 116, 117, 130, 141
235. Maks Ovsjanikov, Étienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH 2017 Courses*. 2017. 21, 22, 81, 119
236. Cengiz Öztireli, Marc Alexa, and Markus Gross. Spectral sampling of manifolds. *ACM Transactions on Graphics (TOG)*, 29(6):168, 2010. 130
237. Mikhail Panine and Achim Kempf. Towards spectral geometric methods for euclidean quantum gravity. *Physical Review D*, 93(8):084033, 2016. 130
238. Daniele Panozzo, Yaron Lipman, Enrico Puppo, and Denis Zorin. Fields on symmetric surfaces. *ACM Trans. Graph.*, 31(4):111:1–111:12, July 2012. 94
239. Daniele Panozzo, Enrico Puppo, Marco Tarini, and Olga Sorkine-Hornung. Frame Fields: Anisotropic and non-orthogonal cross fields. *ACM Trans. Graph.*, 33(4):134:1–134:11, July 2014. 94
240. Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 11
241. Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 17
242. Nick Pears, Yonghuai Liu, and Peter Bunting. *3D Imaging, Analysis and Applications*. Springer, 2012. 75
243. Morten Goodwin Per-Arne Andersen and Ole-Christoffer Granmo. Deep rts: A game environment for deep reinforcement learning in real-time strategy games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, 2018. 144
244. David Pickup, Xianfang Sun, Paul L Rosin, Ralph R Martin, Z Cheng, Zhouhui Lian, Masaki Aono, A Ben Hamza, A Bronstein, M Bronstein, et al. Shape retrieval of non-rigid 3D human models. *International Journal of Computer Vision*, 120(2):169–193, Nov 2016. 52, 78
245. Nico Pietroni, Marco Tarini, and Paolo Cignoni. Almost isometric mesh parameterization through abstract domains. *IEEE Transaction on Visualization and Computer Graphics*, 16(4):621–635, July/August 2010. 93
246. Ulrich Pinkall and Konrad Polthier. Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics*, 2(1):15–36, 1993. 15, 132
247. Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3D human modeling. *Pattern Recognition*, 67(C):276–286, July 2017. 45, 53, 77
248. Joshua Podolak, Aleksey Golovinskiy, and Szymon Rusinkiewicz. Symmetry-enhanced remeshing of surfaces. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 235–242, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. 94
249. Jonathan Pokrass, Alexander M Bronstein, Michael M Bronstein, Pablo Sprechmann, and Guillermo Sapiro. Sparse modeling of intrinsic correspondences. 32(2pt4):459–468, 2013. 24
250. Gerard Pons-Moll, Javier Romero, Naureen Mahmood, and Michael J Black. Dyna: A model of dynamic human shape in motion. *ACM Transactions on Graphics, (Proc. SIGGRAPH)*, 34(4):120:1–120:14, August 2015. 17
251. Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 120

252. Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 12, 18, 122, 141
253. Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS,Äô17, page 5105,Äì5114, Red Hook, NY, USA, 2017. Curran Associates Inc. 12, 18
254. Umer Rafi, Juergen Gall, and Bastian Leibe. A semantic occlusion model for human pose estimation from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 67–74, 2015. 30
255. Arianna Rampini, Irene Tallini, Maks Ovsjanikov, Alex M Bronstein, and Emanuele Rodolà. Correspondence-free region localization for partial shape similarity via hamiltonian spectrum alignment. In *International Conference on 3D Vision (3DV)*, 2019. 15, 128, 132, 136, 138, 140
256. Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *European Conference on Computer Vision (ECCV)*, 2018. 135, 136, 141
257. Nicolas Ray, Wan Chiu Li, Bruno Lévy, Alla Sheffer, and Pierre Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25(4):1460–1485, October 2006. 93
258. Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018. 24, 58, 59, 68, 82, 112
259. Martin Reuter. Hierarchical shape segmentation and registration via topological features of laplace-Beltrami eigenfunctions. *International Journal of Computer Vision*, 89(2-3):287–308, 2010. 130, 132
260. Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. Laplace-spectra as fingerprints for shape matching. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 101–106. ACM, 2005. 130
261. Sebastian Risi and Mike Preuss. Behind deepmind,Äôs alphastar ai that reached grandmaster level in starcraft ii. *KI-Künstliche Intelligenz*, 34(1):85–86, 2020. 144
262. Kathleen M Robinette, Hans Daanen, and Eric Paquet. The caesar project: a 3-d surface anthropometry survey. In *Proc. Second International Conference on 3-D Digital Imaging and Modeling*, pages 380–386, Washington, DC, USA, oct 1999. IEEE. 17, 52, 77
263. E. Rodolà, M. Moeller, and D. Cremers. Point-wise map recovery and refinement from functional correspondence. In *Vision, Modeling and Visualization*, VMV, pages 25–32, Chichester, UK, oct 2015. The Eurographics Association. 23
264. Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum*, 36(1):222–236, 2017. 24, 48
265. Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017. 24, 57, 59, 112, 116, 124, 125
266. Emanuele Rodolà, Zorah Lähner, Alexander M Bronstein, Michael M Bronstein, and Justin Solomon. Functional maps representation on product manifolds. *Computer Graphics Forum*, 38(1):678–689, 2019. 23
267. Emanuele Rodolà, Michael Moeller, and Daniel Cremers. Regularized pointwise map recovery from functional correspondence. In *Computer Graphics Forum*, volume 36, pages 700–711, Chichester, UK, 2017. Wiley Online Library, The Eurographics Association and John Wiley & Sons Ltd. 23
268. Emanuele Rodolà, S Rota Bulò, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 4177–4184, Washington, DC, USA, jun 2014. IEEE. 52, 71
269. Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017. 17
270. German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 10
271. Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1627, 2019. 24, 116, 136

272. Raif M Rustamov. Laplace-Beltrami eigenfunctions for deformation invariant shape representation. In *Proc. SGP*, pages 225–233, Aire-la-Ville, Switzerland, 2007. Eurographics Association. 79, 85, 117
273. Raif M. Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)*, 32(4), 2013. 130, 131
274. Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020. 19
275. Y. Sahillio*ğ*ülu and Y. Yemez. Partial 3-d correspondence from shape extremities. *Computer Graphics Forum*, 33(6):63–76, 2014. 45
276. Carlos Sánchez-Belenguer, Simone Ceriani, Pierluigi Taddei, Erik Wolfart, and Vítor Sequeira. Global matching of point clouds for scan registration and loop detection. *Robotics and Autonomous Systems*, 123:103324, 2020. 115
277. Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 144
278. Nikolaos Sarafianos, Bogdan Boteanu, Bogdan Ionescu, and Ioannis A. Kakadiaris. 3D human pose estimation: A review of the literature and analysis of covariates. *Computer Vision and Image Understanding*, 152:1–20, 2016. 27, 29, 30
279. István Sárándi, Timm Linder, Kai O Arras, and Bastian Leibe. How robust is 3D human pose estimation to occlusion? In *IROS Workshop - Robotic Co-workers 4.0*, 2018. 27, 30
280. Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008. 18
281. Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 17
282. Hermann Amandus Schwarz. Über ein die flächen kleinsten flächeninhalts betreffendes problem der variationsrechnung. In *Gesammelte Mathematische Abhandlungen*, pages 223–269. Springer, 1890. 12
283. Guy L Scott. The alternative snake-and other animals. In *Alvey Vision Conference*, pages 1–8. Citeseer, 1987. 16
284. Nicholas Sharp, Yousuf Soliman, and Keenan Crane. Navigating intrinsic triangulations. *ACM Trans. Graph.*, 38(4):55:1–55:16, July 2019. 142
285. Meged Shoham, Amir Vaxman, and Mirela Ben-Chen. Hierarchical functional maps between subdivision surfaces. In *Computer Graphics Forum*, volume 38, pages 55–73. Wiley Online Library, 2019. 24
286. Markos Sigalas, Maria Pateraki, and Panos Trahanias. Full-body pose tracking - the top view reprojection approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2015. 36, 37, 39
287. David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. 144
288. Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 31
289. Ayan Sinha, Asim Unmesh, Qi-Xing Huang, and Karthik Ramani. SurfNet: Generating 3D shape surfaces using deep residual networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 791–800, 2017. 131
290. Dmitriy Smirnov, Matthew Fisher, Vladimir G. Kim, Richard Zhang, and Justin Solomon. Deep parametric shape predictions using distance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 11
291. Shuran Song and Jianxiong Xiao. Tracking revisited using rgbd camera: Unified benchmark and baselines. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013. 36, 39
292. Olga Sorkine. Differential representations for mesh processing. In *Computer Graphics Forum*, volume 25, pages 789–807. Wiley Online Library, 2006. 13, 14, 15
293. Vinkle Srivastav, Thibaut Issenhuth, Abdolrahim Kadkhodamohammadi, Michel de Mathelin, Afshin Gangi, and Nicolas Padoy. MVOR: A multi-view RGB-D operating room dataset for 2D and 3D human pose estimation. *arXiv preprint*, 2018. 28, 36
294. Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 18

295. Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 23(3):399–405, August 2004. 61
296. Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. 45, 79, 85, 130
297. Jian Sun, Maks Ovsjanikov, and Leonidas J. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. 24
298. Kenshi Takayama, Daniele Panozzo, Alexander Sorkine-Hornung, and Olga Sorkine-Hornung. Sketch-based generation and editing of quad meshes. *ACM Trans. Graph.*, 32:97:1–97:8, 2013. 93
299. Marco Tarini, Enrico Puppo, Daniele Panozzo, Nico Pietroni, and Paolo Cignoni. Simple quad domains for field aligned mesh parametrization. *ACM Trans. Graph.*, 30(6):142:1–142:12, December 2011. 94
300. Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995. 14
301. Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH, page 351,Äì358, New York, NY, USA, 1995. Association for Computing Machinery. 20
302. James Thewlis, Samuel Albanie, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of landmarks by descriptor vector exchange. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6361–6371, 2019. 24
303. Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *International Conference on Computer Vision (ICCV)*, pages 356–369, 2010. 80, 81, 85
304. Warren S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952. 13
305. H. Tung, H. Wei, E. Yumer, and K. Fragkiadaki. Self-supervised learning of motion capture. In *Neural Information Processing Systems (NIPS)*, 2017. 27, 29, 30
306. Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library, 2008. 20
307. Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. 19
308. Kiran Varanasi, Andrei Zaharescu, Edmond Boyer, and Radu Horaud. Temporal surface tracking using mesh evolution. In *10th European Conference on Computer Vision*, ECCV, pages 30–43, Berlin, Heidelberg, oct 2008. Springer, Springer Berlin Heidelberg. 42
309. Gül Varol, Duygu Ceylan, Bryan Russell, Jimei Yang, Ersin Yumer, Ivan Laptev, and Cordelia Schmid. Bodynet: Volumetric inference of 3D human body shapes. In *European Conference on Computer Vison (ECCV)*, 2018. 27, 30, 37
310. Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017. 10, 30, 31, 122
311. Amir Vaxman and Mirela Ben-Chen. Dupin meshing: A parameterization approach to planar hex-dominant meshing. Technical report, Tech. Rep. CS-2015-01, Department of Computer Science, technion-IIT, 2015. 17, 2015. 94
312. Amir Vaxman, Marcel Campen, Olga Diamanti, David Bommes, Klaus Hildebrandt, Mirela Ben-Chen, and Daniele Panozzo. Directional field synthesis, design, and processing. In *SIGGRAPH ASIA 2016 Courses*, SA '16, pages 15:1–15:30, New York, NY, USA, 2016. ACM. 94
313. Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodola, Alex Bronstein, Michael Bronstein, Ron Kimmel, et al. Efficient deformable shape correspondence via kernel matching. In *2017 International Conference on 3D Vision (3DV)*, pages 517–526. IEEE, 2017. 58, 59
314. 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 *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, volume 2017-January of *CVPR*, pages 6681–6690, Washington, DC, USA, jan 2017. IEEE. 23
315. 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 *Proc. CVPR*, pages 6681–6690, 2017. 68

316. Thomas Vetter and Tomaso Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):733–742, 1997. 16
317. Daniel Vlasic, Ilya Baran, Wojciech Matusik, and Jovan Popović. Articulated mesh animation from multi-view silhouettes. *ACM Trans. Graph.*, 27(3):97:1–97:9, August 2008. 58
318. Alex Waibel, Toshiyuki Hanazawa, Geoffrey Hinton, Kiyohiro Shikano, and Kevin J Lang. Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339, 1989. 18
319. Fu-Dong Wang, Nan Xue, Yipeng Zhang, Gui-Song Xia, and Marcello Pelillo. A functional representation for graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 115
320. Max Wardetzky, Saurabh Mathur, Felix Kälberer, and Eitan Grinspun. Discrete laplace operators: no free lunch. In *Symposium on Geometry processing*, pages 33–37. Aire-la-Ville, Switzerland, 2007. 20
321. Lingyu Wei, Qixing Huang, Duygu Ceylan, Étienne Vouga, and Hao Li. Dense human body correspondences using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1544–1553, 2016. 24, 116
322. Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional pose machines. In *CVPR*, 2016. 31
323. Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016. 18, 131
324. Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. SAGNet: Structure-aware generative network for 3D-shape modeling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2019)*, 38(4):91:1–91:14, 2019. 131
325. Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 18
326. Stefanie Wuhrer, Chang Shu, and Pengcheng Xi. Landmark-free posture invariant human shape correspondence. *The Visual Computer*, 27(9):843–852, Sep 2011. 44
327. Stefanie Wuhrer, Pengcheng Xi, and Chang Shu. Human shape correspondence with automatically predicted landmarks. *Machine Vision and Applications*, 23(4):821–830, Jul 2012. 44
328. Yun-Peng Xiao, Yu-Kun Lai, Fang-Lue Zhang, Chunpeng Li, and Lin Gao. A survey on deep geometry learning: From a representation perspective. *Computational Visual Media*, 6(2):113–133, 2020. 9
329. Hongyi Xu, Eduard Gabriel Bazavan, Andrei Zanfir, William T Freeman, Rahul Sukthankar, and Cristian Sminchisescu. GHUM & GHUML: Generative 3D human shape and articulated pose models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6184–6193, 2020. 19
330. Zongyi Xu, Qianni Zhang, and Shiyang Cheng. Multilevel active registration for kinect human body scans: from low quality to high quality. *Multimedia Systems*, 24(3):257–270, Jun 2018. 52, 78
331. Dong-Ming Yan, Guanbo Bao, Xiaopeng Zhang, and Peter Wonka. Low-resolution remeshing using the localized restricted voronoi diagram. *TVCG*, 20(10):1418–1427, 2014. 82
332. Dong-Ming Yan, Bruno Lévy, Yang Liu, Feng Sun, and Wenping Wang. Isotropic remeshing with fast and exact computation of restricted voronoi diagram. *Computer Graphics Forum*, 28(5):1445–1454. 93
333. Dong-Ming Yan and Peter Wonka. Non-obtuse remeshing with centroidal voronoi tessellation. *IEEE Transactions on Visualization and Computer Graphics*, 22(9):2136–2144, Sept 2016. 93
334. Yipin Yang, Yao Yu, Yu Zhou, Sidan Du, James Davis, and Ruigang Yang. Semantic parametric reshaping of human body models. In *2nd International Conference on 3D Vision*, volume 2 of *3DV*, pages 41–48, Washington, DC, USA, Dec 2014. IEEE. 52
335. Yipin Yang, Yao Yu, Yu Zhou, Sidan Du, James Davis, and Ruigang Yang. Semantic parametric reshaping of human body models. In *2014 2nd International Conference on 3D Vision*, volume 2, pages 41–48. IEEE, 2014. 69, 77
336. Li Yi, Lin Shao, Manolis Savva, Haibin Huang, Yang Zhou, Qirui Wang, Benjamin Graham, Martin Engelcke, Roman Klokov, Victor Lempitsky, et al. Large-scale 3D shape reconstruction and segmentation from shapenet core55. *arXiv preprint arXiv:1710.06104*, 2017. 141

337. Yusuke Yoshiyasu, Eiichi Yoshida, and Leonidas Guibas. Symmetry aware embedding for shape correspondence. *Computers & Graphics*, 60:9–22, 2016. 23
338. Shin Yoshizawa, Alexander Belyaev, and Hans-Peter Seidel. A fast and simple stretch-minimizing mesh parameterization. In *Shape Modeling Applications, 2004. Proceedings*, pages 200–208. IEEE, 2004. 102
339. Chao Zhang, Sergi Pujades, Michael Black, and Gerard Pons-Moll. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR*, pages 5484–5493, Washington, DC, USA, 2017. IEEE. 44
340. Hao Zhang, Alla Sheffer, Daniel Cohen-Or, Quan Zhou, Oliver Van Kaick, and Andrea Tagliasacchi. Deformation-driven shape correspondence. *Computer Graphics Forum*, 27(5):1431–1439, 2008. 44
341. Yan Zhang, Mohamed Hassan, Heiko Neumann, Michael J. Black, and Siyu Tang. Generating 3D people in scenes without people. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6194–6204, June 2020. 30
342. Liang Zheng, Hengheng Zhang, Shaoyan Sun, Manmohan Krishna Chandraker, and Qi Tian. Person re-identification in the wild. pages 3346–3355, 2017. 27
343. J. Zhou, M. J. Wang, W. D. Mao, M. L. Gong, and X. P. Liu. SiamesePointNet: A siamese point network architecture for learning 3D shape descriptor. *Computer Graphics Forum*, 39(1):309–321, 2020. 24
344. Siyuan Zhu, Cheng Shang, Jingjing Fan, Xin Wang, and Meili Wang. Bas-reliefs modelling based on learning deformable 3D models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, 2020. 17
345. Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3D Reconstruction with RGB-D cameras. *Computer Graphics Forum (Eurographics State of the Art Reports 2018)*, 37(2), 2018. 28, 29, 75
346. Denis Zorin, Peter Schröder, T De Rose, L Kobbelt, A Levin, and W Sweldens. Subdivision for modeling and animation. *SIGGRAPH 2000 Course Notes*, 2000. 65, 66
347. Silvia Zuffi and Michael J Black. The stitched puppet: A graphical model of 3D human shape and pose. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, pages 3537–3546, Washington, DC, USA, June 2015. IEEE. 44, 45, 55
348. Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 17, 103, 139, 141