Dissertation

Robust Deformable 3D Shape Correspondence and Interpolation

Marvin Eisenberger

Chair for
Computer Vision and
Artificial Intelligence

Department of
Informatics

Technische
Universität
München

Technische Universität München

TUM School of Computation, Information and Technology

Robust Deformable 3D Shape Correspondence and Interpolation

Marvin Marc Eisenberger

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

Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigten Dissertation.

Vorsitz:	Prof. Dr. Rüdiger Westermann
Prüfer der Dissertation:	1. Prof. Dr. Daniel Cremers
	2. Prof. Dr. Emanuele Rodolà
	3. Prof. Dr. Maks Ovsjanikov

Die Dissertation wurde am 20.12.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Computation, Information and Technology am 03.07.2024 angenommen.

To my family.
– *For all the love and support.*

The first principle is that you must not fool yourself and you are the easiest person to fool.

Richard P. Feynman

Abstract

Modeling and analyzing deformable 3D geometric objects is a fundamental challenge in computer vision. In this thesis, we consider the specific tasks of 3D shape correspondence and shape interpolation. The former aims at predicting dense point-to-point maps between pairs of input objects, whereas the latter yields continuous sequences of intermediate poses. There are numerous potential applications of such approaches, including 3D reconstruction, virtual and augmented reality, shape retrieval, predicting object dynamics, information transfer, 3D content creation, and computer graphics pipelines.

We provide an extensive overview of the field of shape analysis, summarizing common concepts and mathematical tools for processing 3D shapes. We further define the challenges of shape correspondence and interpolation, outlining specific techniques and possible solutions relevant for our own contributions. Finally, we introduce the framework of optimal transport, as a central paradigm for computing distances between arbitrary probability distributions. Beyond its basic definition, we include several applications in shape analysis for different, concrete distance metrics.

While classical shape analysis methods obtain solutions through optimization-based algorithms, many recent works leverage geometric deep learning. In practice, both approaches have distinct advantages, informed by a trade-off between offline and online computation cost, as well as the robustness to samples outside the training distribution. Hence, the main contributions of this thesis comprise both classes of approaches.

Despite significant synergies, shape correspondence and interpolation are often addressed as separate challenges in the existing literature. Instead, we propose several novel, holistic techniques that integrate intrinsic matching with extrinsic shape alignment. In this manner, the extrinsic alignment serves as a strong prior constraining the space of admissible correspondence maps, and vice versa.

The main contributions of this dissertation comprise a number of specific practical algorithms, as well as general technical contributions in three major areas. For once, we introduce a shape deformation technique based on parameterized, volumetric, divergence-free vector fields. We find that such vector fields yield a compact yet expressive representation for volume preserving shape morphing. In our second technical contribution, we perform multi-scale shape registration in the product space of extrinsic coordinates and intrinsic spectral features. This coarse-to-fine alignment yields high-quality maps, even for challenging, non-isometric pairs. Finally, we propose several techniques that enable self-supervised learning on geometric data. Such models are trained on collections of 3D shapes without requiring ground truth correspondence annotations.

Zusammenfassung

Das Modellieren und Analysieren geometrischer 3D-Objekte ist eine fundamentale Herausforderung in der Computer Vision. In dieser Doktorarbeit behandeln wir die konkreten Problemstellungen der Korrespondenz und Interpolation von 3D-Formen. Ersteres bezieht sich auf das Bestimmen von dichten Punkt-zu-Punkt Abbildungen zwischen Objektpaaren, während zweiteres kontinuierliche Zwischensequenzen berechnet. Es existieren zahlreiche potentielle Anwendungen solcher Ansätze, z.B. 3D-Rekonstruktion, Virtuelle und Augmentierte Realität, das Finden von Formen in Datenbanken, Modellierung von Objektdynamiken, Informationstransfer, 3D-Inhalteerstellung, und Computergrafik-Pipelines.

Wir bieten einen umfassenden Überblick gängiger Konzepte und mathematischer Werkzeuge zur Verarbeitung von 3D-Formen. Des Weiteren definieren wir die konkreten Herausforderungen der Formen-Korrespondenz und Interpolation, wobei spezielle Verfahren und mögliche Lösungsansätze umrissen werden. Anschließend geben wir eine Einleitung in das Feld des Optimalen Transports, was einen grundlegenden Formalismus zur Berechnung von Distanzen zwischen beliebigen Wahrscheinlichkeitsverteilungen darstellt. Zusätzlich zur grundlegenden Definition, beschreiben wir einige konkrete Anwendungen für die Analyse von 3D-Formen, wobei jeweils verschiedene Distanz-Metriken verwendet werden.

Während klassische Ansätze optimierungsbasierte Algorithmen entwickeln, verwenden neuere Methoden oft Geometrisches Deep Learning. Beide Ansätze haben verschieden Vorteile in der Praxis, bezüglich der benötigten Rechenressourcen, und der Robustheit für Beispiele außerhalb der Trainingsdatenverteilung. Die Hauptbeiträge dieser Dissertation behandeln daher beide Ansätze.

Trotz klarer Synergien, werden Korrespondenz und Interpolation von Formen in existierender Literatur oft als unabhängige Herausforderungen behandelt. Stattdessen definieren wir verschiedene holistische Ansätze, welche intrinsisches Matching mit extrinsischer Ausrichtung kombinieren. Auf diese Weise haben die beiden Komponenten einen regularisierenden Effekt aufeinander.

Die Hauptbeiträge dieser Dissertation behandeln sowohl mehrere konkrete, praktische Algorithmen, als auch allgemeine technische Beiträge in drei Bereichen. Zum einen stellen wir einen Ansatz zur Deformation von Formen vor, basierend auf parametrisierten, volumetrischen, divergenzfreien Vektorfeldern. Solche Felder bieten eine kompakte und flexible Repräsentation volumenerhaltender Verformungen. Unser zweiter Beitrag bezieht sich auf eine mehrskalige Registrierung von 3D-Formen, im Produktraum der extrinsischen Koordinaten und der intrinsischen Frequenzkomponenten. Sogar für nicht-isometrische Paare liefert dieser Ansatz hochwertige Korrespondenzabbildungen. Schließlich stellen wir verschiedene Ansätze vor, die selbstüberwachtes Lernen auf geometrischen Daten ermöglichen. Derartige Modelle werden auf Sammlungen von 3D-Formen trainiert, wobei keine annotierten Korrespondenzen benötigt werden.

Acknowledgements

My personal journey leading up to this thesis was characterized by countless setbacks and successes, new experiences and important learnings – both painful and transformative. Above all, I was privileged to meet and work with many amazing people over the last few years, advisors, colleagues, collaborators, and new friends. All of the work discussed in this thesis was largely made possible through their relentless support.

First and foremost, I want to thank Prof. Daniel Cremers for giving me the opportunity to pursue my PhD studies under his supervision, for the freedom of allowing me to follow my passions and choose my own path in my research career, while providing his ceaseless support and guidance.

It was a superb experience to be part of the Computer Vision Group (CVG) at the Technical University of Munich (TUM), surrounded by numerous talented researchers. I want to thank Zorah Lähner for mentoring me in the early stages of my PhD. I further thank my exceptional co-authors for their invaluable contributions to our joint research projects, including Abhishek Saroha, Andrea Vedaldi, Aysim Toker, Christian Koke, Daniel Cremers, David Novotny, Dongliang Cao, Florian Bernard, Gael Kerchenbaum, Laura Leal-Taixé, Lukas Kondmann, Maolin Gao, Mark Weber, Michael Moeller, Nafie El Amrani, Natalia Neverova, Patrick Labatut, Paul Roetzer, Tarun Yenamandra, Viktoria Ehm, Yuesong Shen, Xiaoxiang Zhu, and Zorah Lähner.

In fall of 2020, I had the opportunity of pursuing an internship at Facebook AI Research, London. It was a profound experience, despite the necessity of moving it to a virtual setting, particularly thanks to Gael Kerchenbaum, Matan Atzmon, Nan Yang, Patrick Labatut, and Roman Shapovalov. Special thanks goes to my exceptional advisors Andrea Vedaldi, David Novotny, and Natalia Neverova, which provided me with the support and inspiration to successfully conclude our joint project in a few short weeks.

I had an amazing time at the TUM with many cherished memories, such as traveling to international conferences and other scientific events with colleagues, afternoon sports sessions with Björn Häfner and Nikolaus Demmel, group skiing trips, rafting on the Isar, hikes, bouldering, barbecues, dinners, and many competitive rounds of kicker (table football) during breaks. It was a pleasure to share an office with several amazing colleagues over the years, including Dominik Mühle, Emanuel Laude, Felix Wimbauer, John Chiotellis, Linus Härenstam-Nielsen, Rudolph Triebel, and Zorah Lähner.

Finally, I want to thank my family and friends, whose unwavering support forms the cornerstone of my journey. My heartfelt thanks to my parents, my brothers, and my wife Aysim for their patience and encouragement, accompanying me through many lows and highs and providing their unconditional love.

Contents

Part I

Introduction and Preliminaries

Chapter 1

Introduction

Reasoning about the 3D world is a long-standing challenge for machine learning systems. As humans, we effortlessly predict the spatial layout of real-world scenes and behaviors of everyday objects. This accomplishment, however, is far from trivial, considering the intricate physical properties and complex dynamics underlying common object interactions. Not surprisingly, it has long been established that a significant percentage of the human brain's cortex is devoted to vision and visuospatial processing [111]. In recent times, there has been a significant increase in the availability of digitized 3D data, from advances in acquisition devices, image-based 3D reconstruction, and generative machine learning models. On the other hand, measuring and extracting the 3D geometry of observed objects is only the first step towards building a functional model of the

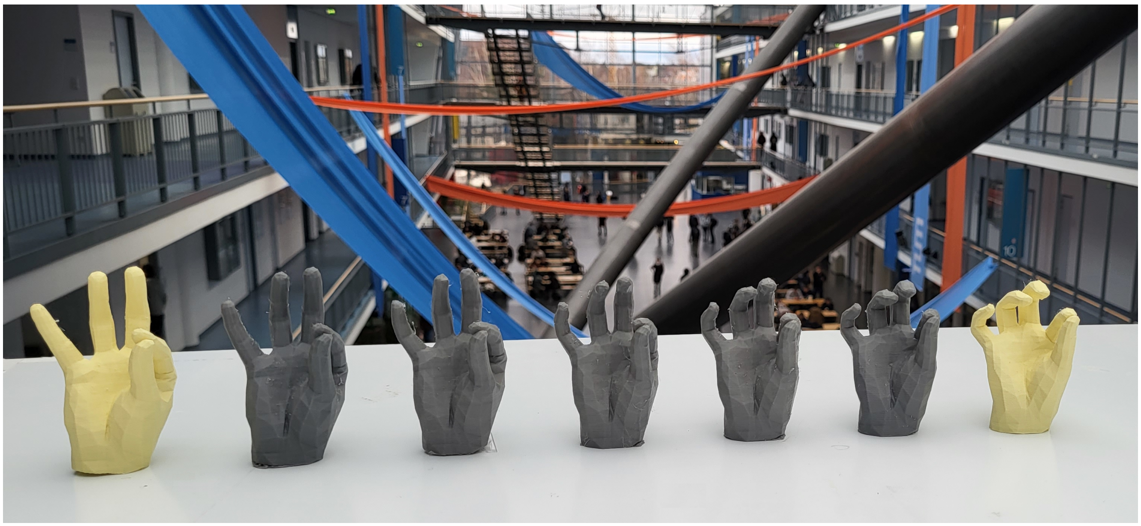


Figure 1.1: A sample shape interpolation sequence (gray) between a pair of input poses (yellow), showcased as 3D prints and photographed against the backdrop of the Mathematics & Informatics building at the Technical University of Munich.

world. The crucial missing link is devising techniques that help us gain a practical, semantic understanding of perceived objects and their environments. The field of shape analysis provides a framework for studying properties of geometric shapes, measuring similarities, reasoning about statistics in object collections, and extracting compact representations. Hence, it offers a promising avenue for capturing the various modes of objects' behaviors. In Fig. 1.1, we visualize a shape interpolation sequence capturing the continuous motion between two hand geometries.

A fundamental challenge in this context is establishing maps between different observations. Depending on the specific application, common formulations involve instances of the same object in different poses, or semantically similar but distinct shapes. According to an often cited anecdote, the famous researcher Takeo Kanade, after being asked about the three most important challenges in computer vision, once replied: "Correspondence, correspondence, correspondence!" [130]. Indeed, we find that many relevant properties of 3D shapes naturally arise from comparing salient features and examining shape variations between multiple object instances. Obtaining correspondence maps allows us quantify similarities and define metrics in shape collections, fuse multiple sensor observations, enhance 3D reconstruction, identify related objects in different contexts, and extrapolate acquired information across object categories.

1.1 Motivation

Shape correspondence and interpolation are fundamental challenges in shape analysis that lie at the core of many 3D computer vision applications. Naturally, there is a large body of literature on both topics [104, 124, 138], documenting a plethora of possible approaches and methodologies. On the other hand, despite targeting similar domains of practical applications, a majority of existing approaches address shape correspondence and interpolation as separate tasks. For instance, many shape interpolation approaches require dense correspondence maps as input [26, 37, 58–60, 68, 132]. Conversely, shape interpolation can be interpreted as a special case of extrinsic shape registration, where a given object instance is continuously deformed into a target pose. In this manner, correspondence maps can be extracted from an interpolation, by propagating individual points along the deformation sequence. We propose to leverage these evident synergies and jointly address the challenges of correspondence and interpolation.

In mathematical terms, the nearly-isometric shape matching problem specifies a quadratic assignment problems (QAP) [72] – predicting optimal permutations that preserve the pairwise distance metric. While this formulations occasionally results in ambiguous matchings in the context of intrinsic symmetries, for the most part, its solutions are well-defined and accurate. On the other hand, the QAP in its basic form is NP-hard and, thus, computationally intractable for most practical applications. While there exist different relaxations in the literature [13, 34, 67, 88, 98, 127], any concrete algorithm necessarily introduces simplifying assumptions, leading to approximate solutions. A key observation of our work is that, in most relevant applications, nearly-isometric correspondences between two surfaces can be explained through an extrinsic shape deformation. Several existing matching methods [21, 53, 79] leverage this assumption by modeling shape alignment in terms of an explicit template mesh or a parametric deformable model [74, 78, 139]. In most cases, such approaches rely on additional domain knowledge and are limited to specific shape categories, such as human bodies [78]. Instead, we propose hybrid approaches that simultaneously predict an intrinsic correspondence map, as well as an extrinsic shape interpolation sequence, *cf*. Chapters 4 and 8. Moreover, to extend this methodology to non-isometric matching tasks, we propose to combine intrinsic functional maps [88] with extrinsic shape registration in a multi-scale framework, *cf*. Chapters 5, 7 and 9. Again, the extrinsic deformation acts as a regularizer constraining the space of admissible correspondence maps. In either case, we find that the holistic approach of combining intrinsic matching with extrinsic alignment often yields superior results, compared to modeling the two tasks separately.

Classical approaches in the field of shape analysis often devise optimization frameworks, comprising energy functions that summarize relevant prior assumptions about a considered task. Specific common formulations of shape correspondence and interpolation in the literature are assignment problems [70, 72] and geodesics in shape space [60, 68, 132], respectively. On the other hand, more recent approaches often define machine learning models, trained on large collections of geometric data. For instance, in the context of shape correspondence, geometric deep learning techniques [29] enable learned feature representations, exhibiting superior predictive accuracy compared to hand-crafted descriptors in classical pipelines. For shape interpolation, autoencoder-type 3D generative models produce compressed latent space representations [69], capturing the different deformation modes of an object category.

In recent times, the availability of geometric data saw a marked increase, further benefiting the development of machine learning techniques. Moreover, querying a learned model typically requires less computational resources than solving an optimization problem for the same task, reducing the runtime cost for unseen test poses. On the other hand, specific assumptions are often more explicit in optimization approaches, which makes their predictions more transparent and controllable. They further do not require an initial offline training phase, which constitutes the bulk of the computation cost in most learning methods. Moreover, since optimization pipelines are not specifically tuned to the training set distribution, they often lead to more robust predictions for out-of-domain test data. Hence, they are readily deployed to diverse practical scenarios. The main contributions of this dissertation include aspects of both classical optimization-based approaches (*cf*. Chapters 4 to 6) and geometric deep learning (*cf*. Chapters 7 to 9).

Despite the significant progress, to date there is still a domain gap between state-of-the-art shape correspondence algorithms and noisy real-world scans. Many benchmarks considered in earlier works [15, 22, 28] comprise synthetic meshes with the same number of vertices and a consistent triangulation. One advantage of such synthetic data is that it simplifies evaluation, since the ground truth maps correspond to the identity matrix. However, for more recent learning-based approaches this bears the risk of overfitting, limiting generalization to unseen test poses. In recent times, real-world 3D data has become readily available from different sources through advances in acquisition devices, 3D reconstruction techniques, and 3D generative models. Predicting correspondences for real scans leads to distinct challenges, including non-isometric deformations, topological changes, partial views, self-occlusions, disconnected components, outliers, and varying degrees of scanning noise.

Compared to synthetic data, obtaining ground truth maps for real scans necessitates extensive manual annotation by human experts. Thus, in order to circumvent this classical annotation bottleneck, we advocate for learning correspondences in a self-supervised manner. To this end, we leverage geometric cues for obtaining correspondence maps, such as optimal transport alignment (*cf*. Chapter 7), extrinsic shape interpolation (*cf*. Chapter 8), or multi-matching (*cf*. Chapter 9).

1.2 Thesis Outline

This cumulative thesis is structured as follows.

In the remainder of **Part I**, we introduce the considered research challenges of deformable shape correspondence and shape interpolation, as well as providing an overview of the existing literature and central references. In Chapter 2, we outline our major contributions in three main areas, focusing on vector field based shape registration [1, 3], multi-scale shape correspondence [2, 5], and self-supervised learning approaches [4, 6], respectively. In Chapter 3, we define important concepts and mathematical tools that form the basis of our work. Specifically, we introduce the topic of shape analysis and properties of 3D shapes. We further formulate the challenges of shape correspondence and interpolation, summarize techniques relevant for our contributions, and outline existing approaches in the literature. Subsequently, we introduce the framework of optimal transport which is a fundamental formalism in the correspondence literature, and central to many of our own approaches.

In Part II, we include all six core publications which constitute the main part of this thesis. Firstly, Chapter 4 defines a divergence-free vector field formulation of extrinsic shape deformation. In Chapter 5, we introduce a multi-scale shape registration approach, which is further applied in a learning setting in Chapter 7. In Chapter 6, we discuss a Hamiltonian dynamics approach to shape interpolation, which leverages a momentum preservation term to obtain time-dependent, divergence-free deformation fields. In Chapter 8, we specify a self-supervised learning approach capable of jointly predicting correspondence maps and interpolation sequences in a single forward pass. Moreover, in Chapter 9 we devise a learnable multi-shape matching approach that simultaneously predicts correspondences for two or more shapes, while enforcing cycle-consistency.

In Part III, we conclude the thesis, summarize our key findings, and provide reflections on relevant directions for future research.

Lastly, in Part IV we include the supplementary materials of our main publications.

Chapter 2

Contributions

We propose several novel techniques aimed at advancing the field of shape analysis, with a specific focus on the challenges of shape correspondence and shape interpolation. In the following, we provide an overview of individual publications and the major contributions that constitute the core of the thesis.

2.1 List of Publications

A summary of the the publications over the course of this thesis is provided in Table 2.1 with the core publications highlighted in Table 2.2. This work was done in collaboration with the exceptional researchers Daniel Cremers, Aysim Toker, Zorah Lähner, Laura Leal-Taixé, Florian Bernard, Maolin Gao, Paul Roetzer, Viktoria Ehm, Michael Moeller, Lukas Kondmann, Mark Weber, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, and Andrea Vedaldi. Each reference is the result of extensive research conducted at the Technical University of Munich [7, 1–3, 8, 5, 6, 9, 10] and Facebook AI Research [4], as well as collaborations with the University of Oxford, the University of Bonn, Nvidia, Planet Labs, the University of Siegen, and the German Aerospace Center. Individual references correspond to peer-reviewed publications at esteemed computer vision and machine learning venues, including CVPR, NeurIPS, ECCV, ICCV, 3DV, and the Symposium on Geometry Processing (SGP). Several of the works received either a spotlight [1] or full oral presentation [2, 3]. The publication NeuroMorph [4] was developed in the fall of 2020 as part of an internship at Facebook AI Research, London.

M. Eisenberger, Z. Lähner, and D. Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38 of number 5, pages 1–12, 2019 [3]

M. Eisenberger, Z. Lahner, and D. 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 [2]

M. Eisenberger and D. Cremers. Hamiltonian dynamics for real-world shape interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 179–196. Springer International Publishing, 2020 [1]

M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. Deep shells: unsupervised shape correspondence with optimal transport. *Advances in Neural information processing systems*, 33:10491–10502, 2020 [5]

M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi. Neuromorph: unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021 [4]

A. Toker, L. Kondmann, M. Weber, M. Eisenberger, A. Camero, J. Hu, A. P. Hoderlein, Ç. Şenaras, T. Davis, D. Cremers, et al. Dynamicearthnet: daily multi-spectral satellite dataset for semantic change segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21158–21167, 2022 [10]

M. Eisenberger, A. Toker, L. Leal-Taixé, F. Bernard, and D. Cremers. A unified framework for implicit sinkhorn differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 509–518, 2022 [8]

M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. G-msm: unsupervised multi-shape matching with graph-based affinity priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22762–22772, 2023 [6]

M. Gao, P. Roetzer, M. Eisenberger, Z. Lähner, M. Moeller, D. Cremers, and F. Bernard. Sigma: scale-invariant global sparse shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 645–654, 2023 [9]

V. Ehm, P. Roetzer, M. Eisenberger, M. Gao, F. Bernard, and D. Cremers. Geometrically consistent partial shape matching. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024 [7]

Table 2.1: A full list of peer-reviewed publications over the course of the thesis. The list is in chronological order, according to the date of publication. While we include all references for completeness, only the six publications highlighted in black contribute to this dissertation.

1. M. Eisenberger, Z. Lähner, and D. Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38 of number 5, pages 1–12, 2019 [3]

2. M. Eisenberger, Z. Lahner, and D. 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 [2]

3. M. Eisenberger and D. Cremers. Hamiltonian dynamics for real-world shape interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 179–196. Springer International Publishing, 2020 [1]

4. M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. Deep shells: unsupervised shape correspondence with optimal transport. *Advances in Neural information processing systems*, 33:10491–10502, 2020 [5]

5. M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi. Neuromorph: unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021 [4]

6. M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. G-msm: unsupervised multi-shape matching with graph-based affinity priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22762–22772, 2023 [6]

Table 2.2: A list of core publications that contribute to this dissertation. The list is in chronological order, according to the date of publication.

2.2 Major Contributions

We include a total of six major publications that contribute towards this thesis [1–6]. Each of these works constitutes a fundamental building block towards the goal of applying shape correspondence and interpolation algorithms to real-world geometric data – beyond the classical assumption of near-isometry. Our contributions include both works on classical, optimization based approaches [1–3], as well as geometric deep learning models trained on collections of shapes [4–6].

We distinguish between three major formalisms, where we propose tools to model shape deformation in terms of divergence-free vector fields (Sec. 2.2.1), compute multi-scale correspondences in hybrid embedding spaces of both intrinsic and extrinsic geometric features (Sec. 2.2.2), and devise self-supervised geometric deep learning techniques to enable representation learning for shape analysis tasks (Sec. 2.2.3). All three components are central to several of our proposed approaches, and indeed there is a substantial overlap between the different areas. For a complete picture, we outline the contributions of two key publications for each topic.

2.2.1 Vector Field Based Shape Registration

Divergence-Free Shape Correspondence by Deformation [3] In Chapter 4, we introduce a novel formalism of modeling shape deformation in terms of parameterized, volumetric vector fields. One of our key contributions is representing of such fields as a linear combination of spatially dense, divergence-free basis functions. The individual components of this basis are constructed by applying Helmholtz's theorem, where the corresponding potential fields are defined as Fourier basis functions – ordered from low to high frequencies. Since individual basis functions are divergence-free, any possible linear combination retains this property. Thus, the volume preservation property is built into the parameterization as a hard constraint. For a pair of input surfaces $\mathcal{X}$ and $\mathcal{Y}$, we obtain a deformation of $\mathcal{X}$ by propagating its vertices along a given vector field through time-integration. We then optimize for the unknown correspondences and vector field coefficients, such that the deformed shape coincides with a second reference pose $\mathcal{Y}$.

Hamiltonian Dynamics for Real-World Shape Interpolation [1] While the vector field representation yields compact and highly expressible shape deformations, in the formulation presented in [3] it is limited to stationary fields. This poses a significant limitation, since, for many relevant types of motion, different parts of an object move through the same location in 3D space at different times. In Chapter 6, we extend this formulation to time-dependent fields, by introducing additional assumptions about the dynamics of the considered, deformable surfaces $\mathcal{X}$ and $\mathcal{Y}$. To this end, we formulate the Hamiltonian energy of an object in terms of its local surface distortion, as well as its kinetic energy. Moreover, each individual surface point on $\mathcal{X}$ is embedded in the product space of its position at each time, as well as its momentum represented as a divergence-free field. The principles of Hamiltonian mechanics then yield laws of motion for an object $\mathcal{X}$ and a given initial velocity field, which we integrate over time with an implicit Euler scheme. To obtain a shape interpolation, we optimize for the initial velocity field that minimizes the distance between the deformed shape $\mathcal{X}$ and the reference pose $\mathcal{Y}$.

2.2.2 Multi-Scale Shape Correspondence

Multi-Scale Shape Registration with Functional Maps [2] In Chapter 5, the smooth shells formalism is introduced, which is defined as a sigmoid weighted low-pass filtering of a 3D shape. This procedure yields a simplified geometry, which can be either slightly smoothed or entirely devoid of fine-scale details, depending on the frequency parameter of the filtering. Based on this smoothing operator, we then devise a multi-scale matching approach. For a pair of input shapes $\mathcal{X}$ and $\mathcal{Y}$, we initially match very coarse approximations, and then gradually upsample the level of detail in an iterative scheme. In each step, we compute dense correspondence maps, as well as shape registrations in the product space of extrinsic shells, and intrinsic spectral features. We further propose a specialized initialization algorithm based on Markov chain Monte Carlo sampling. Overall, our experiments demonstrate that the coarse-to-fine optimization strategy yields superior local minima, even for challenging non-isometric matching tasks.

Unsupervised Shape Correspondence with Optimal Transport [5] In Chapter 7, the smooth shells methodology is applied in the context of geometric deep learning. To this end, a differentiable variant of the multi-scale matching algorithm in [2] is introduced, based on optimal transport. Specifically, the distance between two shapes in the product space of smooth shell geometries and spectral features is defined as the Wasserstein 2-metric, with an additional entropy regularization term. Overall, this results in a matching operator that produces a transport plan, *i.e.*, a soft correspondence map, between a pair of 3D shapes $\mathcal{X}$ and $\mathcal{Y}$. To initialize the optimal transport matching, we additionally introduce a feature extraction backbone based on learnable spectral convolution layers. The resulting model can be trained in a self-supervised manner, where the training loss is defined as the Wasserstein 2-metric in the joint embedding space. Compared to the optimization-based approach [2], the deep shells method [5] has distinct advantages. While [2] requires several minutes of optimization, a single forward pass in [5] typically takes only a few seconds. On the other hand, the performance of [5] depends, to a certain degree, on the quality of the training data. Moreover, [2] can be applied to any matching problem without requiring an initial offline training phase.

2.2.3 Self-Supervised Learning Approaches

Unsupervised Shape Interpolation and Correspondence in One Go [4] In Chapter 8, the self-supervised learning approach NeuroMorph is outlined, which jointly predicts correspondence maps and interpolation sequences in a single forward pass. The concrete network architecture is based on a novel graph neural network model, interleaving EdgeConv message passing layers with global feature pooling. Overall, the proposed pipeline integrates two consecutive modules, responsible for predicting correspondences and interpolated poses, respectively. Related to our observations discussed in Sec. 2.2.1, we find that there are significant synergies between the two challenges of correspondence and interpolation. In this manner, the interpolation module serves as a strong prior constraining the correspondence maps, which enables self-supervised learning in an alignment-based formulation. Moreover, NeuroMorph is computationally inexpensive, compared to our previous approaches involving an expectation maximization algorithm [3], or an inner optimization loop during each forward pass [5]. On the other hand, in contrast to [5], the resulting correspondence maps contain a degree of fine-scale noise and, hence, require post-processing with [2] in certain scenarios.

Unsupervised Multi-Shape Matching with Graph-based Affinity Priors [6] A majority of learning-based matching models in the shape analysis literature, including [4, 5], are trained in a pairwise manner – predicting correspondence maps for random pairs of shapes $\mathcal{X}$ and $\mathcal{Y}$ during each forward pass. In contrast, in Chapter 9 we present a novel approach that jointly predicts correspondences for a collection of $N > 2$ input shapes. To this end, we construct a weighted, undirected shape graph $\mathcal{G}$, whose scalar edge weights reflect similarity scores between two surfaces. During training, we enforce cycle-consistency along shortest paths in the shape graph, promoting optimal multi-matches. The overall architecture consists of a learnable feature extractor, a multi-scale pairwise matching module based on [5], as well as the shape graph multi-matching module. We find that the multi-shape formulation increases the correspondence accuracy over earlier, pairwise approaches [4, 5] on several benchmarks. These improvements are especially significant for real scans corrupted by topological changes from self-contact, where the additional poses help to infer the latent topology of a considered shape category.

Chapter 3

Theoretical Background

We now define important concepts and terminology on shape analysis and relevant tasks in geometry processing. These will form the basis for the mathematical toolbox used throughout our own contributions in Part II. We further provide an overview of the pertinent literature.

3.1 3D Shapes

The primary objective of the field of shape analysis is studying geometric objects $\mathcal{X}$. In the following, we focus on the specific case of compact, 2-dimensional Riemannian manifolds [43] with an explicit embedding in 3D space $\mathcal{X} \subset \mathbb{R}^3$. This perspective closely aligns with real-world sensor data from observed objects (3D scans) that we try to model. On the other hand, it is often beneficial to devise algorithms that are fully intrinsic, *i.e.*, are agnostic to the specific coordinate embedding. In the following, we provide definitions of central properties and tools used to analyze 3D shapes $\mathcal{X}$.

3.1.1 Invariance

Consider the equivalence class $[\mathcal{X}]_{\mathcal{T}} = \{T(\mathcal{X}) \subset \mathbb{R}^3 | T \in \mathcal{T}\}$ induced by a given category of transformations $\mathcal{T}$. We then call any property $f(\cdot)$ of the shape $\mathcal{X}$ *invariant* under the class of transformations $\mathcal{T}$, if it holds $\forall \mathcal{X}' \in [\mathcal{X}]_{\mathcal{T}} : f(\mathcal{X}) = f(\mathcal{X}')$. One common example is the class of rigid-body transformations $T \in \mathrm{SE}(3)$

$$T : \mathbb{R}^3 \to \mathbb{R}^3, \mathbf{p} \mapsto \mathbf{R}\mathbf{p} + \mathbf{t}, \tag{3.1}$$

where $\mathbf{R} \in \mathrm{SO}(3)$ is a rotation matrix and $\mathbf{t} \in \mathbb{R}^3$ a translation vector. The rotation and translation determine an object's position, in absolute terms, and thus reflect coordinate conventions and the viewpoint of the observer. Indeed, we intuitively assume many properties of real-world objects to be preserved under such rigid maps, *e.g.*, the surface area of $\mathcal{X}$ or the pairwise Euclidean distance of two points on the surface. In practice, it is often beneficial to constrain the space of solutions and promote useful inductive biases by restricting computations to such invariant properties of geometric objects [29].

3.1.2 Isometry

Another useful class of transformations in the context of deformable object categories are *isometric* maps $T : \mathcal{X} \to \mathcal{X}'$. If we consider the metric space $(\mathcal{X}, d_{\mathcal{X}})$ with the geodesic distance metric $d_{\mathcal{X}}$, then this property can be defined as the set of distance preserving maps $d_{\mathcal{X}}(\mathbf{p}, \mathbf{q}) = d_{\mathcal{X}'}(T(\mathbf{p}), T(\mathbf{q}))$ for $\mathbf{p}, \mathbf{q} \in \mathcal{X}$. From the perspective of differential geometry, such maps can be characterized as diffeomorphisms T that preserve the metric tensor [43] of the manifold $\mathcal{X}$. A simple example are rigid transformations defined in Eq. (3.1), which are a special case of isometric maps. Isometries are a powerful tool to model deformable object categories, such as humans, hands, or animal bodies. They closely align with our intuition of real-world dynamics, since the distance along the shortest path between points on the surface $d_{\mathcal{X}}$ is approximately preserved across different poses. On the other hand, under real-world conditions the isometry property is often violated, to a certain extent, due to small degrees of surface stretching and scanning noise. To reflect these deviations, such transformations are commonly referred to as *nearly-isometric* maps [17, 27, 48, 88] in the applied literature.

3.1.3 Intrinsic/Extrinsic

We call a property $f(\cdot)$ of shapes $\mathcal{X}$ *intrinsic*, if it does not depend on the explicit coordinate embedding of the underlying Riemannian manifold $\mathcal{X}$. Since isometric maps T preserve the metric tensor [43] on $\mathcal{X}$, we assert that intrinsic properties f are invariant under isometric transformations $f(\mathcal{X}) = f \circ T(\mathcal{X})$. In contrast, we call f *extrinsic*, if it is influenced by the specific coordinate embedding of $\mathcal{X} \subset \mathbb{R}^3$. Common examples of extrinsic properties are, *e.g.*, the outer normals and the local curvature of the surface.

Intrinsic shape descriptors are useful, in practice, to constrain the space of admissible solutions and simplify optimization. By design, they have a built-in robustness to challenging non-rigid deformations that we commonly observe in deformable shape dynamics. On the other hand, there are several well-known failure modes where intrinsic descriptors are insufficient to capture all the necessary information about an observed surface. For many real-world objects, there exist so-called intrinsic symmetries, which are isometric self-maps $T : \mathcal{X} \to \mathcal{X}$ not equal to the identity map. For instance, humans are approximately mirror-symmetric w.r.t. the sagittal plane (left-right symmetry). Extrinsic information is thereby required to convey the left-right orientation. Moreover, the isometry assumption is often violated for real scans, *e.g.*, due to topological changes from self-contact, corrupted scans/noise, or partial views.

3.1.4 Laplace-Beltrami Operator

The Laplace-Beltrami operator (LBO), sometimes dubbed the 'Swiss Army knife' of geometry processing [114], is a ubiquitous tool in the shape analysis literature, defined as

$$\Delta_{\mathcal{X}} f := \mathrm{div}(\nabla f), \tag{3.2}$$

where $f \in C^2(\mathcal{X})$ is a twice continuously differentiable scalar function and div and ∇ are the divergence and gradient operators defined on the Riemannian manifold $\mathcal{X}$, respectively. The LBO is a generalization of the Laplacian in Euclidean space and allows us to extend familiar concepts, such as heat diffusion, fluid flow, or wave propagation, to curved Riemannian domains. We summarize a few of its relevant key properties:

Proposition 1. *Let $\Delta_{\mathcal{X}}$ be the LBO defined in Eq.* (3.2) *and let $f, g \in C^2(\mathcal{X})$ and $a, b \in \mathbb{R}$ be arbitrary, then it holds that $\Delta_{\mathcal{X}}$ is*

i. Linear: $\Delta_{\mathcal{X}}(af + bg) = a\Delta_{\mathcal{X}} f + b\Delta_{\mathcal{X}} g$.

ii. Self-adjoint: $\langle f, \Delta_{\mathcal{X}} g \rangle_{\mathcal{X}} = \langle \Delta_{\mathcal{X}} f, g \rangle_{\mathcal{X}}$.

iii. Negative semidefinite: $\langle f, \Delta_{\mathcal{X}} f \rangle_{\mathcal{X}} \leq 0$.

iv. Intrinsic, as defined in Sec. 3.1.3.

In the geometry processing literature, the LBO has been applied to a range of tasks, including mesh editing [118], smoothing [42], deformation [116], geodesic distance computation [38], shape descriptors [16, 120], correspondence [88], and interpolation [133].

3.1.5 Spectral Decomposition

Several applications model shape properties through the LBO's spectrum. To this end, we consider the eigenvalue problem [33]

$$-\Delta_{\mathcal{X}} \phi_k = \lambda_k \phi_k, \tag{3.3}$$

where λ_k and $\phi_k : \mathcal{X} \to \mathbb{R}$ are the k-th eigenvalue and the corresponding eigenfunction, respectively. The eigenpairs further admit the following properties:

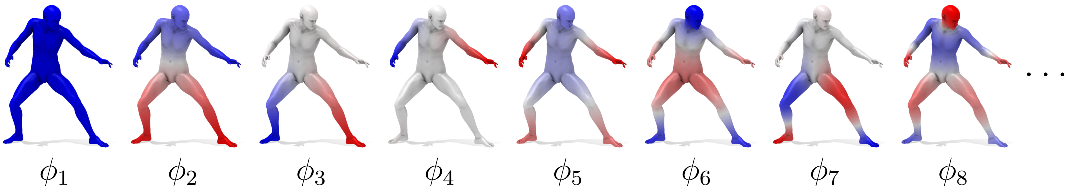


Figure 3.1: We visualize the first eight LBO eigenfunctions $\phi_1, \dots, \phi_8$ for an example shape $\mathcal{X}$ from the TOSCA dataset [28]. Each $\phi_k : \mathcal{X} \to \mathbb{R}$ is a scalar field, displayed with a color coding showing $\phi_k > 0$ as blue, $\phi_k < 0$ as red, and $\phi_k = 0$ as white. According to Proposition 2, the eigenfunctions ϕ_k are ordered w.r.t. the frequency values $\lambda_k \geq 0$. For shapes without boundary, as the one shown above, $\phi_1 \equiv C$ is a constant function and $\lambda_1 = 0$.

Proposition 2. *Let* $(\lambda_1, \phi_1), (\lambda_2, \phi_2), \dots$ *be the eigenpairs defined in Eq.* (3.3)*, then*

i. $0 \leq \lambda_1 \leq \lambda_2 \leq ... < \infty$, *where* $\lim_{k\to\infty} \lambda_k = \infty$.

ii. The set $\{\phi_1, \phi_2, \dots\}$ *forms an orthonormal basis of* $L^2(\mathcal{X})$.

These results are a special case of the spectral theorem [56] and follow from the fact that $\Delta_{\mathcal{X}}$ is a compact, self-adjoint, negative-semidefinite operator on $\mathcal{X}$. Refer to [33, Ch. 1] for a formal proof. As a common convention, in *i.*, the eigenvalues λ_k are ordered increasingly w.r.t. their magnitude. This results in a natural ordering of the eigenbasis $\{\phi_1, \phi_2, \dots\}$ from low to high frequencies, see Fig. 3.1 for a visualization. While there exist repeated eigenvalues, the eigenspaces associated with each distinct value are finite. As a direct consequence of *ii.*, we can represent arbitrary square-integrable functions $f \in L^2(\mathcal{X})$ in terms of the orthonormal eigenbasis

$$f = \sum_{k=1}^{\infty} a_k \phi_k, \quad \text{where} \quad a_k := \langle f, \phi_k \rangle_{\mathcal{X}} = \int_{\mathcal{X}} f(p)\phi_k(p)dp. \tag{3.4}$$

This identity is analogous to Fourier analysis for signal processing in Euclidean domains. According to the min-max principle, the basis representation in Eq. (3.4) is optimal for compressing smooth functions $f \in L^2(\mathcal{X})$, when only a fixed, finite number $K \in \mathbb{N}$ of basis functions are retained $f \approx \sum_{k=1}^{K} a_k \phi_k$, refer to [92, Ch. 10] for more details.

3.1.6 Discretization

So far, we defined 3D shapes $\mathcal{X}$ as continuous, differentiable manifolds [43]. In practice, we need to introduce discrete approximations of such manifold surfaces to obtain a computationally tractable, finite parameterization. To this end, we focus on the widely used polygonal mesh representation [24].

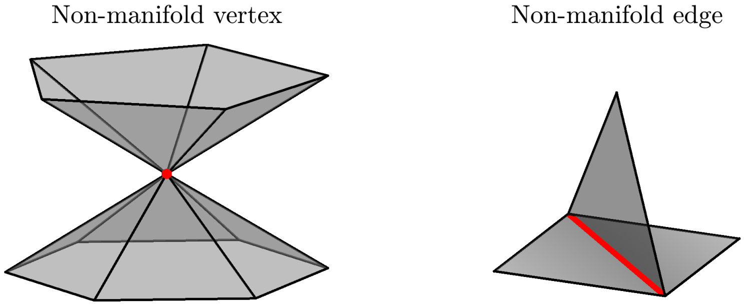


Figure 3.2: Examples for the two main cases of degenerate triangular meshes, refer to [24, Ch. 1.3] for a detailed discussion. Non-manifold vertices are connected to two separate surface segments 'pinched' together at a single point, whereas non-manifold edges coincide with more than two triangle faces.

Specifically, we consider triangular meshes specified by a finite number of vertices $\mathcal{V} = \{\mathbf{v}_1, \dots, \mathbf{v}_{|\mathcal{V}|}\} \subset \mathbb{R}^3$ and faces $\mathcal{F} = \{\boldsymbol{\tau}_1, \dots, \boldsymbol{\tau}_{|\mathcal{F}|}\} \subset \mathcal{V} \times \mathcal{V} \times \mathcal{V}$. For convenience, we often denote $\mathbf{V} \in \mathbb{R}^{|\mathcal{V}| \times 3}$ and $\mathbf{T} \in \mathbb{N}^{|\mathcal{F}| \times 3}$ in matrix notation, where $\mathbf{V}_{i,:} = \mathbf{v}_i$ for any $i \in \{1, \dots, |\mathcal{V}|\}$ and $\mathbf{T}_{i,:} = [j, k, l]^\top$ for any $\boldsymbol{\tau}_i = (\mathbf{v}_j, \mathbf{v}_k, \mathbf{v}_l) \in \mathcal{F}$ and $i \in \{1, \dots, |\mathcal{F}|\}$. Not every possible simplicial complex $(\mathcal{V}, \mathcal{F})$ constitutes a valid approximation of a 2D manifold surface $\mathcal{X}$. We call a triangular mesh *2-manifold*, if it is free of non-manifold vertices and edges [24, Ch. 1.3], see Fig. 3.2 for a visualization. Moreover, we require that a given mesh is free of self-intersections, meaning that triangles shall only intersect at adjacent edges. Avoiding these degenerate cases ensures that, in the limit $|\mathcal{F}| \to \infty$, the mesh $(\mathcal{V}, \mathcal{F})$ approximates a smooth manifold surface.

Scalar fields $f : \mathcal{X} \to \mathbb{R}$ on $\mathcal{X}$ are commonly approximated as piecewise linear functions on the triangles $\mathcal{F}$. They are fully specified by scalar coefficients $\mathbf{f}_i$ at each vertex $\mathbf{v}_i$, resulting in the coefficient vector $\mathbf{f} \in \mathbb{R}^{|\mathcal{V}|}$. Likewise, linear operators on $\mathcal{X}$, such as the LBO $\Delta_{\mathcal{X}}$ defined in Eq. (3.2), can be discretized as matrices $\mathbf{L} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$. For the specific case of the LBO, we utilize the standard cotangent discretization based on finite element analysis. This results in a matrix $\mathbf{L} := \mathbf{M}^{-1}\mathbf{S}$ defined in terms of an invertible, sparse, symmetric mass matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ and a sparse, symmetric stiffness matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$, refer to [93] for technical details on how to construct $\mathbf{M}$ and $\mathbf{S}$.

Applying the discrete operator $\mathbf{L}$ to a scalar field $\mathbf{f}$ simply amounts to a matrix-vector multiplication $\mathbf{g} := \mathbf{L}\mathbf{f} \in \mathbb{R}^{|\mathcal{V}|}$. On the other hand, explicitly storing the dense matrix $\mathbf{L} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ can be computationally expensive, depending on the resolution $|\mathcal{V}|$. Hence, for many applications, it is preferable to directly retain the sparse mass and stiffness matrices $\mathbf{M}$ and $\mathbf{S}$. For instance, we can discretize the eigenpairs (λ_k, ϕ_k)

defined in Eq. (3.3) by solving a generalized eigenvalue problem

$$-\mathbf{S}\boldsymbol{\phi}_k = \lambda_k \mathbf{M}\boldsymbol{\phi}_k. \tag{3.5}$$

Since $\mathbf{M}$ is invertible, this expression is mathematically equivalent to the standard eigenvalue problem $-\mathbf{L}\boldsymbol{\phi}_k = \lambda_k\boldsymbol{\phi}_k$, while allowing iterative solvers to take advantage of the sparsity of $\mathbf{M}$ and $\mathbf{S}$. In matrix notation, we denote $\boldsymbol{\Phi} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ where $\boldsymbol{\Phi}_{:,k} = \boldsymbol{\phi}_k$ and $\boldsymbol{\Lambda} := \mathrm{diag}\left(\lambda_1, \dots, \lambda_{|\mathcal{V}|}\right) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$. The eigenvectors $\boldsymbol{\Phi}$ form an orthonormal basis w.r.t. the inner product associated with the mass matrix $\mathbf{M}$, *i.e.*, $\boldsymbol{\Phi}^\top\mathbf{M}\boldsymbol{\Phi} = \mathbf{I}$.

3.2 Shape Correspondence

Computing correspondences between 3D shapes is a central challenge in shape analysis. There is a growing demand for such algorithms since, in recent times, applications involving 3D avatars and digitized objects are becoming increasingly wide-spread. Establishing correspondence maps allows us to reason about similarity in shape collections, understand object dynamics, perform sensor fusion of multiple observations, detect anomalies, or perform information and style transfer in computer graphics applications. Moreover, such geometric data is becoming broadly available due to improvements in acquisition devices, 3D reconstruction techniques, and 3D generative models.

3.2.1 Problem Formulation

We consider the problem of estimating dense correspondence maps $\pi : \mathcal{X} \to \mathcal{Y}$ between pairs of deformable shapes $\mathcal{X}$ and $\mathcal{Y}$. Depending on the specific application, $\mathcal{X}$ and $\mathcal{Y}$ either depict different poses of the same object or instances of semantically similar, but distinct objects. To determine π, we require that corresponding points $\mathbf{p} \in \mathcal{X}$ and $\mathbf{q} = \pi(\mathbf{p}) \in \mathcal{Y}$ are identical or semantically equivalent.

It is often useful to make additional assumptions about the class of transformations π. For instance, in the case of rigid object registration, we can apply the popular iterative closest point algorithm [20], limiting the optimization to the 6-dimensional space of transformations defined in Eq. (3.1). Some approaches assume that π are diffeomorphisms, defined as bijective maps where both π and π^{-1} are differentiable. However, the resulting optimization problems are computationally expensive and thus limited to coarse resolutions of a few hundred triangles [100, 131]. Moreover, the diffeomorphism assumption is often violated in noisy real-world data due to partial views, self-occlusions, disconnected components, scanning noise, and topological changes from self-contact.

In most cases, the dynamics of non-rigid shapes are accurately approximated by nearly-isometric transformations as defined in Sec. 3.1.2. This assumption can be leveraged, in practice, by parameterizing π with intrinsic functional maps (*cf*. Sec. 3.2.2) or through intrinsic local descriptors (*cf*. Sec. 3.2.3). On the other hand, there are many applications where the isometry assumption is too restrictive, *e.g.*, considering noisy scans, or for inter-class pairs with distinct body shapes. For practical algorithms, we find that integrating both intrinsic and extrinsic information often yields the most robust results. This is a common theme in many of our proposed approaches discussed in Chapter 2.

In the discrete case, we formulate the correspondence problem for a pair of triangular meshes $\left(\mathcal{V}^{(\mathcal{X})}, \mathcal{F}^{(\mathcal{X})}\right)$ and $\left(\mathcal{V}^{(\mathcal{Y})}, \mathcal{F}^{(\mathcal{Y})}\right)$ as finding a point-to-point map $\pi : \mathcal{V}^{(\mathcal{X})} \to \mathcal{V}^{(\mathcal{Y})}$. In practice, we represent π with binary row-stochastic matrices $\mathbf{\Pi} \in \{0,1\}^{|\mathcal{V}^{(\mathcal{X})}|\times|\mathcal{V}^{(\mathcal{Y})}|}$, where $\mathbf{\Pi}\mathbf{1} = \mathbf{1}$ and $\mathbf{1}$ is the vector of ones. An entry $\mathbf{\Pi}_{ij} = 1$ indicates a match between $\mathbf{v}_i^{(\mathcal{X})}$ and $\mathbf{v}_j^{(\mathcal{Y})}$. For deep learning approaches and robust optimization, we sometimes relax the binary constraint $\{0,1\}$ to $[0,1]$, resulting in soft correspondence matrices $\mathbf{\Pi}$.

3.2.2 Functional Maps

We provide a brief introduction to the functional map formalism [88] here and refer to the SIGGRAPH ASIA 2016 course notes [89] for an in-depth discussion.

3.2.2.1 Definition

The main idea of *functional maps* (FM) is to reframe the problem of determining a pointwise map $\pi : \mathcal{X} \to \mathcal{Y}$ as finding a linear functional $\mathcal{C} : L^2(\mathcal{X}) \to L^2(\mathcal{Y})$ from functions $f \in L^2(\mathcal{X})$ to functions $g = \mathcal{C}(f) \in L^2(\mathcal{Y})$. According to Eq. (3.4), any such functions can be represented compactly in their respective LBO eigenbases $f = \sum_k a_k\phi_k$ and $g = \sum_l b_l\psi_l$. In this basis representation it then holds

$$\begin{aligned}\sum_l b_l\psi_l =& g = \mathcal{C}(f) = \mathcal{C}\left(\sum_k a_k\phi_k\right) \overset{\text{linearity}}{=} \sum_k a_k\mathcal{C}\left(\phi_k\right)\\ =& \sum_k a_k \sum_l \underbrace{\left\langle \mathcal{C}\left(\phi_k\right), \psi_l\right\rangle}_{=:c_{lk}} \psi_l. \end{aligned} \tag{3.6}$$

To make this expression computationally tractable, we commonly consider truncated eigenbases, retaining only a finite basis of $K \in \mathbb{N}$ dimensions. Assuming that $f \in \operatorname{span}\{\phi_1, \dots, \phi_K\}$ and $g \in \operatorname{span}\{\psi_1, \dots, \psi_K\}$, we obtain from Eq. (3.4) that

$$b_l \approx \sum_{k=1}^{K} c_{lk}a_k, \quad \text{for all} \;\; l \in \{1, \dots, K\}. \tag{3.7}$$

In matrix notation, we can simply write $\mathbf{b} \approx \mathbf{C}\mathbf{a}$ for the coefficient vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^K$ and the matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$. Thus, the problem of estimating the functional $\mathcal{C} : L^2(\mathcal{X}) \to L^2(\mathcal{Y})$ is simplified to estimating the coefficient matrix $\mathbf{C}$.

While other choices of bases are possible, the LBO basis is the standard choice in the literature due to its natural ordering from low to high frequencies. Specifically, the min-max principle [92, Ch. 10] guarantees that the finite-dimensional subspace $\operatorname{span}\{\phi_1, \dots, \phi_K\} \subset L^2(\mathcal{X})$ is provably optimal for representing smooth functions. Since the LBO is known to be intrinsic (*cf*. Prop. 1), the eigenbasis representation of $\mathbf{C}$ is naturally biased towards nearly-isometric maps. Hence, a relatively small number of retained basis functions $K \approx 100$ often suffices, in practice, to capture the variability of nearly-isometric poses – significantly reducing the number of variables $K^2 \ll |\mathcal{V}^{(\mathcal{X})}| \cdot |\mathcal{V}^{(\mathcal{Y})}|$.

3.2.2.2 Optimization

In order to estimate $\mathbf{C}$ for a given pair of shapes $\mathcal{X}$ and $\mathcal{Y}$, we consider the following optimization problem [88]

$$\arg\min_{\mathbf{C}} \; \|\mathbf{C}\mathbf{A} - \mathbf{B}\|_F^2 + \lambda_{\text{reg}} E_{\text{reg}}(\mathbf{C}), \tag{3.8}$$

consisting of a descriptor preservation term of some spectral features $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{K \times L}$ and a regularization energy E_{reg}. One major advantage of the functional map representation is that assumptions about the shapes $\mathcal{X}$ and $\mathcal{Y}$ and correspondence maps π can be formulated in terms of the matrix representation $\mathbf{C}$. We summarize several examples for a pair of triangular meshes $\big(\mathcal{V}^{(\mathcal{X})}, \mathcal{F}^{(\mathcal{X})}\big)$ and $\big(\mathcal{V}^{(\mathcal{Y})}, \mathcal{F}^{(\mathcal{Y})}\big)$ and their truncated eigenbases $\mathbf{\Phi}_K \in \mathbb{R}^{|\mathcal{V}^{(\mathcal{X})}| \times K}$ and $\mathbf{\Psi}_K \in \mathbb{R}^{|\mathcal{V}^{(\mathcal{Y})}| \times K}$ in the following.

Local Descriptors If we have some local descriptor fields $\mathbf{f}_i \in \mathbb{R}^{|\mathcal{V}^{(\mathcal{X})}|}$ and $\mathbf{g}_i \in \mathbb{R}^{|\mathcal{V}^{(\mathcal{Y})}|}$ available on the input meshes, then we can formulate a local descriptor similarity term between $\mathbf{f}_i$ and $\mathbf{g}_i$ by setting the spectral features $\mathbf{a}_i := \mathbf{\Phi}_K^\dagger \mathbf{f}_i$ and $\mathbf{b}_i := \mathbf{\Psi}_K^\dagger \mathbf{g}_i$.

Input Landmarks If a sparse set of corresponding landmarks $\mathbf{v}_i^{(\mathcal{X})}$ and $\mathbf{v}_j^{(\mathcal{Y})}$ are given as input, we can again incorporate this information by defining corresponding scalar fields $\mathbf{f}_i$ and $\mathbf{g}_i$, *e.g.*, in terms of a smooth indicator function around the landmark vertices.

Operator Preservation In the case of isometric transformations, functional maps are known to commute with the LBO [88, Thm. 5.1 (2)]. Hence, we can devise a regularization term based on the LBO eigenvalues $E_{\text{reg}}(\mathbf{C}) := \big\|\mathbf{\Lambda}^{(\mathcal{Y})}\mathbf{C} - \mathbf{C}\mathbf{\Lambda}^{(\mathcal{X})}\big\|_F^2$. Other popular variants include preservation of the resolvent operator $(\Delta_{\mathcal{X}} - \alpha \mathrm{Id})^{-1}$ introduced by [97], or slanted diagonal regularization for partial shape correspondence [99].

Area Preservation If we assume that a given correspondence map is locally area-preserving, it was shown in [88, Thm. 5.1 (1)] that we can limit the optimization in Eq. (3.8) to orthogonal matrices $\mathbf{C}^\top \mathbf{C} = \mathbf{I}$. This can be enforced as a hard constraint through Procrustes analysis, or by adding a soft penalty $E_{\mathrm{reg}}(\mathbf{C}) := \left\|\mathbf{C}^\top \mathbf{C} - \mathbf{I}\right\|_F^2$.

3.2.2.3 Pointwise Map Recovery

For most constraints discussed in Sec. 3.2.2.2, the optimization problem in Eq. (3.8) can be solved in closed form – either through linear least squares or Procrustes analysis. After obtaining an optimal functional map $\mathbf{C}$, we still need to convert it to a pointwise correspondence map $\mathbf{\Pi} \in \{0,1\}^{|\mathcal{V}^{(\mathcal{X})}| \times |\mathcal{V}^{(\mathcal{Y})}|}$. To this end, we consider the following optimization problem

$$\arg\min_{\mathbf{\Pi}\mathbf{1}=\mathbf{1}} \left\|\mathbf{\Phi}_K \mathbf{C}^\top - \mathbf{\Pi}\mathbf{\Psi}_K\right\|_F^2 . \tag{3.9}$$

Minimizing this energy for $\mathbf{\Pi}$ entails a per-vertex nearest neighbor search, which can be solved efficiently with kd-trees.

Overall, obtaining a correspondence map $\mathbf{\Pi}$ for a pair of shapes $\mathcal{X}$ and $\mathcal{Y}$ requires two individual computations where we minimize Eq. (3.8) and Eq. (3.9), respectively. In [88], the authors additionally introduce an iterative refinement technique, where the correspondences obtained in Eq. (3.9) are utilized as landmark features which, in turn, improve the functional maps $\mathbf{C}$ obtained in Eq. (3.8). If the functional maps are restricted to orthogonal matrices $\mathbf{C} \in \mathrm{O}(K)$, then the resulting iterative scheme constitutes a direct extension of the iterative closest point (ICP) algorithm [20] to K dimensions. Thus, the problem of non-rigid correspondence in 3D space is effectively reformalized as a rigid alignment problem in the K-dimensional spectral domain. This approach was further improved in [83] by combining the alternating optimization with a progressive upsampling of the spectral resolution K for iterative map refinement.

3.2.3 Shape Descriptors

Most correspondence methods rely on local descriptor fields $\mathbf{F} \in \mathbb{R}^{|\mathcal{V}| \times L}$ to convey information about geometric features of a given input mesh $(\mathcal{V}, \mathcal{F})$. For each vertex, $\mathbf{F}$ specifies an L-dimensional feature vector. The utility of $\mathbf{F}$ for correspondence algorithms is contingent on a few central properties. For once, $\mathbf{F}$ should be distinctive, allowing us to differentiate between different vertices. We further expect $\mathbf{F}$ to be robust to common shape deformation modes to make reliable predictions across different observed poses. Other considerations concern the spatial localization of features, and the computation cost of obtaining and storing $\mathbf{F}$. Designing appropriate features is a topic of ongoing research and generally requires a trade-off between these different properties. For instance, leveraging fully intrinsic shape properties yields descriptors that are robust to nearly-isometric transformations, but they often lack distinctiveness due to self-similarities and intrinsic symmetries of $\mathcal{X}$.

Two examples of intrinsic local descriptors are the heat kernel signature (HKS) [120] and the wave kernel signature (WKS) [16] features defined as

$$f(\mathbf{p}) := \sum_{k=1}^{\infty} w(\lambda_k)\phi_k(\mathbf{p})^2, \tag{3.10}$$

where (λ_k, ϕ_k) are the LBO eigenpairs defined in Eq. (3.3) and $w : \mathbb{R} \to \mathbb{R}^L$ is a vector valued function. Depending on the exact choice of w [16, 120], we obtain features whose entries correspond to the heat kernel for HKS or to quantum particle distributions on the surface $\mathcal{X}$ for WKS. To obtain a finite expression, we can approximate Eq. (3.10) by using a truncated basis of eigenfunctions $\{\phi_1, \dots, \phi_K\}$. Another popular class of shape descriptors are histogram-based features [103, 123] which count point and outer normal occurrences in specific regions subdividing a fixed sphere $\{\mathbf{q} \in \mathbb{R}^3 | \|\mathbf{p} - \mathbf{q}\|_2 < r\}$ with radius $r > 0$ around a given point $\mathbf{p} \in \mathcal{X}$.

Given local feature descriptors, we can quantify the similarity of points via the standard Euclidean metric $d(\cdot,\cdot)$ in the L-dimensional feature space. If we assume perfectly distinctive and robust features $\mathbf{F}$ and $\mathbf{G}$, the correspondence problem for a pair of input meshes can be addressed through a simple algorithm, based on a nearest neighbor search in the shared feature space $\mathbf{\Pi}_{ij^*} = 1 \iff j^* = \arg\min_j d(\mathbf{f}_i, \mathbf{g}_j)$. In practice, however, the direct feature correspondences are prone to mismatches and local noise, requiring additional refinement and filtering [127].

3.2.4 Representation Learning

Rather than defining shape descriptors manually, many recent works devise data-driven approaches that obtain optimal geometric descriptors through deep feature learning [44, 54, 75, 81, 108]. A central challenge, in this context, is designing appropriate neural network architectures capable of processing triangular mesh data. The umbrella term geometric deep learning [29] summarizes several such techniques appropriate for learning on irregular non-Euclidean domains. In the following, we focus on specific approaches relevant to our own contributions summarized in Part II.

3.2.4.1 Feature Extraction

One of the first building blocks of geometric deep learning pipelines often involves deep feature extraction, specified as a learnable mapping

$$\Theta : (\mathcal{V}, \mathcal{F}) \mapsto \mathbf{F} \in \mathbb{R}^{|\mathcal{V}| \times L}, \tag{3.11}$$

that computes a local descriptor field $\mathbf{F}$. In contrast to hand-crafted descriptors introduced in Sec. 3.2.3, the mapping Θ is parameterized by learnable weights, which are adjusted during training through gradient-based optimization. There are several desirable properties relevant for defining concrete modules Θ, such as robustness and distinctiveness. These considerations are, however, for the most part analogous to our discussion of hand-crafted descriptors in Sec. 3.2.3. Designing efficient and flexible architectures is still a topic of ongoing research. We now provide brief descriptions for several examples of such models.

Optimal Spectral Descriptors The definitions of the HKS and WKS features are based on two concrete choices of vector-valued transfer functions $w : \mathbb{R} \to \mathbb{R}^L$ in Eq. (3.10). While the motivations for both approaches are based on concrete physical processes, there are no guarantees that these choices of w yield descriptors that are optimal for the correspondence task. Instead, [76] proposes to parameterize each entry of w as a linear combination of a B-spline basis, which can be optimized for specific tasks.

Refining Hand-Crafted Features Several works [54, 75, 102] devise set-learning architectures based on refining hand-crafted input descriptor fields, such as the ones defined in Sec. 3.2.3. While different definitions are possible, one simplest example is [75], which refines hand-crafted SHOT features [123] through seven consecutive deep residual layers [57]. Each such layer processes local features independently per point. One advantage of this approach is that the predicted features inherit certain desirable properties from the input descriptors, such as rotation-invariance [123]. On the other hand, the quality of input features varies across different datasets as shown in [82], limiting the stability of learned features when generalizing to unseen test poses.

Charting-based Methods Another class of approaches aim at learning features through local geodesic patch operators [23, 81, 86, 94]. The primary motivation is to define feature refinement on irregular mesh domains analogous to convolution filters in Euclidean space. The obtained patch operators are intrinsic, and often consider a region with a fixed geodesic radius around each point. Possible choices of weighting functions include geodesic polar weights [81], anisotropic heat kernels [23], and generic parameteric kernel functions [86].

Point Cloud Learning A special case of set-learning architectures are point-based models [95, 96, 122], which treat the input vertices $\mathcal{V}$ of a mesh as an unstructured point cloud. While such models do not utilize all available information – omitting the connectivity $\mathcal{F}$ – they are nevertheless common in the shape correspondence literature due to their simplicity and flexibility [44, 53, 62, 80, 107]. Notably, the obtained features are, in general, not invariant to the rigid pose, necessitating careful data augmentation.

Heat Diffusion A recent approach yielding state-of-the-art results in correspondence problems is the DiffusionNet [108] architecture. It is based on the heat-equation on the surface $\mathcal{X}$, defined as

$$\frac{\partial}{\partial t}f(\mathbf{p},t)=\Delta_{\mathcal{X}}f(\mathbf{p},t). \tag{3.12}$$

Intuitively, simulating the heat dispersion process results in local information flow, which can be leveraged for feature refinement. Specifically, the updated feature fields $f(\cdot,t)$ are obtained as solutions of Eq. (3.12) for an input feature field $f(\cdot,0)$ and time intervals $t>0$. The support of the resulting filters depends on the evaluation time t, which is learned adaptively. The heat diffusion is further combined with an anisotropic spatial gradient term and a pointwise multilayer perceptron (MLP) refinement layer. The resulting network architecture is robust to different meshing densities, and can even be applied to different input representation such as point clouds, through an alternative discretization of the Laplacian [109].

3.2.4.2 Deep Functional Maps

The deep functional maps framework [75] is a dominant paradigm in recent shape correspondence approaches. The main idea is to combine a learnable feature extractor with a differentiable functional maps layer. For a pair of input meshes $\left(\mathcal{V}^{(\mathcal{X})},\mathcal{F}^{(\mathcal{X})}\right)$ and $\left(\mathcal{V}^{(\mathcal{Y})},\mathcal{F}^{(\mathcal{Y})}\right)$, most existing models adopt variants of the following procedure.

1. Compute local feature embeddings $\mathbf{F}:=\Theta\left(\mathcal{V}^{(\mathcal{X})},\mathcal{F}^{(\mathcal{X})}\right)$ and $\mathbf{G}:=\Theta\left(\mathcal{V}^{(\mathcal{Y})},\mathcal{F}^{(\mathcal{Y})}\right)$ through Eq. (3.11), in a siamese manner.

2. Project the obtained descriptors onto the truncated eigenbases to convert them to the spectral domain $\mathbf{A}:=\mathbf{\Phi}_K^{\dagger}\mathbf{F}$ and $\mathbf{B}:=\mathbf{\Psi}_K^{\dagger}\mathbf{G}$.

3. Compute the functional map $\mathbf{C}$ through Eq. (3.8).

The inner optimization in step 3. requires solving a linear system, resulting in a differentiable operation. Overall, the steps 1.-3. specify an end-to-end trainable model, mapping the two input meshes to a functional map $\mathbf{C} \in \mathbb{R}^{K \times K}$. Existing works in the literature vary in the choice of the concrete feature extractor Θ, as discussed in Sec. 3.2.4.1, as well as the training loss function. Supervised loss functions measure the discrepancy between the model's predictions and the ground truth correspondence maps [44, 75]. Unsupervised approaches, on the other hand, leverage geometric priors as training losses, promoting the preservation of geodesic distances [54] or structural properties of $\mathbf{C}$ [32, 102]. During test time, the predicted functional map $\mathbf{C}$ can be converted to a pointwise correspondence map $\mathbf{\Pi}$ through Eq. (3.9).

3.2.5 Multi-Shape Correspondence

The objective of multi-shape correspondence is to jointly compute correspondence maps $\mathbf{\Pi}^{(i,j)} \in \{0,1\}^{|\mathcal{V}^{(i)}| \times |\mathcal{V}^{(j)}|}$ between all pairs of poses in a collection of $N \in \mathbb{N}$ shapes $\{(\mathcal{V}^{(1)}, \mathcal{F}^{(1)}), \dots, (\mathcal{V}^{(N)}, \mathcal{F}^{(N)})\}$. Compared to standard pairwise matching $N = 2$, the predicted correspondences are required to be compatible with each other, defined as a cycle-consistency constraint

$$\mathbf{\Pi}^{(i_1,i_2)}\mathbf{\Pi}^{(i_2,i_3)} \dots \mathbf{\Pi}^{(i_{K-1},i_K)} \approx \mathbf{\Pi}^{(i_1,i_K)}, \quad \text{where } i_1, \dots, i_K \in \{1, \dots, N\}. \tag{3.13}$$

Intuitively, we expect that mapping a given point repeatedly across different shapes yields comparable correspondences for fixed endpoints $\mathcal{V}^{(i_1)}$ and $\mathcal{V}^{(i_K)}$, irrespective of the intermediate sequence. One straightforward approach of enforcing cycle-consistency in learning frameworks is to add a soft penalty based on Eq. (3.13). Since the number of possible sequences increases exponentially in the path length, this approach is often limited to cycles up to a certain length, *e.g.*, three-cycles.

A prevalent solution to the multi-matching problem is modeling correspondences through canonical embeddings [18, 31, 48, 62, 63]. To this end, we define auxiliary maps $\mathbf{\Pi}^{(i)} \in \{0,1\}^{|\mathcal{V}^{(i)}| \times |\bar{\mathcal{V}}|}$ from each pose $(\mathcal{V}^{(i)}, \mathcal{F}^{(i)})$ to an implicit universe shape $(\bar{\mathcal{V}}, \bar{\mathcal{F}})$. The pairwise correspondences can then be extracted by concatenating the respective shape-to-universe maps $\mathbf{\Pi}^{(i,j)} := \mathbf{\Pi}^{(i)} \left(\mathbf{\Pi}^{(j)}\right)^\top$. For simplicity, we assume here that the correspondence maps $\mathbf{\Pi}^{(i)}$ are permutation matrices, where all considered shapes have the same number of vertices. Under these simplified conditions, cycle-consistency is guaranteed for the pairwise maps $\mathbf{\Pi}^{(i,j)}$ in the following sense:

Proposition 3. *The cycle-consistency constraint in Eq.* (3.13) *holds exactly for any sequence of poses* $i_1, \dots, i_K \in \{1, \dots, N\}$, *iff all pairwise matches can be decomposed into shape-to-universe maps* $\mathbf{\Pi}^{(i,j)} = \mathbf{\Pi}^{(i)} \left(\mathbf{\Pi}^{(j)}\right)^\top$.

Refer to [64, Prop. 1] for a formal proof. In the literature, there are different variants of this approach, allowing for partial permutation matrices [18, 31] or defining canonical functional maps [48, 62, 63].

3.3 Shape Interpolation

We define shape interpolation as the task of determining a family of intermediate poses between two given 3D shapes, continuously deforming one into the other. It is closely related to the challenge of shape correspondence; in both cases, we model meaningful semantic relationships between pairs of shapes $\mathcal{X}$ and $\mathcal{Y}$. While correspondences are specified by an intrinsic point-to-point map $\pi : \mathcal{X} \to \mathcal{Y}$, interpolation sequences consider the underlying shape dynamics to obtain plausible extrinsic pose predictions.

3.3.1 Problem Formulation

For a given pair of 3D shape poses $\mathcal{X}, \mathcal{Y} \subset \mathbb{R}^3$, we define shape interpolation as predicting a sequence of poses $\mathcal{Z}(t) \subset \mathbb{R}^3$ for the time interval $t \in [0, 1]$, respecting the boundary conditions $\mathcal{Z}(0) = \mathcal{X}$ and $\mathcal{Z}(1) = \mathcal{Y}$. Since such sequences continuously morph $\mathcal{X}$ into $\mathcal{Y}$, we often characterize $\mathcal{Z}$ in terms of a displacement field $\varphi : [0, 1] \times \mathcal{X} \to \mathbb{R}^3$ deforming the first input shape $\mathcal{Z}(t) := \varphi(t, \mathcal{X})$. In this formulation, the first boundary condition simplifies to $\varphi(0, \cdot) = \mathrm{Id}(\cdot)$. Individual deformations $\varphi(t, \cdot)$ are defined as diffeomorphic maps, specifying dense correspondences between $\mathcal{X}$ and $\mathcal{Z}(t)$. Considering the special case of $t = 1$, this formulation also includes the correspondence map $\varphi(1, \cdot)$ between $\mathcal{X}$ and $\mathcal{Y}$, confirming strong synergies between the two tasks. We further require that the resulting trajectory of each initial vertex $\mathbf{p} \in \mathcal{X}$ is a smooth differentiable path, *i.e.*, $\varphi(\cdot, \mathbf{p}) \in C^\infty([0, 1], \mathbb{R}^3)$.

3.3.2 Eulerian Flow

For a given displayement field φ, we obtain the corresponding extrinsic Eulerian flow field as [60]

$$v(t, \mathbf{p}) := \frac{\partial}{\partial t} \varphi(t, \varphi^{-1}(t, \mathbf{p})), \tag{3.14}$$

where $\varphi^{-1}(t, \cdot) : \mathcal{Z}(t) \to \mathcal{X}$ denotes the inverse of the diffeomorpism φ at time t. The resulting field v denotes the direction of the time-dependent flow for each surface point $\mathbf{p} \in \mathcal{Z}(t)$. Vice versa, we can define a concrete parameterization of φ via the extrinsic volumetric flow $v : [0, 1] \times \mathbb{R}^3 \to \mathbb{R}^3$. We then recover the displacement fields φ from v

through time integration of each individual initial vertex $\mathbf{p} \in \mathcal{X}$

$$\begin{cases} \frac{\partial}{\partial t}\varphi(t, \mathbf{p}) = v(t, \varphi(t, \mathbf{p})), \\ \varphi(0, \mathbf{p}) = \mathbf{p}. \end{cases} \tag{3.15}$$

In this formulation, we obtain an interpolation sequence $\mathcal{Z}(t)$ by optimizing for the flow field $v(t, \mathbf{p})$. One advantage of this approach is that we can incorporate certain assumptions about the resulting shape deformation, such as volume preservation, in terms of mathematical properties of the flow v [35]. The extrinsic flow formulation has further been employed in learning frameworks for 4D reconstruction [87] and modeling object dynamics [66], where v is parameterized by a neural network. In the time-discrete setting, we can solve the ordinary differential equation in Eq. (3.15) through a numeric integration method, such as the explicit Euler method, or the Runge-Kutta method [49].

3.3.3 Geodesics in Shape Space

We now consider a pair of discrete meshes $\left(\mathcal{V}^{(\mathcal{X})}, \mathcal{F}\right)$ and $\left(\mathcal{V}^{(\mathcal{Y})}, \mathcal{F}\right)$ with an identical number of vertices $D := \left|\mathcal{V}^{(\mathcal{X})}\right| = \left|\mathcal{V}^{(\mathcal{Y})}\right|$ and compatible triangulation $\mathcal{F}$. We further assume that the vertices are in a canonical ordering, such that the ground truth correspondences are the identity map $\mathbf{\Pi} = \mathbf{I}$. In matrix notation, we then define the $3 \cdot D$ dimensional *shape space* corresponding to the triangles $\mathcal{F}$ as [68]

$$\mathcal{S}_{\mathcal{F}} := \left\{(\mathbf{V}, \mathcal{F}) | \mathbf{V} \in \mathbb{R}^{D \times 3}\right\}. \tag{3.16}$$

Interpreting individual poses $\mathbf{V}$ as points in the space of vertex embeddings, the shape space $(\mathcal{S}_{\mathcal{F}}, g)$ specifies a $3 \cdot D$ dimensional Riemannian manifold [58, 59, 61, 65] with the metric tensor $g_{\mathbf{V}} : \mathbb{R}^{D \times 3} \times \mathbb{R}^{D \times 3} \to \mathbb{R}$ at each point $(\mathbf{V}, \mathcal{F}) \in \mathcal{S}_{\mathcal{F}}$. There are several possible choices for g depending on the specific deformation model [25, 50, 113, 117]. For an exact derivation of g for the thin shell surface model, refer to [58, Thm. 1].

Considering piecewise linear displacement fields φ per triangle, we obtain interpolation sequences that are fully specified by the vertex displacements $\mathbf{V}(t) := \varphi(t, \mathbf{V}^{(\mathcal{X})})$. We further quantify the path length of $\{(\mathbf{V}(t), \mathcal{F}) \in \mathcal{S}_{\mathcal{F}} | t \in [0, 1]\}$ on the shape space manifold as [119, Ch. 9]

$$\Lambda\left(\mathbf{V}(t)|_{t \in [0,1]}\right) := \int_0^1 \sqrt{g_{\mathbf{V}(t)}\left(\dot{\mathbf{V}}(t), \dot{\mathbf{V}}(t)\right)} \mathrm{d}t, \tag{3.17}$$

where $\dot{\mathbf{V}}(t) := \frac{\mathrm{d}}{\mathrm{d}t}\mathbf{V}(t)$ denotes the local flow at $\mathbf{V}(t)$. We then obtain *geodesics* on the

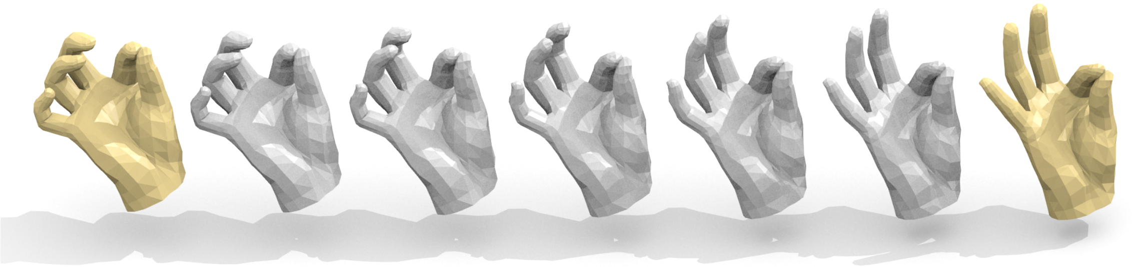


Figure 3.3: The predicted interpolation sequence (gray, 2nd-6th) for a sample pair of shapes from the MANO [101] dataset (yellow, left and right), obtained by solving Eq. (3.19) with the as-rigid-as-possible [117] deformation model $\mathcal{W}_{\text{arap}}$.

shape space manifold $\mathcal{S}_{\mathcal{F}}$ as curves that minimize the path length

$$\underset{\mathbf{V}(t)}{\arg\min}\ \Lambda\left(\mathbf{V}(t)|_{t\in[0,1]}\right) \quad \text{s.t.} \quad \mathbf{V}(0) \overset{!}{=} \mathbf{V}^{(\mathcal{X})}, \mathbf{V}(1) \overset{!}{=} \mathbf{V}^{(\mathcal{Y})}, \tag{3.18}$$

where the boundary conditions at times $t \in \{0, 1\}$ are specified by the two input shape poses $\left(\mathbf{V}^{(\mathcal{X})}, \mathcal{F}\right), \left(\mathbf{V}^{(\mathcal{Y})}, \mathcal{F}\right) \in \mathcal{S}_{\mathcal{F}}$.

3.3.4 Surface Deformation Models

For practical implementation, we introduce a temporal discretization of interpolation sequences $\mathbf{V}_k := \mathbf{V}\left(\frac{k}{K}\right)$ for an equidistant set of timesteps $k = 0, \dots, K$. Based on Eq. (3.17), we then obtain interpolation sequences as solutions of [60, 68]

$$\underset{(\mathbf{V}_0,\dots,\mathbf{V}_K)}{\arg\min} \sum_{k=0}^{K-1} \mathcal{W}\left(\mathbf{V}_k, \mathbf{V}_{k+1}\right) \quad \text{s.t.} \quad \mathbf{V}_0 \overset{!}{=} \mathbf{V}^{(\mathcal{X})}, \mathbf{V}_K \overset{!}{=} \mathbf{V}^{(\mathcal{Y})}. \tag{3.19}$$

There are several possible choices for the concrete deformation model $\mathcal{W}$ in the literature [25, 50, 113, 117]. We introduce the as-rigid-as-possible deformation model [117], which is most relevant to our own contributions summarized in Part II, defined as

$$\mathcal{W}_{\text{arap}}\left(\mathbf{V}_k, \mathbf{V}_{k+1}\right) := \frac{1}{2} \sum_{i=1}^{D} \min_{\mathbf{R}_i \in \mathrm{SO}(3)} \sum_{j \in \mathcal{N}(i)} \left\| \mathbf{R}_i(\mathbf{v}_{k,j} - \mathbf{v}_{k,i}) - (\mathbf{v}_{k+1,j} - \mathbf{v}_{k+1,i}) \right\|_2^2, \tag{3.20}$$

where $\mathcal{N}(i)$ denotes the set of neighboring vertices $\mathbf{v}_j$ of $\mathbf{v}_i$ in the graph induced by the triangulation $\mathcal{F}$. In Fig. 3.3, we show a sample interpolation sequence, obtained by minimizing Eq. (3.19) based on the as-rigid-as-possible deformation model from Eq. (3.20), using a standard first-order optimizer.

3.3.5 Variational Autoencoder Formulation

Autoencoders offer a convenient formulation of shape interpolation for a collection of input poses. The key idea is to learn a compressed representation $\mathbf{z} := \mathcal{E}(\mathbf{V}, \mathcal{F}) \in \mathbb{R}^L$ for any shape in a considered shape space $(\mathbf{V}, \mathcal{F}) \in \mathcal{S}_{\mathcal{F}}$. For the concrete choice of encoder network $\mathcal{E} : \mathcal{S}_{\mathcal{F}} \to \mathbb{R}^L$, we can employ any of the local feature extraction architectures described in Sec. 3.2.4.1, in combination with a global pooling operation over the vertex dimension such as max pooling, average pooling, or attention [126].

The corresponding decoder architecture $\mathcal{D} : \mathbb{R}^L \to \mathcal{S}_{\mathcal{F}}$ then generates an output shape $(\hat{\mathbf{V}}, \hat{\mathcal{F}}) := \mathcal{D}(\mathbf{z}) \in \mathcal{S}_{\mathcal{F}}$ from the latent code representation $\mathbf{z} \in \mathbb{R}^L$. Devising generative models, *i.e.*, architectures capable of synthesizing novel geometric objects, is a topic of ongoing research with output formats ranging from voxel grids [36, 55], template mesh deformation [53, 66, 129], skinning-based deformable models [74, 78, 101, 139] and point clouds [11, 47, 136] to neural implicit representations [51, 85, 91]. Since we assume a fixed, global meshing $\mathcal{F}$, a simple choice in this context is modeling $\mathcal{D}$ as a fully connected multilayer perceptron (MLP) network with $3 \cdot D$ output dimensions, as well as setting $\hat{\mathcal{F}} := \mathcal{F}$.

To train the encoder $\mathcal{E}$ and decoder $\mathcal{D}$ networks for a given collection of input meshes $\left\{\left(\mathbf{V}^{(1)}, \mathcal{F}\right), \dots, \left(\mathbf{V}^{(N)}, \mathcal{F}\right)\right\} \subset \mathcal{S}_{\mathcal{F}}$, we devise multiple training loss signals. For once, we enforce that $\mathcal{D} \circ \mathcal{E} \approx \mathrm{Id}$ with a reconstruction loss

$$\ell_{\mathrm{rec}}(\mathbf{V}, \mathcal{F}) := \left\|\hat{\mathbf{V}} - \mathbf{V}\right\|_F^2, \text{ where } (\hat{\mathbf{V}}, \mathcal{F}) := \mathcal{D}\left(\mathcal{E}(\mathbf{V}, \mathcal{F})\right). \tag{3.21}$$

The learned latent space yields a compressed global representation $L \ll 3 \cdot D$ which is to a certain degree lossy, *i.e.*, there exist input shape poses $(\mathbf{V}, \mathcal{F}) \in \mathcal{S}_{\mathcal{F}}$ such that $\ell_{\mathrm{rec}}(\mathbf{V}, \mathcal{F}) \neq 0$. Beyond the dimensionality reduction, variational autoencoders [69] further regularize the learned latent space to endow it with specific algebraic properties. For instance, to learn shape spaces that are useful for shape interpolation, we can enforce an interpolation loss [37]

$$\ell_{\mathrm{interp}}(\mathbf{V}^{(i)}, \mathbf{V}^{(j)}; \mathcal{F}) := \int_0^1 \left\|\mathbf{D}_\alpha - \left((1-\alpha)\hat{\mathbf{D}}^{(i)} + \alpha\hat{\mathbf{D}}^{(j)}\right)\right\|_F^2 \mathrm{d}\alpha, \tag{3.22}$$

where $\hat{\mathbf{D}}^{(i)}, \hat{\mathbf{D}}^{(j)}, \mathbf{D}_\alpha \in \mathbb{R}^{D \times D}$ denote the pairwise geodesic distance matrices between $(\hat{\mathbf{V}}^{(i)}, \mathcal{F})$, $(\hat{\mathbf{V}}^{(j)}, \mathcal{F})$, and $(\mathbf{V}_\alpha, \mathcal{F})$, respectively, and where $\mathbf{V}_\alpha := \mathcal{D}\big((1-\alpha)\mathbf{z}^{(i)} + \alpha\mathbf{z}^{(j)}\big)$. Intuitively, minimizing the loss ℓ_{interp} enforces geodesic metric interpolation along straight paths in the learned latent shape space. Hence, we can obtain continuous interpolation sequences by decoding linear interpolation paths in the latent encoding space $\mathcal{Z}(\alpha) := \mathcal{D}\left((1-\alpha)\mathcal{E}\left(\mathbf{V}^{(\mathcal{X})}, \mathcal{F}\right) + \alpha\mathcal{E}\left(\mathbf{V}^{(\mathcal{Y})}, \mathcal{F}\right)\right)$ for any $\alpha \in [0, 1]$.

3.4 Optimal Transport

The framework of optimal transport involves techniques to establish geometrically meaningful distance measures between arbitrary probability distributions. This has direct implications for shape analysis, where optimal transport has been successfully applied for both correspondence [90, 110, 137] and interpolation tasks [40, 115]. Similar methodologies lie at the core of several of our own contributions in Part II.

3.4.1 Basic Definition

Optimal transport describes the problem of computing distances between two generic measure spaces (X,μ) and (Y,ν) [128]. In this context, we assume that μ and ν are probability measures, *i.e.*, $\mu(X)=\nu(Y)=1$ and $\mu(A),\nu(B)\in[0,1]$ for any measurable subsets $A\subset X$ and $B\subset Y$. We further require a measurable cost function c mapping from $X\times Y$ to $(-\infty,\infty]$ that defines the transportation cost between the two domains. In this context, values $c(A,B)=\infty$ denote that transporting from A to B is precluded. A *transportation plan* is defined as a probability measure π on the product space of the two input domains $X\times Y$. We then define Kantorovich's formulation of optimal transport as solutions to the following optimization problem [128]

$$d(\mu,\nu):=\inf_{\pi\in\Pi(\mu,\nu)}\int_{X\times Y}c(x,y)\mathrm{d}\pi(x,y). \tag{3.23}$$

The set of admissible transportation plans is further specified as

$$\Pi(\mu,\nu):=\left\{\pi\text{ prob. measure on }X\times Y\middle|\int_X\mathrm{d}\pi(x,y)=\mathrm{d}\nu(y),\int_Y\mathrm{d}\pi(x,y)=\mathrm{d}\mu(x)\right\}. \tag{3.24}$$

The optimal transport plans determined in Eq. (3.23) indicate, how much probability mass $\pi(A,B)\in[0,1]$ is transported between measurable sets $A\subset X$ and $B\subset Y$. Constraining π to $\Pi(\mu,\nu)$ further ensures, that the total probability mass transported from X to Y at each point matches the input marginal distributions. In Fig. 3.4, we provide a toy example visualization of the resulting plan π where we consider two input marginals μ,ν defined on $X=Y=[0,1]$.

3.4.2 Application to Shape Analysis

When applying the optimal transport framework to shape analysis tasks, we typically model X and Y as pairs of input surfaces in $\mathbb{R}^3$. While it is possible to specify the optimization problem in Eq. (3.23) for continuous surfaces, we focus on the discrete

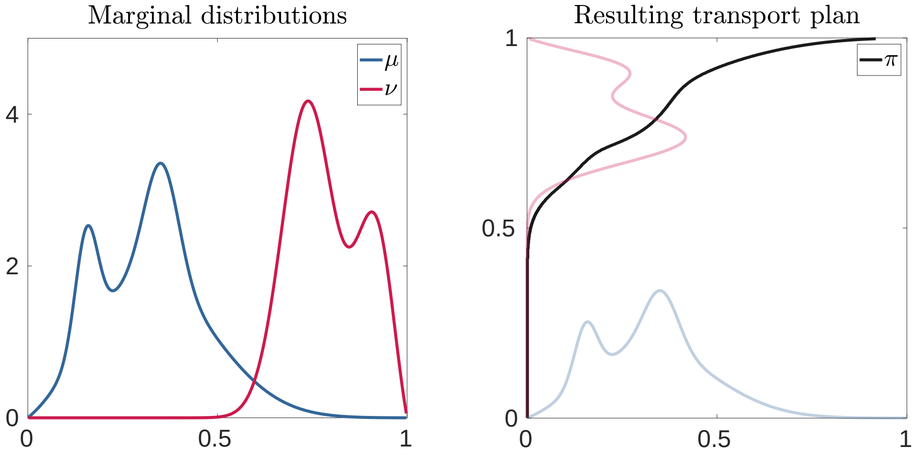


Figure 3.4: (left) We provide a visualization of two continuous marginal distributions μ and ν, defined on $X = Y = [0,1]$. We then solve the optimal transport problem defined in Eq. (3.23) with a squared Euclidean cost metric $c(x,y) := d(x,y)^2$. (right) The resulting optimal transportation plan π describes a curve in the product space $X \times Y$. For further details, refer to [112].

setting common in concrete practical algorithms. To this end, we formulate the optimal transport problem between the sets of vertices $X := \mathcal{V}^{(\mathcal{X})}$ and $Y := \mathcal{V}^{(\mathcal{Y})}$ with a cost matrix $\mathbf{C} := \mathbb{R}^{|\mathcal{V}^{(\mathcal{X})}|\times|\mathcal{V}^{(\mathcal{Y})}|}$. The unknown transport plans $\mathbf{P}$ are further contained in the transport polytope, defined as

$$\bar{\Pi}(\mathbf{a},\mathbf{b}) := \left\{\mathbf{P} \in [0,1]^{|\mathcal{V}^{(\mathcal{X})}|\times|\mathcal{V}^{(\mathcal{Y})}|} \middle| \mathbf{P}\mathbf{1} = \mathbf{a}, \mathbf{P}^\top\mathbf{1} = \mathbf{b}\right\}, \tag{3.25}$$

where $\mathbf{a} \in [0,1]^{|\mathcal{V}^{(\mathcal{X})}|}$ and $\mathbf{b} \in [0,1]^{|\mathcal{V}^{(\mathcal{Y})}|}$ are the input marginals, specifying discrete probability distributions $\mathbf{a}^\top\mathbf{1} = \mathbf{b}^\top\mathbf{1} = 1$. The discrete optimal transport problem is then defined as

$$d(\mathbf{a},\mathbf{b};\mathbf{C}) := \min_{\mathbf{P}\in\bar{\Pi}(\mathbf{a},\mathbf{b})} \langle\mathbf{P},\mathbf{C}\rangle_F\,. \tag{3.26}$$

There are several potential applications in shape analysis, depending on the concrete choices of cost matrices $\mathbf{C}$ and marginals $\mathbf{a}$, $\mathbf{b}$. We provide two examples in the following.

Shape Correspondence To model correspondences between $\mathcal{V}^{(\mathcal{X})}$ and $\mathcal{V}^{(\mathcal{Y})}$, a common approach is to define $\mathbf{C}$ as the squared Euclidean feature distance matrix

$$\mathbf{C}_{i,j} := \|\mathbf{F}_{i,:} - \mathbf{G}_{j,:}\|_2^2, \tag{3.27}$$

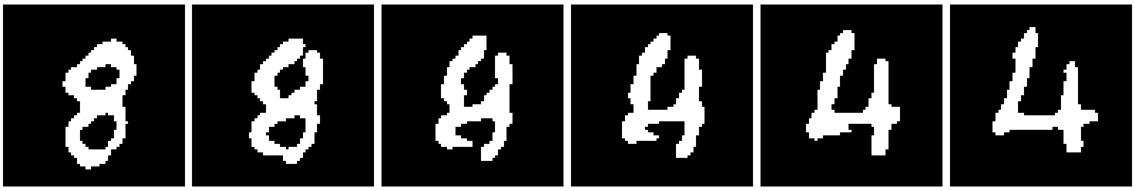


Figure 3.5: An example interpolation (2nd-5th sample) of two binary images from the MNIST dataset (left, right), obtained by minimizing the Wasserstein barycenter objective in Eq. (3.28).

where $\mathbf{F} \in \mathbb{R}^{|\mathcal{V}^{(\mathcal{X})}|\times L}$ and $\mathbf{G} \in \mathbb{R}^{|\mathcal{V}^{(\mathcal{Y})}|\times L}$ are generic L-dimensional feature embeddings per input vertex. Under these conditions, Eq. (3.26) is often referred to as the Wasserstein 2-distance in feature space. The resulting optimal transportation plan $\mathbf{P} \in [0,1]^{|\mathcal{V}^{(\mathcal{X})}|\times|\mathcal{V}^{(\mathcal{Y})}|}$ further specifies a soft correspondence matrix. Informally, we can interpret $\mathbf{P}_{i,j} \in [0,1]$ as the probability of matching the vertices $\mathbf{v}_i^{(\mathcal{X})}$ and $\mathbf{v}_j^{(\mathcal{Y})}$.

Although different vertex weightings are possible, for convenience, we often choose uniform marginal distributions $\mathbf{a} := \frac{1}{|\mathcal{V}^{(\mathcal{X})}|}\mathbf{1}_{|\mathcal{V}^{(\mathcal{X})}|}$ and $\mathbf{b} := \frac{1}{|\mathcal{V}^{(\mathcal{Y})}|}\mathbf{1}_{|\mathcal{V}^{(\mathcal{Y})}|}$. Moreover, if the two input shapes have an equal number of vertices $|\mathcal{V}^{(\mathcal{X})}| = |\mathcal{V}^{(\mathcal{Y})}|$, the optimization in Eq. (3.26) simplifies to the special case of a linear assignment problem. In this setting, there exist explicit algorithms with a polynomial runtime complexity of $\mathcal{O}(|\mathcal{V}^{(\mathcal{X})}|^3)$, such as the auction algorithm [19] or the Hungarian algorithm [70].

Barycentric Interpolation Optimal transport defines geometrically meaningful distance measures between probability distributions. This makes it viable as a shape deformation prior, related to our discussion in Sec. 3.3.4. In the literature, this is commonly referred to as barycentric interpolation [14, 40, 115]. Instead of surface-based representations, we consider explicit grid-based representations in 2D or 3D space (*i.e.* pixels or voxels), where individual entries of the marginals $\mathbf{a}$ indicate either high ≈ 1 or low occupancy ≈ 0 values of the i-th grid point. In this setting, we can compute intermediate shapes between a collection of input poses $\mathbf{b}_1, \dots, \mathbf{b}_N$ as

$$\underset{\mathbf{a}\in[0,1]^D}{\arg\min} \sum_{i=1}^{N} w_i d(\mathbf{a}, \mathbf{b}_i; \mathbf{C}) \quad \text{s.t.} \quad \mathbf{a}^\top \mathbf{1} = 1, \tag{3.28}$$

where $\mathbf{C}$ is a Euclidean distance matrix similar to Eq. (3.27), defined between the grid point center positions. The scalars $w_i \geq 0$ define a weighted average of the input poses $\mathbf{b}_i$. In Fig. 3.5, we provide an example interpolation for two binary images $\mathbf{b}_1, \mathbf{b}_2$ from the MNIST dataset [73], where we choose interpolation weights $(w_1, w_2) := (1-\alpha, \alpha)$ for $\alpha \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$. Intuitively, the deformation prior $d(\cdot,\cdot;\mathbf{C})$ yields average shapes that require minimal transportation cost in terms of Euclidean distances, but do not minimize local surface distortions such as [50]. Hence, while this definition allows for compelling interpolation results for 2D and 3D objects [115], the scope is different compared to Sec. 3.3, focusing mostly on non-deformable object categories.

3.4.3 Entropy Regularization

To make the optimization problem in Eq. (3.26) computationally tractable, a common solution involves additional regularization terms. For the popular entropy regularizer, this is defined as [39]

$$d(\mathbf{a},\mathbf{b};\mathbf{C}) := \min_{\mathbf{P}\in\bar{\Pi}(\mathbf{a},\mathbf{b})} \langle \mathbf{P},\mathbf{C}\rangle_F - \tau h(\mathbf{P}), \tag{3.29}$$

where the entropy function $h(z) := -z\,(\log(z)-1)$ is applied to the unknown transport plan $\mathbf{P} \in [0,1]^{|\mathcal{V}^{(\mathcal{X})}|\times|\mathcal{V}^{(\mathcal{Y})}|}$ per entry. Intuitively, this term h promotes values in the interior of the transport polytope $\mathbf{P}\in\bar{\Pi}(\mathbf{a},\mathbf{b})$. Since $\frac{\mathrm{d}}{\mathrm{d}z}\left(-h(z)\right)|_{z=0} = -\infty$, any value $\mathbf{P}_{i,j} = 0$ incurs a steep cost, encouraging non-binary entries $\mathbf{P}_{i,j} \notin \{0,1\}$. For any $\tau > 0$, it has been shown in [39] that the regularized objective in Eq. (3.29) can be optimized efficiently with Sinkhorn's algorithm, specified as

Algorithm 1: Sinkhorn's algorithm

```
   Input: Cost matrix $\mathbf{C}$, marginals $\mathbf{a},\mathbf{b}$.
   Output: Resulting (regularized) transport plan $\mathbf{P}$.
1  Initialize $\mathbf{u}=\mathbf{1}_{|\mathcal{V}^{(\mathcal{X})}|}$ and $\mathbf{v}=\mathbf{1}_{|\mathcal{V}^{(\mathcal{Y})}|}$ as vectors of ones ;
2  Compute $\mathbf{K}=\exp(-\mathbf{C}/\tau)$;
3  while not converged do
4      $\mathbf{v}\leftarrow\mathbf{b}\oslash(\mathbf{K}^\top\mathbf{u})$;
5      $\mathbf{u}\leftarrow\mathbf{a}\oslash(\mathbf{K}\mathbf{v})$;
6  end while
7  $\mathbf{P}=\mathrm{diag}(\mathbf{u})\mathbf{K}\mathrm{diag}(\mathbf{v})$;
8  return $\mathbf{P}$;
```

The main computation steps of the algorithm involve an iterative scheme of alternating row-wise and column-wise normalizations of $\mathrm{diag}(\mathbf{u})\mathbf{K}\mathrm{diag}(\mathbf{v})$, denoted with the Hadamard division $\oslash$. There are different potential choices of stopping criteria in Line 3, refer to [39, Ch. 5] for examples. Besides yielding a simple and efficient explicit algorithm, the regularized objective in Eq. (3.29) further makes the solution operator $(\mathbf{C},\mathbf{a},\mathbf{b})\mapsto\mathbf{P}$ differentiable. Hence, it allows for gradient-based optimization, making it possible to incorporate it as a building block in deep neural networks to model soft permutations [12, 41, 46, 84, 105] or for learning correspondences [77, 90, 106, 135, 137].

Part II

Own Publications

Chapter 4

Divergence-Free Shape Correspondence by Deformation

Computing correspondence between pairs of 3D shapes generally requires that we specify the underlying class of transformations. Making such additional assumptions often results in more compact parameterizations, *e.g.*, six degrees of freedom for rigid transformations. This further helps constrain the optimization and ultimately yields more accurate results. A popular class of non-rigid deformations are nearly-isometric maps, which preserve the intrinsic surface distance metric. In this case, we often consider the different non-rigid poses of the same deformable object. For most real-world examples, such pose differences can be explained through explicit extrinsic object dynamics. We include this assumption in our approach by modeling explicit deformation fields $v : \mathbb{R}^3 \to \mathbb{R}^3$ for two given input shapes $\mathcal{X}$ and $\mathcal{Y}$. We then continuously deform the initial pose $\mathcal{X}$ into $\mathcal{Y}$ through time-integration of v. Besides the shape alignment, we also obtain a pose interpolation, *i.e.*, a sequence of intermediate poses. One advantage of our framework is that properties about the underlying transformation can be specified directly in terms of v. For instance, we obtain volume preserving maps by requiring that the deformation fields are divergence-free $\nabla \cdot v = 0$. Beyond the total volume of the input shape $\mathcal{X}$, the local volume of any possible domain $\Omega \subset \mathbb{R}^3$ is preserved, leading to strong structural preservation throughout the predicted pose sequence. To parameterize the deformation field v, we leverage Helmholtz's theorem, and decompose the associated potential fields as a linear combination of a finite number of Fourier basis functions. This representation has a natural ordering from low to high frequencies and results in spatially continuous fields v with an explicit analytical expression. For a given pair of poses, we optimize for v by using an expectation-maximization algorithm alternating between computing the optimal correspondence map and deformation field. In our experiments, we obtain compelling results on several shape benchmarks, both in terms of correspondence accuracy and qualitative interpolation results.

Individual contributions

Leading role in realizing the scientific project.

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

Copyright

Marvin Eisenberger, Zorah Lähner, and Daniel Cremers

Divergence-Free Shape Correspondence by Deformation

2019 Eurographics Symposium on Geometry Processing (SGP), Computer Graphics Forum

DOI: 10.1111/cgf.13785

JOHN WILEY AND SONS LICENSE
TERMS AND CONDITIONS

Dec 04, 2023

This Agreement between Mr. Marvin Eisenberger ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

License Number	5681821190966
License date	Dec 04, 2023
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Computer Graphics Forum
Licensed Content Title	Divergence-Free Shape Correspondence by Deformation
Licensed Content Author	D. Cremers, Z. Lähner, M. Eisenberger
Licensed Content Date	Aug 12, 2019
Licensed Content Volume	38
Licensed Content Issue	5
Licensed Content Pages	12
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Electronic

Portion	Full article
Will you be translating?	No
Title of new work	Fully Automatic Deformable 3D Shape Correspondence and Interpolation
Institution name	Technical University of Munich
Expected presentation date	May 2024
Order reference number	license_divergence_free
Requestor Location	Mr. Marvin Eisenberger Muehlfeldweg 22 Garching, Bavaria 85748 Germany Attn: Mr. Marvin Eisenberger
Publisher Tax ID	EU826007151
Billing Type	Invoice
Billing Address	Mr. Marvin Eisenberger Muehlfeldweg 22 Garching, Germany 85748 Attn: Mr. Marvin Eisenberger
Total	0.00 USD
Terms and Conditions	

TERMS AND CONDITIONS

This copyrighted material is owned by or exclusively licensed to John Wiley & Sons, Inc. or one of its group companies (each a"Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at http://myaccount.copyright.com).

Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.

- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley Materials for the purpose specified in the licensing process. This license, **and any CONTENT (PDF or image file) purchased as part of your order,** is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this license must be completed within two years of the date of the grant of this license (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.

- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner.**For STM Signatory Publishers clearing permission under the terms of the STM Permissions Guidelines only, the terms of the license are extended to include subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted figures or extracts,** You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.

- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto

- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE

MATERIALS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU.

- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.

- You shall indemnify, defend and hold harmless WILEY, its Licensors and their respective directors, officers, agents and employees, from and against any actual or threatened claims, demands, causes of action or proceedings arising from any breach of this Agreement by you.

- IN NO EVENT SHALL WILEY OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, PROVISIONING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

- Should any provision of this Agreement be held by a court of competent jurisdiction to be illegal, invalid, or unenforceable, that provision shall be deemed amended to achieve as nearly as possible the same economic effect as the original provision, and the legality, validity and enforceability of the remaining provisions of this Agreement shall not be affected or impaired thereby.

- The failure of either party to enforce any term or condition of this Agreement shall not constitute a waiver of either party's right to enforce each and every term and condition of this Agreement. No breach under this agreement shall be deemed waived or excused by either party unless such waiver or consent is in writing signed by the party granting such waiver or consent. The waiver by or consent of a party to a breach of any provision of this Agreement shall not operate or be construed as a waiver of or consent to any other or subsequent breach by such other party.

- This Agreement may not be assigned (including by operation of law or otherwise) by you without WILEY's prior written consent.

- Any fee required for this permission shall be non-refundable after thirty (30) days from receipt by the CCC.

- These terms and conditions together with CCC's Billing and Payment terms and conditions (which are incorporated herein) form the entire agreement between you and WILEY concerning this licensing transaction and (in the absence of fraud) supersedes all prior agreements and representations of the parties, oral or written. This Agreement may not be amended except in writing signed by both parties. This Agreement shall be binding upon and inure to the benefit of the parties' successors, legal representatives, and authorized assigns.

- In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall prevail.

- WILEY expressly reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.

- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of Creative Commons Licenses. The license type is clearly identified on the article.

The Creative Commons Attribution License

The Creative Commons Attribution License (CC-BY) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-

Creative Commons Attribution Non-Commercial License

The Creative Commons Attribution Non-Commercial (CC-BY-NC)License permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The Creative Commons Attribution Non-Commercial-NoDerivs License (CC-BY-NC-ND) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)

Use by commercial "for-profit" organizations

Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee.

Further details can be found on Wiley Online Library http://olabout.wiley.com/WileyCDA/Section/id-410895.html

Other Terms and Conditions:

v1.10 Last updated September 2015

Questions? customercare@copyright.com.

Eurographics Symposium on Geometry Processing 2019
D. Bommes and H. Huang
(Guest Editors)

Volume 38 (2019), Number 5

Divergence-Free Shape Correspondence by Deformation

M. Eisenberger, Z. Lähner and D. Cremers

Technical University of Munich, Germany

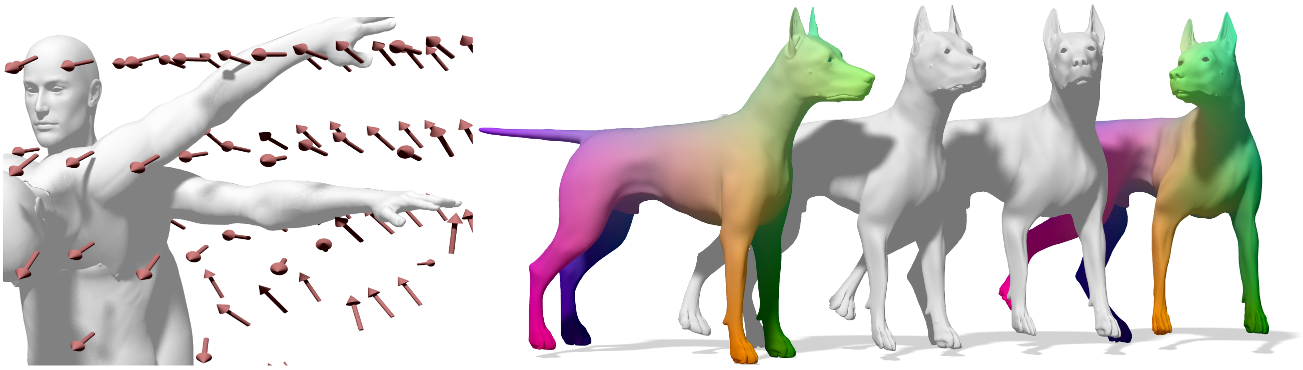


Figure 1: Given two input shapes we propose to morph the source shape along a divergence-free deformation field in order align it with the target. (Left) Example of a deformation field in 3D. (Right) Example of the results of our framework. We alternate between optimizing for the deformation field and calculating correspondences and therefore generate highly accurate correspondences (color coded) as well a sequence of natural intermediate shapes as a by-product (white). The translation is only added for visibility.

Abstract

We present a novel approach for solving the correspondence problem between a given pair of input shapes with non-rigid, nearly isometric pose difference. Our method alternates between calculating a deformation field and a sparse correspondence. The deformation field is constructed with a low rank Fourier basis which allows for a compact representation. Furthermore, we restrict the deformation fields to be divergence-free which makes our morphings volume preserving. This can be used to extract a correspondence between the inputs by deforming one of them along the deformation field using a second order Runge-Kutta method and resulting in an alignment of the inputs. The advantages of using our basis are that there is no need to discretize the embedding space and the deformation is volume preserving. The optimization of the deformation field is done efficiently using only a subsampling of the orginal shapes but the correspondence can be extracted for any mesh resolution with close to linear increase in runtime. We show 3D correspondence results on several known data sets and examples of natural intermediate shape sequences that appear as a by-product of our method.

1. Introduction

Handling non-rigid, nearly isometric deformations of 3D shapes is at the heart of numerous problems in computer vision and graphics. Applications range from shape comparison, information and style transfer to the automatic generation of new, meaningful shapes. In comparison to rigid shape registration, the complexity increases significantly in the presence of non-rigid deformations. Many methods rely on purely intrinsic or very local measures because these are robust under extreme extrinsic changes [OBS*12; ADK16]. While this helps to reduce the complexity of the correspondence problem, it also often leads to artifacts in the matching coming from intrinsic symmetries, which are indistinguishable in the intrinsic view, or areas with indiscriminative features.

A different line of work aims at deforming shapes by directly manipulating their geometry in the embedding space [vFTS06; MS10; MZT*14]. Having an explicit notion of extrinsic deformations yields more regular, continuous matchings which is relevant in many applications. In particular, this approach allows for the creation of new, intermediate versions of the input shapes. On the other hand, these methods are in general more prone to get stuck in local minima and therefore dependent on a good initial alignment of the inputs. Unfortunately, many extrinsic matching methods use linear mappings to model surface deformations [MS10; MZT*14]. While this is feasible for small changes, it is often not compatible with how objects deform in the real world. On the other hand, finding a physically correct morphing between two shapes is highly complex and computationally intense, even when the perfect correspondence or prior knowledge about the input is given [WBRS11; GCLX17]. In this paper, we propose a more plausible morphing model that takes into account volume-preservation during the entire deformation. This is possible by modeling volume-preservation through zero divergence in a deformation field. This property makes our intermediate shapes more natural and our results are less likely to end up in a local minimum than with a linear mapping. In our method, the deformation field is represented in a spatially continuous, coarse-to-fine basis which allows for an efficient optimization. Moreover, we can apply the final deformation to shapes of arbitrary resolution with a minimal increase in complexity.

2. Related Work

2.1. Shape Registration and Matching

Much work has been done in the direction of shape registration and matching and we would like to point the interested reader to in-depth surveys of these topics for an overview [vKZHC11; SMFF07; TCL*13]. Here we will focus on work that is directly related to our approach.

One recent line of work in shape matching is based on spectral decomposition of the surface Laplace-Beltrami operator [DK10]. This is popular because it reduces the dimensionality of the problem from the number of vertices to the number of basis functions chosen [OBS*12]. Nevertheless, extracting the correspondence from the low dimensional representation is still a complex problem and often retrieved solutions are noisy or hard to compute [RMC15]. One major problem with purely spectral approaches is that intrinsic symmetries can not be distinguished, [RPWO18] being one of few exceptions. We also use a spectral approach but, instead of a basis for functions on the surface, we represent deformation fields in the embedding space using the eigenfunctions of the standard Laplacian. Among other things the embedding space allows us to distinguish between intrinsincally symmetric but opposite points.

Methods based on Multi-Dimensional Scaling find correspondences by reembedding and then aligning shapes in a (possibly smaller) embedding space where the complexity is reduced [BBK06; ADK16]. [CK15] calculate a robust non-rigid registration based on Markov random fields but cannot retrieve a continuous deformation which we do. In [MS10] and [MZT*14] the authors address the non-rigid registration problem by modeling one point cloud as a Gaussian mixture model, similar to our method. Moreover, they also determine the correspondences and point mappings in an alternating manner using a expectation maximization algorithm. This work is strongly related to our framework but no intermediate deformation is modeled. There also exist extensions of this method which additionally include descriptor values [MZY16; MJLL17]. [HAWG08] achieve accurate non-rigid alignments but rely on good initial correspondence and expensive geodesic distance computation to find these.

2.2. Deformation Fields

Deformation fields have a long history in image registration. One of the first approaches in that direction is the LDDMM framework [BMTY05]. Ashburner and colleagues made use of deformation fields for autonomous shape morphing [Ash07]. They consider temporally constant deformation fields offering limited flexibility to capture more complex deformations. Solving for a space and time dependent deformation field is a highly underdetermined problem. A remedy for this issue is provided by the geodesic shooting approach advocated by [MTY06] which only estimates the initial velocity field for each pixel and then how the velocity has to propagate in the image domain in order to preserve the kinetic energy and the momentum of the whole system. Further improvements of this framework were proposed in subsequent work, including a Gauss-Newton approach [AF11] and a particularly efficient adjoint calculation [VRRC12].

Closely related to our work is [vFTS06] in which the authors also model volume preserving shape deformations using divergence-free vector fields. Here, deformation fields are constructed from hand crafted templates which are meant to be used as interactive shape transformation tools whereas our method is fully automated. As in our work, in [AOW*08] the deformations are based on a subsampling of the input shapes and can be efficiently applied to the full resolution but the correspondence is assumed to be given.

Probabilistic interpretations of deformation fields are a popular formulation. Such a model for image registration and 2D shape registration with a Gaussian process modeling of the correspondence mapping is proposed in [ALV08]. Further work [LJGV16; DGL*17] specified how one can extend this approach to Gaussian processes on the surface of a three dimensional shape. The authors in [BHB00], [THB08], [ALV08] and [PDS*09] also model non-rigid transformations using a PCA type representation of permitted motions. Analogously, [MS10] and [MZT*14] pursue a reproducing kernel Hilbert space approach to model the vector field interpolation. However, for all these references the respective vector fields are not defined on the whole embedding space surrounding the shapes but rather only at the elements of the considered point clouds and they do not admit an interpretation as a deformation field which makes is harder to impose global properties, e.g. volume-preservation.

Another classical approach to shape deformation is based on a rotation invariant representation of triangle meshes [LSLC05]. In [ZSC*08] this deformation model is used to compute a sparse set of correspondences but this method is hard to scale to high resolutions.

3. Contribution

We introduce a mathematical framework which solves the correspondence problem on two shapes with approximately the same volume. For this purpose, we propose to alternatingly estimate the correspondences and a smooth 3D deformation field aligning the two input shapes. Our shape morphing model solves an initial value problem to shift the first shape along this deformation field. Numerically, this differential equation is integrated using a second order Runge-Kutta scheme. Our framework allows us to incorporate

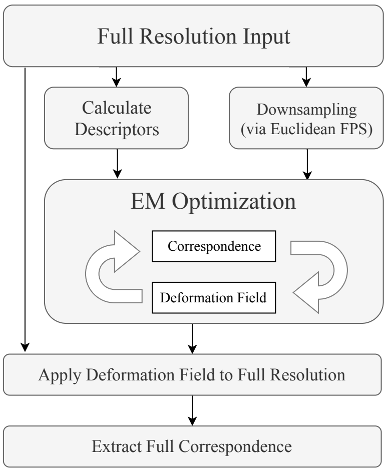

Full Resolution Input
Calculate Descriptors
Downsampling (via Euclidean FPS)
EM Optimization
Correspondence
Deformation Field
Apply Deformation Field to Full Resolution
Extract Full Correspondence

Figure 2: Overview over our complete pipeline.

physical assumptions about the deformations by directly building them into our model. We suggest to impose volume preservation by enforcing the deformation fields to have zero divergence. More specifically, we define a coarse-to-fine basis representation of these vector fields where each basis function is divergence-free. This allows us to reduce the complexity by optimizing only over the most significant coefficients. We use an expectation-maximization approach to simultaneously compute a subset of the unknown point-to-point correspondences and the optimal deformation field coefficients. A schematic diagram of the complete pipeline can be found in Figure 2. We demonstrate that the proposed framework can be used to solve for correspondences which are on par with state-of-the-art methods. Moreover, our method can produce a sequence of reasonable intermediate shapes between the inputs as a by-product. Both can be scaled up to arbitrary resolution without a significant increase in complexity which we demonstrate on on a datset of real scans with over 100k vertices.

4. Problem Formulation

In the following, we define the problem we want to solve and the mathematical background we use in later sections. In general we consider two point clouds $\mathcal{X} = \{x_1, \dots, x_N\} \subset \Omega$ and $\mathcal{Y} = \{y_1, \dots, y_M\} \subset \Omega$ contained in a compact domain $\Omega \subset \mathbb{R}^D$. In practice we choose $\Omega = [0,1]^D$. These points x_n and y_m are samples from the surface of two similar $(D-1)$-dimensional Riemannian manifolds embedded in $\mathbb{R}^D$. Our method aims at aligning the point clouds $\mathcal{X}$ and $\mathcal{Y}$ in a meaningful manner. In particular, we are looking for a mapping $f : \mathcal{X} \to \Omega$ which provides the coordinates for a new embedding of each point on $\mathcal{X}$. In the end, $f(\mathcal{X})$ should be well aligned with $\mathcal{Y}$.

4.1. Deformation field shape morphing

We propose to model the shape morphing $f : \mathcal{X} \to \Omega$ using the following initial value problem:

$$\begin{cases} \dot{x}(t) = v(x(t)). \\ x(0) = x_{\text{init}}. \end{cases} \tag{1}$$

In this context, $v : \Omega \to \mathbb{R}^D$ is some fixed deformation field shifting any point $x_{\text{init}} \in \Omega$ over time. If we solve this differential equation until some fixed time t_{eval}, we get the flow $\varphi : [0, t_{\text{eval}}] \times \Omega \to \Omega$ of Equation (1). The flow φ morphs the space Ω over time, it maps any input point x_{init} to its destination $\varphi(t, x_{\text{init}})$ at time $t \in [0,1]$. Applying Equation (1) to all points $x_{\text{init}} := x_n \in \mathcal{X}$ yields a morphing model for the source shape $\mathcal{X}$:

$$f(x_n) := f_n := \varphi(t_{\text{eval}}, x_n). \tag{2}$$

In order to make those shape deformations more plausible, we require them to be smooth in space and in time. For this purpose, we assume that the deformation fields $v \in C^\infty(\Omega, \mathbb{R}^D)$ which, according to the Picard-Lindelöf Theorem, yields smooth point trajectories $x(\cdot) := \varphi(\cdot, x_{\text{init}}) \in C^\infty([0,1], \Omega)$, see [Tes12, Lemma 2.3,Theorem 2.5]. For convenience we choose $t_{\text{eval}} = 1$ in our experiments.

Our morphing model computes natural shape deformations which can be transformed into correspondences through nearest neighbor search (See Section 5.3). Due to the time dependency of the flow, we additionally get intermediate poses of the input shape at times $t \in (0,1)$ which constitute the underlying transformation. Those are typically more meaningful than naive approaches like linear interpolation between the points. We believe that having a continuous correspondence and a natural deformation are inherently connected and solving for both simultaneously improves the results considerably.

4.2. Divergence-free deformations

One advantage of our morphing model (1) is that it allows us to incorporate assumptions about the deformation fields v into our model. In our framework we restrict these velocity fields to be divergence-free, an assumption that is commonly used in mathematical modeling of incompressible fluids [CM93]:

$$\nabla \cdot v = 0. \tag{3}$$

A well known consequence of this local property is that it yields volume preservation over time for any subpart $U \subset \Omega$ of the embedding space. In particular, we can consider the set of solutions of Equation (1):

$$U(t) := \left\{ \varphi(t, x_{\text{init}}) \in \Omega \middle| x_{\text{init}} \in U \right\}. \tag{4}$$

Then the assumption in Equation (3) yields that each morphed set $U(t)$ has the same volume as U [Tes12, Lemma 8.8]. Therefore, each subvolume of the input shape $\mathcal{X}$, as well as of the embedding space, is preserved at any given time. Notice that this property is stronger than global volume preservation of the interior of $\mathcal{X}$ only. In general, two very differently shaped objects can have the same volume. However, for our method the volume of all, potentially very small, subparts is preserved. In our experiments, we found that this is a reasonable assumption for real world deformations and it provides a good regularization of our morphing model (1).

4.3. Helmholtz decomposition

Helmholtz's theorem [Ari62] implies that any sufficiently smooth vector field on the compact domain Ω can be decomposed into the sum of a curl-free, a divergence-free and a harmonic component. It furthermore provides us with an explicit construction of the divergence-free component that we are interested in:

$$v := \nabla \times \Phi. \tag{5}$$

In this context, $\Phi : \Omega \to \mathbb{R}^D$ is a potential function and $\nabla \times \cdot$ is the curl operator. Indeed,[CNT15, Lemma 2.2] shows that such a Φ exists for any divergence-free, C^∞ vector field $v : \Omega \to \mathbb{R}^D$ with no outflow at the boundary:

$$\langle v, n \rangle = 0 \text{ on } \partial\Omega. \tag{6}$$

Furthermore, for a given Φ we always get a divergence-free vector field v as a basic property of the curl operator:

$$\nabla \cdot (\nabla \times \Phi) = 0. \tag{7}$$

To further restrict the space of admissible deformation fields, we additionally require the potential functions to admit Dirichlet boundary conditions $\Phi|_{\partial\Omega} = 0$. This guarantees that the potential functions are tangential to the outer normals at $\partial\Omega$ which is a necessary condition in the existence proof [CNT15, Lemma 2.2]. Moreover, we are only interested in a high expressibility in the interior of Ω and choosing Dirchlet boundary conditions makes the representation of our deformation fields even more compact. Intuitively, it guarantees that for the resulting deformation fields v there is no flow in and out of the domain Ω (see Equation (6)). In the case of $D = 3$ spatial dimensions the construction of v in (5) admits the following form:

$$v = \begin{pmatrix} \partial_2\Phi_3 - \partial_3\Phi_2 \\ \partial_3\Phi_1 - \partial_1\Phi_3 \\ \partial_1\Phi_2 - \partial_2\Phi_1 \end{pmatrix} = \begin{pmatrix} 0 \\ \partial_3\Phi_1 \\ -\partial_2\Phi_1 \end{pmatrix} + \begin{pmatrix} -\partial_3\Phi_2 \\ 0 \\ \partial_1\Phi_2 \end{pmatrix} + \begin{pmatrix} \partial_2\Phi_3 \\ -\partial_1\Phi_3 \\ 0 \end{pmatrix}. \tag{8}$$

Remark. The harmonic component in the Helmholtz decomposition corresponds to global translations of the input shape $\mathcal{X}$ but we refrain from including them in our framework. For once, we would like the flow $\varphi : [0,1] \times \Omega \to \Omega$ to map all points $x_n \in \Omega$ back to the same domain. Furthermore, modeling global translations is not necessary because we shift the input shapes a priori such that their empirical mean corresponds to the center of Ω.

5. Method

In the following, we outline the core components of our method. First, we construct a coarse-to-fine deformation field basis with certain built-in properties like volume preservation (Section 5.1). Then, we show how to integrate the initial value problem of Equation (1) (Section 5.2). Finally, we provide details about our expectation maximization algorithm (Section 5.3) where we simultaneously optimize for the unknown correspondences and an appropriate deformation field. We refer the reader to the supplementary material for full details. Regarding relevant applications, we will mainly restrict ourselves to the case of 2D shapes embedded in $\mathbb{R}^3$. However, extensions to higher dimensions $D > 3$ or $D = 2$ are straightforward.

5.1. Spatial representation

Standard discretizations of vector fields v using voxel grids have cubic complexity which makes them too costly for any reasonable resolution. To get a more compact representation, we introduce a low rank basis $\{v_1, ..., v_K\}$ of spatially dense, divergence-free deformation fields. The number of basis functions can be adjusted for either speed or expressiveness. Without loss of generality we set the domain to a D-dimensional cube $\Omega := [0,1]^D$. In practice, we then translate and scale any shape to generously fit inside. We begin with defining a basis for the potential fields Φ. For this purpose, consider the eigenfunctions $\{\phi_1, \phi_2, ...\}$ and eigenvalues $\{\lambda_1, \lambda_2, ...\}$ of the scalar Laplacian Δ on Ω:

$$\Delta\phi_k = \lambda_k^\Delta \phi_k. \tag{9}$$

This basis of eigenfunctions $\{\phi_1, \phi_2, ...\}$ is ordered with descending eigenvalues $0 \geq \lambda_1^\Delta \geq \lambda_2^\Delta \geq ....$ Furthermore, we require the potential fields to admit Dirichlet boundary conditions $\Phi|_{\partial\Omega} = 0$. These ϕ_k can be computed analytically, they are exactly the sine elements of the Fourier basis:

$$\mathcal{B}_\phi = \left\{ \phi : [0,1]^D \to \mathbb{R},\ x \mapsto \prod_{d=1}^{D} \frac{1}{2} \sin(x_d \pi j_d) \middle| j \in \mathbb{N}^D \right\}. \tag{10}$$

The set $\mathcal{B}_\phi = \{\phi_1, \phi_2, ...\}$ is ordered by ascending Dirichlet energy of the ϕ_k. These ϕ_k form an orthonormal basis wrt. the $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$ inner product for scalar functions on Ω. We can now use $\mathcal{B}_\phi$ to construct a basis for the deformation fields $\mathcal{B}_v$ according to Equation (5). Note that the basis $\mathcal{B}_\phi$ consists of scalar functions while the potential functions $\Phi : \Omega \to \mathbb{R}^D$ are vector valued. However, due to the linearity of the curl $\nabla \times \cdot$ we obtain a basis by using (5) one entry at a time. For $D = 3$ this can be done as follows:

$$\mathcal{B}_v = \bigcup_{k=1}^{\infty} \left\{ \nabla \times \begin{pmatrix} \phi_k \\ 0 \\ 0 \end{pmatrix}, \nabla \times \begin{pmatrix} 0 \\ \phi_k \\ 0 \end{pmatrix}, \nabla \times \begin{pmatrix} 0 \\ 0 \\ \phi_k \end{pmatrix} \right\} =$$
$$\bigcup_{k=1}^{\infty} \left\{ \begin{pmatrix} 0 \\ \partial_3\phi_k \\ -\partial_2\phi_k \end{pmatrix}, \begin{pmatrix} -\partial_3\phi_k \\ 0 \\ \partial_1\phi_k \end{pmatrix}, \begin{pmatrix} \partial_2\phi_k \\ -\partial_1\phi_k \\ 0 \end{pmatrix} \right\}. \tag{11}$$

We get three deformation basis functions for each ϕ_k in (10). Analogously to the potential fields, the basis elements $\mathcal{B}_v = \{v_1, v_2, ...\}$ are again sorted according to the eigenvalues λ_k^Δ of the corresponding ϕ_k in descending order. Note that there are in general multiple basis functions v_k for each eigenvalue λ_k^Δ. Overall, we obtain arbitrary deformation fields v as the linear combination of the first K basis elements v_k with some coefficients a_k:

$$v(x) = \sum_{k=1}^{K} v_k(x) a_k. \tag{12}$$

Remark One aspect we would like to discuss in this context is our choice of domain $\Omega = [0,1]^D$. The first basis function v_1 in Figure 3 is equivalent up to first order to a rotation around the x_3 axis. This especially holds near the center of the domain Ω and deteriorates at its boundary $\partial\Omega$. Those considerations raise the question whether a cubic domain Ω is the best choice for our purposes. Following the work in [ZB07; ZB08] we could pursue our approach in a spherical

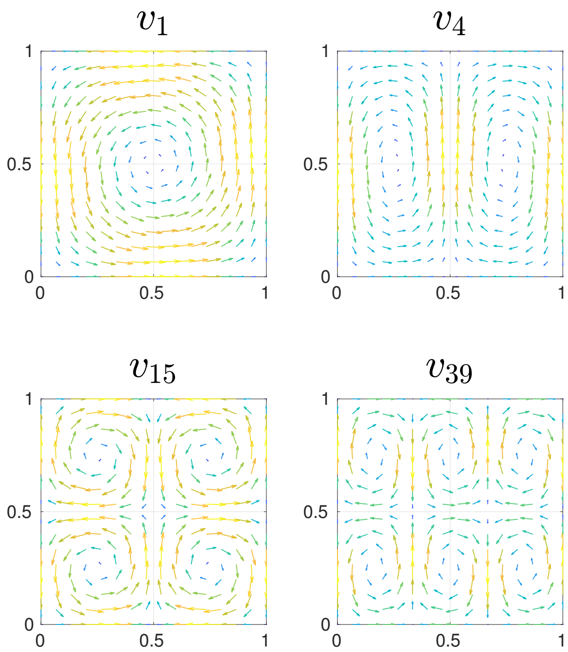


Figure 3: Cross section of some deformation field basis functions $v_k : \Omega \to \mathbb{R}^3$ at $x_3 = 0.5$. Notice the low frequency structures for low k and increasing frequencies with higher indices. Furthermore, one can see that our deformation fields have no flow in and out of the domain Ω at the boundary.

domain. This would lead to more complex basis functions v_k but the first three eigenfunctions would span the space of rotations without undesirable artifacts at the boundaries of the domain. Although this would be a nice theoretical property, we refrain from using these basis functions here due their complex structure.

5.2. Temporal discretization

In order to evaluate the correspondence mapping f in (2) we have to solve the initial value problem (1) with a numerical integration scheme. The simplest choice in this context is the explicit Euler method. However, we decided to use a second order Runge-Kutta method [GH10, Ch. 9] because it has a significantly higher accuracy and therefore allows for a coarser time discretization. We subdivide the time domain in an equidistant grid with $T \in \mathbb{N}$ intervals and set the step size $h = \frac{1}{T}$. This yields an explicit iteration scheme:

$$\begin{cases} x_n^{(0)} := x_n. \\ x_n^{(t+1)} := x_n^{(t)} + hv\left(x_n^{(t)} + \frac{h}{2}v\left(x_n^{(t)}\right)\right). \\ f_n := x_n^{(T)}. \end{cases} \tag{13}$$

We typically choose $T \in \{1,...,100\}$ in our experiments. In general, we have to make a trade off between runtime and accuracy when selecting an appropriate number of steps T. If we choose T too small, we lose some key properties of our framework like the volume preservation. This effect is illustrated in Figure 4 for the 2D shape of a bat transformed by a 90 degree rotation around the center. Note that the deformation field corresponding to this transformation is only approximately contained in our framework due to our choice of domain and boundary conditions, see discussion in the previous subsection. If we choose too few time steps T, the shape shifts outward and the area expands. On the other hand, this effect becomes insignificantly small if we choose $T \geq 10$.

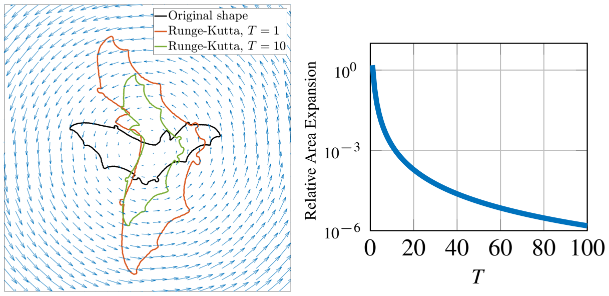


Figure 4: Area expansion with different step sizes using the Runge-Kutta integration. Left: Rotation around 90 degrees on a bat shape of the MPEG-7 dataset [Ral] (black). If executed in one step ($T = 1$) the shape expands (red) whereas for ten steps $T = 10$ the area of the interior stays nearly the same (green). Right: Relative area expansion when performing the same deformation with an increasing number of steps T.

5.3. Optimization

In the previous sections we derived a coherent description of shape morphing using volume preserving deformation fields. We can now use this framework to construct an algorithm that matches two given point clouds $\mathcal{X}$ and $\mathcal{Y}$ by calculating a volume preserving deformation field between them. In order to do that we simultaneously optimize for the deformation field coefficients a and the unknown correspondences.

Similar to [MS10] and [MZT*14] we approach shape registration in a probabilistic manner. We interpret the point cloud $\mathcal{X}$ as a Gaussian mixture model with the means located at the shifted points $f_n = x_n^{(T)}$ and the covariance $\sigma^2 I_D \in \mathbb{R}^{D \times D}$ for some $\sigma > 0$. This enables us to simultaneously determine the deformation field coefficients $a \in \mathbb{R}^K$ and the correspondences $W \in [0,1]^{N \times M}$ by applying an expectation maximization approach. Expecation maximization alternates between optimizing the deformation field coefficients and the correspondence while assuming the other to be fixed. The full derivation of each step, as well as additional implementation details, can be found in the supplementary material.

Expectation step The expectation step calculates correspondences for a fixed deformation. We represent the correspondences between the morphed $f(\mathcal{X}) = \{f_1, \dots, f_N\}$ and the reference pointcloud $\mathcal{Y} = \{y_1, \dots, y_M\}$ as soft correspondence matrices $W \in [0,1]^{N \times M}$ which arise naturally from the Gaussian mixture model assumption. High values of $W_{nm} \approx 1$ indicate a high correspondence probability

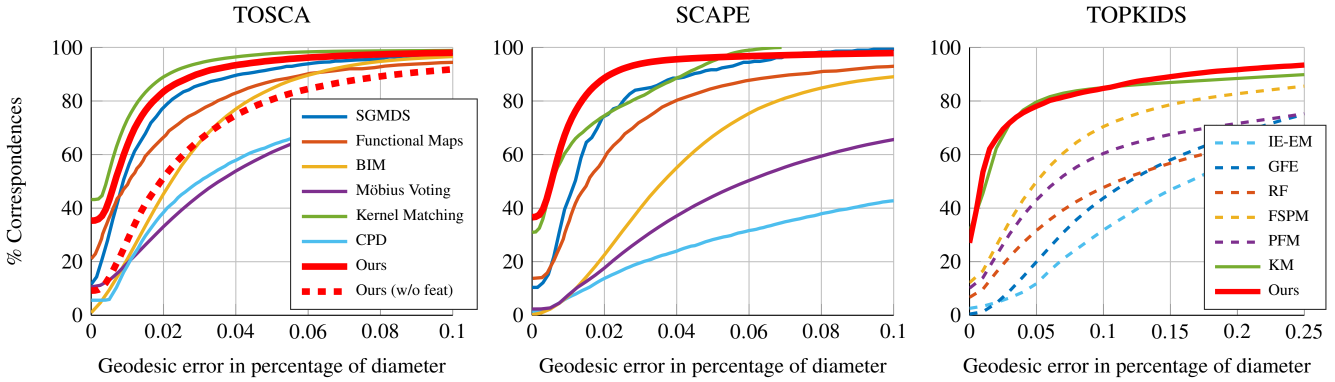


Figure 5: Quantitative evaluation using the Princeton benchmark protocol on the TOSCA data set [BBK08] (left), the SCAPE data set [ASK*05] (middle), and the high-resolution TOPKIDS [LRB*16] (right). On TOSCA and SCAPE we compare against Spectral Generalized Multi-Dimensional Scaling (SGMDS) [ADK16], Functional Maps [OBS*12], Blended Intrinsic Maps (BIM) [KLF11], Möbius Voting [LF09], Coherent Point Drift (CPD) [MS10] and Kernel Matching [VLB*17]. On TOPKIDS we compare against the competitors of the original paper (Isometric Embedding via Expectation-Maximization (IE-EM) [SY12], Green's Function Embedding Alignment (GFE) [BDK17], Random Forests (RF) [RRW*14]), Fully Spectral Partial Matching (FSPM) [LRBB17], Partial Functional Maps (PFM) [RCB*17] and Kernel Matching (KM) [VLB*17]. Both the TOSCA as well as the TOPKIDS dataset contain cases which are critical for our method but our results are still on a par with state-of-the-art. See Section 6.1 for details. On the TOSCA data set we additionally evaluate our method without using features. The drop in performance shows that these are crucial to avoid unwanted optima.

for the point pair (x_n, y_m) while values close to zero indicate low probability. The expectation maximization framework then yields an explicit update rule for W given the deformation coefficients a:

$$W_{nm} := \frac{\exp\left(-\frac{1}{2\sigma^2} \mathrm{d}_{nm}^2\right)}{(2\pi\sigma^2)^{\frac{D}{2}} + \sum_{\tilde{n}=1}^{N} \exp\left(-\frac{1}{2\sigma^2} \mathrm{d}_{\tilde{n}m}^2\right)}. \tag{14}$$

For a derivation of this formula, see the supplementary material. Intuitively, W_{nm} describes the value of a Gaussian with center f_n and variance σ at point y_m.Similar to [MS10], the normalization factor in the denominator comes from the mixture model assumption combined with an explicit modeling of outliers. In order to prevent our method from getting stuck in incorrect local optima, we include SHOT descriptor [TSS10] with standard parameters from the authors' implementation. We combine them with Euclidean distances to define a metric for pairs of points x_n and y_m:

$$\mathrm{d}_{nm}^2 := \left\|y_m - f_n\right\|_2^2 + \bar{\mathrm{d}}\left\|\mathrm{SHOT}(x_n) - \mathrm{SHOT}(y_m)\right\|_2^2. \tag{15}$$

We introduce the factor $\bar{\mathrm{d}} \geq 0$ to ensure that both metrics have a comparable scaling, in particular we require both summands to have the same mean value for all point pairs $\mathcal{X}$ and $\mathcal{Y}$. Note that we use descriptor values $\mathrm{SHOT}(x_n)$ on the original shape $\mathcal{X}$ instead of the morphed shape $f(\mathcal{X})$ in order to not recompute them at every iteration.

Maximization step The maximization step updates the deformation field for given soft correspondences W. Intuitively, we are looking for the deformation field coefficients a that best align points with high correspondence probability W_{nm}. For this purpose, we interpret the coefficients $a = (a_1, .., a_K)^\top$ as random variables with a normal distribution $a \sim \mathcal{N}(0, L)$, where $L := \mathrm{diag}(\lambda_1, ..., \lambda_K)$. If we compute the pushforward of this Gaussian according to Equation (12), we get a prior distribution of deformation fields v. The weights λ_k are constructed from the eigenvalues λ_k^Δ as follows:

$$\lambda_k := \left(-\lambda_k^\Delta\right)^{-\frac{D}{2}} = \left(\pi^2 \sum_{d=1}^{D} j_d^2\right)^{-\frac{D}{2}}. \tag{16}$$

The mathematical background of this choice for the weights λ_k is provided by the Karhunen-Loève expansion [Sul15, Ch. 11] which is an extension of the principal component analysis (PCA) for function spaces, see the supplementary material for more details. Intuitively, this kind of weighting promotes a damping of the high frequency components and smoothness of the deformation field v. The maximization step optimizes the coefficients a for their posterior distribution given the current correspondences which describes how well the deformation field of a explains W. This results in the following energy for a:

$$E(a) := \frac{\sigma^2}{2} a^\top L^{-1} a + \sum_{m=1}^{M} \sum_{n=1}^{N} W_{nm} \rho(\|y_m - f_n\|_2). \tag{17}$$

This energy E is the sum of the negative log prior including the weights λ_k (left term) and the negative log likelihood (right term) of a. The function $\rho : \mathbb{R} \to [0, \infty)$ is the Huber loss [Hub64] which helps to account for outliers and makes the deformation field estimation more robust:

$$\rho(r) = \begin{cases} \frac{1}{2} r^2 & |r| \leq r_0. \\ r_0 |r| - \frac{1}{2} r_0^2 & \text{otherwise.} \end{cases} \tag{18}$$

In our experiments, we choose the outer slope as $r_0 := 0.01$. Furthermore, we apply a Gauss-Newton type approach to minimize

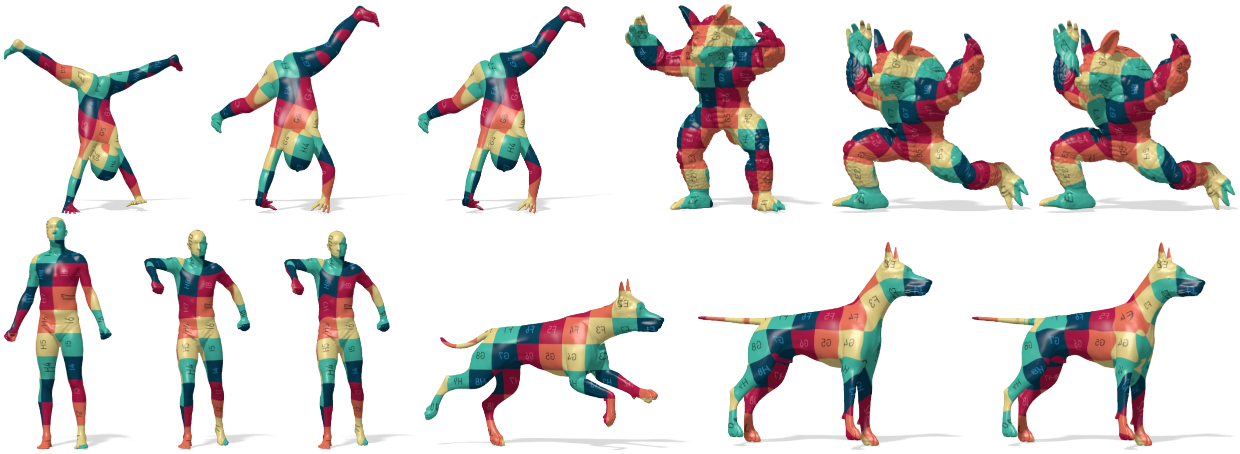


Figure 6: Examples of texture transfer done with our method. For each object the first image shows the source shape and texture, the second image the texture transferred with the ground-truth map and the third image the texture transferred with our correspondences. Our results are nearly identical to the ground-truth except for the dog which shows some artifacts on tail and chest.

the energy in (17). This results in an iterative method similar to the Levenberg-Marquardt algorithm [Lev44]. For this purpose, the residual term $\|y_m - f_n\|_2$ is linearized in each iteration. This requires a differentiation of the Runge-Kutta scheme (13) wrt. the weights a, see the supplementary for an explicit formulation of the derivative $\frac{\mathrm{d}}{\mathrm{d}a} f_n$ and the Gauss-Newton update step for the energy in Equation (17).

To summarize, our method alternates between computing the weights $W^{(i)}$ according to (14) and performing one Gauss-Newton update step for (17) to obtain $a^{(i)}$. To initialize the algorithm we set the deformation field to zero $a^{(0)} := 0$.

6. Experiments

We evaluate our method for several applications to show that it is general and flexible. Although we handle shapes with up to 200k and more vertices, the computation of the deformation field is always done on a downsampled version of the inputs with 3000 vertices and then applied to the full resolution. We use Euclidean farthest point sampling. The downsampled shape should include points of all relevant large and fine scale structures in order for the deformation field to move these correctly but we found 3000 sufficient for our applications. As a preprocessing step we shift both inputs such that the mean of their vertex positions is in the center of the domain and align them using PCA. To avoid wrong alignments along the principle component axes we choose the orientation that minimizes Eq. (15). When averaging over all experiments presented here, our algorithm takes about 370 seconds to compute the deformation and correspondences for one pair of shapes. Due to our a priori downsampling the runtime is only linearly dependent on the number of vertices, see Section 6.4 for a discussion of this property. All experiments were performed with MATLAB on a system with an Intel Core i7-3770 CPU clocked at 3.40GHz, 32 GB RAM and a GeForce GTX TITAN X graphics card running a recent Linux distribution. In all our experiments we only use the raw shape data and in particular do not need any ground truth information or user input.

6.1. Matching

We verify our method using the TOSCA [BBK08], SCAPE [ASK*05] and high-resolution TOPKIDS [LRB*16] data sets. All these shapes are synthetic and therefore the exact intraclass correspondences are known. TOSCA contains 76 triangular meshes with 8 classes of humans and animals, SCAPE consists of 72 poses of the same person and TOPKIDS contains 26 poses of the same person in which topological merging, as it might appear in real scanning, is imitated.

We set the hyperparameters $\sigma^2 := 0.01, T := 20$ and choose $K = 3000$ basis functions for the deformation field. Because $W^{(i)}$ only contains 3000 correspondences we perform a nearest-neighbor search with respect to the metric in Eq. (15) to obtain a dense mapping. The evaluation is done with the Princeton benchmark protocol [KLF11]. Given the ground-truth match $(x, y^*) \in \mathcal{X} \times \mathcal{Y}$, the error of the calculated match (x, y) is given by the geodesic distance between y and y^* normalized by the diameter of $\mathcal{Y}$:

$$\varepsilon(x) = \frac{\mathrm{d}_{\mathcal{Y}}^{\mathrm{Geo}}(y, y^*)}{\sqrt{\mathrm{area}(\mathcal{Y})}}$$

We plot cumulative curves showing the percentages of matches that are below an increasing threshold. As zero is the value for ground-truth matches, the ideal curve would be constant at 100. See Figure 5 for our results and Figure 6 for example matching results showing texture transfer. On SCAPE we are able to reach state-of-the-art results whereas on TOSCA the intrinsic Kernel Matching methods is slightly better. Our extrinsic approach makes self-touching poses more challenging and these cases occur fairly often in TOSCA. Although TOPKIDS is still synthetic the self-touching poses are actually merged in the geometry which makes it more challenging. On this dataset we are slightly better than Kernel

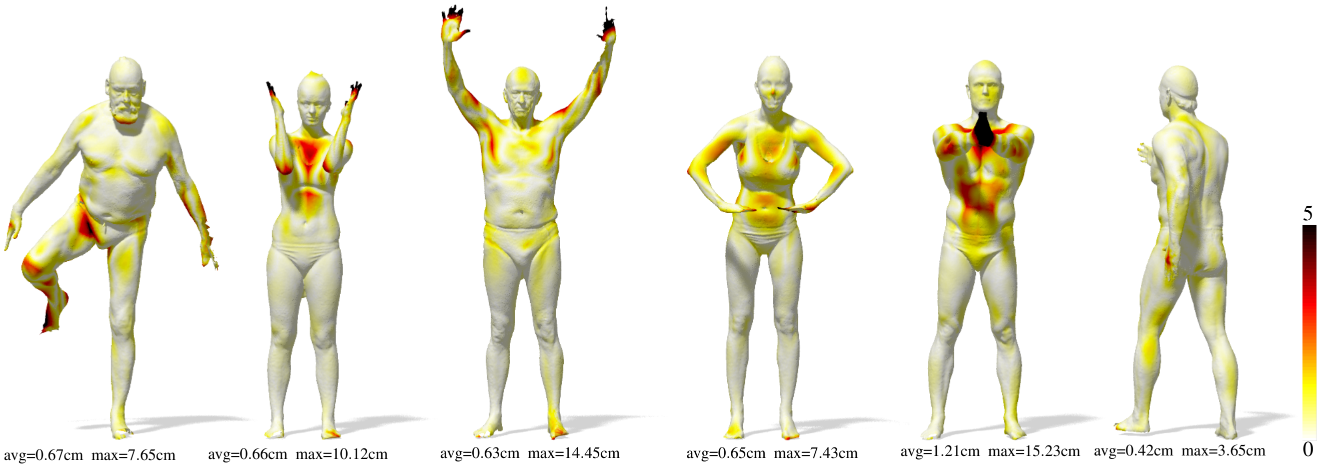


Figure 7: Example registrations from the FAUST scan data set. The surface color corresponds to the Euclidean surface distance between scan and registration. The scale of the scans is in real cm values and the same on all plots. We report the average and maximum error under each image. Many errors occur due to the SHOT descriptors being corrupted at holes and in noisy areas (e.g. the hands). Furthermore, in some case the assumption of exact volume preservation is too restrictive for real scans with noise and topological changes (see especially second to the right).

Matching (see Figure 5). See the supplementary material for an example.

To show the influence of features on the results we do an evaluation of our method without using features at any point during the optimization. Instead the distance of Equation (15) is replaced with the pure Euclidean distance between the coordinates. The result can be seen in Figure 5. The performance without features decreases substancially because the Euclidean distance is a weak indicator when large deformations take place. Therefore, our method gets stuck in local optima more often.

6.2. Registration

We apply our framework to the FAUST Scan dataset [BRLB14] which contains data from scans of real humans in different poses. Each of these shapes has approximately $200k$ vertices, they are sampled inconsistently and some of them are severely affected by scanning noise, holes and topological changes. We match the null shape of every person to its other poses. In Figure 7 we display the surface distance of the morphed shapes to the goal shape for some examples. We reach very tight alignments except in very challenging cases like topological changes. Furthermore, the scanned volume varies slightly even between different poses of the same humans which induces small errors in our method.

6.3. Effect of the basis size

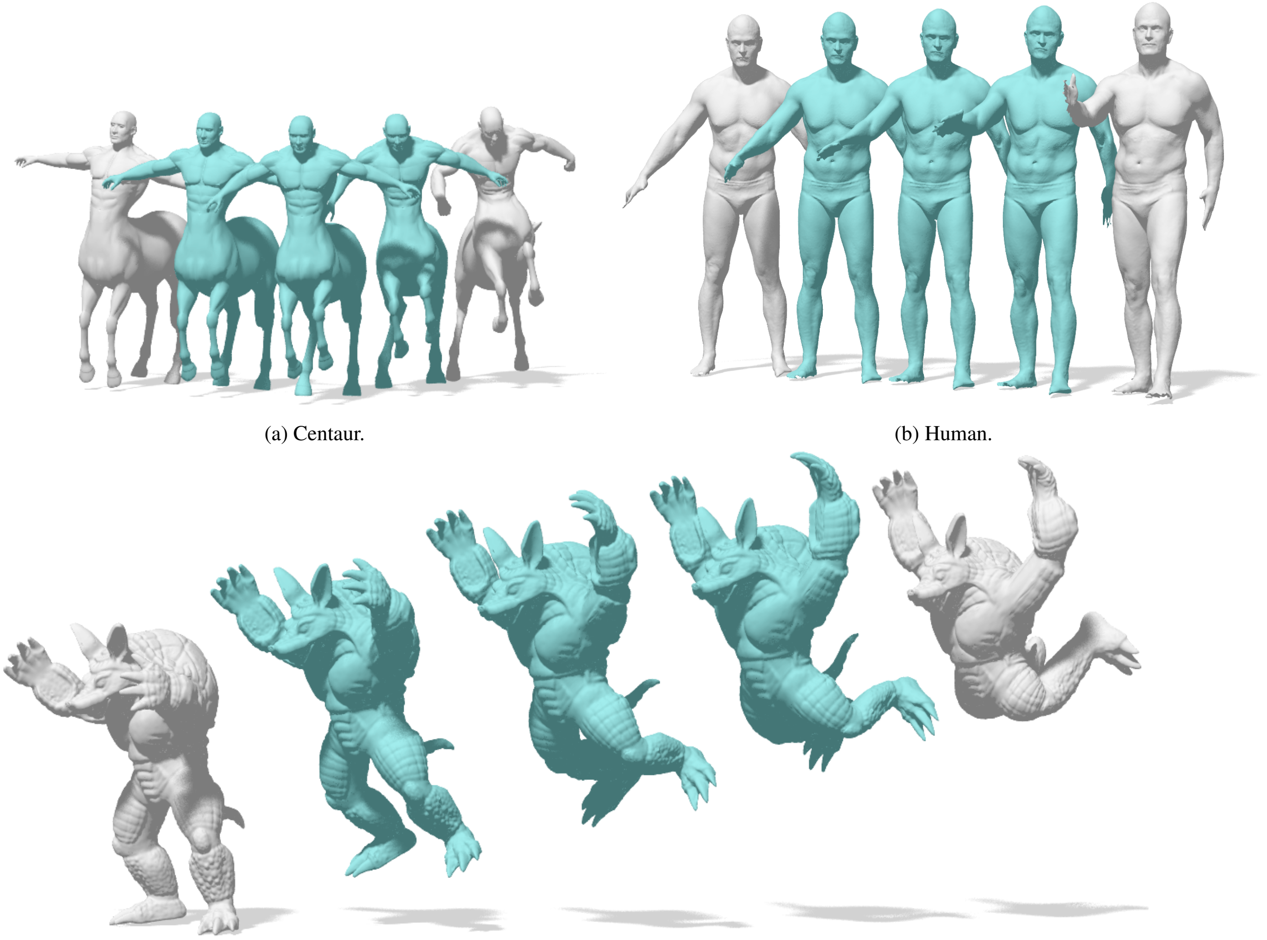
In our evaluations we consistently use $K = 3000$ deformation field basis functions. To justify this choice empirically, we compute the mean geodesic errors of each TOSCA pair for several basis sizes $K \in \{1,\dots,3000\}$, see Figure 9. We observe that while the accuracy increases significantly for small $K \leq 1000$, after some point it starts plateauing. In our evaluations, we choose $K = 3000$ because we aim for a high accuracy. However, for some applications where runtime is more important than accuracy a smaller basis size $K < 3000$ might be sufficient.

6.4. Runtime for high resolution

One major advantage of our method is that it is scalable to high resolution input shapes like those from FAUST because we optimize for the deformation field on downsampled shapes (3000 vertices). One point that we want to stress in this context is that this is not the same as computing matchings only on low resolution shapes. For many matching methods this scaling to the full resolution is challenging, most methods need to come up with a custom coarse-to-fine strategy. In general, it is not straightforward to extend a shape matching or deformation from a downsampled shape to the rest of the vertices. However, for our method this upscaling is trivial because the deformation field basis functions (11) are defined densely on the whole embedding space, therefore they can be evaluated anywhere in Ω. This upsampling scales linearly in N because the Runge-Kutta method (13) is computed independently for all vertices x_n. See Figure 10 for an empirical verification of this property. Here, the runtime for the shape deformation is computed for various downsampled versions of one high resolution shape. To sum it up, the runtime for computing shape morphings is relatively low and increases only linearly in the number of vertices N which makes our method scalable for high resolution input shapes.

6.5. Shape Interpolation

Interpolation Our method morphs the input shape $\mathcal{X}$ by solving the ODE (1) up to time $t_{\text{eval}} = 1$. If we now instead evaluate it at an intermediate time $t \in (0,1)$, we get interpolated shapes as a byproduct of our matching pipeline. Just like the morphed shapes $f(\mathcal{X})$ those intermediate shape morphings are smooth and volume preserving which makes them look natural. Three examples with



(a) Centaur.

(b) Human.

(c) Armadillo.

Figure 8: Three examples of shapes that are morphed into one another according to the initial value problem of Eq. (1). The centaur (a) and the human (b) are from the TOSCA [BBK08] and FAUST [BRLB14] dataset respectively. The armadillo (c) is from the AIM@SHAPE shape repository [AIM]. (b) is a scan of a real person and very high resolution (214k vertices). The source and target shape are shown in white and the interpolations at times $t = 0.25, 0.5, 0.75$ in blue. The translation is not part of our deformation and was only introduced for clarity in the figures.

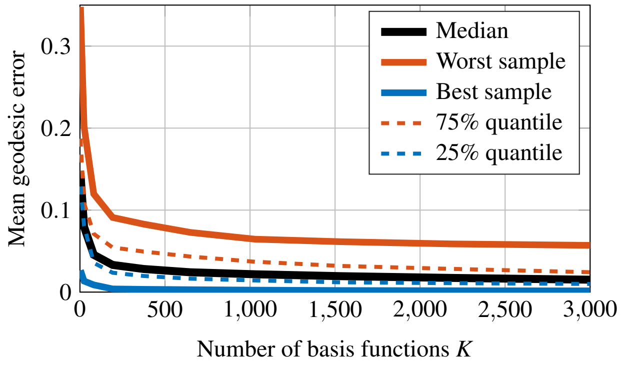


Figure 9: Dependency of the mean geodesic errors on TOSCA on different basis sizes $K \in \{1, \ldots, 3000\}$. In particular, we show the elements at the 0%, 25%, 50%, 75% and 100% quantile.

interpolated shapes are displayed in Figure 8 and videos can be found in the supplementary material.

Extrapolation Similarly to the idea of interpolating shapes as a byproduct of our method we can also use the computed deformation field v to solve the initial value problem (1) up to times $t > 1$. This results in extrapolated shapes, see Figure 11. In contrast to shape interpolation, extrapolation is a severely underdetermined task and it is hard to evaluate quantitatively. Nevertheless, we observed that for moderate time spans $t \in [1, 1.5]$ our method produces reasonable results. In general, the morphing speed slows down at some point, especially when the shape is moving in previously unoccupied space. Intuitively, for the optimization there no incentive to impose any particular movement on these parts of the domain Ω, if it is not relevant for the surface alignment. Still, our

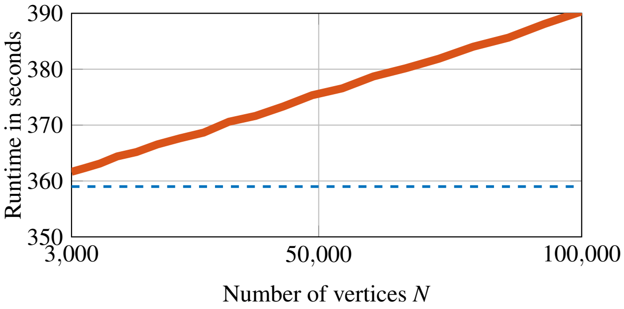


Figure 10: Runtime of our method for the full resolution shape deformation for different number of vertices $N \in \{3000, \dots, 100000\}$. The full pipeline has two steps: (1) a fixed size optimization over 3000 vertices which takes around 360 seconds on average (blue dashed line), (2) applying the deformation field to the full resolution shape and extracting the correspondence for the full shape. The plot shows that our method scales linearly in the number of vertices and is therefore still feasible for very detailed shapes with over $100k$ vertices.

extrapolated shapes are visually appealing and not too severely affected by distortions.

7. Conclusion

We presented a novel extrinsic approach to shape matching. Besides computing dense surface correspondences, we also determine a smooth, volume preserving deformation field between the input shapes. Our morphing model shifts the source shape $\mathcal{X}$ along this deformation field using a second order Runge-Kutta integration scheme in order to align it with the reference shape $\mathcal{Y}$. Additionally to aligning the inputs, this model can also be used to efficiently calculate plausible interpolated shapes.

Our method addresses the coupled problem of finding an unknown deformation and correspondence with an expectation maximization approach. Furthermore, we represent our morphing model with a low rank deformation field basis which reduces the degrees of freedom and makes the optimization problem well constrained. This then allows for a subsampling of the inputs which makes it computationally feasible, even for high resolution meshes, with only a linear increase in runtime. Quantitative evaluations for shape correspondence partly prove state-of-the-art performance of our method. Moreover, we show convincing examples of shape interpolation and extrapolation that arise naturally from our pipeline.

7.1. Limitations

Due to our choice of basis the deformation field is forced to be volume preserving. This makes sense in applications with the same object but prevents inter-class matchings - for example between two humans with different body shapes. The volume preservation property applies to every subregion of the domain Ω, including the intermediate space between parts of the shape. Therefore, separating two touching parts (for example two hands) is in theory possible but requires many high frequency deformation basis elements which would make the optimization costly.

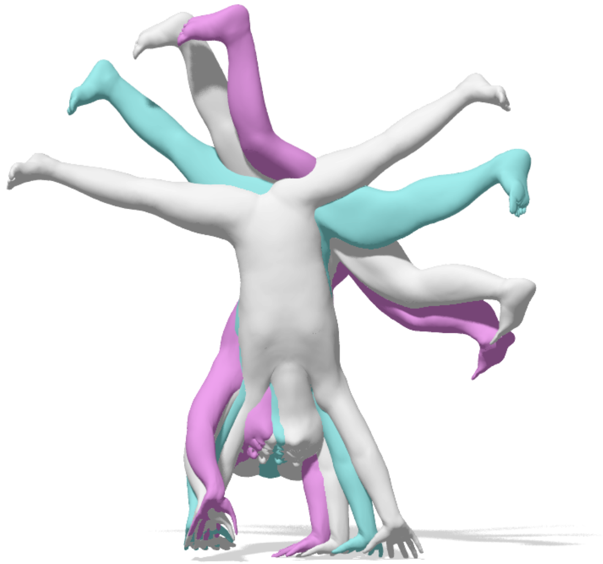


Figure 11: Example of an extrapolated shape from the KIDS dataset [RBW*14]. It can be determined using the temporally fixed deformation field v for simulating the initial value problem (1) up to the time $t = 1.3$. Source and target shape are white, one interpolated shape is shown in blue and the extrapolation is pink. The deformation field is usually magnified in the area between the input shapes and fans out in several directions further away from the input shapes. Therefore, choosing a really high time does not lead to broken shapes but the movement slows down more and more until it basically stops.

The assumption of (1) being autonomous can be problematic if different parts of the shape move through the same region of the embedding space in a contradictory manner. One example for this is a hand closing to a fist. At first the index and middle finger occupy parts of the embedding space before the thumb moves in the same area but in a different direction. See Figure 12. A possible remedy for this problem is making the deformation fields time dependent.

Furthermore, since there is not one unique, volume preserving deformation between two shapes, our interpolation is not guaranteed to be as-rigid-as-possible which is a plausible assumption in many applications. If the displacement is spatially far, we might end up with squeezed intermediate states that are volume preserving but are affected by undesirable distortions. This is also visible in Figure 12 where the tip of the thumb becomes flat.

7.2. Future Work

Right now, our method will always find a solution that is globally volume preserving. This allows to find good deformations fields in the case of severe non-rigid deformations but is not applicable to partial data. In the future, we want to extend this method to work on real scans, for example from the Kinect, which naturally only show partial shapes. A promising approach for this is making the deformation divergence-free in certain parts of the domain while still allowing volume changes in some places and imposing an as-Killing-as-possible constraint in the optimization. Additionally, we want to construct an adaptive basis for this depending on the input shapes. This might also help with the separation of close parts and handling non volume preserving deformations like style or class changes. Furthermore, we only calculate one time independent field for the entire deformation which means mass at one spatial point always needs to move in the same direction at a later time step. This restricts the complexity of the deformations that our method can handle, especially for large-scale motions over a longer period of time. Future versions should allow more flexible types of deformation fields to extend it to a broader range of applications. We could for example associate different parts of the shape with different deformations fields or let them vary over time to address more difficult tasks.

Acknowledgements

We thank Emanuele Rodolá, Nikolaus Demmel and Emanuel Laude for useful discussions. We gratefully acknowledge the support of the ERC Consolidator Grant "3D Reloaded".

References

[ADK16] Aflalo, Yonathan, Dubrovina, Anastasia, and Kimmel, Ron. "Spectral generalized multi-dimensional scaling". *IJCV* 118.3 (2016), 380–392 1, 2, 6.

[AF11] Ashburner, John and Friston, Karl J. "Diffeomorphic registration using geodesic shooting and Gauss–Newton optimisation". *NeuroImage* 55.3 (2011), 954–967 2.

[AIM] AIM@SHAPE repository. *AIM@SHAPE repository*. http://visionair.ge.imati.cnr.it:8080/ontologies/shapes/viewgroup.jsp?id=657-Armadillo_1 9.

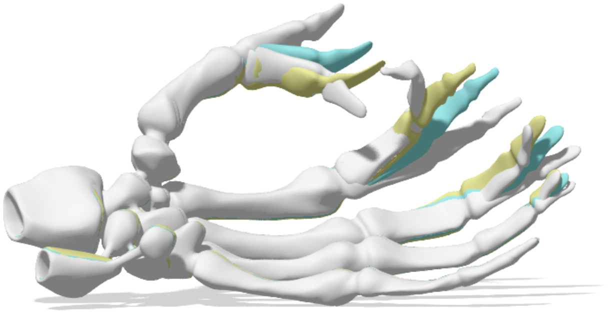


Figure 12: Example of a failure case. The source and target shapes are white, the interpolated shape at $t = 0.5$ is blue and the resulting shape at $t = 1$ is yellow. The yellow shape is supposed to be as close to the target as possible but the fingers are pushed away from each other instead. Here, the thumb and index finger are supposed to move in spatially very close areas. We are only calculating one deformation field for all time steps, therefore it is not clear for our method what motion to assign for contradicting motions. In order to resolve this problem, we need to either assign different motions to different subparts of the objects or make the deformation fields time dependent.

[ALV08] Albrecht, Thomas, Lüthi, Marcel, and Vetter, Thomas. "A statistical deformation prior for non-rigid image and shape registration". *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), 1–8 2.

[AOW*08] Adams, Bart, Ovsjanikov, Maks, Wand, Michael, et al. "Meshless Modeling of Deformable Shapes and their Motion". *Eurographics/SIGGRAPH Symposium on Computer Animation*. Ed. by Gross, Markus and James, Doug. The Eurographics Association, 2008 2.

[Ari62] Aris, Rutherford. *Vectors, tensors, and the basic equations of fluid mechanics*. Englewood Cliffs, N.J., Prentice-Hall, 1962 3.

[Ash07] Ashburner, John. "A fast diffeomorphic image registration algorithm". *Neuroimage* 38.1 (2007), 95–113 2.

[ASK*05] Anguelov, Dragomir, Srinivasan, Praveen, Koller, Daphne, et al. "SCAPE: shape completion and animation of people". *ACM transactions on graphics (TOG)*. Vol. 24. 3. ACM. 2005, 408–416 6, 7.

[BBK06] Bronstein, Alexander M, Bronstein, Michael M, and Kimmel, Ron. "Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching". *PNAS* 103.5 (2006), 1168–1172 2.

[BBK08] Bronstein, Alexander M, Bronstein, Michael M, and Kimmel, Ron. *Numerical geometry of non-rigid shapes*. http://tosca.cs.technion.ac.il/book/resources_data.html. Springer, 2008 6, 7, 9.

[BDK17] Burghard, Oliver, Dieckmann, Alexander, and Klein, Reinhard. "Embedding Shapes with Green's functions for Global Shape Matching". *Computers & Graphics* 68C (2017), 1–10 6.

[BHB00] Bregler, Christoph, Hertzmann, Aaron, and Biermann, Henning. "Recovering non-rigid 3D shape from image streams". *Proceedings IEEE Conference on Computer Vision and Pattern Recognition* 2 (2000), 690–696 2.

[BMTY05] Beg, M Faisal, Miller, Michael I, Trouvé, Alain, and Younes, Laurent. "Computing large deformation metric mappings via geodesic flows of diffeomorphisms". *International journal of computer vision* 61.2 (2005), 139–157 2.

[BRLB14] Bogo, Federica, Romero, Javier, Loper, Matthew, and Black, Michael J. "FAUST: Dataset and evaluation for 3D mesh registration". *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE, June 2014 8, 9.

[CK15] Chen, Qifeng and Koltun, Vladlen. "Robust Nonrigid Registration by Convex Optimization". *IEEE International Conference on Computer Vision (ICCV)*. 2015, 2039–2047 2.

[CM93] Chorin, Alexandre J and Marsden, Jerrold E. *A Mathematical Introduction to Fluid Mechanics*. Springer, 1993 3.

[CNT15] Creusé, Emmanuel, Nicaise, Serge, and Tang, Zuqi. "Helmholtz decomposition of vector fields with mixed boundary conditions and an application to a posteriori finite element error analysis of the Maxwell system". *Mathematical Methods in the Applied Sciences* 38.4 (2015), 738–750 4.

[DGL*17] Dölz, Jürgen, Gerig, T, Lüthi, Marcel, et al. "Efficient computation of low-rank Gaussian process models for surface and image registration". (2017) 2.

[DK10] Dubrovina, Anastasia and Kimmel, Ron. "Matching shapes by eigendecomposition of the Laplace-Beltrami operator". 2010 2.

[GCLX17] Gao, Lin, Chen, Shu-Yu, Lai, Yu-Kun, and Xia, Shihong. "Data-Driven Shape Interpolation and Morphing Editing". *Comput. Graph. Forum* 36 (2017), 19–31 2.

[GH10] Griffiths, David and Higham, Desmond J. *Numerical Methods for Ordinary Differential Equations. Initial Value Problems*. Springer, 2010 5.

[HAWG08] Huang, Qi-Xing, Adams, Bart, Wicke, Martin, and Guibas, Leonidas. "Non-Rigid Registration Under Isometric Deformations". *Computer Graphics Forum* 27 (2008), Number 5 2.

[Hub64] HUBER, PETER J. "Robust Estimation of a Location Parameter". *Annals of Statistics* 53.1 (1964), 73–101 6.

[KLF11] KIM, VLADIMIR G., LIPMAN, YARON, and FUNKHOUSER, THOMAS A. "Blended intrinsic maps". *Trans. Graphics* 30.4 (2011) 6, 7.

[Lev44] LEVENBERG, KENNETH. "A method for the solution of certain non-linear problems in least squares". *Quarterly of Applied Mathematics* 2.2 (1944), 164–168 7.

[LF09] LIPMAN, YARON and FUNKHOUSER, THOMAS. "Möbius voting for surface correspondence". *Trans. Graphics*. Vol. 28. 3. 2009, 72 6.

[LJGV16] LÜTHI, MARCEL, JUD, CHRISTOPH, GERIG, THOMAS, and VETTER, THOMAS. "Gaussian Process Morphable Models". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 99 (2016) 2.

[LRB*16] LÄHNER, Z., RODOLA, E., BRONSTEIN, M. M., et al. "SHREC'16: Matching of Deformable Shapes with Topological Noise". *Proc. of Eurographics Workshop on 3D Object Retrieval (3DOR)*. May 2016 6, 7.

[LRBB17] LITANY, OR, RODOLÀ, EMANUELE, BRONSTEIN, ALEX, and BRONSTEIN, MICHAEL. "Fully Spectral Partial Shape Matching". *Computer Graphics Forum* 36.2 (2017), 1681–1707 6.

[LSLC05] LIPMAN, YARON, SORKINE, OLGA, LEVIN, DAVID, and COHEN-OR, DANIEL. "Linear rotation-invariant coordinates for meshes". *ACM Transactions on Graphics (TOG)*. Vol. 24. 3. ACM. 2005, 479–487 2.

[MJLL17] MA, JIAYI, JIANG, JUNJUN, LIU, CHENGYIN, and LI, YANSHENG. "Feature guided Gaussian mixture model with semi-supervised EM and local geometric constraint for retinal image registration". *Information Sciences* 417 (2017), 128–142 2.

[MS10] MYRONENKO, ANDRIY and SONG, XUBO. "Point Set Registration: Coherent Point Drift". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (2010), 2262–2275 1, 2, 5, 6.

[MTY06] MILLER, MICHAEL I, TROUVÉ, ALAIN, and YOUNES, LAURENT. "Geodesic shooting for computational anatomy". *Journal of mathematical imaging and vision* 24.2 (2006), 209–228 2.

[MZT*14] MA, JIAYI, ZHAO, JI, TIAN, JINWEN, et al. "Robust point matching via vector field consensus". *IEEE Transactions on Image Processing* 23.4 (2014), 1706–1721 1, 2, 5.

[MZY16] MA, JIAYI, ZHAO, JI, and YUILLE, ALAN L. "Non-rigid point set registration by preserving global and local structures". *IEEE Transactions on image Processing* 25.1 (2016), 53–64 2.

[OBS*12] OVSJANIKOV, MAKS, BEN-CHEN, MIRELA, SOLOMON, JUSTIN, et al. "Functional maps: a flexible representation of maps between shapes". *ACM Transactions on Graphics (TOG)* 31.4 (2012), 30 1, 2, 6.

[PDS*09] PALADINI, MARCO, DEL BUE, ALESSIO, STOSIC, MARKO, et al. "Factorization for non-rigid and articulated structure using metric projections". (June 2009), 2898–2905 2.

[Ral] RALPH, RICHARD. *MPEG-7 Core Experiment CE-Shape-1 Test Set*. `http://www.dabi.temple.edu/~shape/MPEG7/dataset.html` 5.

[RBW*14] RODOLA, EMANUELE, BULO, SAMUEL ROTA, WINDHEUSER, THOMAS, et al. "Dense Non-Rigid Shape Correspondence Using Random Forests". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 10.

[RCB*17] RODOLÀ, E., COSMO, L., BRONSTEIN, M. M., et al. "Partial Functional Correspondence". *Computer Graphics Forum* 36.1 (2017), 222–236 6.

[RMC15] RODOLÀ, EMANUELE, MOELLER, MICHAEL, and CREMERS, DANIEL. "Point-wise Map Recovery and Refinement from Functional Correspondence". *Vision, Modeling and Visualization*. The Eurographics Association, 2015 2.

[RPWO18] REN, JING, POULENARD, ADRIEN, WONKA, PETER, and OVSJANIKOV, MAKS. "Continuous and Orientation-preserving Correspondences via Functional Maps". *ACM Trans. Graph.* 37.6 (Nov. 2018), 248:1–248:16 2.

[RRW*14] RODOLÀ, EMANUELE, ROTA BULÒ, SAMUEL, WINDHEUSER, THOMAS, et al. "Dense non-rigid shape correspondence using random forests". *Proc. CVPR*. 2014 6.

[SMFF07] SALVI, JOAQUIM, MATABOSCH, CARLES, FOFI, DAVID, and FOREST, JOSEP. "A Review of Recent Range Image Registration Methods with Accuracy Evaluation". *Image Vision Comput.* 25.5 (2007), 578–596 2.

[Sul15] SULLIVAN, T J. *Introduction to Uncertainty Quantification*. Springer, 2015 6.

[SY12] SAHILLIOĞLU, YUSUF and YEMEZ, YÜCEL. "Minimum-Distortion Isometric Shape Correspondence Using EM Algorithm". *IEEE Trans. Pattern Anal. Mach. Intell.* 34.11 (2012), 2203–2215 6.

[TCL*13] TAM, GARY K. L., CHENG, ZHI-QUAN, LAI, YU-KUN, et al. "Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid". *IEEE Transactions on Visualization and Computer Graphics* 19.7 (July 2013), 1199–1217 2.

[Tes12] TESCHL, GERALD. *Ordinary Differential Equations and Dynamical Systems*. AMS, 2012 3.

[THB08] TORRESANI, LORENZO, HERTZMANN, AARON, and BREGLER, CHRISTOPH. "Nonrigid structure-from-motion: estimating shape and motion with hierarchical Priors". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.5 (2008), 878–892 2.

[TSS10] TOMBARI, FEDERICO, SALTI, SAMUELE, and STEFANO, LUIGI DI. "Unique signatures of histograms for local surface description". *In Proceedings ECCV* 16.9 (2010), 356–369 6.

[vFTS06] Von FUNCK, WOLFRAM, THEISEL, HOLGER, and SEIDEL, HANS-PETER. "Vector field based shape deformations". *ACM Transactions on Graphics (TOG)*. Vol. 25. 3. ACM. 2006, 1118–1125 1, 2.

[vKZHC11] Van KAICK, OLIVER, ZHANG, HAO, HAMARNEH, GHASSAN, and COHEN-OR, DANIEL. "A Survey on Shape Correspondence". *Computer Graphics Forum* 30.6 (2011), 1681–1707 2.

[VLB*17] VESTNER, MATTHIAS, LÄHNER, ZORAH, BOYARSKI, AMIT, et al. "Efficient Deformable Shape Correspondence via Kernel Matching". *International Conference on 3D Vision (3DV)*. Oct. 2017 6.

[VRRC12] VIALARD, FRANÇOIS-XAVIER, RISSER, LAURENT, RUECKERT, DANIEL, and COTTER, COLIN J. "Diffeomorphic 3D image registration via geodesic shooting using an efficient adjoint calculation". *International Journal of Computer Vision* 97.2 (2012), 229–241 2.

[WBRS11] WIRTH, BENEDIKT, BAR, LEAH, RUMPF, MARTIN, and SAPIRO, GUILLERMO. "A Continuum Mechanical Approach to Geodesics in Shape Space". *International Journal of Computer Vision* 93.3 (July 2011), 293–318. ISSN: 1573-1405 2.

[ZB07] ZHAO, CHUNYU and BURGE, JAMES H. "Orthonormal vector polynomials in a unit circle, Part I: basis set derived from gradients of Zernike polynomials". *Opt. Express* 15.26 (2007), 18014–18024 4.

[ZB08] ZHAO, CHUNYU and BURGE, JAMES H. "Orthonormal vector polynomials in a unit circle, Part II: completing the basis set". *Opt. Express* 16.9 (2008), 6586–6591 4.

[ZSC*08] ZHANG, HAO, SHEFFER, ALLA, COHEN-OR, DANIEL, et al. "Deformation-driven shape correspondence". *Computer Graphics Forum*. Vol. 27. 5. Wiley Online Library. 2008, 1431–1439 2.

Chapter 5

Smooth Shells: Multi-Scale Shape Registration with Functional Maps

The shape correspondence literature commonly differentiates between intrinsic approaches, which are invariant to the explicit coordinate embedding of a surface, and extrinsic approaches sensitive to this embedding. Intrinsic maps allow for a compact parameterization through functional maps, which are linear maps in the spectral domain between the Fourier eigenbasis of $\mathcal{X}$ and $\mathcal{Y}$. While this representation helps constrain the optimization, there are well-known failure modes, such as intrinsic symmetries, non-isometric distortions, or topological changes from self-contact. We propose a hybrid approach integrating both intrinsic and extrinsic information. To this end, we model shape poses in the product space of the extrinsic vertex coordinates and the intrinsic Laplace-Beltrami eigenfunctions. The space of admissible transformations is then parameterized through a linear map in low-rank basis representation and a functional map, respectively. For the vertex embedding, we introduce the shell operator $\mathcal{S}_K$, computing a sigmoid weighted low-pass filtering of an input shape $\mathcal{X}$, where the level of detail is modulated by the number of retained eigenfunctions K. For a given pair of surfaces, we optimize for both the unknown correspondence map, and the product space transformation in an alternating optimization scheme. We additionally regularize the extrinsic vertex deformations with an as-rigid-as-possible deformation energy and the functional maps with a local feature preservation term. After each alternating optimization step, we gradually increase the number of Laplace-Beltrami eigenfunctions, resulting in a multi-scale scheme. Since the resulting optimization problem is highly non-convex, we introduce an initialization algorithm based on Markov chain Monte Carlo sampling that performs stochastic search on the space of initial maps. In our experiments, our approach demonstrates superior performance on classical nearly-isometric datasets, as well as specialized benchmarks focusing on different meshing densities, topological changes, and inter-class poses.

Individual contributions

Leading role in realizing the scientific project.

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

Copyright

Marvin Eisenberger, Zorah Lähner, and Daniel Cremers
Smooth Shells: Multi-Scale Shape Registration with Functional Maps
2020 Conference on Computer Vision and Pattern Recognition (CVPR)
DOI: 10.1109/CVPR42600.2020.01228
We include the accepted version of the original publication [2].

RightsLink

Sign in/Register

Smooth Shells: Multi-Scale Shape Registration With Functional Maps

Conference Proceedings:
2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Author: [::Marvin::] [::Eisenberger::]; Zorah Lähner; Daniel Cremers

Publisher: IEEE

Date: 13-19 June 2020

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis online.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

Smooth Shells: Multi-Scale Shape Registration with Functional Maps

Marvin Eisenberger Zorah Lähner Daniel Cremers
Technical University of Munich

Abstract

We propose a novel 3D shape correspondence method based on the iterative alignment of so-called smooth shells. Smooth shells define a series of coarse-to-fine shape approximations designed to work well with multiscale algorithms. The main idea is to first align rough approximations of the geometry and then add more and more details to refine the correspondence. We fuse classical shape registration with Functional Maps by embedding the input shapes into an intrinsic-extrinsic product space. Moreover, we disambiguate intrinsic symmetries by applying a surrogate based Markov chain Monte Carlo initialization. Our method naturally handles various types of noise that commonly occur in real scans, like non-isometry or incompatible meshing. Finally, we demonstrate state-of-the-art quantitative results on several datasets and show that our pipeline produces smoother, more realistic results than other automatic matching methods in real world applications.

1. Introduction

The wide selection of affordable 3D scanning devices in recent years has led to an enormous growth in the amount of 3D shapes and scans available. In contrast to synthetic shapes, real-world scans are often noisy and many properties cannot be guaranteed. For example, topological noise might appear in self-touching areas or the meshing density varies depending on scanning conditions. These distortions were proven to be difficult for state-of-the-art shape correspondence methods [24, 34]. Many traditional methods focus only on the (nearly) isometric case, clearly defined extensions of this like partiality [28], or learn matching for different classes of shapes under certain perturbations [18]. Unfortunately, this requires training data and knowledge about what deformations and noise are to be expected.

In general one can distinguish between intrinsic and extrinsic correspondence methods. Intrinsic methods only use surface properties that are independent of the embedding, for example the Laplace-Beltrami operator. On the other hand, extrinsic methods directly use the 3D embedding of the shapes. While intrinsic methods are invariant to large scale, near-isometric deformations, extrinsic alignment is often more suitable for pairs with topological changes or other non-isometric deformations. A natural step would be to combine both to get the best of both worlds but only few previous approaches venture in this direction [11, 13].

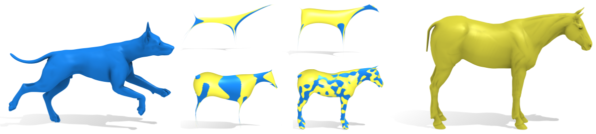


Figure 1: Given a source (left) and target (right) shape we propose a hierarchical smoothing procedure to iteratively align the inputs. First, we align very coarse approximations and then refine until we get correspondences for the original inputs. Among other things, we can handle challenging interclass pairs like matching a dog to a horse and our method is fully automatic, i.e. we do not use any additional information like hand-selected landmarks.

Contribution In this paper we combine intrinsic and extrinsic information by embedding the input shapes into the product space of intrinsic (spectral) and extrinsic (xyz) coordinates. Then, we iteratively align smooth approximations of the two input shapes in this product space which we call smooth shells. Moreover, we propose a Markov chain Monte Carlo initialization strategy to find a meaningful local minimum and disambiguate self-similarities. Overall, we obtain a robust matching pipeline that works out of the box for a broad range of inputs beyond the isometry assumption and in the presence of various types of noise.

2. Related Work

2.1. Shape Correspondence and Registration

Shape correspondence is an extensively studied topic with various applications in Computer Vision and Graphics. Surveys of state-of-the-art methods [55, 47, 52, 46] give a broader overview of existing approaches but here we focus on work that is immediately related to ours.

The Functional Maps [38] paper proposes an elegant formalism to model shape correspondences. The main idea is to model mappings of functions on the input shape to functions on the output shape instead of point-to-point maps. This allows for a compact matrix representation in a low rank basis. Over the last years, the original framework has been extended and applied to various applications [45, 29, 42, 1]. A major challenge in this context is extracting a point-wise correspondence from a Functional Map [44]. Several methods extend the original formalism but most of them are computationally heavy or make restrictive assumptions about the inputs [44, 42, 37]. Another common approach is to take noisy correspondences obtained from a Functional Map and denoise them [16, 57]. However, this only works if the input map is sufficiently accurate. Finally, all the methods listed above are by design prone to produce faulty matches in the presence of intrinsic self-similarities.

Extrinsic methods explicitly deform and align the input shapes in the 3D embedding space. [36, 31] model the deformation with a linear mapping in a low rank basis on the surface of one shape. Like our approach, [15] alternates between calculating a deformation field and correspondences but the volume-preservation constraint restricts the applicability. Many deformation-based methods require expensive preprocessing to apply the deformation model, for example with a deformation graph [49], structural rods [3] or deep learning [18]. Non-rigid ICP methods iteratively align shapes but they rely on a good initialization [27, 4]. However, for many applications we do not have such a previous alignment and in general there is no trivial way to obtain it.

There exist accurate methods to register certain classes of shapes, e.g. humans [7, 2]. Unfortunately, these are highly specialized and depend on class specific features [33], or learn statistical models from data [41]. While these methods perform extremely well for shapes within their classes, they usually do not generalize to arbitrary examples.

2.2. Shape Approximation and Simplification

The idea of mesh simplification by smoothing is investigated thoroughly in previous work. [54] use manifold harmonics for the smoothing. In surface deformation modeling this is usually a two stage algorithm. First, a smoothed version of a shape is deformed and then the details are added back to the surface, see [8] for an overview. Some classical works on shape modeling with smoothing are [19] and [23]. [9] combines this approach with differential coordinates. Although our smooth shells are related to smoothing technique like [54], none of the mentioned approaches use a series of approximations. We propose a novel hierarchical shape smoothing method that is particularly suitable for coarse-to-fine matching.

Shape skeletons offer a lower-dimensional description of the rough geometry of a shape. A recent survey of 3D skeleton methods can be found in [51]. Although the skeletons are usually designed to be easily aligned between different shapes from similar classes, most methods typically only define a single, unique skeleton for each shape. This is useful for a rough matching but does not allow for an iterative refinement of the surface alignment. Similar to our method, [12] extracts a skeleton based on Laplacian-based contraction but aims at getting a unique curve skeleton. Some methods create an entire class of skeletons for each shape [43]. Our method differs from the previously mentioned in that we do not introduce a fixed skeleton for each shape. Instead we construct a whole class of approximations with an increasing level of detail.

3. Background

A correspondence between two input shapes $\mathcal{X}$ and $\mathcal{Y}$ is defined as a point-to-point mapping $\mathbf{P}:\mathcal{X}\to\mathcal{Y}$. Here, a shape is a 2D Riemannian manifold with an embedding in $\mathbb{R}^3$. We use triangular meshes to discretize the surfaces $\mathcal{X}$ and $\mathcal{Y}$ and denote the coordinate matrices as $X\in\mathbb{R}^{N\times 3}$ and $Y\in\mathbb{R}^{M\times 3}$ with N and M vertices respectively.

3.1. Laplace-Beltrami Operator

The Laplace-Beltrami operator $\Delta=\text{div}(\nabla\cdot)$ is an extension of the standard Euclidean Laplacian to manifold domains $\mathcal{X}$. Computing solutions of $\Delta\phi_k=\lambda_k\phi_k$ yields the Laplace-Beltrami eigenfunctions $\{\phi_k\}_{k\in\mathbb{N}}$ which form an orthonormal basis of $L^2(\mathcal{X})$. This allows for a spectral representation of functions $f\in L^2(\mathcal{X})$:

$$f\approx\tilde{f}=\sum_{k=1}^{K}\langle f,\phi_k\rangle\phi_k. \qquad (1)$$

According to the min-max principle, $\tilde{f}$ is an optimal compact approximation of smooth functions $f\in L^2(\mathcal{X})$ [39] with a fixed basis size K. To compute the Laplace-Beltrami operator on triangular meshes, we use a cotangent discretization $\Delta\in\mathbb{R}^{N\times N}$ with lumped mass matrix [40] and we denote its first K eigenvectors as $\Phi_K=(\phi_1,\dots,\phi_N)\in\mathbb{R}^{N\times K}$ (analogously $\Psi_K\in\mathbb{R}^{M\times K}$ for $\mathcal{Y}$).

3.2. Functional Maps

The Functional Map framework [38] is a popular approach to solve for correspondences $\mathbf{P}:\mathcal{X}\to\mathcal{Y}$. In Functional Maps, $\mathbf{P}$ is replaced with a mapping of functions to functions $\mathcal{C}:L^2(\mathcal{X})\to L^2(\mathcal{Y})$. $\mathcal{C}$ is linear and can therefore be compactly written as a matrix $\mathbf{C}\in\mathbb{R}^{K\times K}$:

$$\mathcal{C}=\Psi_K\mathbf{C}\Phi_K^{\dagger}. \qquad (2)$$

To compute $\mathbf{C}$ for a pair of input shapes we need additional information to constrain the solution. Given pairs of

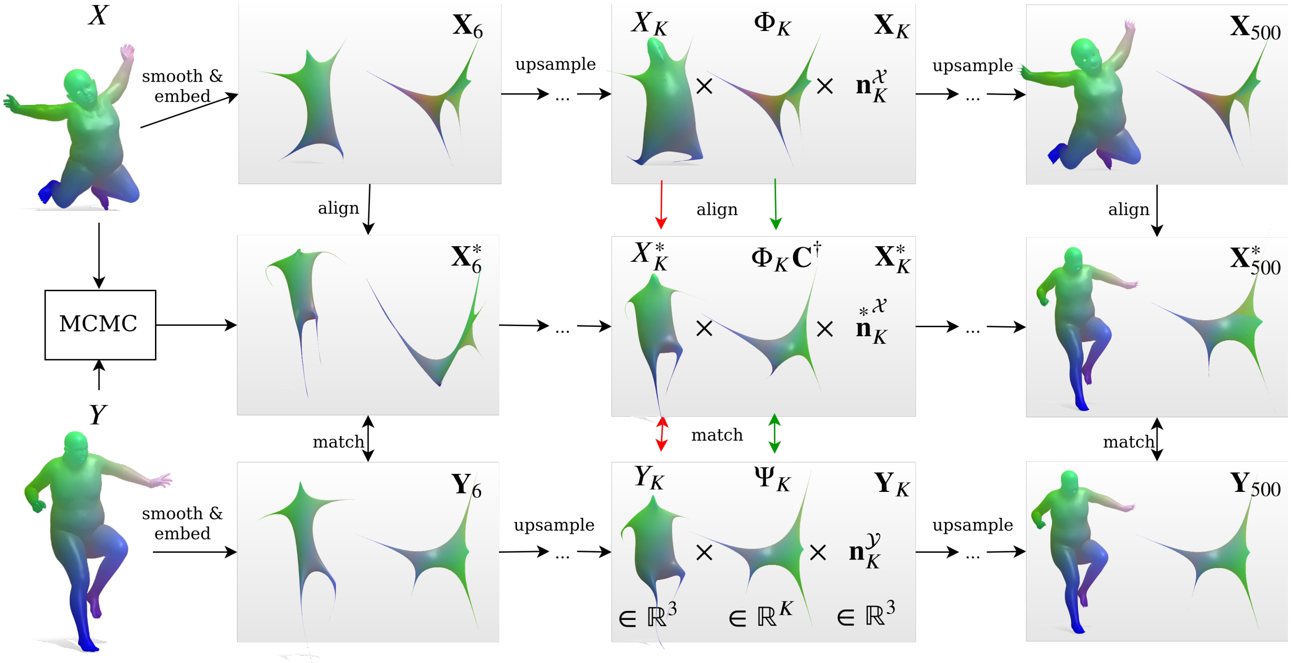


Figure 2: Overview of our pipeline. (Column 1) We initialize our method with the alignment X_6^* from our Markov chain Monte Carlo initialization algorithm, see Section 5 for details. (Columns 2-4) On each level K we embed both shapes in the $K+6$ dimensional product space of the smoothed extrinsic coordinates $X_K \in \mathbb{R}^3$, the intrinsic spectral coordinates $\Phi_K \in \mathbb{R}^K$ and the outer normals $\mathbf{n}_K^{\mathcal{X}} \in \mathbb{R}^3$. In this space we can align $\mathbf{X}_K$ and $\mathbf{Y}_K$ by computing an extrinsic morphing $\tau \in \mathbb{R}^{K\times 3}$ for X_K and a Functional Map $\mathbf{C} \in \mathbb{R}^{K\times K}$ for Φ_K. To visualize the spectral embedding Φ_K only the first three dimensions are shown. Finally, using the aligned $\mathbf{X}_K^*$ and $\mathbf{Y}_K$ we obtain a point-to-point matching $\mathbf{P} : \mathcal{X} \to \mathcal{Y}$ with a nearest neighbor search in $\mathbb{R}^{K+6}$. We repeat this process for 50 iterations with smoothing levels on a logarithmic scale between $K = 6 : 500$. Each iteration is initialized with the previous alignment.

corresponding functions $f_i \in \mathbb{R}^N$ and $g_i \in \mathbb{R}^M$ on the two surfaces, an energy to optimize for $\mathbf{C}$ is:

$$E_{\text{feat}}(\mathbf{C}) := \|\mathbf{C}\Phi_K^\dagger F - \Psi_K^\dagger G\|_F^2. \tag{3}$$

Here, F, G are matrices whose columns are the feature functions f_i, g_i. Possible choices for those features range from pointwise descriptors or surface texture to input landmarks. Another common assumption is that the mapping $\mathbf{P}$ is area preserving which leads to orthogonal Functional Maps $\mathbf{C}^\top\mathbf{C} = \mathbf{I}$, see [38, Theorem 5.1].

3.3. Shape deformation

A different approach is to align the surfaces in the embedding space instead of calculating the correspondence directly. We denote the deformed version of $\mathcal{X}$ with $\mathcal{X}^*$ and impose that $\mathcal{X}^*$ should align with $\mathcal{Y}$. A common choice model is a linear displacement in a low rank basis [36, 31], e.g. the Laplace-Beltrami eigenbasis:

$$X^* = X + \Phi_K \tau. \tag{4}$$

$\tau \in \mathbb{R}^{K\times 3}$ are some displacement coefficients that parameterize the deformation. In the discrete case, the pointwise correspondence is represented by the matrix $\mathbf{P} \in \{0,1\}^{M\times N}$ with $\mathbf{P}^\top\mathbf{1} = \mathbf{1}$. Using the aligned shape X^*, we can recover $\mathbf{P}$ by minimizing the following energy:

$$E_{\text{align,3D}}(\mathbf{P}) := \|\mathbf{P}X^* - Y\|_F^2. \tag{5}$$

This is equivalent to a nearest neighbor search in $\mathbb{R}^3$. In order to get a meaningful correspondence with this approach we need to additionally regularize the deformations $\mathcal{X}^*$. One possibility is to assume that the deformations are as-rigid-as-possible on a local scale:

$$\begin{aligned} E_{\text{arap}}(\tau) := \int_{\mathcal{X}} \int_{\mathcal{N}(x)} \big\| R(x)\big(X(x) - X(y)\big) - \\ \big(X_\tau^*(x) - X_\tau^*(y)\big)\big\|_2^2 \mathrm{d}y\mathrm{d}x. \end{aligned} \tag{6}$$

$\mathcal{N}(x)$ denotes the neighborhood of $x \in \mathcal{X}$ and $R : \mathcal{X} \to \mathrm{SO}(3)$ describes the local rotation, for details see [48].

4. Method

We propose to compute shape correspondences by iteratively aligning a series of coarse-to-fine approximations of the input surfaces $\mathcal{X}$ and $\mathcal{Y}$. This is based on the idea that the alignment of two shapes can end up in unwanted local optima in the presence of non-consistent small scale features. In many cases the rough structure of $\mathcal{X}$ and $\mathcal{Y}$, like the number of extremities, is similar while the fine scale details can differ, see Figure 1. After matching the global features, the local features can be used to refine the alignment. The smooth shells we use as coarse shape approximations are defined in Section 4.1. Section 4.2 explains how we combine extrinsic and intrinsic shape embeddings and Section 4.3 defines our complete matching algorithm.

4.1. Smooth Shells

In this section, we propose a novel shape smoothing operator $\mathcal{S}_K$ that yields smoothed shapes similar to those from spectral surface reconstruction [25]. In comparison, our operator leads to smoother transitions between $\mathcal{S}_K$ and $\mathcal{S}_{K+1}$ which makes it more suitable for a hierarchical alignment.

Spectral Reconstruction Spectral reconstruction [25] smoothes $\mathcal{X}$ by projecting its coordinate function X onto the first K Laplace-Beltrami eigenfunctions:

$$T_K := \mathcal{T}_K(X) = \sum_{k=1}^{K} \big(\phi_k \otimes \phi_k\big) X. \tag{7}$$

Here, $\phi_k \otimes \phi_k$ denotes the outer product of ϕ_k with itself which results in the projection of X onto ϕ_k. Since the eigenfunctions are ordered by frequency, this creates a coarse-to-fine approximation of the original shape. The level of detail is controlled by the number of eigenfunctions K. For small K only the rough geometry is reconstructed, whereas for $K \to \infty$, T_K converges to the original X.

Shell Operator The gradual smoothing from Eq. (7) is useful for hierarchical shape matching. In each iteration we increase K and use the alignment from the previous iteration as an initialization. However, in many cases the refinement with spectral reconstruction leads to undesirable artifacts. Especially the first few K projections from Eq. (7) cause large disparities between reconstructions. This makes the alignment from the previous iteration less useful for the next step. We introduce the *shell operator $\mathcal{S}_K$* to circumvent this issue:

$$X_K := \mathcal{S}_K(X) := \sum_{k=1}^{\infty} \frac{1}{1+\exp\big(\sigma(k-K)\big)} \big(\phi_k \otimes \phi_k\big) X. \tag{8}$$

Just like spectral reconstruction, $\mathcal{S}_K$ smooths X using a projection on ϕ_k. However, instead of truncating the spectral coordinates at a certain K, we introduce a gradual truncation with sigmoid weights. Those are close to 1 if $k \ll K$ and decay to 0 when $k \gg K$. This guarantees that the displacement from $\mathcal{S}_K$ to $\mathcal{S}_{K+1}$ is reasonably bounded. For small σ the transition becomes smoother, whereas for $\sigma \to \infty$ the sigmoid function converges to the indicator function $\mathbf{1}_{\{k \leq K\}}$ which corresponds to spectral reconstruction. In particular, we can show the following smoothness result for $\mathcal{S}$:

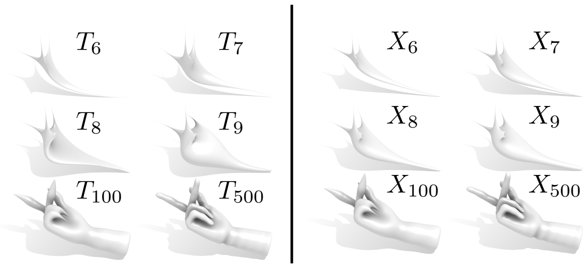


Figure 3: At first glance there is no significant difference between spectral reconstruction T_K and smooth shells $X_K := \mathcal{S}_K(X)$. They both converge to X for $K \to \infty$, and for high indices $K \gg 50$ they are indistinguishable. The crucial difference lies in their upsampling behaviour. While smooth shells transition smoothly from X_K to X_{K+1}, consecutive shapes T_K tend to have large displacements and are therefore less suitable for iterative alignment methods.

Theorem 1. *(Transition smoothness of $\mathcal{S}$)*
Let $\mathcal{X}$ be a shape with coordinate function $X \in L^2$, then the geometric difference of state X_K and X_{K+1} is bounded by the upsampling variance σ in following way:

$$\frac{\big\|\mathcal{S}_{K+1}(X) - \mathcal{S}_K(X)\big\|_{L^2}}{\big\|\mathcal{S}_{K+1}(X)\big\|_{L^2}} \leq |1 - e^{-\sigma}| = \mathcal{O}(\sigma), \sigma \to 0. \tag{9}$$

We provide a proof in Appendix B. See Figure 3 for an illustration of the practical implications and Table 1 for a quantitative comparison to spectral reconstruction.

4.2. Intrinsic-extrinsic Embedding

Intrinsic and extrinsic methods are often depicted as opposing viewpoints and, although there are some notable exceptions [11, 13], only few methods try to combine them. Our deformation model combines shape alignment in both intrinsic and extrinsic space. Functional Maps is based on rigid ICP alignment of the spectral coordinates $\Phi_K \in \mathbb{R}^{N \times K}$ of $\mathcal{X}$ and $\Psi_K \in \mathbb{R}^{M \times K}$ of $\mathcal{Y}$ in the K-dimensional spectral domain [38].

$$\Phi_K \mathbf{C}^\dagger \approx \Psi_K.$$

On the other hand, extrinsic methods typically align the 3-dimensional geometry as described in Eq. (4):

$$X_K^* = X_K + \Phi_K \tau \approx Y_K.$$

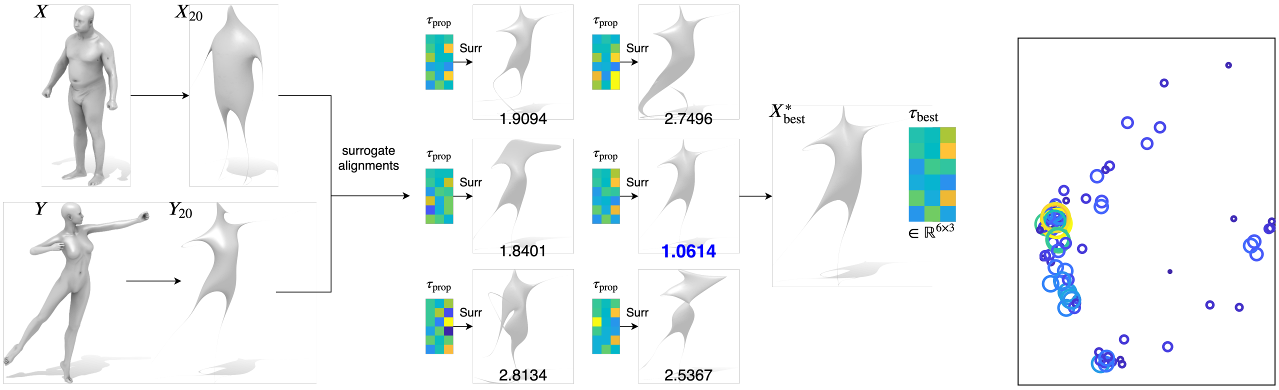


Figure 4: (left) Overview of of our MCMC initialization method. We sample potential initial alignments $\tau_{\text{prop}} \in \mathbb{R}^{6\times3}$ and rate them using surrogate runs (see Section 5). Each proposal τ_{prop} is assigned a mark $E(\text{Surr}(\tau_{\text{prop}})) > 0$ based on the alignment quality of the current surrogate. In the shown example, the best objective is $E = 1.0614$ and indeed this sample visually shows the tightest alignment of X^*_{20} and Y_{20}. (right) 2D embedding with multi-dimensional scaling of all τ_{prop} used for one initialization. Samples with small objective values have big, yellow circles and big objectives correspond to small, blue circles. Evidently there is a big cluster around the optimal (yellow) circle which shows that our algorithm is able to determine the optimal initialization with high confidence.

We combine intrinsic and extrinsic alignment in order to gain both their advantages. To this end, we embed the inputs $\mathcal{X}$ and $\mathcal{Y}$ in the product space of the *intrinsic* (spectral coordinates Φ_K) and the *extrinsic* (smoothed Cartesian coordinates X_K and outer normals $\mathbf{n}_K^{\mathcal{X}}$ of $\mathcal{X}$) coordinates:

$$\mathbf{X}_K := \left(\Phi_K, X_K, \mathbf{n}_K^{\mathcal{X}}\right) \in \mathbb{R}^{N\times(K+6)}. \tag{10a}$$

$$\mathbf{Y}_K := \left(\Psi_K, Y_K, \mathbf{n}_K^{\mathcal{Y}}\right) \in \mathbb{R}^{M\times(K+6)}. \tag{10b}$$

Using the normals makes the embedding more descriptive because they convey information about the inside-outside orientation of each point. Using both the Functional Map $\mathbf{C}$ and the extrinsic deformation τ (see Eq. (4)) now yields the morphed embedding $\mathbf{X}_K^*$:

$$\mathbf{X}_K^* := \left(\Phi_K \mathbf{C}^\dagger, X_K + \Phi_K \tau, \overset{*}{\mathbf{n}}{}_K^{\mathcal{X}}\right) \in \mathbb{R}^{N\times(K+6)}. \tag{11}$$

$\overset{*}{\mathbf{n}}{}_K^{\mathcal{X}}$ are the normals of $X_K + \Phi_K \tau$. The next section will go into detail on how to compute C and τ.

4.3. Hierarchical Matching

Putting everything together, we can define a hierarchical correspondence algorithm with the following energy:

$$E(\mathbf{P}, \mathbf{C}, \tau) := \|\mathbf{P}\mathbf{X}_K^* - \mathbf{Y}_K\|_F^2 + \lambda_{\text{feat}} E_{\text{feat}}(\mathbf{C}) + \lambda_{\text{arap}} E_{\text{arap}}(\tau). \tag{12}$$

The regularization terms E_{feat} and E_{arap} are defined in Eq. (3) and Eq. (6) respectively. For the former we use the SHOT [53] and HKS [50] descriptors. To minimize the energy E we choose an alternating optimization strategy. In particular, we first fix the correspondences $\mathbf{P}$ and optimize for the alignment $(\mathbf{C}, \tau)$ and then do the same vice versa. This is a common approach for both intrinsic [38] and extrinsic [31, 32, 36] matching methods. Our overall matching algorithm is the following:

Algorithm 2. *(Hierarchical Matching)*

```
1. Input: $\mathcal{X}, \mathcal{Y}$
2. For $k \in \{0, \dots, |\mathcal{K}| - 1\}$:
    2.1 $\mathbf{P}_{k+1} := \arg\min_{\mathbf{P}_{mn} \in \{0,1\}, \mathbf{P}^\top \mathbf{1} = \mathbf{1}} E(\mathbf{P}, \mathbf{C}_k, \tau_k)$.
    2.2 $(\tau_{k+1}, \mathbf{C}_{k+1}) := \arg\min_{\tau, \mathbf{C}^\top \mathbf{C} = \mathbf{I}} E(\mathbf{P}_{k+1}, \mathbf{C}, \tau)$.
3. Output: $\mathbf{P}_{|\mathcal{K}|}, \mathbf{X}^*_{|\mathcal{K}|}$.
```

The decomposition of the optimization problem E results in more tractable subproblems. For $\mathbf{P}$ this is a nearest neighbor search, for $\mathbf{C}$ a Procrustes problem and for τ a nonlinear least squares problem. The first two can be solved in closed form, for the last one we use Gauss-Newton optimization. Our method now repeatedly solves those optimization problems with shells of an increasing level of detail $K \in \mathcal{K}$ on a logarithmic scale between $K_{\text{init}} = 6$ and $K_{\text{max}} = 500$. See Figure 2 for a visualization of Algorithm 2.

5. Initialization: Surrogate based Markov chain Monte Carlo Sampling

Self-similarities are still a challenging problem for state-of-the-art shape correspondence methods and many struggle to distinguish them without proper initialization [56, 17, 35]. Even for humans it is difficult to distinguish between the legs/arms of an animal without any context. In other words, our energy from Eq. (12) is highly non-convex with a multitude of local minima. Unfortunately, there is no obvious way to compute a meaningful initial alignment for all classes of shapes. We propose an indirect approach to this using Markov chain Monte Carlo (MCMC) sampling.

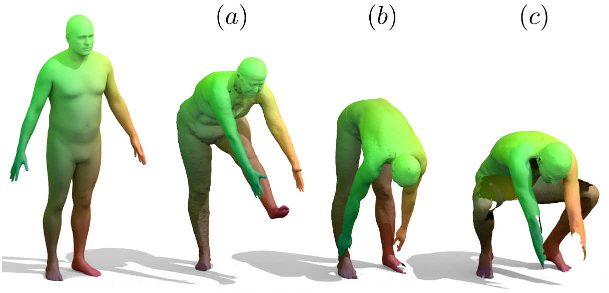


Figure 6: Example matchings for real scans from the FAUST [6] dataset. The shapes have very high resolution (200k vertices) and contain scanning noise. The FAUST interclass challenge consists of (a) different humans that are (b) subject to topological changes and (c) extreme degrees of noise and partiality. Like [18] we match a template (left) to each target. Here, correspondences are color coded such that matching points have the same color.

Surrogate runs This approach is based on efficiently exploring the space of initial poses instead of heuristically picking one. We assign a probability distribution to the displacement parameter $\tau \in \mathbb{R}^{K_{\text{init}} \times 3}$ and sample from this distribution. In particular, we set the prior for τ to the standard normal distribution $\mathcal{N}(0, \mathbf{I})$ and the negative log likelihood proportional to the objective value E. By design, this yields samples τ that have a high objective value E. Each τ is ranked according to the objective function E from Eq. (12) and the lowest energy result is used to initialize the full pipeline.

To evaluate E, we run a low cost version of the full pipeline, a *surrogate run*, with $K_{\max} = 20$, no regularizers $\lambda_{\text{feat}}, \lambda_{\text{arap}} := 0$ and downsampled versions of the input shapes to 1000 vertices. We evaluate $N_{\text{prop}} = 100$ different proposals τ_{prop}. Those can be run in parallel with an average runtime of 0.46 seconds per surrogate. See Figure 4 for a visualization of this strategy and see Appendix A for pseudo code of our MCMC algorithm as well as the implementation in the supplementary material.

6. Experiments

We apply our pipeline to various, challenging matching tasks using two metrics to measure the quality of a matching. The first one is the *accuracy*, defined as the geodesic distance to ground truth matches, see Section 6.1. The second is the *smoothness* of the correspondence $\mathbf{P}$ which we quantify using the conformal distortion of triangles, see Section 6.2. To show that our method can be used out of the box, we use the same set of parameters for all experiments and do not require additional information except for the inputs $\mathcal{X}$ and $\mathcal{Y}$. See our implementation in the supplementary material for more details. Additionally, we perform an ablation study in Section 6.3 and a runtime analysis in the Appendix C to further investigate our method. Finally, there are more qualitative examples of matchings and style transfer in the Appendix.

6.1. Shape correspondence

We evaluate the matching accuracy of our method according to the Princeton benchmark protocol [22] on multiple datasets. Given the ground-truth match (x, y^*), the error of the calculated match (x, y) is given by the geodesic dis-

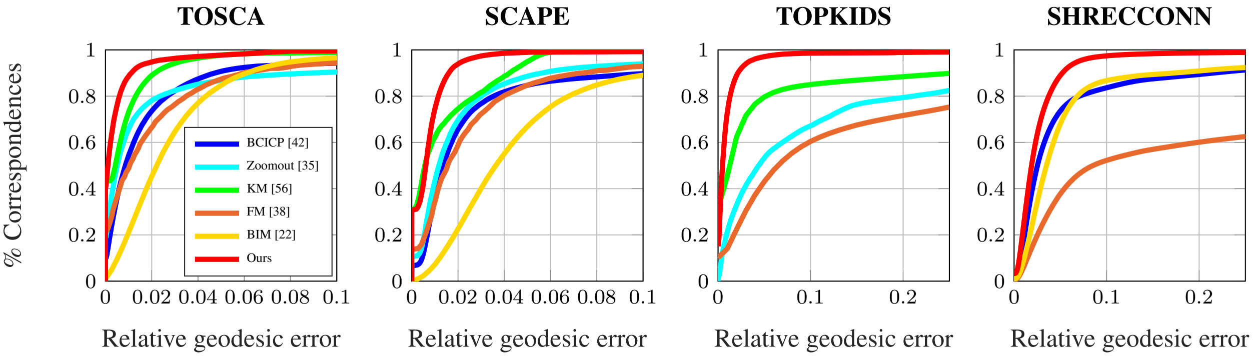


Figure 5: Our matching accuracies for four datasets in comparison to other popular fully automatic shape correspondence methods. For more details on the datasets, see Section 6.1.

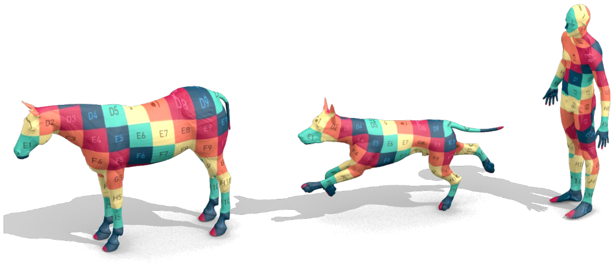


Figure 7: Example texture transfers with our method for challenging interclass examples. The texture defined on the source shape (horse) is transferred to two individual target shapes (dog and human).

tance between y and y^* normalized by the diameter of $\mathcal{Y}$:

$$\epsilon(x) = \frac{\mathrm{d}_{geo}(y, y^*)}{\sqrt{\mathrm{area}(\mathcal{Y})}} \tag{13}$$

TOSCA, SCAPE, TOPKIDS, SHRECCONN Two datasets contain synthetic shapes with isometric pairs, TOSCA [10] contains 76 shapes of humans and animals, SCAPE [5] contains 72 poses of the same person. TOPKIDS [24] contains 25 poses of a human child with self intersections. These shapes are also synthetic but topological changes from real scans are simulated with merged meshing at self-touching areas. The SHREC'19 Connectivity [34] dataset contains 430 pairs of human shapes from different classes with severe differences in the meshing ranging from template sized shapes ($N \approx 5000$ vertices) to real scans ($N > 200$K), and varying vertex densities in different areas. We compare our matching accuracy on these datasets to other fully-automatic matching methods, see Figure 5.

FAUST The FAUST [6] dataset contains 300 real scans of different humans in various poses, see Figure 6. Besides being high resolution $N \approx 200$K with non-compatible meshing, the shapes are noisy and highly non-isometric. Additionally, there are various poses with topological changes due to self touching parts. To address this issue, we do not compute the correspondence directly for a given pair of shapes but, like 3D-CODED [18], FARM [33] and LBS-AE [26], use an intermediate template from [30] to compute correspondences for two scans. This allows our method to separate topological changes and deal with noisy geometry, otherwise the as-rigid-as-possible assumption leads to faulty deformations. The accuracy (in cm) of the best methods for the FAUST [6] interclass challenge reported on the website *faust.is.tue.mpg.de* are:

Method	3D-CODED	SP	Ours	LBS	FARM	FMNet
Error	2.878 (4.883*)	3.126	3.929	4.079	4.123	4.826

For 3D-CODED (*) refers to the unsupervised version. The striking observation is that our method is on par with the state-of-the-art without specializing on the class of human shapes. Ours is the only method listed here that does not train on human shapes (3D-CODED, FMNet) or makes strong modelling assumptions holding only for humans (Stitched Puppet (SP), LBS-AE, FARM). We did not specifically tune parameters for this challenge.

6.2. Map Smoothness

A map with good accuracy can still produce artifacts when transferring information form surface $\mathcal{X}$ to $\mathcal{Y}$ because small scale noise typically does not have a severe effect on the geodesic matching error (13). This behavior is, however, prohibitive for applications like meshing, texture or normal map transfer, see Figure 7. The conformal distortion of each triangle after deformation measures the local consistency of a matching [21, Eq. (3)]. This allows for a quantification of the smoothness of the map $\mathbf{P}$.

Figure 8 shows the conformal triangle distortion of our method on the SCAPE dataset. Remarkably, the deformations obtained with our method are even smoother than the ground-truth provided in [5]. The reason for that lies in the way the authors construct this ground-truth. In order to transfer the meshes they use a classical nonrigid registration algorithm [20] to register a template in a canonical pose to 71 noisy scans of a person. This method requires ~ 150 markers to get a faithful alignment, some of which are hand-selected. The main concern was to obtain a possibly tight alignment of the markers. However, in practice the markers are not perfectly placed and these small deviations lead to distorted triangles. In comparison to that, we align the templates without any markers while explicitly using an ARAP penalization term. This evidently leads to smoother deformations and the few triangles that get distorted are typically not artifacts of random noise but rather in meaningful places like the armpits or the abdomen of the person in Figure 8.

6.3. Ablation study

We assess the effect of the different components of our method in the ablation study in Table 1. The main insight is that there is an intricate interplay of the different subparts of our method and the accuracy drops significantly if any part is removed. In particular, the MCMC initialization strategy is vital. Without it our deformation based approach is extremely prone to run into suboptimal local minima which leads to a failure rate of over 83%. Remarkably, even when our rigid initialization strategy (see end of Appendix A) is replaced with random rigid poses the failure rate is only around $38 - 50\%$. In many cases, our MCMC algorithm is able to find the correct pose, even in the presence of large scale rotational displacements of the inputs.

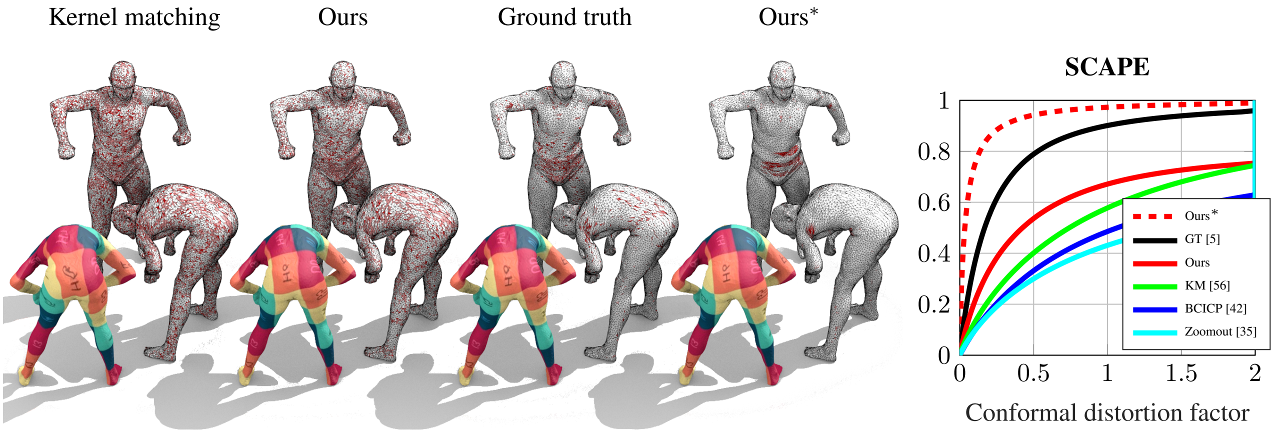


Figure 8: Correspondence smoothness measured on the SCAPE dataset using conformal distortion of triangles. (Left) Qualitative comparison – red triangles are distorted whereas white triangles preserve the angles. Our deformed mesh shows the most meaningful, artifact-free result. Additionally, we provide an example texture transfer to prove that our deformed mesh (Ours*) is the most useful one. (Right) Accumulated distortions for all 71 pairs in the dataset. There are two possibilities to transfer the mesh of $\mathcal{X}$ to the reference pose $\mathcal{Y}$. Either we use the deformed geometry $\mathcal{X}^*$ directly (Ours*) or we snap it to the surface $\mathcal{Y}$ post alignment using the map $\mathbf{P} : \mathcal{X} \to \mathcal{Y}$ (Ours). The former is only possible for our method because it is the only one here to calculate a deformation instead of only a correspondence. See Section 6.2 for more details.

SCAPE	Ours	$\lambda_{\text{feat}} = 0$	$\lambda_{\text{arap}} = 0$	Extr. only	Intr. only	w/o normals	w/o MCMC	random rigid	spectral rec.
Avg. error	0.0088	0.0211	0.0147	0.0344	0.0121	0.0115	0.0568	0.1163	0.0139
Failure rate	0	0.2676	0.0282	0.7606	0.2254	0.0141	0.8310	0.4930	0.0282
Avg. Distortion	0.1287	0.1171	0.1604	0.1322	0.1539	0.1310	0.2594	0.2055	0.1305
TOSCA	Ours	$\lambda_{\text{feat}} = 0$	$\lambda_{\text{arap}} = 0$	Extr. only	Intr. only	w/o normals	w/o MCMC	random rigid	spectral rec.
Avg. error	0.0056	0.0075	0.0066	0.0441	0.0205	0.0076	0.1039	0.0694	0.0098
Failure rate	0	0.2083	0.0833	0.7500	0.4028	0.0694	0.8611	0.3889	0.1111
Avg. Distortion	0.1654	0.1641	0.1926	0.1710	0.2239	0.1716	0.3485	0.2829	0.1666

Table 1: Ablation study on TOSCA and SCAPE. We turn off certain parts of the method or replace it with an alternative to assess its necessity and compare the average geodesic error in $\%$ of the diameter, the failure rate and the average conformal distortion for each setting. The failure rate is the number of pairs in $\%$ where the geodesic error is twice as high than (Ours). $\lambda_{\text{feat}}, \lambda_{\text{arap}} = 0$ turns off the regularizers, *Extr./Intr. only, w/o normals* removes one part of the embedding (see Eq. (10)), *w/o MCMC* removes the initialization, *random rigid* replaces our rigid alignment strategy with random rigid poses, and *spectral rec.* replaces smooth shells with spectral reconstruction (see Eq. (7)). The main insight is that the accuracy decreases whenever one of the components is removed. The conformal distortion is rather stable, except when a big percentage of the results are totally broken (e.g. w/o MCMC) or the as-rigid-as-possible regularizer is removed.

7. Conclusion

We have presented a novel approach to shape correspondence that combines geometric and spectral alignment by embedding the input shapes into an extrinsic-intrinsic product space. Our method introduces smooth shells as a coarse-to-fine shape approximation with minimal geometry changes between iterations. This is valuable for hierarchical approaches. Furthermore, we solve the problem of self-similarities by starting with an efficient surrogate based Markov chain Monte Carlo approach in which the deformation energy is used to find the optimal initialization. Finally, our method produces state-of-the-art results on established isometry datasets as well as two datasets which focus on specific noise, namely different meshing and topology changes. In the FAUST real-scan interclass challenge we are on par with the state-of-the-art although we do not train specifically for this set-up. All results were achieved with the same set of parameters which shows great generality.

Acknowledgements

We would like to thank Emanuele Rodolà for useful discussions and Jing Ren for help with running the ZoomOut experiments. We gratefully acknowledge the support of the ERC Consolidator Grant ”3D Reloaded”.

References

[1] Hierarchical functional maps between subdivision surfaces. *Computer Graphics Forum*, 35(5):55 – 73, 2019.

[2] Non-rigid registration under anisotropic deformations. *Computer Aided Geometric Design*, 71:142 – 156, 2019.

[3] Ibraheem Alhashim, Kai Xu, Yixin Zhuang, Junjie Cao, Patricio Simari, and Hao Zhang. Deformation-driven topology-varying 3d shape correspondence. *ACM Trans. Graph.*, 34(6):236:1–236:13, Oct. 2015.

[4] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[5] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.

[7] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: Registering human bodies in motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[8] Mario Botsch and Olga Sorkine. On linear variational surface deformation methods. *IEEE transactions on visualization and computer graphics*, 14(1):213–230, 2007.

[9] Mario Botsch, Robert Sumner, Mark Pauly, and Markus Gross. Deformation transfer for detail-preserving surface editing. In *Vision, Modeling & Visualization*, pages 357–364. Citeseer, 2006.

[10] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes.* Springer, 2008. http://tosca.cs.technion.ac.il/book/resources$_data.html$.

[11] Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Topology-invariant similarity of nonrigid shapes. *International Journal of Computer Vision*, 81:281–301, 2009.

[12] Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhinxun Su. Point cloud skeletons via laplacian based contraction. In *Proceedings of the 2010 Shape Modeling International Conference*, Shape Modeling International Conference (SMI), pages 187–197, Washington, DC, USA, 2010. IEEE Computer Society.

[13] Etienne Corman, Justin Solomon, Mirela Ben-Chen, Leonidas Guibas, and Maks Ovsjanikov. Functional characterization of intrinsic and extrinsic geometry. *ACM Trans. Graph.*, 36(2), Mar. 2017.

[14] RM Dyke, C Stride, Y-K Lai, Paul L Rosin, Mathieu Aubry, Amit Boyarski, Alexander M Bronstein, Michael M Bronstein, Daniel Cremers, Matthew Fisher, et al. Shape correspondence with isometric and non-isometric deformations. 2019.

[15] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38, pages 1–12. Wiley Online Library, 2019.

[16] Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. *Computer Graphics Forum (CGF)*, 36(5), August 2017.

[17] Danielle Ezuz, Behrend Heeren, Omri Azencot, Martin Rumpf, and Mirela Ben-Chen. Elastic correspondence between triangle meshes. In *Computer Graphics Forum (CGF)*, volume 38, 2019.

[18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] Igor Guskov, Wim Sweldens, and Peter Schröder. Multiresolution signal processing for meshes. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 325–334. Citeseer, 1999.

[20] Dirk Haehnel, Sebastian Thrun, and Wolfram Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *IJCAI*, volume 3, pages 915–920, 2003.

[21] Kai Hormann and Günther Greiner. Mips: An efficient global parametrization method. Technical report, Erlangen-Nuernberg University (Germany) Computer Graphics Group, 2000.

[22] Vladimir G Kim, Yaron Lipman, and Thomas A Funkhouser. Blended intrinsic maps. *Transactions on Graphics (TOG)*, 30(4), 2011.

[23] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *ACM SIGGRAPH*, volume 98, pages 105–114, 1998.

[24] Zorah Lähner, Emanuele Rodolà, Michael M Bronstein, Daniel Cremers, Oliver Burghard, Luca Cosmo, Andreas Dieckmann, Reinhard Klein, and Yusuf Sahillioglu. Shrec'16: Matching of deformable shapes with topological noise. *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2:11, 2016.

[25] Bruno Lévy and Hao Richard Zhang. Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses*, page 8. ACM, 2010.

[26] 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.

[27] Hao Li, Robert W. Sumner, and Mark Pauly. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (CGF)*, 27(5), 2008.

[28] Or Litany, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2):1681–1707, 2017.

[29] Or Litany, Emanuele Rodolà, Alex M Bronstein, Michael M Bronstein, and Daniel Cremers. Non-rigid puzzles. *Computer Graphics Forum (CGF), Proceedings of Symposium on Geometry Processing (SGP)*, 35(5), 2016.

[30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015.

[31] Jiayi Ma, Ji Zhao, Jinwen Tian, Alan L Yuille, and Zhuowen Tu. Robust point matching via vector field consensus. *IEEE Transactions on Image Processing*, 23(4):1706–1721, 2014.

[32] Jiayi Ma, Ji Zhao, and Alan L Yuille. Non-rigid point set registration by preserving global and local structures. *IEEE Transactions on image Processing*, 25(1):53–64, 2016.

[33] Riccardo Marin, Simone Melzi, Emanuele Rodolà, and Umberto Castellani. FARM: functional automatic registration method for 3d human bodies. *CoRR*, abs/1807.10517, 2018.

[34] Simone Melzi, Riccardo Marin, Emanuele Rodolà, and Umberto Castellani. Matching humans with different connectivity. *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2019.

[35] Simone Melzi, Jing Ren, Emanuele Rodola, Maks Ovsjanikov, and Peter Wonka. Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019.

[36] Andriy Myronenko and Xubo Song. Point set registration: Coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, 2010.

[37] Dorian Nogneng, Simone Melzi, Emanuele Rodolà, Umberto Castellani, Michael M Bronstein, and Maks Ovsjanikov. Improved functional mappings via product preservation. *Computer Graphics Forum (CGF)*, 37(2), May 2018.

[38] 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, 2012.

[39] Beresford N Parlett. *The symmetric eigenvalue problem*, volume 20. siam, 1998.

[40] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *EXPERIMENTAL MATHEMATICS*, 2:15–36, 1993.

[41] 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, aug 2015.

[42] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6):248:1–248:16, Dec. 2018.

[43] Dennie Reniers, Jarke van Wijk, and Alexandru Telea. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):355–368, Mar. 2008.

[44] Emanuele Rodolà, Michael Moeller, and Daniel Cremers. Point-wise Map Recovery and Refinement from Functional Correspondence. In *Vision, Modeling and Visualization*. The Eurographics Association, 2015.

[45] Emanuele Rodolà, Luca Cosmo, Michael Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum (CGF)*, 2016.

[46] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, pages 1–17, 2019.

[47] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.*, 25(5):578–596, 2007.

[48] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[49] Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. 2007.

[50] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[51] Andrea Tagliasacchi, Thomas Delame, Michaela Spagnuolo, Nina Amenta, and Alexandru Telea. 3d skeletons: A state-of-the-art report. In *Computer Graphics Forum (CGF) (Proceedings of Eurographics)*, volume 35, 2016.

[52] Gary K. L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, July 2013.

[53] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. *In Proceedings of European Conference on Computer Vision (ECCV)*, 16(9):356–369, 2010.

[54] Bruno Vallet and Bruno Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library, 2008.

[55] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[56] Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodolà, Alex M. Bronstein, Michael M. Bronstein, Ron Kimmel, and Daniel Cremers. Efficient deformable shape correspondence via kernel matching. In *International Conference on 3D Vision (3DV)*, October 2017.

[57] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex M Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Chapter 6

Hamiltonian Dynamics for Real-World Shape Interpolation

A fundamental limitation of shape interpolation approaches in the literature is their reliance on perfect correspondence maps. For a given pair of input poses $\mathcal{X}$ and $\mathcal{Y}$, existing approaches often define interpolation sequences as geodesics in an underlying shape space. While this formulation is elegant and successful in a broad range of interpolation tasks, it is mostly designed for computer graphics applications with synthetic data, requiring exact input correspondences. We address this fundamental limitation and devise a robust, physically meaningful interpolation approach. Our framework is based on Hamiltonian dynamics, modeling interpolation sequences in the six-dimensional product space of surface points and momenta. We specify our Hamiltonian energy by combining a surface distortion energy, motivated by shape space approaches, with a momentum preservation term. For any initial velocity field, we then integrate Hamilton's equations with an implicit Euler scheme. In order to predict an interpolation sequence that deforms $\mathcal{X}$ into $\mathcal{Y}$, we optimize for the initial velocities with gradient descent. We parameterize the velocity fields at each time-step with divergence-free vector fields proposed in [3]. We further introduce an anisotropic variant of the popular as-rigid-as-possible deformation energy to allow for more complex local distortions. Our method yields physically plausible interpolation sequences, where the Hamiltonian dynamics deformation prior serves as a strong regularizer for the intermediate poses. Thus, compared to existing shape space approaches, it does not require exact input correspondences. We demonstrate its robustness on several deformable shape datasets, where imperfect, noisy input maps are obtained from state-of-the-art correspondence approaches. Moreover, our parameterization of shape deformation with extrinsic, divergence-free fields guarantees that the obtained poses are free of self-intersections and preserve volume. Finally, we show compelling qualitative results, including shape interpolation, shape extrapolation, correspondence refinement, and interpolation of partial poses.

Individual contributions

Leading role in realizing the scientific project.

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

Copyright

Reprinted, with permission, from Springer, Cham.
Marvin Eisenberger, and Daniel Cremers
Hamiltonian Dynamics for Real-World Shape Interpolation
The 2020 European Conference on Computer Vision (ECCV)
DOI: 10.1007/978-3-030-58548-8_11

SPRINGER NATURE LICENSE
TERMS AND CONDITIONS

Dec 05, 2023

This Agreement between Mr. Marvin Eisenberger ("You") and Springer Nature ("Springer Nature") consists of your license details and the terms and conditions provided by Springer Nature and Copyright Clearance Center.

License Number	5682370642016
License date	Dec 05, 2023
Licensed Content Publisher	Springer Nature
Licensed Content Publication	Springer eBook
Licensed Content Title	Hamiltonian Dynamics for Real-World Shape Interpolation
Licensed Content Author	Marvin Eisenberger, Daniel Cremers
Licensed Content Date	Jan 1, 2020
Type of Use	Thesis/Dissertation
Requestor type	academic/university or research institute
Format	electronic
Portion	full article/chapter
Will you be translating?	no
Circulation/distribution	1 - 29
Author of this Springer Nature content	yes

Title of new work	Fully Automatic Deformable 3D Shape Correspondence and Interpolation
Institution name	Technical University of Munich
Expected presentation date	May 2024
Order reference number	license_hamiltonian
Requestor Location	Mr. Marvin Eisenberger Muehlfeldweg 22 Garching, Bavaria 85748 Germany Attn: Mr. Marvin Eisenberger
Billing Type	Invoice
Billing Address	Mr. Marvin Eisenberger Muehlfeldweg 22 Garching, Germany 85748 Attn: Mr. Marvin Eisenberger
Total	0.00 USD
Terms and Conditions	

Springer Nature Customer Service Centre GmbH Terms and Conditions

The following terms and conditions ("Terms and Conditions") together with the terms specified in your [RightsLink] constitute the License ("License") between you as Licensee and Springer Nature Customer Service Centre GmbH as Licensor. By clicking 'accept' and completing the transaction for your use of the material ("Licensed Material"), you confirm your acceptance of and obligation to be bound by these Terms and Conditions.

1. Grant and Scope of License

1. 1. The Licensor grants you a personal, non-exclusive, non-transferable, non-sublicensable, revocable, world-wide License to reproduce, distribute, communicate to the public, make available, broadcast, electronically transmit or create derivative works using the Licensed Material for the purpose(s) specified in your RightsLink Licence Details only. Licenses are granted for the specific use requested in the order and for no other use, subject to these Terms and Conditions. You acknowledge and agree that the rights granted to you under this License do not include the right to

modify, edit, translate, include in collective works, or create derivative works of the Licensed Material in whole or in part unless expressly stated in your RightsLink Licence Details. You may use the Licensed Material only as permitted under this Agreement and will not reproduce, distribute, display, perform, or otherwise use or exploit any Licensed Material in any way, in whole or in part, except as expressly permitted by this License.

1. 2. You may only use the Licensed Content in the manner and to the extent permitted by these Terms and Conditions, by your RightsLink Licence Details and by any applicable laws.

1. 3. A separate license may be required for any additional use of the Licensed Material, e.g. where a license has been purchased for print use only, separate permission must be obtained for electronic re-use. Similarly, a License is only valid in the language selected and does not apply for editions in other languages unless additional translation rights have been granted separately in the License.

1. 4. Any content within the Licensed Material that is owned by third parties is expressly excluded from the License.

1. 5. Rights for additional reuses such as custom editions, computer/mobile applications, film or TV reuses and/or any other derivative rights requests require additional permission and may be subject to an additional fee. Please apply to journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

2. Reservation of Rights

Licensor reserves all rights not expressly granted to you under this License. You acknowledge and agree that nothing in this License limits or restricts Licensor's rights in or use of the Licensed Material in any way. Neither this License, nor any act, omission, or statement by Licensor or you, conveys any ownership right to you in any Licensed Material, or to any element or portion thereof. As between Licensor and you, Licensor owns and retains all right, title, and interest in and to the Licensed Material subject to the license granted in Section 1.1. Your permission to use the Licensed Material is expressly conditioned on you not impairing Licensor's or the applicable copyright owner's rights in the Licensed Material in any way.

3. Restrictions on use

3. 1. Minor editing privileges are allowed for adaptations for stylistic purposes or formatting purposes provided such alterations do not alter the original meaning or intention of the Licensed Material and the new figure(s) are still accurate and representative of the Licensed Material. Any other changes including but not limited to, cropping, adapting, and/or omitting material that affect the meaning, intention or moral rights of the author(s) are strictly prohibited.

3. 2. You must not use any Licensed Material as part of any design or trademark.

3. 3. Licensed Material may be used in Open Access Publications (OAP), but any such reuse must include a clear acknowledgment of this permission visible at the same time as the figures/tables/illustration or abstract and which must indicate that the Licensed Material is not part of the governing OA license but has been reproduced with permission. This may be indicated according to any standard referencing system but must include at a minimum 'Book/Journal title, Author, Journal Name (if applicable), Volume (if applicable), Publisher, Year, reproduced with permission from SNCSC'.

4. STM Permission Guidelines

4. 1. An alternative scope of license may apply to signatories of the STM Permissions Guidelines ("STM PG") as amended from time to time and made available at https://www.stm-assoc.org/intellectual-property/permissions/permissions-guidelines/.

4. 2. For content reuse requests that qualify for permission under the STM PG, and which may be updated from time to time, the STM PG supersede the terms and conditions contained in this License.

4. 3. If a License has been granted under the STM PG, but the STM PG no longer apply at the time of publication, further permission must be sought from the Rightsholder. Contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

5. Duration of License

5. 1. Unless otherwise indicated on your License, a License is valid from the date of purchase ("License Date") until the end of the relevant period in the below table:

Reuse in a medical communications project	Reuse up to distribution or time period indicated in License
Reuse in a dissertation/thesis	Lifetime of thesis
Reuse in a journal/magazine	Lifetime of journal/magazine
Reuse in a book/textbook	Lifetime of edition
Reuse on a website	1 year unless otherwise specified in the License
Reuse in a presentation/slide kit/poster	Lifetime of presentation/slide kit/poster. Note: publication whether electronic or in print of presentation/slide kit/poster may require further permission.
Reuse in conference proceedings	Lifetime of conference proceedings
Reuse in an annual report	Lifetime of annual report
Reuse in training/CME materials	Reuse up to distribution or time period indicated in License
Reuse in newsmedia	Lifetime of newsmedia
Reuse in coursepack/classroom materials	Reuse up to distribution and/or time period indicated in license

6. Acknowledgement

6. 1. The Licensor's permission must be acknowledged next to the Licensed Material in print. In electronic form, this acknowledgement must be visible at the same time as the figures/tables/illustrations or abstract and must be hyperlinked to the journal/book's homepage.

6. 2. Acknowledgement may be provided according to any standard referencing system and at a minimum should include "Author, Article/Book Title, Journal name/Book imprint, volume, page number, year, Springer Nature".

7. Reuse in a dissertation or thesis

7. 1. Where 'reuse in a dissertation/thesis' has been selected, the following terms apply: Print rights of the Version of Record are provided for; electronic rights for use only on institutional repository as defined by the Sherpa guideline (www.sherpa.ac.uk/romeo/) and only up to what is required by the awarding institution.

7. 2. For theses published under an ISBN or ISSN, separate permission is required. Please contact journalpermissions@springernature.com or bookpermissions@springernature.com for these rights.

7. 3. Authors must properly cite the published manuscript in their thesis according to current citation standards and include the following acknowledgement: '*Reproduced with permission from Springer Nature*'.

8. License Fee

You must pay the fee set forth in the License Agreement (the "License Fees"). All amounts payable by you under this License are exclusive of any sales, use, withholding, value added or similar taxes, government fees or levies or other assessments. Collection and/or remittance of such taxes to the relevant tax authority shall be the responsibility of the party who has the legal obligation to do so.

9. Warranty

9. 1. The Licensor warrants that it has, to the best of its knowledge, the rights to license reuse of the Licensed Material. **You are solely responsible for ensuring that the material you wish to license is original to the Licensor and does not carry the copyright of another entity or third party (as credited in the published version).** If the credit line on any part of the Licensed Material indicates that it was reprinted or adapted with permission from another source, then you should seek additional permission from that source to reuse the material.

9. 2. EXCEPT FOR THE EXPRESS WARRANTY STATED HEREIN AND TO THE EXTENT PERMITTED BY APPLICABLE LAW, LICENSOR PROVIDES THE LICENSED MATERIAL "AS IS" AND MAKES NO OTHER REPRESENTATION OR WARRANTY. LICENSOR EXPRESSLY DISCLAIMS ANY LIABILITY FOR ANY CLAIM ARISING FROM OR OUT OF THE CONTENT, INCLUDING BUT NOT LIMITED TO ANY ERRORS, INACCURACIES, OMISSIONS, OR DEFECTS CONTAINED THEREIN, AND ANY IMPLIED OR EXPRESS WARRANTY AS TO MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL LICENSOR BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR EXEMPLARY DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, VIEWING OR USE OF THE LICENSED MATERIAL REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION APPLIES NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

10. Termination and Cancellation

10. 1. The License and all rights granted hereunder will continue until the end of the applicable period shown in Clause 5.1 above. Thereafter, this license will be terminated and all rights granted hereunder will cease.

10. 2. Licensor reserves the right to terminate the License in the event that payment is not received in full or if you breach the terms of this License.

11. General

11. 1. The License and the rights and obligations of the parties hereto shall be construed, interpreted and determined in accordance with the laws of the Federal Republic of Germany without reference to the stipulations of the CISG (United Nations Convention on Contracts for the International Sale of Goods) or to Germany´s choice-of-law principle.

11. 2. The parties acknowledge and agree that any controversies and disputes arising out of this License shall be decided exclusively by the courts of or having jurisdiction for Heidelberg, Germany, as far as legally permissible.

11. 3. This License is solely for Licensor's and Licensee's benefit. It is not for the benefit of any other person or entity.

Questions? For questions on Copyright Clearance Center accounts or website issues please contact springernaturesupport@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777. For questions on Springer Nature licensing please visit https://www.springernature.com/gp/partners/rights-permissions-third-party-distribution

Other Conditions:

Version 1.4 - Dec 2022

Questions? customercare@copyright.com.

Hamiltonian Dynamics for Real-World Shape Interpolation

Marvin Eisenberger, Daniel Cremers

Technical University of Munich, Garching, Germany
marvin.eisenberger@in.tum.de

Abstract. We revisit the classical problem of 3D shape interpolation and propose a novel, physically plausible approach based on Hamiltonian dynamics. While most prior work focuses on synthetic input shapes, our formulation is designed to be applicable to real-world scans with imperfect input correspondences and various types of noise. To that end, we use recent progress on dynamic thin shell simulation and divergence-free shape deformation and combine them to address the inverse problem of finding a plausible intermediate sequence for two input shapes. In comparison to prior work that mainly focuses on small distortion of consecutive frames, we explicitly model volume preservation and momentum conservation, as well as an anisotropic local distortion model. We argue that, in order to get a robust interpolation for imperfect inputs, we need to model the input noise explicitly which results in an alignment based formulation. Finally, we show a qualitative and quantitative improvement over prior work on a broad range of synthetic and scanned data. Besides being more robust to noisy inputs, our method yields exactly volume preserving intermediate shapes, avoids self-intersections and is scalable to high resolution scans.

Keywords: Shape interpolation, registration, 3D computer vision

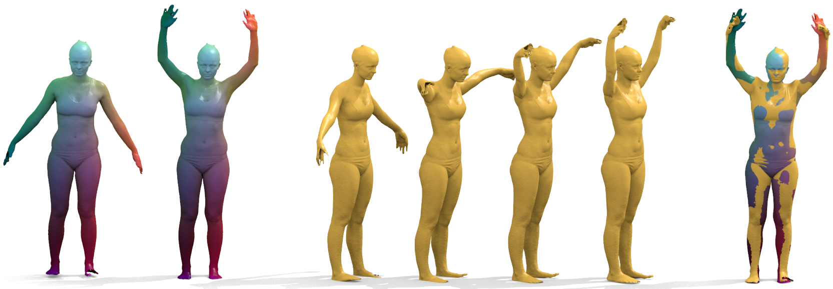


Fig. 1. An example interpolation (middle) on real scans from the FAUST dataset [7] and the final overlap (right). Here, the input correspondences were computed with Deep Functional Maps [36] (left).

1 Introduction

Modeling realistic deformations of 3D shapes is at the heart of many computer vision applications. The central motivation in this context is to give meaning to sparse observations of a dynamically moving 3D object. Depending on the application, these measurements are given in the form of a point cloud, a triangle mesh, a voxel grid or a signed distance function.

In many cases, the sampling is not consistent over time and finding commonalities between observations is not a trivial task. While there are a lot of approaches that try to fuse scanned data for 3D reconstruction [45, 53, 54], relatively few work was dedicated to modeling the temporal transformation of the observed object directly. In this work, we revisit this classical challenge of 3D shape interpolation. Although there exists a multitude of elegant formulations, we will show that a lot of these approaches are mainly designed for synthetic shapes and therefore lack robustness to noisy real-world measurements.

The classical formulation is to define an interpolation as a sequence of shapes with minimal local distortion between consecutive frames [12, 28, 32]. While this is undoubtedly a reasonable assumption, it does not suffice in practice to account for the peculiarities of real-world data. For synthetic 3D objects, the ground-truth correspondences are typically known. For real scans, on the other hand, we need to first estimate them, e.g. by using a shape matching method. In practice, the resulting correspondences are not perfect and contain both outliers and fine-scale noise. This is problematic for an interpolation method that minimizes the local distortion between neighboring frames, because the noise from the faulty correspondences tends to distort the local geometry throughout the whole sequence. Moreover, most classical approaches do not model the global geometry of an object which can lead to artifacts like self-intersections.

Contribution We propose a novel framework for real-world shape interpolation that is systematically derived from Hamiltonian dynamics. It resolves the above challenges by introducing additional, physically plausible modeling assumptions like volume preservation and momentum conservation. More specifically, we formulate shape interpolation as the inverse problems of a dynamic thin shell simulation. The Eulerian time-varying deformation fields are represented in a low rank manner which allows us to build volume preservation directly into our model. In qualitative and quantitative experiments, we demonstrate that our method gives rise to high-quality interpolations for real-world inputs.

2 Related work

Shape interpolation has a long tradition in computer graphics. Originally it was developed for planar shapes [2, 42, 52] with [16] being a more recent formalism. A common approach for 3D surfaces is to define an interpolating trajectory as a geodesic in some higher dimensional shape space [12, 26, 27, 60, 61]. Most of these methods use some kind of deformation measure and then optimize for a sequence such that the local distortion between any two consecutive shapes is low. In [32] this is done with an as-Killing-as-possible energy and in [28] with a discrete shell energy motivated by [24]. Other popular examples of non-linear shape deformation are PriMo [9] and as-rigid-as-possible [56]. For a more thorough introduction to shape spaces, we refer the reader to the book of Younes [63].

An alternative approach to shape interpolation is to interpolate intrinsic quantities like dihedral angles before reconstructing the extrinsic geometry [2, 6, 62]. One class of such intrinsic quantities are rotation-invariant or differential coordinates [1, 34, 35, 51].

Sometimes shape deformation is stated as the time-dependent gradient flow wrt. some surface functional. Typically these functionals promote a smooth flow [14, 15, 18, 58] but most of these methods focus on shape matching with less emphasis on the quality of the intermediate shapes.

Recently, more and more work was dedicated to processing collections of shapes in order to make interpolation more efficient. This can e.g. be achieved by constructing a low-dimensional subspace of admissible poses [3, 21, 29, 57, 65]. In practice, this greatly helps to reduce the computational cost of shape interpolation and even allows for interactive applications [47].

A common assumption of interactive shape deformation modeling is volume preservation. This can be obtained by defining a deformation as the flow of a divergence-free Eulerian vector field [4, 22]. Recently, [19] extended this idea by constructing a divergence-free vector field basis that can be used to interpolate 3D objects. We will make use of this vector field representation and additionally formulate shape interpolation as the inverse problem of a dynamical thin shell simulation. The forward simulation corresponding to this is a well-known problem in computer graphics [44] with applications like cloth [23] or fluid [37] simulation. A recent formulation of this problem that is akin to our approach is projective dynamics [10, 11]. Here, the Lagrangian gradient flow of a dynamical system is restated using the variational form of implicit Euler integration from [38] which leads to an efficient and extremely robust thin shell simulation.

3 Background

We briefly review important preliminary work on shape deformation and interpolation of non-rigidly deforming 3D objects. In this work, we focus on surface-based models like point clouds and 3D meshes. This allows for a compact representation and is in coherence with the output of real-world sensors. In particular, the set of observations $p = (p_1, \dots, p_n)^\top \in \mathbb{R}^{n \times 3}$ consists of n points sampled from a two-dimensional Riemannian manifold $\mathcal{X}$. Depending on the application, these points are either part of a triangle mesh or embedded in a (knn-)graph.

3.1 Physical assumptions for shape deformation

In order to find similarities between two non-rigid poses of an object, it helps to model geometric assumptions about the expected deformations directly. We review two common assumptions, namely small local distortions and volume preservation.

Local distortion A popular deformation energy to quantify the distortion between p and a deformed counterpart $p^* \in \mathbb{R}^{n\times 3}$ is the as-rigid-as-possible (arap) energy [56]:

$$\mathcal{W}_{\text{arap}}\big(p, p^*; (R_i)_{1\leq i\leq n}\big) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)} \big\|R_i(p_j - p_i) - (p_j^* - p_i^*)\big\|_2^2. \tag{1}$$

The assumption behind this functional is that the local deformation of the geometry in the neighborhood $\mathcal{N}(i)$ of every vertex p_i is approximately rigid. I.e. one can find a rotation matrix $R_i \in SO(3)$ that approximately captures the transformation of the neighboring edges $p_j - p_i$. In turn, deviations of the deformation p^* from the approximate rigidity are penalized. The neighborhood $\mathcal{N}(i)$ for a given vertex i is defined as some set of adjacent vertices j to i.

There are multiple popular alternatives with the same flavor as $\mathcal{W}_{arap}$, including PriMo [9], discrete shells [24] and as-Killing-as-possible [55]. Most techniques penalize deformations of the local geometry and each one of them has certain advantages. In our formulation, we choose the arap energy because it is applicable directly for point clouds and because the optimization for p^* and R_i can be done efficiently in closed form.

Volume preservation Another common assumption for shape deformation is that the volume of the observed object is preserved over time [22]. This can be obtained by prescribing that the deformation is the flow induced by an underlying Eulerian deformation field $v : \mathbb{R}^3 \to \mathbb{R}^3$ which is divergence-free $\text{div}(v) = 0$. Recently, [19] proposed a formulation of a coarse-to-fine vector field basis that has the volume preservation built in as a hard constraint. A flow field v is then obtained as the linear combination of a finite subset of those divergence-free basis functions:

$$\text{v}(x; c) = \sum_{k=1}^{K} c_k\phi_k(x), \text{ where } \text{div}(\phi_k) = 0. \tag{2}$$

These deformation fields v are exactly volume preserving because the divergence is a linear operator, see [19] for more details. In practice, a relatively small number $K \approx 1000$ of coefficients $c = (c_1, \dots, c_K)^\top \in \mathbb{R}^K$ suffices to represent arbitrary smooth, volume preserving vector fields v. We make use of this compact representation in this work. However, while [19] only considered stationary vector fields $v(x)$, in this work we consider time-dependent vector fields $v(t, x)$ in order to account for more complex shape variation.

3.2 Shape interpolation

Computing an interpolation of two 3D objects $p = p^{(0)}$ and $q = p^{(T)}$ is a common problem in computer graphics and vision. In general, it is not a well-defined problem because there are typically infinitely many conceivable paths between $p^{(0)}$ and $p^{(T)}$. Therefore, we need to make additional assumptions about plausible sequences like small local distortions or volume preservation. The common way to do this is to define a deformation energy for the whole, time-discrete sequence $p^{(0)}, \dots, p^{(T)}$ of intermediate shapes [12, 27, 28, 31, 32]:

$$E\big(p^{(1)}, \dots, p^{(T-1)}\big) = \sum_{t=0}^{T-1} \mathcal{W}\big(p^{(t)}, p^{(t+1)}\big). \tag{3}$$

Here, $\mathcal{W}$ is some local distortion measure like $\mathcal{W}_{\text{arap}}$ from Eq. (1). For symmetry reasons, the optimization is commonly done jointly for both the standard and the inverse sequence $p^{(T)}, \dots, p^{(0)}$. W.l.o.g. we will consider the time interval $[0, t_{\max}] = [0, 1]$ which leads to a discrete step size $\tau = \frac{1}{T}$.

4 Interpolation of real-world objects

The implicit assumption behind most shape interpolation approaches is that the exact point-to-point correspondences between the two input surfaces p and q are known. While there is a lot of synthetic data where this is feasible, for scanned data the sampling of two given objects is typically not consistent, even if they approximate the same real-world surface $\mathcal{X}$. Not even the number of points of the two surfaces $p \in \mathbb{R}^{n \times 3}$ and $q \in \mathbb{R}^{m \times 3}$ is necessarily the same in the most general case. In order to compute an interpolation for this type of input data, we need to first estimate the surface correspondences between p and q.

Computing shape correspondences is a problem in itself and there is a variety of methods that focus on shape matching, either in the classical sense [20, 33, 41, 46, 48, 59] or using machine learning [8, 25, 36, 39, 43, 50]. The output of those

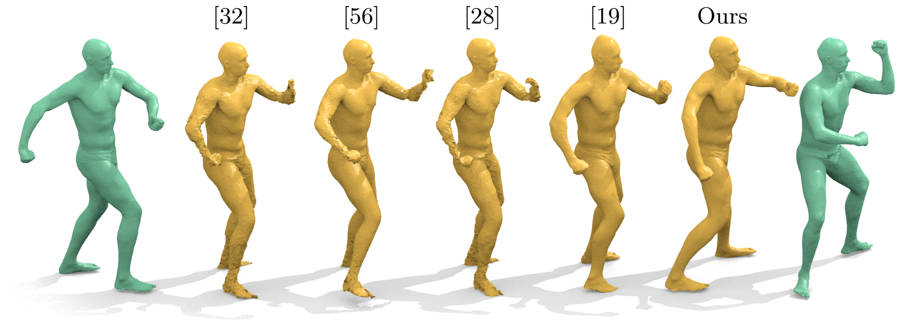


Fig. 2. A qualitative comparison of our approach with other popular shape interpolation methods. Here, we display the intermediate shapes at $t = 0.5$ for an example pair from SCAPE [5] with correspondences from BCICP [48]. Like us, [19] solves for an approximate alignment formulated as an IVP but the stationary vector field leads to slight distortions of the geometry (e.g. at the head and right arm). The other methods [28, 32, 56] solve a BVP and in certain areas the high frequency noise of the correspondences from BCICP leads to a severely degenerate geometry.

methods is a point-to-point assignment of the surface $p \in \mathbb{R}^{n \times 3}$ to $q \in \mathbb{R}^{m \times 3}$ which can be represented with a matrix $\Pi \in \{0,1\}^{n \times m}$. In principle, we can now transfer the points and neighborhood information from p to q and apply a classical interpolation method like [32] or [28] to $p \in \mathbb{R}^{n \times 3}$ and $\Pi q \in \mathbb{R}^{n \times 3}$. However, in practice the correspondences Π are not perfect and contain faulty or noisy matches. We found that most interpolation methods that assume perfect correspondences are not very robust to fine-scale noise, see Figure 2.

One possible way to make interpolation feasible for real scans is to acknowledge that the given matching Π is not perfect and to build this stochastic discrepancy directly into our model. In particular, we add Gaussian random noise η to the vertex position of the second shape Πq:

$$\tilde{q} := \Pi q + \eta, \text{ with } \eta \sim \mathcal{N}(0, \sigma). \tag{4}$$

Instead of finding intermediate shapes by solving a boundary value problem (BVP) as outlined in Eq. (3), we can then define an initial value problem (IVP) similar to [19]. In particular, we will optimize for a sequence $p^{(0)}, \dots, p^{(T)}$ with $p = p^{(0)}$ and $p^{(T)} = \tilde{q} \approx \Pi q$.

5 From Hamiltonian dynamics to Eulerian-Lagrangian shape interpolation

In this work, we model the motions of objects in an inertial frame of reference as a physical phenomenon that is governed by three aspects: internal forces, momentum conservation and volume preservation. Most existing interpolation techniques model internal forces in some way, yet they omit the momentum conservation and volume preservation. Without momentum conservation, the intermediate objects can be plausible but in many cases the motions lack temporal coherence. The volume preservation helps to constrain the optimization and prevents self-intersections, see Figure 3. Our formulation combines the strengths of volume preserving fields [19] and projective dynamics [10] with those of classical interpolation methods [28, 32].

5.1 Deformation model

We systematically derive the evolution of a surface as a physical system from the Hamiltonian energy given by:

$$\mathcal{H}(p, v) = \frac{1}{2} \|v\|_2^2 + \mathcal{W}(p). \tag{5}$$

This energy consists of a kinetic energy term that models momentum conservation (with unit mass per point) and some potential energy component $\mathcal{W}$ that penalizes intrinsic distortions. The principles of Hamiltonian mechanics now prescribe how this system evolves over time:

$$\begin{cases} \dot{p} = \frac{\mathrm{d}\mathcal{H}}{\mathrm{d}v} = v. \\ \dot{v} = -\frac{\mathrm{d}\mathcal{H}}{\mathrm{d}p} = -\nabla \mathcal{W}(p). \end{cases} \tag{6}$$

We couple this with the volume preservation assumption by constraining v to the low rank vector field representation from Eq. (2). This allows us to model displacements of a shape $p(t) = \big(p_1(t), \dots, p_n(t)\big)^\top$ at time t with only $K \ll n$ degrees of freedom:

$$\dot{p}_i(t) = \mathrm{v}\big(p_i(t); c(t)\big) = \sum_{k=1}^{K} c_k(t)\phi_k\big(p_i(t)\big). \tag{7}$$

Besides providing a compact representation, this approach builds volume preservation directly into the deformation model, because $\operatorname{div}(v) = 0$. In [19], the authors model shape deformations in a similar way but with a stationary vector field $\mathrm{v}\big(x; c(t)\big) = \mathrm{v}\big(x; c\big)$. This leads to a well-constrained optimization problem with only K degrees of freedom $c_1, \dots, c_K$ but it is also restrictive and lacks expressivity. Instead of using a constant vector field, following Eq. (6), we define a dynamic flow $v(t, x) = \mathrm{v}(x; c(t))$:

$$\begin{cases} \dot{v}\big(t, p(t)\big) = -\nabla\mathcal{W}\big(p(t)\big). \\ \operatorname{div}(v) = 0. \end{cases} \tag{8}$$

In our formulation, the internal forces are defined as the negative gradient of our anisotropic as-rigid-as-possible potential $\mathcal{W}$ which we define in the next chapter.

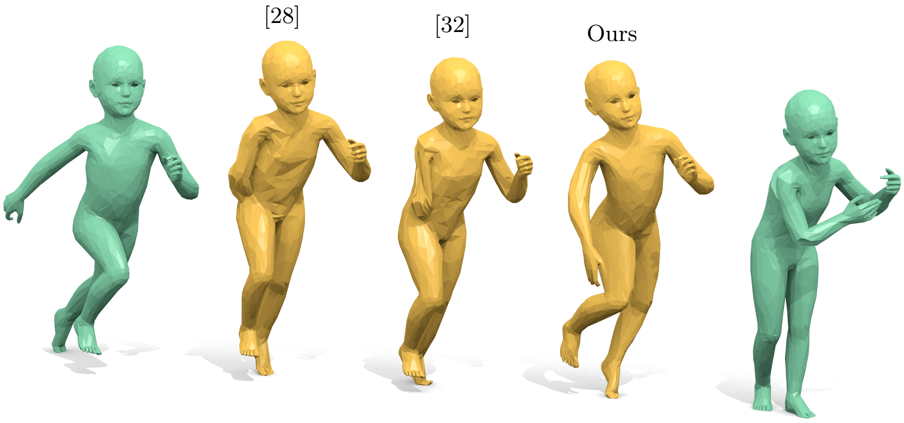


Fig. 3. A pair of synthetic shapes with ground-truth correspondences from the KIDS dataset [50] for which we show the intermediate shapes at $t = 0.5$. This example shows that many classical methods like [32] or [28] cannot detect self-intersections of different subparts. Here, the optimal path that minimizes a local distortion metric makes the right arm of the kid move through itself. Our method, on the other hand, avoids self-intersections by design: All deformations are expressed as a divergence-free Eulerian field, therefore the resulting flow has to be globally consistent in the sense that two close parts cannot have contradictory motions.

5.2 Anisotropic as-rigid-as-possible deformation

For most 3D objects, not all parts are behaving similar in terms of local distortions. For example, regions near joints of a human body allow for more movement than most other parts of the surface. The classical as-rigid-as-possible potential that we reviewed in Eq. (1) penalizes distortions of the geometry uniformly in all directions and equal for all parts of the considered object. We generalize this idea and introduce an anisotropic as-rigid-as-possible energy:

$$\mathcal{W}\big(p(t);(R_i)_i,(\Sigma_i)_i\big)=\frac{1}{2}\sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)}\big\|\big(p_j(0)-p_i(0)\big)-R_i^\top\big(p_j(t)-p_i(t)\big)\big\|_{\Sigma_i}^2. \quad (9)$$

In this context, $\|\cdot\|_{\Sigma_i}$ denotes the standard Mahalanobis norm [40] with an unknown covariance matrix $\Sigma_i\in\mathbb{R}^{3\times3}$. This energy $\mathcal{W}$ allows our model to adapt the appropriate local behavior during the optimization, see Figure 4 for an example. Moreover, the distortion is always computed in the reference frame of the first pose $p(0)$. This means that we only need to compute the distortion model of $p(0)$ and therefore we only need one local distortion matrix per vertex Σ_i for the whole sequence.

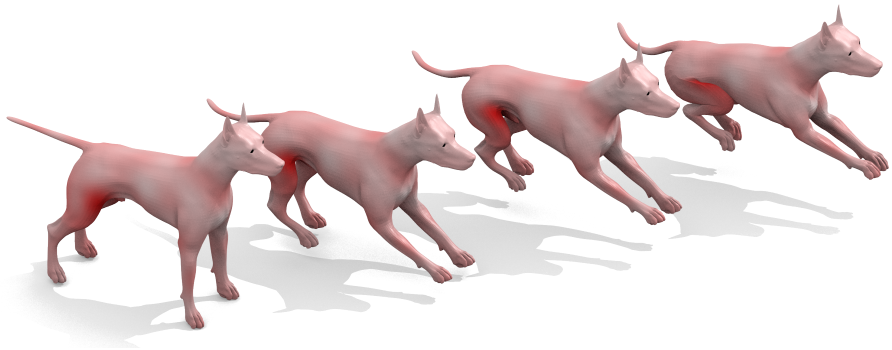


Fig. 4. An example from TOSCA where we color code the log-determinant of the covariance matrix $\log\det(\Sigma_i)$ for each point p_i. Red stands for a low value which corresponds to a high local distortion. This shows how our anisotropic as-rigid-as-possible energy (9) automatically adapts to objects consisting of inhomogeneous parts. Certain regions like joints allow for more local distortion throughout the sequence than others. Notice the difference between the hind legs, the head and the rest of the body.

5.3 Time discretization

In the time-discrete setting, we can approximate Eq. (7) and Eq. (8) using an implicit Euler intergration scheme:

$$\begin{cases} p^{(t+1)} = p^{(t)} + \tau v^{(t+1)}. & \text{(10a)} \\ v^{(t+1)} = v^{(t)} - \tau \nabla \mathcal{W}\big(p^{(t+1)}\big). & \text{(10b)} \\ \operatorname{div}\big(v^{(t+1)}\big) = 0. & \text{(10c)} \end{cases}$$

This is a Eulerian-Lagrangian scheme: The velocity field is represented on the surface $v^{(t)} \in \mathbb{R}^{n \times 3}$ but the divergence-free condition $\operatorname{div}\big(v^{(t+1)}\big) = 0$ is Eulerian. In order to make this interaction tractable, we will use the divergence-free vector field representation from Eq. (2) and combine it with the variational form of implicit Euler integration introduced in [38]. This allows us to restate this scheme as an optimization problem in terms of the vector field coefficients $c \in \mathbb{R}^K$:

$$\begin{cases} c^{(t+1)} = \underset{c,R}{\arg\min} \left\| \mathrm{v}\big(p^{(t)}; c\big) - \bar{v}^{(t)} \right\|_F^2 + \mathcal{W}\left(p^{(t)} + \tau \mathrm{v}\big(p^{(t)}; c\big); R, \Sigma\right). & \text{(11a)} \\ v_i^{(t+1)} = \mathrm{v}\big(p_i^{(t)}; c^{(t+1)}\big) = \sum_{k=1}^{K} c_k^{(t+1)} \phi_k\big(p_i^{(t)}\big). & \text{(11b)} \\ p^{(t+1)} = p^{(t)} + \tau v^{(t+1)}. & \text{(11c)} \\ \bar{v}^{(t+1)} = 2v^{(t+1)} - v^{(t)}. & \text{(11d)} \end{cases}$$

We refer the interested reader to [38] and [10] for more details on how this scheme is derived. The update of the coefficients c in (11a) can be computed using Gauss-Newton optimization. We use an additional extrapolation step (11d) to get a better prediction of the velocity $v^{(t+2)}$ which we justify in the following:

Theorem 1. *For continuously differentiable vector fields, the extrapolation step* (11d) *of Algorithm 11 yields an estimate $\bar{v}^{(t+1)}$ of $v^{(t+2)}$ with an error of order $\mathcal{O}(\tau^2)$. For the alternative scheme without step* (11d) *it is $\mathcal{O}(\tau)$.*

This result implies that (11d) leads to a qualitative improvement because a better estimate $\bar{v}^{(t+1)} \approx v^{(t+2)}$ provides a more faithful approximation in the next update step (11a) of c. See Appendix A for a proof of Thm. 1.

5.4 Interpolation algorithm

We will now use the scheme (11) from last chapter to define an interpolation algorithm for two given shapes p and q. In each iteration, we initialize the scheme with $p^{(0)} := p$ and the unknown variables $c^{(0)} := \hat{c}$ and $(\hat{\Sigma}_i)_{1 \leq i \leq n}$. We then compute the deformed shapes $p^{(0)}, \dots, p^{(T)}$ according to our scheme (11). Overall, this forward pass can be summarized as the differentiable solution operator $\mathcal{S}$:

$$\mathcal{S} : \begin{cases} \mathbb{R}^K \times \mathbb{R}^{n \times 3 \times 3} \to \mathbb{R}^{n \times 3 \times (T+1)}. \\ \big(\hat{c}, \hat{\Sigma}\big) \mapsto \big(p^{(0)}, \dots, p^{(T)}\big). \end{cases} \tag{12}$$

The goal is now to find the input parameters $\hat{c}$ and $\hat{\Sigma}$ that lead to a tight alignment of the deformed shape $p^{(T)}$ with q in accordance with Eq. (4). Together with our regularizer $\mathcal{W}$ from Eq. (9) this leads to the following energy:

$$E\big(p^{(0)},\dots,p^{(T)};\Pi\big) := \frac{1}{2\sigma^2}\big\|p^{(T)} - \Pi q\big\|^2 + \sum_{t=0}^{T}\mathcal{W}\big(p^{(t)}\big). \tag{13}$$

Putting everything together, we can derive the following algorithm:

Algorithm 1 Volume preserving shape interpolation.

```
Require: $p \in \mathbb{R}^{n\times 3}$, $q \in \mathbb{R}^{m\times 3}$
  $\hat{c} \leftarrow 0 \in \mathbb{R}^K$
  $\hat{\Sigma}_i \leftarrow \mathrm{Id}_3 \in \mathbb{R}^{3\times 3}$
  $\Pi \leftarrow \mathrm{match\_shapes}(p,q) \in \{0,1\}^{n\times m}$
  for $i = 1,\dots,N_{\mathrm{it}}$ do
    $\big(\hat{c},\hat{\Sigma}\big) \leftarrow \big(\hat{c},\hat{\Sigma}\big) - \gamma\nabla E\big(\mathcal{S}(\hat{c},\hat{\Sigma});\Pi\big)$
  end for
  return $\big(p^{(0)},\dots,p^{(T)}\big) := \mathcal{S}\big(\hat{c},\hat{\Sigma}\big)$
```

In our implementation, we use a modern automatic differentiation toolbox to compute the gradient $\nabla E \circ S$ wrt. $(\hat{c},\hat{\Sigma})$ in Algorithm 1. The choice of algorithm to compute the input correspondences Π is not further specified here because it is more or less arbitrary. We show various different possibilities in our experiments.

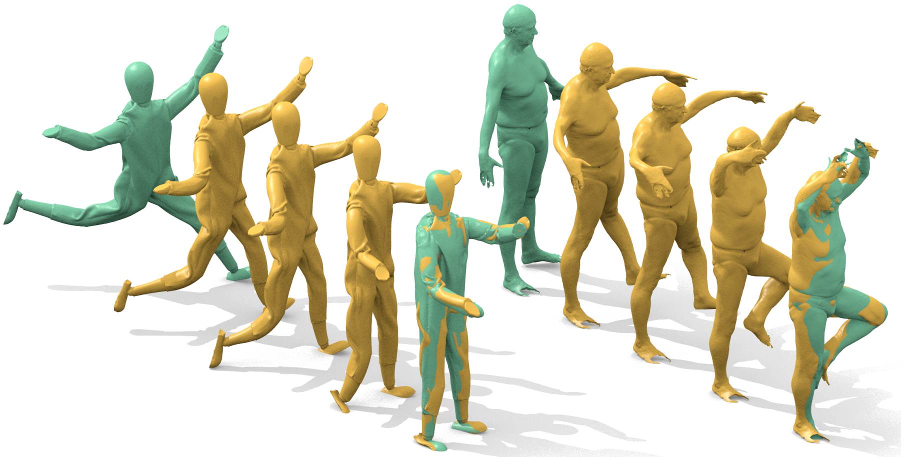


Fig. 5. Two interpolated sequences for real scans of a puppet from SHREC'19 Isometry [17] and a human from FAUST [7] with input correspondences from Smooth Shells [20] and Deep Functional Maps [36] respectively.

6 Experiments

We verify the generality of our method on four different datasets with increasing complexity. The first two are the synthetic datasets TOSCA [13] and SCAPE [5] where we use the ground truth correspondences for the former and correspondences from BCICP [48] for the latter. The last two datasets SHREC'19 Isometry [17] and FAUST [7] contain reals scans of a puppet and different humans respectively, see Figure 5. For those, we use correspondences from Smooth Shells [20] and FMNet [36]. Our experiments show that our formulation is applicable to a wide range of inputs with varying levels of noise. Figure 7 summarizes our quantitative evaluations on all datasets with comparisons to four other popular interpolation methods. The other methods are Geometric Modeling in Shape Space [32], Time-Discrete Geodesics in the Space of Shells [28], Divergence-Free Shape Correspondence by Deformation [19] and As-Rigid-As-Possible Surface Modeling [56]. Although the latter does not describe an interpolation algorithm explicitly, it is trivial to employ its shape deformation procedure in an interpolation pipeline by using Eq. (3). On the surface, our method is similar to [19] in the sense that both approaches compute divergence-free fields in a low rank basis. The decisive difference is that our method is based on a physically plausible formulation which, among other things, allows for time-dependent vector fields $v(t,x)$. This makes our method more expressive, see Figure 6 for an example.

Error metrics In order to quantify the precision of a shape interpolation, we compute three different metrics for each pair of input shapes and plot the resulting cumulative curves in Figure 7. In particular, we measure the conformal distortion [64, Eq. (6)] and volume change [30, Eq. (3)] of intermediate shapes and the Chamfer distance to the target shapes in % of the diameter for our

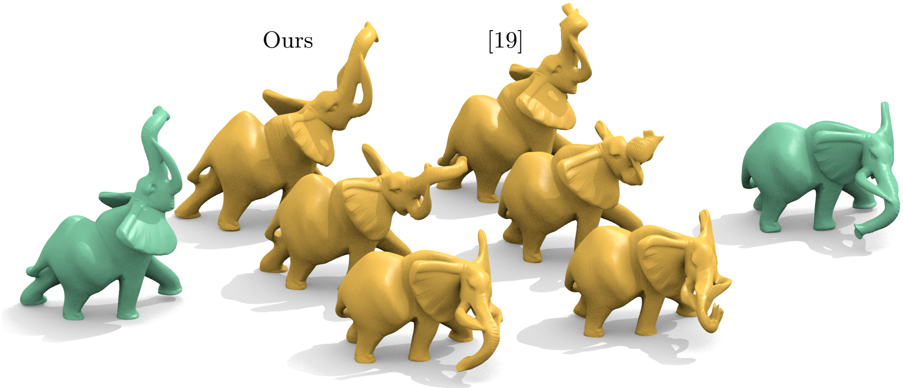


Fig. 6. A comparison of our method and divergence-free interpolation [19] on a pair of synthetic shapes (green). Both methods preserve the volume but for this large scale deformation the stationary vector field in [19] is too restrictive which leads to a distorted geometry for $t \geq 0.5$

method and the second alignment based method [19]. If we are strict, the notion of volume change is only meaningful for watertight meshes, which typically does not hold for real scans. Our argument regarding this is that in theory, a flow induced by a divergence-free deformation field is exactly volume preserving in terms of the underlying watertight real-world manifold $\mathcal{X}$. Remarkably, in this way we can even make sense of the notion of volume for a point cloud, assuming that it was sampled from a closed, continuous surface.

Implementation details The low rank vector field representation of divergence-free fields in our Scheme 11 is entirely decoupled of the input resolutions n and m. Moreover, the vector fields are represented in a spatially dense Eulerian basis

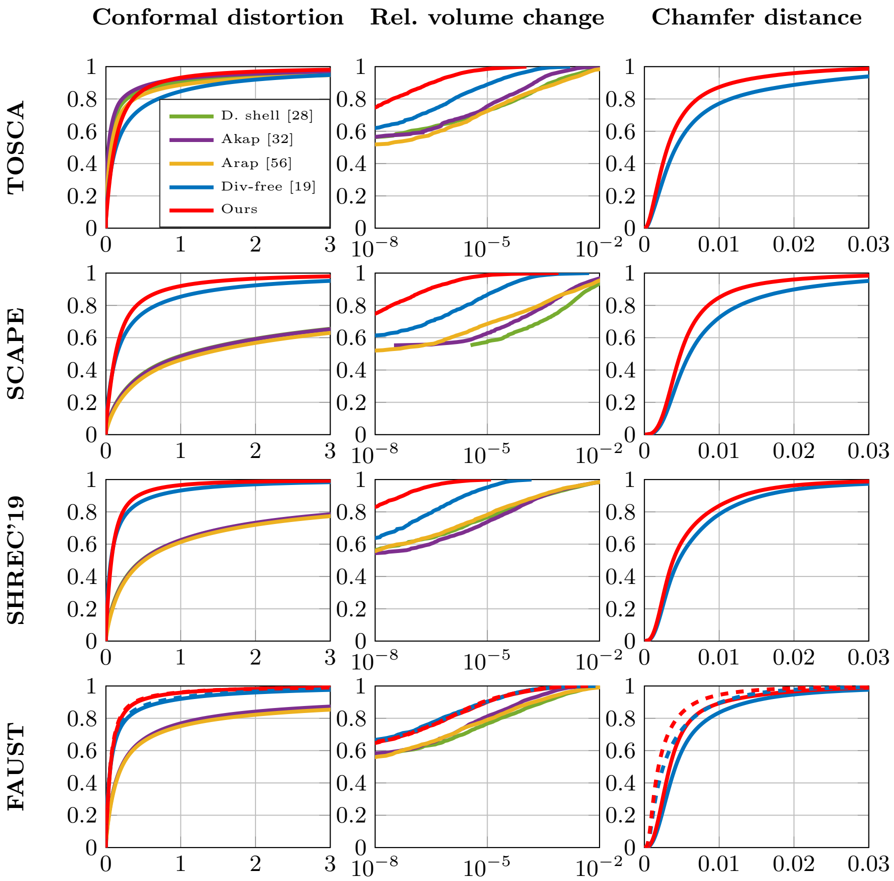


Fig. 7. Quantitative results and comparisons with other methods on four benchmarks. For FAUST, dashed lines correspond to the results on the high resolution scans. Those were only computed for our method and [19] because for the other methods the resolution of around 200k vertices is prohibitively high.

which means that at any discrete time t, the resulting vector field $\mathrm{v}\big(x; c^{(t)}\big)$ can be computed for arbitrary points x in our domain, see [19] for more details. This allows us to efficiently perform the optimization in Algorithm 1 on a subsampled version of the input shapes p and q with a fixed resolution of $2k$ points. Afterwards, the computed vector field can be applied to the full resolution in a single forward pass without any skinning strategy or the like. For once, this makes our approach significantly faster but it also allows for an interpolation of very high resolution objects like those from FAUST (∼200k vertices). Many other classical interpolation methods use some multiscale scheme to allow for higher resolutions [28, 32], but there are still upper limits for them as to what is feasible in terms of computation cost. Our interpolation Algorithm 1 is directly applicable to point clouds, therefore we simply subsample both input shapes using Euclidean farthest point sampling. However, other subsampling strategies like remeshing are also possible if one wants to work directly with meshes. Finally, we use the same set of parameters for all experiments, see our implementation for details.

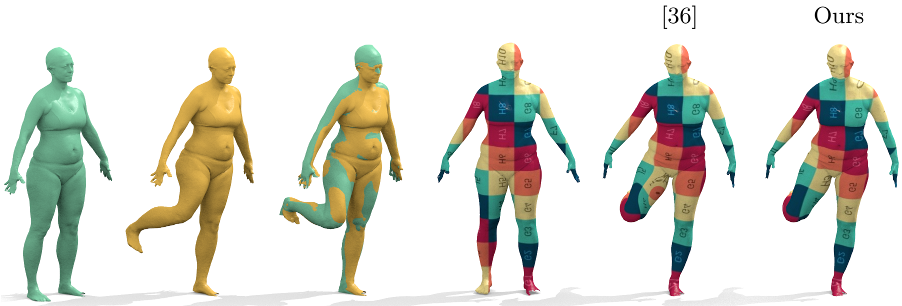


Fig. 8. We show how our method can be used to refine an imperfect shape correspondence. Using the input matching from Deep Functional Maps [36], we compute an interpolation (left half) and use it to recover the improved correspondences using the final alignment at $t = 1$ (right half). We display the matching with a texture map from the first input shape (3rd human from right) to the final pose with both methods.

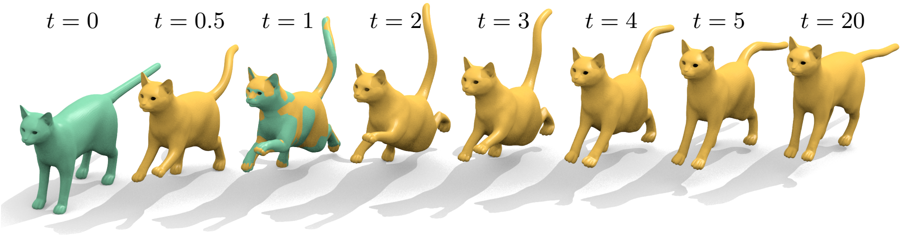


Fig. 9. An example of how our approach can be used to extrapolate the motion prescribed by the two input frames $t = 0$ and $t = 1$. The sequences obtained with our method are physically plausible and remain stable over a long period of time. The cat keeps raising its paw until at $t = 2$, driven by the regularizer (9), the motion reverses.

Additional evaluations As a proof of concept, we show that our physically plausible formulation allows for a broad range of applications beyond shape interpolation. For once, we can use our alignment at $t = 1$ to refine a shape matching which we show for a real scan of FAUST in Figure 8. Furthermore, we can compute plausible shape extrapolations by simply simulating the forward integration for a longer period of time than $t = 1$. Remarkably, this can be done without any additional optimization, we simply compute an interpolation between p at $t = 0$ and q at $t = 1$ and then integrate our Scheme 11 until $t > 1$, see Figure 9. Finally, we show that our method allows for input objects where only parts of the geometry are available, see Figure 10. This is only feasible for an alignment based method, because the classical formulation as a BVP requires that every vertex has a corresponding point on the other surface. Partial shape interpolation is an important preliminary result for many real world applications like scanning of dynamically moving 3D objects.

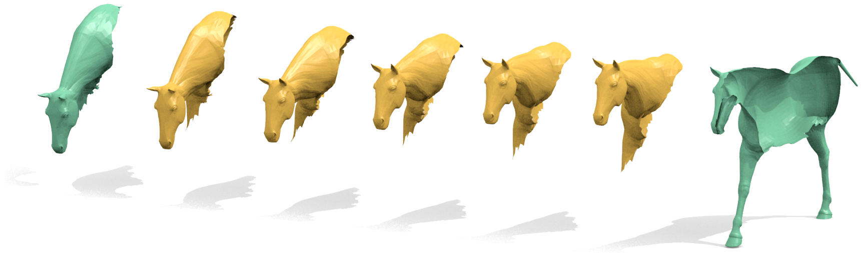


Fig. 10. An example interpolation of a pair of partial shapes from the synthetic TOSCA cuts [49] dataset with our method.

7 Conclusion

We presented a general and flexible approach to shape interpolation that is systematically derived from a formulation of Hamiltonian dynamics. For this, we employ recent advances in dynamic thin shell simulation to get a robust deformation model and solve its inverse problem by optimizing over the initial motion and anisotropic surface properties. We demonstrated that, in comparison to prior work, our approach is able to compute high quality, physically plausible interpolations of noisy real world inputs. In future work, we will apply our setup to a broader range of applications like 3D scanning of actions or mesh compression.

Acknowledgements We would like to thank Zorah Lähner and Aysim Toker for useful discussions. We gratefully acknowledge the support of the ERC Consolidator Grant ”3D Reloaded”.

References

1. Alexa, M.: Differential coordinates for local mesh morphing and deformation. The Visual Computer **19**(2-3), 105–114 (2003)
2. Alexa, M., Cohen-Or, D., Levin, D.: As-rigid-as-possible shape interpolation. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. pp. 157–164. SIGGRAPH '00 (2000)
3. Allen, B., Curless, B., Popović, Z., Hertzmann, A.: Learning a correlated model of identity and pose-dependent body shape variation for real-time synthesis. In: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation. pp. 147–156. Eurographics Association (2006)
4. Angelidis, A., Cani, M.P., Wyvill, G., King, S.: Swirling-sweepers: Constant-volume modeling. Graphical Models **68**(4), 324–332 (2006)
5. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: Scape: shape completion and animation of people. In: ACM transactions on graphics (TOG). vol. 24, pp. 408–416. ACM (2005)
6. Baek, S.Y., Lim, J., Lee, K.: Isometric shape interpolation. Computers & Graphics **46**, 257–263 (2015)
7. Bogo, F., Romero, J., Loper, M., Black, M.J.: FAUST: Dataset and evaluation for 3D mesh registration. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). IEEE, Piscataway, NJ, USA (Jun 2014)
8. Boscaini, D., Masci, J., Rodolà, E., Bronstein, M.: Learning shape correspondence with anisotropic convolutional neural networks. In: Advances in neural information processing systems. pp. 3189–3197 (2016)
9. Botsch, M., Pauly, M., Gross, M.H., Kobbelt, L.: Primo: coupled prisms for intuitive surface modeling. In: Symposium on Geometry Processing. pp. 11–20. No. CONF (2006)
10. Bouaziz, S., Martin, S., Liu, T., Kavan, L., Pauly, M.: Projective dynamics: fusing constraint projections for fast simulation. ACM Transactions on Graphics (TOG) **33**(4), 1–11 (2014)
11. Brandt, C., Eisemann, E., Hildebrandt, K.: Hyper-reduced projective dynamics. ACM Transactions on Graphics (TOG) **37**(4), 1–13 (2018)
12. Brandt, C., von Tycowicz, C., Hildebrandt, K.: Geometric flows of curves in shape space for processing motion of deformable objects. In: Computer Graphics Forum. vol. 35, pp. 295–305. Wiley Online Library (2016)
13. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer (2008), http://tosca.cs.technion.ac.il/book/resources_data.html
14. Charpiat, G., Faugeras, O., Keriven, R.: Approximations of shape metrics and application to shape warping and empirical shape statistics. Foundations of Computational Mathematics **5**(1), 1–58 (2005)
15. Charpiat, G., Maurel, P., Pons, J.P., Keriven, R., Faugeras, O.: Generalized gradients: Priors on minimization flows. International journal of computer vision **73**(3), 325–344 (2007)
16. Chen, R., Weber, O., Keren, D., Ben-Chen, M.: Planar shape interpolation with bounded distortion. ACM Transactions on Graphics (TOG) **32**(4), 1–12 (2013)
17. Dyke, R., Stride, C., Lai, Y., Rosin, P.: Shrec-19: Shape correspondence with isometric and non-isometric deformations (2019)
18. Eckstein, I., Pons, J.P., Tong, Y., Kuo, C.C., Desbrun, M.: Generalized surface flows for mesh processing. In: Proceedings of the fifth Eurographics symposium on Geometry processing. pp. 183–192. Eurographics Association (2007)

19. Eisenberger, M., Lähner, Z., Cremers, D.: Divergence-free shape correspondence by deformation. In: Computer Graphics Forum. vol. 38, pp. 1–12. Wiley Online Library (2019)
20. Eisenberger, M., Lähner, Z., Cremers, D.: Smooth shells: Multi-scale shape registration with functional maps. arXiv preprint arXiv:1905.12512 (2019)
21. Fletcher, P.T., Lu, C., Pizer, S.M., Joshi, S.: Principal geodesic analysis for the study of nonlinear statistics of shape. IEEE transactions on medical imaging **23**(8), 995–1005 (2004)
22. von Funck, W., Theisel, H., Seidel, H.P.: Vector field based shape deformations. In: ACM Transactions on Graphics (TOG). vol. 25, pp. 1118–1125. ACM (2006)
23. Goldenthal, R., Harmon, D., Fattal, R., Bercovier, M., Grinspun, E.: Efficient simulation of inextensible cloth. In: ACM SIGGRAPH 2007 papers, pp. 49–es (2007)
24. Grinspun, E., Hirani, A.N., Desbrun, M., Schröder, P.: Discrete shells. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation. pp. 62–67. Eurographics Association (2003)
25. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: The European Conference on Computer Vision (ECCV) (September 2018)
26. Heeren, B., Rumpf, M., Schröder, P., Wardetzky, M., Wirth, B.: Exploring the geometry of the space of shells. In: Computer Graphics Forum. vol. 33, pp. 247–256. Wiley Online Library (2014)
27. Heeren, B., Rumpf, M., Schröder, P., Wardetzky, M., Wirth, B.: Splines in the space of shells. Computer Graphics Forum **35**(5), 111–120 (2016)
28. Heeren, B., Rumpf, M., Wardetzky, M., Wirth, B.: Time-discrete geodesics in the space of shells. In: Computer Graphics Forum. vol. 31, pp. 1755–1764. Wiley Online Library (2012)
29. Heeren, B., Zhang, C., Rumpf, M., Smith, W.: Principal geodesic analysis in the space of discrete shells. In: Computer Graphics Forum. vol. 37, pp. 173–184. Wiley Online Library (2018)
30. Hormann, K., Greiner, G.: Mips: An efficient global parametrization method. Tech. rep., Erlangen-Nuernberg University (Germany) Computer Graphics Group (2000)
31. Huber, P., Perl, R., Rumpf, M.: Smooth interpolation of key frames in a riemannian shell space. Computer Aided Geometric Design **52**, 313–328 (2017)
32. Kilian, M., Mitra, N.J., Pottmann, H.: Geometric modeling in shape space. In: ACM Transactions on Graphics (TOG). vol. 26, p. 64. ACM (2007)
33. Kim, V.G., Lipman, Y., Funkhouser, T.A.: Blended intrinsic maps. Transactions on Graphics (TOG) **30**(4) (2011)
34. Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rossi, C., Seidel, H.P.: Differential coordinates for interactive mesh editing. In: Proceedings Shape Modeling Applications, 2004. pp. 181–190. IEEE (2004)
35. Lipman, Y., Sorkine, O., Levin, D., Cohen-Or, D.: Linear rotation-invariant coordinates for meshes. ACM Transactions on Graphics (TOG) **24**(3), 479–487 (2005)
36. Litany, O., Remez, T., Rodolà, E., Bronstein, A., Bronstein, M.: Deep functional maps: Structured prediction for dense shape correspondence. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5659–5667 (2017)
37. Macklin, M., Müller, M.: Position based fluids. ACM Transactions on Graphics (TOG) **32**(4), 1–12 (2013)
38. Martin, S., Thomaszewski, B., Grinspun, E., Gross, M.: Example-based elastic materials. In: ACM SIGGRAPH 2011 papers, pp. 1–8 (2011)

39. Masci, J., Boscaini, D., Bronstein, M., Vandergheynst, P.: Geodesic convolutional neural networks on riemannian manifolds. In: Proceedings of the IEEE international conference on computer vision workshops. pp. 37–45 (2015)
40. McLachlan, G.J.: Mahalanobis distance. Resonance **4**(6), 20–26 (1999)
41. Melzi, S., Ren, J., Rodola, E., Ovsjanikov, M., Wonka, P.: Zoomout: Spectral upsampling for efficient shape correspondence. arXiv preprint arXiv:1904.07865 (2019)
42. Michor, P.W., Mumford, D.: Riemannian geometries on spaces of plane curves. arXiv preprint math/0312384 (2003)
43. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5115–5124 (2017)
44. Müller, M., Heidelberger, B., Hennix, M., Ratcliff, J.: Position based dynamics. Journal of Visual Communication and Image Representation **18**(2), 109–118 (2007)
45. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 343–352 (2015)
46. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. ACM Transactions on Graphics (TOG) **31**(4), 30 (2012)
47. von Radziewsky, P., Eisemann, E., Seidel, H.P., Hildebrandt, K.: Optimized subspaces for deformation-based modeling and shape interpolation. Computers & Graphics **58**, 128–138 (2016)
48. Ren, J., Poulenard, A., Wonka, P., Ovsjanikov, M.: Continuous and orientation-preserving correspondences via functional maps. ACM Trans. Graph. **37**(6), 248:1–248:16 (Dec 2018)
49. Rodolà, E., Cosmo, L., Bronstein, M.M., Torsello, A., Cremers, D.: Partial functional correspondence. In: Computer Graphics Forum. vol. 36, pp. 222–236. Wiley Online Library (2017)
50. Rodolà, E., Rota Bulo, S., Windheuser, T., Vestner, M., Cremers, D.: Dense non-rigid shape correspondence using random forests. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
51. Sassen, J., Heeren, B., Hildebrandt, K., Rumpf, M.: Geometric optimization using nonlinear rotation-invariant coordinates. arXiv preprint arXiv:1908.11728 (2019)
52. Sederberg, T.W., Gao, P., Wang, G., Mu, H.: 2-d shape blending: an intrinsic solution to the vertex path problem. In: Proceedings of the 20th annual conference on Computer graphics and interactive techniques. pp. 15–18 (1993)
53. Slavcheva, M., Baust, M., Cremers, D., Ilic, S.: Killingfusion: Non-rigid 3d reconstruction without correspondences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1386–1395 (2017)
54. Slavcheva, M., Baust, M., Ilic, S.: Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2646–2655 (2018)
55. Solomon, J., Ben-Chen, M., Butscher, A., Guibas, L.: As-killing-as-possible vector fields for planar deformation. In: Computer Graphics Forum. vol. 30, pp. 1543–1552. Wiley Online Library (2011)
56. Sorkine, O., Alexa, M.: As-rigid-as-possible surface modeling. In: Symposium on Geometry processing. vol. 4, pp. 109–116 (2007)
57. Sumner, R.W., Zwicker, M., Gotsman, C., Popović, J.: Mesh-based inverse kinematics. ACM transactions on graphics (TOG) **24**(3), 488–495 (2005)

58. Sundaramoorthi, G., Yezzi, A., Mennucci, A.C.: Sobolev active contours. International Journal of Computer Vision **73**(3), 345–366 (2007)
59. Vestner, M., Lähner, Z., Boyarski, A., Litany, O., Slossberg, R., Remez, T., Rodolà, E., Bronstein, A.M., Bronstein, M.M., Kimmel, R., Cremers, D.: Efficient deformable shape correspondence via kernel matching. In: International Conference on 3D Vision (3DV) (October 2017)
60. Wirth, B., Bar, L., Rumpf, M., Sapiro, G.: A continuum mechanical approach to geodesics in shape space. International Journal of Computer Vision **93**(3), 293–318 (2011)
61. Wirth, B., Bar, L., Rumpf, M., Sapiro, G.: A continuum mechanical approach to geodesics in shape space. International Journal of Computer Vision **93**(3), 293–318 (Jul 2011)
62. Xu, D., Zhang, H., Wang, Q., Bao, H.: Poisson shape interpolation. In: Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling. pp. 267–274 (2005)
63. Younes, L.: Shapes and diffeomorphisms, vol. 171. Springer (2010)
64. Zhang, C., Chen, T.: Efficient feature extraction for 2d/3d objects in mesh representation. In: Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205). vol. 3, pp. 935–938. IEEE (2001)
65. Zhang, C., Heeren, B., Rumpf, M., Smith, W.A.: Shell pca: Statistical shape modelling in shell space. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1671–1679 (2015)

Chapter 7

Deep Shells: Unsupervised Shape Correspondence with Optimal Transport

In recent times, geometric deep learning has proven to be a powerful formalism for diverse applications involving 3D shape data. A primary bottleneck for learning dense correspondence maps is obtaining ground truth annotations, required for training supervised approaches. In practice, this often involves manual pose alignment by human experts, or is limited to sparse landmark annotations. We propose an unsupervised learning approach for shape correspondence based on optimal transport. To this end, we leverage the product space representation introduced in [2], where each vertex is associated with both extrinsic coordinates and the intrinsic Laplace-Beltrami eigenfunctions. We then define optimal matchings between surface points in terms of their Wasserstein 2-distance in the joint embedding space. The resulting transport plans encode a soft correspondence, which we threshold at test time to obtain our final output correspondences. We combine the optimal transport matching with a learnable local feature extractor, which enables representation learning by refining noisy input descriptor fields. For the specific architecture, we devise a spectral convolution layer defined as pointwise products in the Fourier domain of a given set of features. The overall pipeline consists of an iterative scheme of optimization steps, alternating between updates of the optimal transport plans and the product space alignment. The learned features are employed to initialize this iterative scheme, and the training loss is defined as the resulting optimal transport distance. Since gradient-based optimization requires that individual optimization steps are differentiable, we augment our optimal transport energy with an additional entropic regularization term. In our experiments, we demonstrate that our approach obtains highly accurate correspondence maps, even outperforming state-of-the-art supervised approaches, and when generalizing across different datasets.

Individual contributions

Leading role in realizing the scientific project.

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

Copyright

Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers
Deep Shells: Unsupervised Shape Correspondence with Optimal Transport
34th Conference on Neural Information Processing Systems (NeurIPS)

Deep Shells: Unsupervised Shape Correspondence with Optimal Transport

Marvin Eisenberger
Technical University of Munich
marvin.eisenberger@in.tum.de

Aysim Toker
Technical University of Munich
aysim.toker@in.tum.de

Laura Leal-Taixé
Technical University of Munich
leal.taixe@tum.de

Daniel Cremers
Technical University of Munich
cremers@tum.de

Abstract

We propose a novel unsupervised learning approach to 3D shape correspondence that builds a multiscale matching pipeline into a deep neural network. This approach is based on smooth shells, the current state-of-the-art axiomatic correspondence method, which requires an a priori stochastic search over the space of initial poses. Our goal is to replace this costly preprocessing step by directly learning good initializations from the input surfaces. To that end, we systematically derive a fully differentiable, hierarchical matching pipeline from entropy regularized optimal transport. This allows us to combine it with a local feature extractor based on smooth, truncated spectral convolution filters. Finally, we show that the proposed unsupervised method significantly improves over the state-of-the-art on multiple datasets, even in comparison to the most recent supervised methods. Moreover, we demonstrate compelling generalization results by applying our learned filters to examples that significantly deviate from the training set.

1 Introduction

Computing shape correspondence is a fundamental component in understanding the 3D world, and is at the heart of many applications in computer vision and graphics. It is also a notoriously hard problem and a multitude of different solutions have been proposed over the years. Classical axiomatic methods make assumptions about the geometric properties of the input surfaces, like small local distortions, to compute correspondences for a well-defined class of objects. Many such methods rely on hand-crafted features [40, 39, 3] and most of them make restrictive assumptions about the discretization, topology or morphology of the considered objects. A promising venture for extending shape matching methods to a broader range of inputs is to apply machine learning to geometric data. While deep learning has achieved great success in the field of image analysis, extending the power of deep networks to non-Euclidean data remains an important and very actively studied open challenge. Although there has been much progress, many methods to date lack a strong geometric foundation that fully acknowledges the underlying structure of 3D surfaces. A recent line of research pioneered by [20] aims at combining a learnable local feature extractor with the axiomatic approach functional maps [26]. This combination of machine learning with axiomatic methods is an important breakthrough since it allows us to integrate geometric priors into our model which can significantly enhance the learning process. Most follow up work inspired by deep functional maps [20] use functional maps as a matching layer. The underlying assumption of these approaches is that the input pairs are nearly isometric. This manifests in energy functions that try to preserve intrinsic quantities like the Laplace-Beltrami operator [34, 10] or geodesic distances [14]. As a consequence,

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

current state-of-the-art methods require expensive postprocessing via an axiomatic method and they typically do not generalize well to previously unseen data and non-isometric pairs. Instead of the purely intrinsic functional maps method, we build upon more recent advances in extrinsic-intrinsic axiomatic shape matching [11].

Contribution In this work, we systematically derive a deepified version of the current state-of-the-art axiomatic shape correspondence method smooth shells [11] from optimal transport. This allows us to integrate it into an end-to-end trainable deep network and learn optimal local features. Most prior work improves hand-crafted input features independently per vertex using shared weights. We show that this approach tends to be unstable for imperfect inputs. Instead, we use a manifold CNN architecture based on smooth spectral filters [7, 15]. Our geometric loss function measures the alignment tightness of the obtained matching which can be computed without any supervision. Finally, we show quantitative results that compare favorably to the state-of-the-art, both in terms of axiomatic and machine learning methods. In comparison to closely related learning approaches, the strong geometric nature of our method yields high-quality correspondences without the need for expensive pre or postprocessing.

2 Background

2.1 Manifold learning

The unprecedented success of convolutional neural networks (CNNs) on tasks like image and natural language processing suggests that there is a big potential of devising similar architectures for non-Euclidean data. Here, we give a non-exhaustive account of these geometric deep learning techniques on Riemannian manifolds $\mathcal{X}$ that directly imitate CNNs. For a more detailed review, we refer the reader to [6]. One straightforward approach is to learn high-level features as local correlation patterns which are explicitly defined as intrinsic patch operators [22, 5, 25, 29, 35]. While these charting based techniques have proven to be useful for 3D shapes, they make rather strong assumptions about the structure of the data and are therefore not trivially transferable to different domains like e.g. graphs or general manifolds. A complementary approach is to compute a convolution of signals $F : \mathcal{X} \to \mathbb{R}^L$ with some filter in the spectral space where it is simply a pointwise multiplication with learnable diagonal coefficient matrices $\Gamma_{l',l}$ [7]:

$$G_{l'} = h\biggl(\sum_{l=1}^{L} \Phi \Gamma_{l',l} \Phi^\dagger F_l\biggr). \tag{1}$$

This indeed corresponds to a convolution on the manifold $\mathcal{X}$ because of the well-known property that a convolution is a linear operator that commutes with the Laplacian. In this context, Φ is the spectral basis and h is some non-linearity. In general, these filters are not transferable between two different surfaces because the Laplacian eigenbasis is domain-dependent. However, it was shown in [15] that using coefficients that are smooth in the frequency domain leads to spatially localized filters:

$$\operatorname{diag}(\Gamma_{l',l}) = B\gamma_{l',l}. \tag{2}$$

Here, $B = \bigl(b_j(\lambda_i)\bigr)_{ij}$ is an alternant matrix with smooth basis functions b_j applied to the Laplacian eigenvalues λ_i and $\gamma_{l',l}$ are the learnable weights. Note that the idea of using such filters is also the foundation of graph neural networks. In this particular case, the b_j's are polynomials which alleviates the need for computing the spectral embedding of the input signals F explicitly [9, 19]. This construction vastly popularised graph neural networks, see [43] for a recent survey of subsequent work on this topic. Unfortunately, the message passing scheme from GNN's is not suitable for the kind of data that we consider in this work. A continuous 3D surface is typically discretized by a triangular mesh or a point cloud. In contrast to GNN's however, we want to learn local features that are independent of this discretization. Therefore, we will consider more general smooth spectral convolution filters in this work.

2.2 Shape correspondence

Axiomatic methods Traditional approaches directly compute a matching $\mathcal{P} : \mathcal{X} \to \mathcal{Y}$ between two input surfaces $\mathcal{X}$ and $\mathcal{Y}$ by making use of certain geometric properties and invariances of deformable shapes. This is an extensively studied topic and we only focus on approaches that are directly related to ours. For a more complete overview, we refer the reader to surveys on shape correspondence [37, 41] with [36] being the most recent one.

The functional maps [26] framework generalizes the classical correspondence problem by looking for a functional $\mathcal{C} : L^2(\mathcal{X}) \to L^2(\mathcal{Y})$ that maps not points, but functions from $\mathcal{X}$ to $\mathcal{Y}$. The matching $\mathcal{P}(x) = y$ can be recovered from $\mathcal{C}$ by mapping delta distributions $\mathcal{C}(\delta_x) = \delta_y$. In practice, we can opt for a compact representation of functions in a finite basis $(\phi_i)_{1 \leq i \leq K}$ which allows us to represent a functional map as a matrix $\mathbf{C} \in \mathbb{R}^{K \times K}$. The most common choice of basis functions ϕ_i are the Laplace-Beltrami eigenfunctions which are provably optimal for representing smooth functions on a manifold [27]. Furthermore, this choice allows us to build certain assumptions about the input objects, like near-isometry or area preservation, directly into the representation [26]. The original framework has been extended to handle partial shapes [32, 21], compute refined point-to-point maps [33], to allow for orientation preserving maps [31] and to iteratively upsample a coarse map [23].

Most recently, [11] proposed smooth shells which combines functional maps with extrinsic shape alignment. This method is able to compute high quality correspondences for challenging pairs but it also requires a good initialization to find a meaningful local minimum. The authors solve this issue by performing a stochastic Markov chain Monte Carlo (MCMC) search over the space of initial poses prior to the main pipeline. While this was shown to work well, it is also costly: In practice, roughly 100 test runs have to be performed to assess the quality of different proposal initializations.

Machine learning methods In Section 2.1, we discussed different manifold learning techniques, many of which have been successfully applied to 3D shape correspondence [22, 5, 25, 29, 35]. A complementary, more task-driven approach is to use a modified axiomatic method where the handcrafted inputs are replaced by learned features. This line of research was pioneered by deep functional maps [20] which takes SHOT [40] features as an input and refines them independently for each point using a fully connected neural network with shared weights. The resulting learned features are then used as inputs to the functional maps method [26]. Overall, this yields an end-to-end trainable pipeline because functional maps with soft correspondences is differentiable. Further extensions have successfully coupled this framework with an unsupervised loss [14, 34] or with point cloud learning techniques [10]. One thing that all of these methods have in common is that they are using the functional maps pipeline as a matching layer. Moreover, all but the last one use the same network architecture like the original deep functional maps [20] which computes features independently for each point. Similar to our approach, [38] and [44] use a differentiable Sinkhorn layer for image matching and rigid point cloud registration respectively. A more specialized approach to compute correspondences for a specific class of 3D shapes was proposed by [13]. The main idea here is to learn how to deform a template of a specific class of objects, e.g. a human template. To that end, the authors use a pointnet spatial encoder [30] and a decoder in some latent deformation parameterization.

3 Deep shells

3.1 Product space embedding

Smooth shells [11] is a hierarchical matching method that operates on the product space of extrinsic coordinates and intrinsic features. In particular, it embeds a given input shape $\mathcal{X}$ into $\mathbb{R}^{K+6}$ using the following coordinate function:

$$\mathbf{X}_k := \left(\Phi_k, X_k, \mathbf{n}_k^{\mathcal{X}}\right) : \mathcal{X} \to \mathbb{R}^{K+6}. \tag{3}$$

The intrinsic features Φ_k are defined as the first k Laplace-Beltrami eigenfunctions, $\mathbf{n}_k^{\mathcal{X}}$ are the outer normals and X_k the smoothed extrinsic coordinates, see [11] for more details. In this embedding space, an alignment between two shapes $\mathcal{X}$ and $\mathcal{Y}$ can be computed by appropriately parameterizing the intrinsic deformation with a functional map [26] and the extrinsic deformation as a pointwise translation in the Laplace-Beltrami eigenbasis Φ_k. Overall, the following energy is minimized:

$$E(\mathcal{P}, \mathbf{C}, \tau) := \left\| \mathbf{X}_k^* - \mathbf{Y}_k \circ \mathcal{P} \right\|_{L_2}^2 = \left\| \left(\Phi_k \mathbf{C}^\dagger, X_k + \Phi_k \tau, \overset{*}{\mathbf{n}}{}_k^{\mathcal{X}}\right) - \mathbf{Y}_k \circ \mathcal{P} \right\|_{L_2}^2. \tag{4}$$

Here, we need to jointly optimize for the unknown correspondences $\mathcal{P} : \mathcal{X} \to \mathcal{Y}$ and the deformation coefficients $\mathbf{C} \in \mathbb{R}^{k \times k}$ and $\tau \in \mathbb{R}^{k \times 3}$. To handle this coupled problem, the authors in [11] define an iterative scheme that gradually increases k with a fixed number of iterations and alternates between optimizing for $\mathcal{P}$ and $(\mathbf{C}, \tau)$.

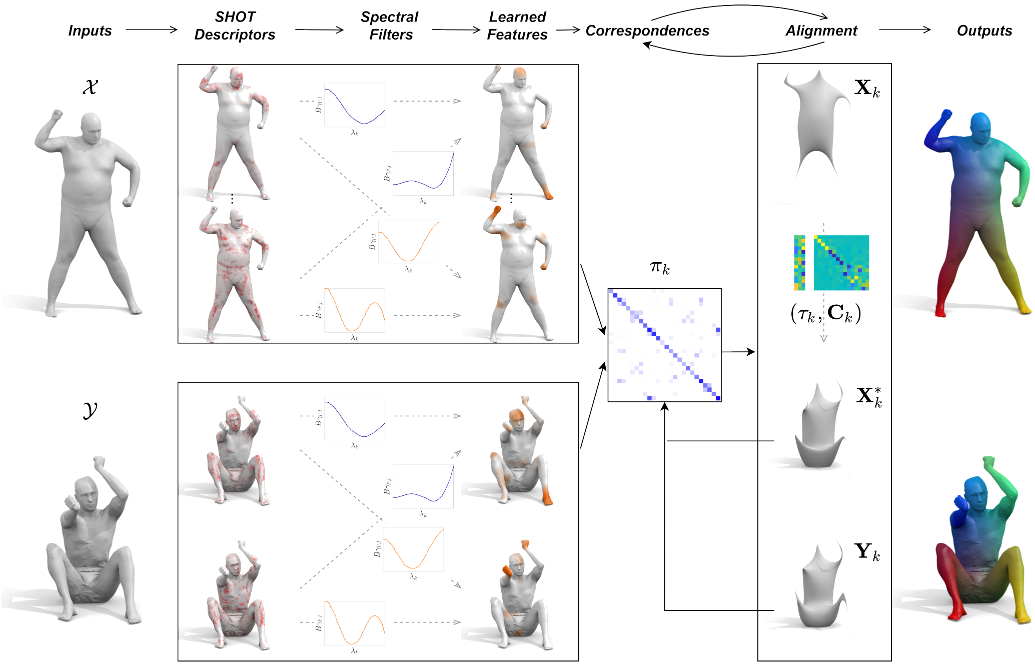


Figure 1: An overview of our network. We compute SHOT [40] descriptors on the two input shapes $\mathcal{X}$ and $\mathcal{Y}$ and apply learnable spectral convolution filters to them to obtain refined local features $G^{\mathcal{X}}$ and $G^{\mathcal{Y}}$. Those are then used to initialize the main matching layer, see Eq. (13). This scheme starts with a coarse approximation of the input geometries $\mathbf{X}_k, \mathbf{Y}_k$ and gradually increases k while alternating between updating π_k and $(\tau_k, \mathbf{C}_k)$ by minimizing Eq. (8). Finally, we output the correspondences and deformed shapes and use them to compute our unsupervised loss, defined as the average alignment tightness (8).

3.2 Deepifying smooth shells

In order to deepify smooth shells, we need to assure that all computational steps are differentiable wrt. the inputs of the method. The update of the deformation coefficients for a fixed mapping $\mathcal{P}$ can be solved as a linear least squares problem and is therefore trivially a differentiable operation:

$$(\mathbf{X}, \mathbf{Y}) \mapsto (\mathbf{C}, \tau) := \arg\min_{(\mathbf{C}, \tau)} E(\mathcal{P}, \mathbf{C}, \tau). \tag{5}$$

Unfortunately, the same does not hold for the update of $\mathcal{P}$. In the discrete case, $\mathcal{P}$ is typically represented by an assignment matrix $\mathbf{P} \in \{0,1\}^{m \times n}$ with $\mathbf{P}^\top \mathbf{1} = \mathbf{1}$. In this representation, the update of the correspondences for a fixed deformation is simply a nearest neighbor search. This, however, leads to a piecewise constant energy function and therefore does not result in a differentiable mapping. In order to make the mapping from the inputs $(\mathbf{X}, \mathbf{Y})$ to the correspondences differentiable, we will replace the strict point-to-point assignment $\mathcal{P} : \mathcal{X} \to \mathcal{Y}$ with a fuzzy correspondence π:

$$\pi \in \Pi(\mathcal{X}, \mathcal{Y}). \tag{6}$$

In this context, $\Pi(\mathcal{X}, \mathcal{Y})$ is defined as the set of probability measures on the product domain $\mathcal{X} \times \mathcal{Y}$ where the marginals correspond to the surface differentials of $\mathcal{X}$ and $\mathcal{Y}$:

$$\int_{\mathcal{X}} \mathrm{d}\pi(x, y) = \mathrm{d}y \ , \quad \int_{\mathcal{Y}} \mathrm{d}\pi(x, y) = \mathrm{d}x. \tag{7}$$

Following the idea proposed in [8], we can then reformulate our matching energy from Eq. (4) using entropy regularized optimal transport (OT):

$$E(\pi, \mathbf{C}, \tau) := \int_{\mathcal{X} \times \mathcal{Y}} \big\| \mathbf{X}_k^*(x) - \mathbf{Y}_k(y) \big\|_2^2 \mathrm{d}\pi(x, y) - \lambda H(\pi). \tag{8}$$

The entropy regularization $H(\pi)$ allows us to efficiently solve for the correspondences π using Sinkhorn's algorithm which leads to an alternating projection scheme:

$$\frac{\mathrm{d}\pi(x,y)}{\mathrm{d}x\mathrm{d}y} = \mathcal{S}^{\mathcal{X}}\big(\mathcal{S}^{\mathcal{Y}}\big(\dots\mathcal{S}^{\mathcal{X}}\big(\mathcal{S}^{\mathcal{Y}}(p_\lambda)\big)\dots\big)\big). \tag{9}$$

In this context, the operators $\mathcal{S}$ are projections of a given probability density $p : \mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ on one of the respective constraints from Eq. (7). Moreover, the input density p_λ is defined as:

$$p_\lambda(x,y) \propto \exp\bigg(-\frac{1}{\lambda}\big\|\mathbf{X}_k^*(x) - \mathbf{Y}_k(y)\big\|_2^2\bigg). \tag{10}$$

This scheme (9) is known to have a linear convergence [12] and we use a fixed number of iterations in practice. More importantly, each individual computation step of the resulting scheme is differentiable and [24] showed that this allows us to build Sinkhorn's algorithm into a neural network. Overall, we are able to minimize the OT energy (8) by alternating between updating π_k and $(\tau_k, \mathbf{C}_k)$ while gradually increasing the level of detail k. For the correspondences π_k, this is done with the Sinkhorn scheme (9), and for the deformation coefficients $(\tau_k, \mathbf{C}_k)$, this is a weighted least squares problem.

3.3 Spectral convolutions

In Equation (1), we discussed how we can define a convolution filter on an input signal $F : \mathcal{X}\to\mathbb{R}^L$ in terms of an element-wise multiplication of the spectral input features $\Phi^\dagger F$. Following prior work on graph convolutions [7, 15], we use spectrally smooth filters in a low rank basis B:

$$G_{l'} = h\bigg(\sum_{l=1}^{L}\Phi_k\big(B\gamma_{l',l}\odot\Phi_k^\dagger F_l\big)\bigg). \tag{11}$$

Here, $\odot$ denotes the pointwise product of two vectors. Note, that we use a truncated spectral basis Φ_k instead of the full basis Φ. In theory, we need infinitely many eigenfunctions $k\to\infty$ to exactly obtain a convolution with using Eq. (11). However, in the discrete world, the possible number of basis functions Φ_k is anyway bounded by the discretization coarseness and very high frequency eigenfunctions get more and more distorted by the discrete geometry. Consequently, using a moderate number of eigenpairs yields filters that are more agnostic to the discretization. Moreover, if we only use a fixed number of eigenfunctions, the approximated Fourier transform $\Phi_k^\dagger F_l$ has linear complexity. We choose the basis B as a variant of the Fourier basis defined on the frequency domain:

$$B_{ij} := b_j(\lambda_i) = \cos\bigg(\frac{\lambda_i\pi j}{T}\bigg). \tag{12}$$

The Fourier basis operates on a compact or periodic domain. This is meaningful in our case because we only use a fixed number of eigenfunctions in Eq. (11). Our overall pipeline takes SHOT [40] features $F : \mathcal{X}\to\mathbb{R}^{352}$ as an input and extracts improved local features G on both shapes using Equation (11). These features are then used to initialize deep shells by converting them to the initial soft correspondences π:

$$E_{\mathrm{init}}(\pi) := \int_{\mathcal{X}\times\mathcal{Y}}\big\|G^{\mathcal{X}}(x) - G^{\mathcal{Y}}(y)\big\|_2^2\mathrm{d}\pi(x,y) - \lambda H(\pi). \tag{13}$$

Our deep shells scheme then alternatingly updates the correspondences π and the deformation coefficients $(\mathbf{C},\tau)$ as described in the previous chapter, see Fig. 1 for an Overview. Finally, our loss function is defined as the objective value $E(\pi,\mathbf{C},\tau)$ from Eq. (8) averaged over all deep shell iterations. Intuitively, this measures the alignment tightness of the output of the main pipeline $\mathbf{X}^*$ with the reference shape $\mathbf{Y}$ which we can compute without any supervision. Our method uses both extrinsic and intrinsic information which makes the alignment tightness a robust indicator of the matching quality without knowing the ground truth correspondences.

4 Results

Implementation details We implemented our network in PyTorch using Adam optimizer [18]. Our pipeline takes 352 dimensional SHOT descriptors [40] as an input that uses geometric features within

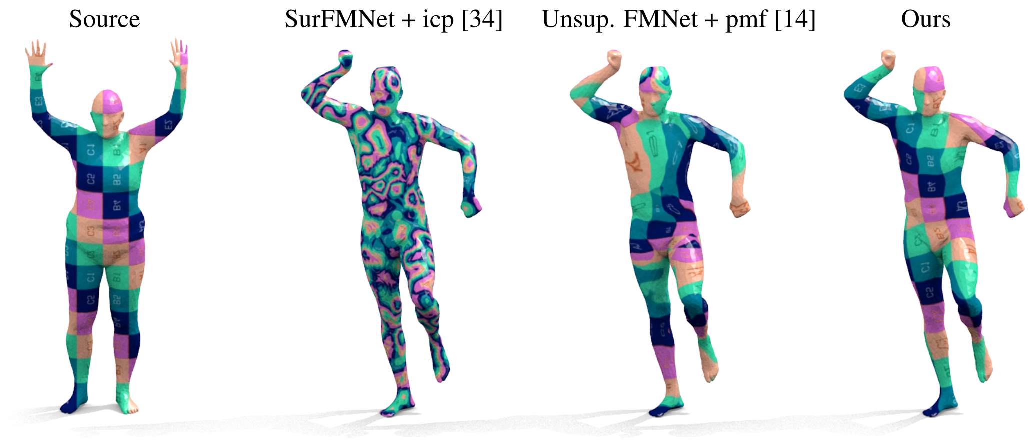

Source SurFMNet + icp [34] Unsup. FMNet + pmf [14] Ours

Figure 2: A qualitative comparison corresponding to our inter-dataset experiment in the last two columns of Table 1. The source shape is from the test set of FAUST, but the target shape is from SCAPE. Our method does not require any postprocessing but still yields the best results.

5% of the shape diameter. The inputs to our method are normalized to a fixed square root area of $\frac{2}{3}$. Furthermore, we compute 500 Laplacian eigenpairs on all inputs as a preprocessing step. In particular, we use a standard cotangent discretization of the Laplace-Beltrami operator on triangular meshes with a lumped mass matrix [28]. Our spectral convolution layer uses 120 filters on the frequency domain represented with 16 cosine basis functions each, see Eq. (12). We use 200 eigenfunctions for the truncated spectral filters from Eq. (11). The high frequency Laplacian eigenfunctions on differentiable manifolds are known to grow approximately linear with a constant incline depending on the total surface area. Consequently, the 200th eigenvalue is more or less stable across surfaces which allows us to choose a fixed frequency domain from 0 to $T = 2e4$ in Eq. (12). Analogously to smooth shells [11], we use 8 iterations from $k = 6$ to $k = 20$ on a logarithmic scale for training and a refined pipeline with up to $k = 500$ eigenfunctions for testing. Finally, we use a fixed number of 10 Sinkhorn projections and the Entropy regularization coefficient $\lambda = 0.12$ in Eq. (9).

Datasets We evaluate our method on the standard benchmarks FAUST [4] and SCAPE [2]. Instead of the normal datasets, we use the more challenging remeshed versions from [31]. These benchmarks are known to be more realistic than the original ones. Ideally, we want correspondence methods to be agnostic to the discretization because scanning of real world objects typically leads to incompatible meshings. Therefore, the remeshed versions are an improvement over the classical FAUST and SCAPE datasets which contain templates with the same number of points and connectivity. We split both datasets into training sets of 80 and 51 shapes respectively and 20 test shapes each and randomly shuffle the 80^2 and 51^2 pairs during training. Although both FAUST and SCAPE contain humans, the

				Test on - Train on			
		FAUST	SCAPE	F - S	S - F	F - F+S	S - F+S
Axiom.	BCICP [31]	6.4	11	-	-	-	-
	ZoomOut [23]	6.1	7.5	-	-	-	-
	Smooth Shells [11]	2.5	4.7	-	-	-	-
Sup.	3D-CODED [13]	2.5	31	33	31	-	-
	FMNet + pmf [20]	11 / 5.9	17 / 6.3	33 / 14	30 / 11	-	-
	GeoFMNet + zo [10]	3.1 / 1.9	4.4 / 3.0	6.0 / 4.3	11 / 9.2	-	-
Unsup.	SurFMNet + icp [34]	15 / 7.4	12 / 6.1.	32 / 23	32 / 19	33 / 32	29 / 24
	Unsup. FMNet + pmf [14]	10 / 5.7	16 / 10	22 / 9.3	29 / 12	11 / 6.2	13 / 7.7
	Ours	**1.7**	**2.5**	**2.7**	**5.4**	**1.6**	**2.4**

Table 1: A summary of our quantitative experiments. For each result, we show the mean geodesic error in $\%$ of the shape diameter. The table is subdivided into three sections with the current state-of-the-art axiomatic, supervised and unsupervised learning approaches. The odd columns show the results on the test set of FAUST remeshed trained on FAUST remeshed, SCAPE remeshed and both datasets respectively. Analogously, the results on SCAPE are in the even columns.

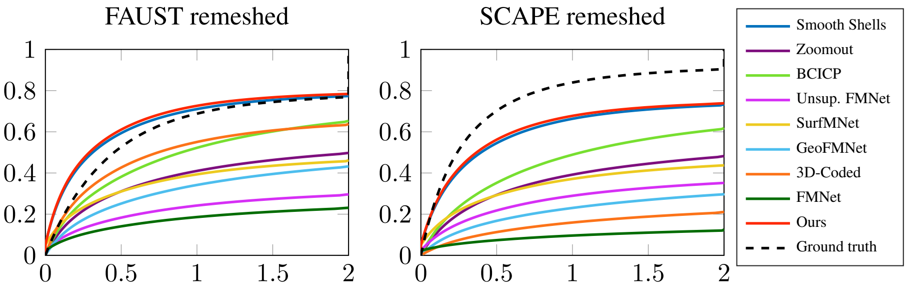


Figure 3: The cumulative conformal distortion of triangles corresponding to the first two columns of Table 1. On both datasets, our curve closely overlaps with the one from smooth shells [11]. This shows that while our method achieves a higher accuracy, it is at the same time able to obtain high quality correspondences that are comparable with smooth shells. The black dashed lines show the distortion that is caused by the remeshing of the two datasets. On FAUST remeshed, the results obtained with our method and [11] are even slightly smoother than the ground-truth.

two datasets are subject to very different challenges. FAUST contains interclass pairs of 10 different people in 10 different poses whereas the 71 SCAPE shapes all show the same person. On the other hand, the poses in SCAPE are more challenging and the geometry has less fine scale details. For example, the hands do not have discernible features like fingers which typically help disambiguate intrinsic symmetries.

Matching accuracy We report the matching accuracy on the test sets in Table 1 for our method and compare it to the current state-of-the-art of both axiomatic and learning approaches. The accuracy, in this context, is defined as the mean geodesic error over all pairs and points in the dataset, normalized by the square root area $\sqrt{\text{area}(\mathcal{Y})}$. All these experiments were conducted following the Princeton benchmark protocol [17]. What is remarkable in this context is that our method outperforms the state-of-the-art, even in comparison to the best supervised methods. Moreover, our method does not require any costly postprocessing because we built a powerful matching method directly into our network. Our method also shows a quantitative improvement over the MCMC strategy from the original smooth shells pipeline [11] and it reduces the query time during testing by a large margin. See Table 2 for a runtime comparison with other methods.

Generalization Aside from evaluating the matching accuracy on the individual benchmarks, we also show generalization results across different datasets, see Table 1. To that end, we apply the filters learned on FAUST remeshed to the test set of SCAPE remeshed, and vice versa. These results

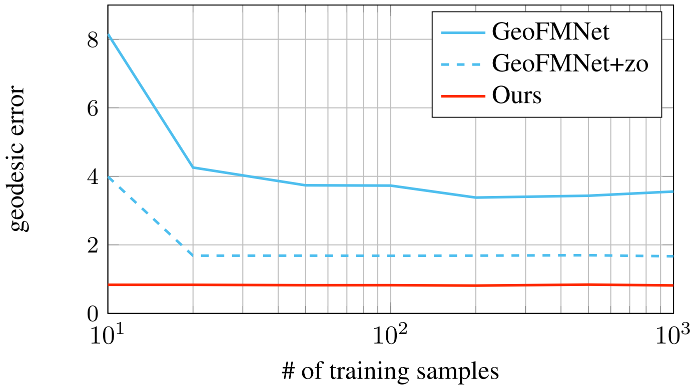


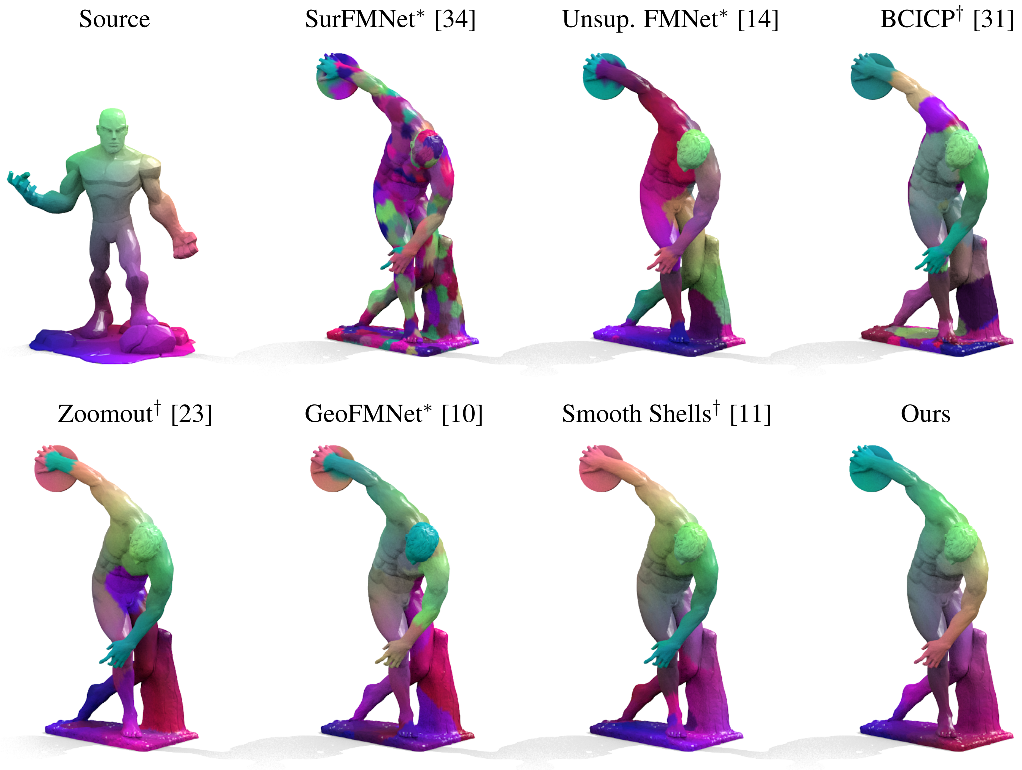
Figure 4: A quantitative comparison of our method and GeoFMNet [10] trained on SURREAL [42] with a varying number of training samples, evaluated on the test set of FAUST [4]. In particular, we show the mean geodesic error in $\%$ of the diameter with training set sizes ranging from 10 to 1000.

B.I. [31]	ZO [23]	Sh. [11]	3D-C. [13]	FM. [20]	G.FM. [10]	S.FM. [34]	U.FM. [14]	Ours
0.9	1.8	8.7	-	4.9	1.2	4.9	4.9	8.7
880.	41.	116.	725.	0.2	0.7	0.8	0.2	4.9
-	-	-	-	223.	35.	43.	216.	-
881.	43.	125.	725.	228.	37.	49.	221.	14.

Table 2: A runtime comparison corresponding to the experiments in the first two columns of Table 1. In the first three rows we display the average pre-processing, query and post-processing time for one pair at test time and in the last row the total time (in seconds). For training, [20] and [14] additionally require geodesic distance matrices which increases the precomputation time by $\sim$ 35s per shape.

show that our learned filters are able to extract robust local features for previously unseen data, even when the local geometry of the inputs varies significantly. The final experiment we present in Table 1 shows that our method can be trained across different datasets. In particular, we train our network on all shapes of the FAUST and SCAPE training sets, including inter-dataset pairs, and test it on the individual datasets. Note, that this is only possible for an unsupervised method because there are no ground-truth labels between FAUST and SCAPE. In comparison to prior work, our method can be trained stably on this challenging setup. Remarkably, it even shows a slight improvement over the results where we trained exclusively on the individual datasets which shows that it can extract additional information from the hybrid pairs.

Training set size Additionally to the results on FAUST and SCAPE, we show a quantative evaluation on synthetic humans from the SURREAL dataset [42], evaluated on the test set of the FAUST [4] registrations. In comparison to GeoFMNet [10], our method produces stable results for as few as 10 training shapes, see Fig. 4. These results suggest that our robust OT matching layer is particularly useful when the amount of training data is limited.



Source

SurFMNet* [34]

Unsup. FMNet* [14]

BCICP† [31]

Zoomout† [23]

GeoFMNet* [10]

Smooth Shells† [11]

Ours

Figure 5: A qualitative comparison on a challenging non-isometric pair. The source shape is from [14], the target shape is a real scan of the statue "Discobolus" from the British Museum in London [1]. Methods with a star (*) require postprocessing and (†) denotes axiomatic approaches.

Qualitative experiments The results in the last two columns of Table 1 indicate that our method is able to learn robust features that are applicable to hybrid pairs between FAUST and SCAPE. Unfortunately, we cannot verify this empirically due to the lack of ground-truth correspondences between the datasets. Instead, we show a qualitative example of an inter-dataset pair in Figure 2 with comparisons to the other two unsupervised methods from Table 1. Additionally, we show a qualitative evaluation on a non-isometric pair in Figure 5. Here, we use the weights trained on FAUST remeshed from our experiments in Table 1 for all but the axiomatic methods.

Map smoothness Apart from the matching accuracy, we also evaluate the map smoothness of our obtained correspondences. This can be quantified as the conformal distortion of individual triangles, see [16, Eq. (3)] for a definition. Having smooth maps is crucial for most applications like information transfer or dense pose labeling because fine scale noise can distort the information. This effect can be visualized by mapping a texture from the surface $\mathcal{X}$ to $\mathcal{Y}$ which shows how faithfully high frequency details are preserved. In Figure 3, we show a quantitative comparison of the conformal distortion with other learning and axiomatic methods. Our results indicate that the quality of our maps is comparable to smooth shells, although we do not require an expensive preprocessing or the as-rigid-as-possible regularizer used by [11].

5 Conclusion

We presented deep shells, a new framework for 3D shape matching that is based on entropy regularized optimal transport. While most prior learning methods on geometric domains use either extrinsic or intrinsic information to obtain correspondences from local features, our approach operates on both domains jointly in a hierarchical pipeline. This embedding fully acknowledges the geometric nature of Riemannian manifolds: It is agnostic to the discretization while using both the extrinsic and intrinsic surface geometry. We show that this greatly increases the robustness of our network, even in an unsupervised setting. In comparison to closely related prior work, we use a spectral CNN feature extractor instead of refining hand-crafted descriptors independently for each vertex. Finally, we show quantitative results on 3D shape matching benchmarks that significantly increase the state-of-the-art. Besides the standard error on individual benchmarks, our method shows compelling generalization results across different datasets.

Acknowledgements

We would like to thank Zorah Lähner and Florian Bernard for useful discussions. This work was supported by the Collaborative Research Center SFB-TRR 109 'Discretization in Geometry and Dynamics', the ERC Consolidator Grant "3D Reloaded", the Humboldt Foundation through the Sofja Kovalevskaja Award and the Helmholtz Association under the joint research school "Munich School for Data Science - MUDS".

Broader Impact

With the ever increasing number of surface acquisition devices and techniques, the demand for algorithms that can process 3D data directly nowadays is higher than ever. The overarching goal is to perform recognition tasks directly on 3D sensory inputs, similarly to the way that we as humans make sense of our environment. In comparison to 2D images, which are a mere projection of the world surrounding us, geometric data is more robust to secondary effects like lighting conditions and general appearances of 3D objects. It is therefore imperative for the vision community to focus its efforts on both 2D and 3D understanding. Regarding ethical aspects, in the context of computer vision algorithms there is always the distinct possibility of dual use, e.g. for military aims. However, we believe that there is not an immediate risk of misuse associated with our algorithm.

References

[1] Scan of discobolus at the british museum. https://www.myminifactory.com/object/3d-print-discobolus-at-the-british-museum-london-7896. Accessed: 2020-05-30.

[2] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[3] Matthieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. *IEEE International Conference on Computer Vision (ICCV) - Workshop on Dynamic Shape Capture and Analysis*, 2011.

[4] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.

[5] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in neural information processing systems*, pages 3189–3197, 2016.

[6] 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.

[7] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[10] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. *arXiv preprint arXiv:2003.14286*, 2020.

[11] 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.

[12] Joel Franklin and Jens Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its applications*, 114:717–735, 1989.

[13] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[14] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019.

[15] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. arxiv (2015). *arXiv preprint arXiv:1506.05163*, 2015.

[16] Kai Hormann and Günther Greiner. Mips: An efficient global parametrization method. Technical report, Erlangen-Nuernberg University (Germany) Computer Graphics Group, 2000.

[17] Vladimir G Kim, Yaron Lipman, and Thomas A Funkhouser. Blended intrinsic maps. *Transactions on Graphics (TOG)*, 30(4), 2011.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[20] 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.

[21] Or Litany, Emanuele Rodolà, Alex M Bronstein, Michael M Bronstein, and Daniel Cremers. Non-rigid puzzles. *Computer Graphics Forum (CGF), Proceedings of Symposium on Geometry Processing (SGP)*, 35(5), 2016.

[22] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[23] 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] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

[25] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[26] 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, 2012.

[27] Beresford N Parlett. *The symmetric eigenvalue problem*, volume 20. siam, 1998.

[28] Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *EXPERIMENTAL MATHEMATICS*, 2:15–36, 1993.

[29] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

[30] 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.

[31] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6):248:1–248:16, December 2018.

[32] Emanuele Rodolà, Luca Cosmo, Michael Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum (CGF)*, 2016.

[33] Emanuele Rodolà, Michael Moeller, and Daniel Cremers. Regularized pointwise map recovery from functional correspondence. In *Computer Graphics Forum*, volume 36, pages 700–711. Wiley Online Library, 2017.

[34] 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.

[35] Klaus Hildebrandt Ruben Wiersma, Elmar Eisemann. Cnns on surfaces using rotation-equivariant features. *Transactions on Graphics*, 39(4), July 2020.

[36] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, pages 1–17, 2019.

[37] Joaquim Salvi, Carles Matabosch, David Fofi, and Josep Forest. A review of recent range image registration methods with accuracy evaluation. *Image Vision Comput.*, 25(5):578–596, 2007.

[38] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020.

[39] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[40] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. *In Proceedings of European Conference on Computer Vision (ECCV)*, 16(9):356–369, 2010.

[41] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[42] Gul 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.

[43] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[44] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11824–11833, 2020.

Chapter 8

NeuroMorph: Unsupervised Shape Interpolation and Correspondence in One Go

Existing work on shape correspondence and shape interpolation of deformable object categories indicates, that there are significant synergies between the two challenges [3]. For instance, most shape interpolation approaches in the literature require exact correspondences as input. In turn, correspondences can be recovered from time-continuous interpolation sequences by mapping points along individual poses. Yet, a majority of existing approaches address the two challenges individually. We devise a novel learning approach that jointly predicts correspondence maps and interpolation sequences in a single forward pass. To this end, we introduce a neural network model, comprising a feature extraction backbone and an interpolator. The feature backbone is based on a novel network architecture, interleaving EdgeConv message passing layers with global feature pooling. To avoid overfitting to a specific input discretization, we apply remeshing augmentation during training. We obtain a soft correspondence map from the learned features by applying the cosine similarity distance and a per-row softmax. The interpolator then maps the predicted correspondences to an interpolation sequence, conditioned on the input vertex positions of $\mathcal{X}$ and the evaluation time $t \in [0, 1]$. Our training loss consists of three terms, requiring that the predicted interpolation overlaps with $\mathcal{Y}$ at $t = 1$, enforcing geodesic distance preservation, and as-rigid-as-possible deformations for the interpolation sequence. The resulting loss is fully unsupervised, *i.e.*, does not require ground truth correspondence maps for training. We evaluate our approach on several 3D shape datasets, both in terms of shape correspondence and interpolation. We further demonstrate that it can be leveraged to synthesize novel intermediate poses as a form of data augmentation when learning generative shape spaces. Finally, we propose a new benchmark of deformable objects with both animal and human shapes, and provide dense ground truth maps between all poses – including inter-class pairs.

Individual contributions

Leading role in realizing the scientific project.

Problem definition	*significantly contributed*
Literature survey	*significantly contributed*
Data curation	*contributed*
Implementation	*significantly contributed*
Experimental evaluation	*significantly contributed*
Preparation of the manuscript	*significantly contributed*

Copyright

©2021 IEEE. Reprinted, with permission, from

Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi

NeuroMorph: Unsupervised Shape Interpolation and Correspondence in One Go

2021 Conference on Computer Vision and Pattern Recognition (CVPR)

DOI: 10.1109/CVPR46437.2021.00739

We include the accepted version of the original publication [4].

RightsLink

Sign in/Register

NeuroMorph: Unsupervised Shape Interpolation and Correspondence in One Go

Requesting permission to reuse content from an IEEE publication

Conference Proceedings:
2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Author:
[::Marvin::] [::Eisenberger::]; David Novotny; Gael Kerchenbaum; Patrick Labatut; Natalia Neverova; Daniel Cremers; Andrea Vedaldi

Publisher: IEEE

Date: 20-25 June 2021

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

NeuroMorph: Unsupervised Shape Interpolation and Correspondence in One Go

Marvin Eisenberger[*,†], David Novotny[*], Gael Kerchenbaum[*], Patrick Labatut[*],
Natalia Neverova[*], Daniel Cremers[†], Andrea Vedaldi[*]

Facebook AI Research[*], Technical University of Munich[†]

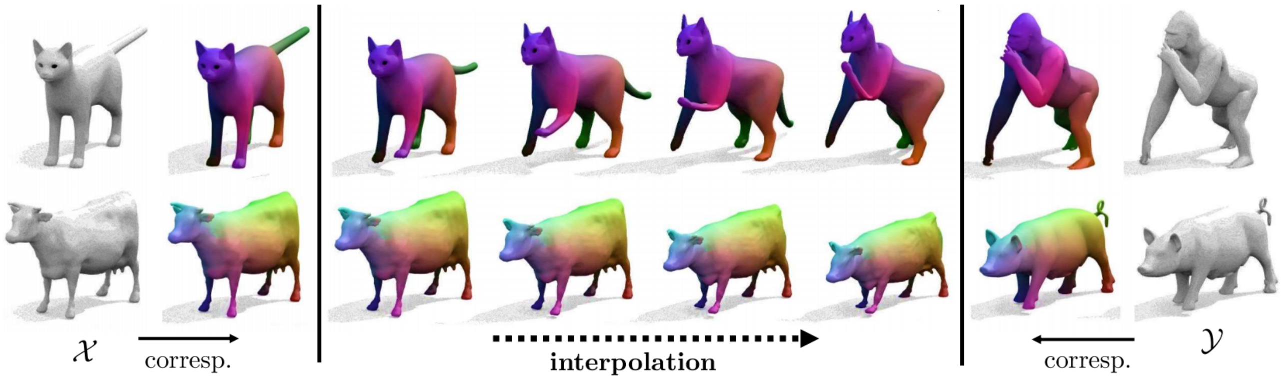


Figure 1: Our **NeuroMorph** neural network takes as input two meshes (left and right) and produces in one go (i.e. in a single feed-forward pass) a continuous **interpolation** and point-to-point **correspondence** between them (color coded). The interpolation, expressed as a displacement field, changes the pose of the source shape while preserving its identity.

Abstract

We present NeuroMorph, a new neural network architecture that takes as input two 3D shapes and produces in one go, i.e. in a single feed forward pass, a smooth interpolation and point-to-point correspondences between them. The interpolation, expressed as a deformation field, changes the pose of the source shape to resemble the target, but leaves the object identity unchanged. NeuroMorph uses an elegant architecture combining graph convolutions with global feature pooling to extract local features. During training, the model is incentivized to create realistic deformations by approximating geodesics on the underlying shape space manifold. This strong geometric prior allows to train our model end-to-end and in a fully unsupervised manner without requiring any manual correspondence annotations. NeuroMorph works well for a large variety of input shapes, including non-isometric pairs from different object categories. It obtains state-of-the-art results for both shape correspondence and interpolation tasks, matching or surpassing the performance of recent unsupervised and supervised methods on multiple benchmarks.

1. Introduction

The ability to relate the 3D shapes of objects is of key importance to fully understand object categories. Objects can change their shape due to articulation, other motions and intra-category variations, but such changes are not arbitrary. Instead, they are strongly constrained by the category of the objects at hand. Seminal works such as [33] express such constraints by learning statistical shape models. In order to do so, they need to put in correspondence large collections of individual 3D scans, which they do by exploiting the fact that individual objects deform continuously in time, and by using some manual inputs to align different object instances. Due to the high complexity of obtaining and preprocessing such 3D data, however, these models remain rare and mostly limited to selected categories such as humans that are of sufficient importance in applications. In this paper, we are thus interested in developing a method that can learn to relate different 3D shapes fully automatically, interpolating a small number of 3D reconstructions, and in a manner which is less specific to a single category (Figure 1).

Due to the complexity of this task, authors have often considered certain sub-problems in isolation. One is to establish *point-to-point correspondences* between shapes, telling which points are either physically identical (for a given articulated object) or at least analogous (for similar objects). A second important sub-problem is *interpolation*, which amounts to continuously deforming a source shape into a target shape. Interpolation must produce a collection of intermediate shapes that are meaningful in their own right, in the sense of being plausible samples from the underlying shape distribution. The interpolation *trajectory* must be also meaningful; for instance, if the deformation between two shapes can be explained by the articulation of an underlying physical object, this solution is preferred.

The correspondence and interpolation problems have been addressed before extensively, by using tools from geometry and, more recently, machine learning. Most of the existing algorithms, however, require at least some manual supervision, for example in the form of an initial set of sparse shape correspondences. Furthermore, correspondence and interpolation are rarely addressed together due to their complexity.

In this paper, we advocate instead for an approach in which the correspondence and interpolation problems are solved *simultaneously*, and in an *unsupervised* manner. To do this, we introduce *NeuroMorph*, a new neural network that solves the two problems in a single feed forward pass. We show that, rather than making learning more difficult, integrating two goals reinforces them, making it possible to obtain excellent empirical results. Most importantly, we show that NeuroMorph can be learned in a *fully unsupervised* manner, given only a collection of 3D shapes as input and certain geometric priors for regularization.

NeuroMorph advances the state of the art in shape matching and interpolation, surpassing by a large margin prior unsupervised methods and often matching the quality of supervised ones. We show that NeuroMorph can establish high-quality point-to-point correspondences without any manual supervision even for difficult cases in which shapes are related by substantial non-isometric deformations (such as between two different types of animals, like a cat and a gorilla, as in Figure 1) which have challenged prior approaches. Furthermore, we also show that NeuroMorph can interpolate effectively between different shapes, acting on the *pose* of a shape while leaving its *identity* largely unchanged. To demonstrate the quality of the interpolation, we use it for *data augmentation*, extending a given dataset of 3D shapes with intermediate ones. Augmenting a dataset in this manner is useful when, as it is often the case, 3D training data is scarce. We show the benefits of this form of data augmentation to supervise other tasks, such as reconstructing continuos surfaces from sparse point clouds.

Our new formulation also gives rise to some interesting applications: Since our method learns a function that produces correspondence and interpolation in a single feed forward pass, it can be used not only to align different shapes, but also for pose transfer, digital puppeteering and other visual effects.

2. Related work

To the best of our knowledge, we are the first to consider the problem of learning a mapping that, given a pair of shapes as input, predicts in a feed-forward manner their correspondences and interpolation. This should be contrasted to other recent approaches to shape understanding such as LIMP [7] that try to learn a shape space. These architectures need to solve the difficult problem of *generating* or *auto-encoding* 3D shapes. Unfortunately, designing good generator networks for 3D shapes remains a challenging problem. In particular, it is difficult for these networks to generalize beyond the particular family of shapes (e.g. humans) experienced during training. By contrast, we do not try to generate shapes outright, but only to *relate* pairs of given input shapes. This replaces the difficult task of shape generation with the easier task of generating a deformation field, working well with a large variety of different shapes.

The rest of the section discusses other relevant work.

Shape correspondences. The problem of establishing correspondences between 3D shapes has been studied extensively (see the recent surveys [55, 52, 49]). Traditional approaches define axiomatic algorithms that focus on a certain subclass of problems like rigid transformations [63, 66], nearly-isometric deformations [39, 1, 57, 44], bounded distortion [35, 13] or partiality [32, 45, 31]. Methods such as functional maps [39] reduce matching to a spectral analysis of 3D shapes.

More recent approaches use machine learning and are often based on developing deep neural networks for non-image data such as point clouds, graphs and geometric surfaces [5]. Charting-based methods define learnable intrinsic patch operators for local feature aggregation [34, 3, 37, 41, 48]. Deep functional maps [30] aim at combining a learnable local feature extractor with a differentiable matching layer based on the axiomatic functional maps framework [39]. Subsequent works [20, 47] extended this idea to the unsupersived setting and combined it with learnable point cloud feature extractors [9, 50]. Moreover, [14] recently proposed to replace the functional maps layer with a multi-scale correspondence refinement layer based on optimal transport. Another related approach is [18] which uses a PointNet [42] encoder to align a human template to point cloud observations to compute correspondences between different human shapes.

Feature extractors for 3D shapes. Several authors have proposed to reduce matching 3D shapes to matching local shape descriptors. A common remedy is learning to refine hand-crafted descriptors such as SHOT [54], e.g. with metric learning [30, 20, 47]. In practice, this approach is highly dependent on the quality of the input features and tends to be unstable due to the noise and the complex variable structures of real 3D data. More recently, authors have thus looked at *learning* such descriptors directly [9, 50] with point cloud feature extractors [53, 43]. Another possibility is to interpret a 3D mesh as a graph and use graph convolutional neural networks [29, 8]. The challenge here is that the specific graph used to represent a 3D shape is partially arbitrary (because we can triangulate a surface in many different ways), and graph convolutions must discount geometrically-irrelevant changes (this is often done empirically by re-meshing as a form of data augmentation).

Shape spaces, manifolds and interpolation. 3D shapes can be interpreted as low-dimensional manifolds in a high-dimensional embedding space [27, 62, 23, 22]. The low-dimensional manifold can, for example, capture the admissible pose of an articulated object [65, 58, 24]. Given a shape manifold, interpolation can then be elegantly formulated as finding geodesic paths between two shapes. However, building shape manifolds may be difficult in practice, especially if the input shapes are not in perfect correspondence. Therefore, also inspired by LIMP [7], for training NeuroMorph we follow approaches such as [12, 11] that avoid building a shape manifold explicitly and instead directly construct geodesic paths that originate at the source shapes and terminate in the vicinity of the target shapes.

Generative shape models. While manifolds provide a geometric characterization of a shape space, generative models provide a statistical one. One particular challenge in this context is designing shape-decoder architectures that can generate 3D surfaces from a latent shape representation. A straightforward solution is predicting occupancy probabilities on a 3D voxel grid [6], but the cost of dense, volumetric representations limits the resolution. Other approaches decode point clouds [15, 64] or 3D meshes [19, 16] directly. A recent trend is encoding an implicit representation of a 3D surface in a neural network [36, 40]. This allows for a compact shape representation and a decoder that can generate shapes of an arbitrary topology. Following the same methodology, [38] predicts a time-dependent displacement field that can be used to interpolate 3D shapes. This approach is related to ours, but it requires 4D supervision during training, whereas our method is trained on a sparse set of poses. ShapeFlow [26] predicts dense velocity fields for template-based reconstruction. Similarly, [60] computes an intrinsic displacement field to align a pair of input shapes, but they do not predict an intermediate sequence.

3. Method

Let $\mathcal{X}$ and $\mathcal{Y}$ be 3D shapes, respectively called the *source* and the *target*, expressed as triangular meshes with vertices $\mathbf{X} = (\mathbf{x}_i)_{1\leq i\leq n} \in \mathbb{R}^{n\times 3}$ and $\mathbf{Y} = (\mathbf{y}_j)_{1\leq j\leq m} \in \mathbb{R}^{m\times 3}$, respectively. Our goal is to learn a function

$$f \,:\, (\mathcal{X}, \mathcal{Y}) \longmapsto (\Pi, \Delta),$$

that, given the two shapes as input, predicts 'in one go' a correspondence matrix Π and an interpolation flow Δ between them. The matrix $\Pi \in [0,1]^{n\times m}$ sends probabilistically the vertices $\mathbf{x}_i$ of the source mesh $\mathcal{X}$ to corresponding vertices $\mathbf{y}_j$ in the target mesh $\mathcal{Y}$ and is thus row-stochastic (i.e. $\Pi\mathbf{1} = \mathbf{1}$). The interpolating flow $\Delta(t) \in \mathbb{R}^{n\times 3}$, $t \in [0,1]$ shifts continuously the vertices of the source mesh, forming trajectories:

$$\mathbf{X}(t) := \mathbf{X} + \Delta(t), \tag{1}$$

that take them from their original locations $\mathbf{X}(0) = \mathbf{X}$ to new locations $\mathbf{X}(1) \approx \Pi\mathbf{Y}$ close to the corresponding vertices in the target mesh.

The function f is given by two deep neural networks. The first, discussed in Section 3.1, establishes the correspondence matrix Π and the second, discussed in Section 3.2, outputs the shifts $\Delta(t)$ for arbitrary values of $t \in [0,1]$. Both networks are trained end-to-end in an unsupervised manner, as described in Section 3.3.

3.1. Correspondences and vertex features

The correspondence matrix Π between meshes $\mathcal{X}$ and $\mathcal{Y}$ is obtained by extracting and then matching features of the mesh vertices. The features are computed by a deep neural network: $\tilde{\mathbf{X}} = \Phi(\mathcal{X}) \in \mathbb{R}^{n\times d}$ that takes the shape $\mathcal{X}$ as input and outputs a matrix $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_i)_{1\leq i\leq n}$ with a feature vector $\tilde{\mathbf{x}}_i$ for each vertex i of the mesh. Given analogous features $\tilde{\mathbf{Y}} = (\tilde{\mathbf{y}}_i)_{1\leq i\leq m} \in \mathbb{R}^{m\times d}$ for the target shape, the correspondence matrix is obtained by comparing features via the cosine similarity and normalizing the rows using the softmax operator:

$$\Pi_{ij} := \frac{\exp(\sigma s_{ij})}{\sum_{k=1}^{m}\exp(\sigma s_{ik})} \;\text{ s.t. }\; s_{ij} := \frac{\langle \tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_j \rangle_2}{\|\tilde{\mathbf{x}}_i\|_2 \|\tilde{\mathbf{y}}_j\|_2}, \tag{2}$$

with temperature $\sigma \in \mathbb{R}^+$. In this way, $\Pi\mathbf{1} = \mathbf{1}$ and Π can be interpreted as a soft assignment of source vertices $\mathbf{x}_i$ in $\mathcal{X}$ to target vertices $\mathbf{y}_j$ in $\mathcal{Y}$.

Feature extractor network. Next, we describe the neural network $\Phi(\mathcal{X})$ that extracts the feature vectors that appear in Equation (2) (this network is also illustrated in Figure 2). While different designs are conceivable, we propose here one based on successive local feature aggregation and global feature pooling. The architecture makes use of the mesh vertices $\mathbf{X}$ as well as the mesh topology. The

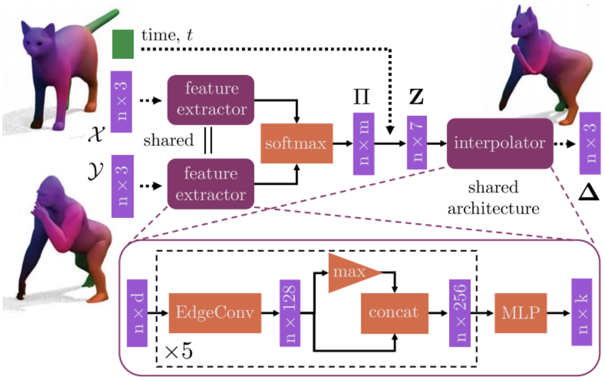


Figure 2: **NeuroMorph.** An overview of our model.

latter is specified by the neighborhood structure $\mathcal{E}$, where $(i, j) \in \mathcal{E} \subset \{1, \dots, n\}^2$ means that vertex $\mathbf{x}_j$ is connected to vertex $\mathbf{x}_i$ by a triangle edge. Thus, the mesh is fully specified by the pair $\mathcal{X} = (\mathbf{X}, \mathcal{E})$.

The layers of the network Φ are given by EdgeConv [61] graph convolution operators implemented via residual subnetworks [21]. In more detail, each EdgeConv layer takes as input vertex features $\tilde{\mathbf{X}} = (\tilde{\mathbf{x}}_i)_{1 \le i \le n}$ on $\mathcal{X}$ and computes an improved set of features $\tilde{\mathbf{X}}' = (\tilde{\mathbf{x}}'_i)_{1 \le i \le n}$ via the expression:

$$\tilde{\mathbf{x}}'_i := \max_{j:(i,j)\in\mathcal{E}} h_\phi(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i). \tag{3}$$

Here, a small residual network h_ϕ is used to combine the feature $\tilde{\mathbf{x}}_i$ of the i-th vertex with the feature $\tilde{\mathbf{x}}_j - \tilde{\mathbf{x}}_i$ of one of the edges adjacent to it. This is repeated for all edges incident on the i-th vertex and the results are aggregated via component-wise max-pooling over the mesh neighborhood $\{j : (i, j) \in \mathcal{E}\}$, resulting in an updated vertex feature $\tilde{\mathbf{x}}'_i$.

The EdgeConv layer can effectively learn the local geometric structures in the vicinity of a point. However, that alone is not sufficient to resolve dependencies in terms of the global geometry, since the message passing only allows for a local information flow. Therefore, we append a global feature vector to the point features after each EdgeConv refinement, by applying the max pooling operator globally:

$$\tilde{\mathbf{x}}''_i := \big(\tilde{\mathbf{x}}'_i, \max_{1\le i\le n} \tilde{\mathbf{x}}'_i\big). \tag{4}$$

The network Φ is given by a succession of these layers, forming a chain $\tilde{\mathbf{X}} \to \tilde{\mathbf{X}}' \to \tilde{\mathbf{X}}'' \to \dots$ alternating global (3) and local (4) update steps. The input features $\tilde{\mathbf{X}} = (\mathbf{X}, \mathbf{N})$ are given by the concatenation of the absolute position of the mesh vertices $\mathbf{X}$ with the outer normals at the vertices $\mathbf{N} = (\mathbf{n}_i)_{1 \le i \le n}$ (the normal vectors $\mathbf{n}_i$ are computed by averaging over face normals adjacent to $\mathbf{x}_i$).

3.2. Interpolator

We are now ready to describe the interpolator component of our model. Recall that the goal is to predict a displacement operator Δ such that the trajectory $\mathbf{X}(t) = \mathbf{X} + \Delta(t)$ smoothly shifts the point of the first mesh to points in the second. Notice that $\Delta(t) \in \mathbb{R}^{n\times 3}$ is just a collection of 3D vectors associated to each mesh vertex, just like the vertex positions, normals and feature vectors in the previous section. Thus, we offload the calculation of the displacements to a similar convolutional neural network and write: $\Delta(t) = \Psi(\mathcal{X}, \mathcal{Y}, t)$. The difference is the input to the network Ψ, which is now given by the 7-dimensional feature vectors $\mathbf{Z} \in \mathbb{R}^{n\times 7}$:

$$\mathbf{Z} := \big(\mathbf{X},\, \Pi\mathbf{Y} - \mathbf{X},\, \mathbf{1}t\big). \tag{5}$$

These feature vectors consist of the vertices $\mathbf{X}$ of the source shape $\mathcal{X}$, the offset vectors $\Pi\mathbf{Y} - \mathbf{X}$ predicted by the correspondence module of Section 3.1[1], and the time variable t ('broadcast' to all vertices by multiplication with a vector of all ones). Just like the network Φ in Section 3.1, the network Ψ alternates global (3) and local (4) update steps to compute a sequence of updated features $\mathbf{Z} \to \mathbf{Z}' \to \mathbf{Z}'' \dots \to \mathbf{V}$ terminating in a matrix $\mathbf{V} \in \mathbb{R}^{n\times 3}$. The final displacements are then given by a scaled version of $\mathbf{V}$, and are set to $\Delta(t) = t\mathbf{V}(t)$.

In this manner, the network can immediately obtain a trivial (degenerate) solution to the interpolation problem by setting $\mathbf{V}(t) = \Pi\mathbf{Y} - \mathbf{X}$, which amounts to copying verbatim part of the input features $\mathbf{Z}$. This result is a simple linear interpolation of the mesh vertices, trivially satisfying the boundary conditions of the interpolation:

$$\mathbf{X}(0) = \mathbf{X} + 0 \cdot \mathbf{V}(0) = \mathbf{X}, \tag{6}$$

$$\mathbf{X}(1) = \mathbf{X} + 1 \cdot (\Pi\mathbf{Y} - \mathbf{X}) = \Pi\mathbf{Y}. \tag{7}$$

Linear interpolation provides a sensible initialization, but is in itself a degenerate solution as we wish to obtain 'geometrically plausible' deformations of the mesh. To prevent the network from defaulting to this case, we thus need to incentivize geometrically meaningful deformations during training, which we do in the next section.

3.3. Learning

In this section, we show how we can train the model in an *unsupervised*[2] *manner*. That is, given only a collection of example meshes with no manual annotations, our method simultaneously learns to interpolate and establish

[1] Note that $\Delta(1) = \Pi\mathbf{Y} - \mathbf{X}$.

[2] In practice, the only assumption we make about the input objects is that they are in an approximately canonical rigid pose in terms of the up-down and front-back orientation. For most existing benchmarks this holds trivially without any further preprocessing. The recent paper by [50] calls this setup weakly supervised.

point-to-point correspondences between them. This sets it apart from prior work on shape interpolation which either require dense correspondences during training or, in the case of classical axiomatic interpolation methods, even at test time. Learning comprises three signals, encoded by three corresponding losses:

$$\ell := \lambda_{\mathrm{reg}}\ell_{\mathrm{reg}} + \lambda_{\mathrm{arap}}\ell_{\mathrm{arap}} + \lambda_{\mathrm{geo}}\ell_{\mathrm{geo}}. \tag{8}$$

The loss ℓ_{reg} ensures that correspondences and interpolation correctly map the source mesh on the target mesh, and the other two ensure that this is done in a geometrically meaningful way. The latter is done by constraining the trajectory $(\mathbf{X}(t))_{t\in[0,1]}$ generated by the model. Recall that the model can be queried for an arbitrary value $t \in [0,1]$, and it is thus able to produce interpolations that are truly continuous in time. During training, in order to compute our losses, we sample predictions $\mathbf{X}_0, \dots, \mathbf{X}_T$ for an equidistant set of discrete time steps $\mathbf{X}_k := \mathbf{X}(k/T)$ where $k = 0, \dots, T$.

Registration loss. Requirement Equation (6) holds trivially as $\Delta(0) = 0$ is built into our model definition (see Equation (1)). For Equation (7), we introduce the registration loss: $\ell_{\mathrm{reg}}\big(\mathbf{X}_T, \mathbf{Y}, \Pi\big) := \|\Pi\mathbf{Y} - \mathbf{X}_T\|_2^2$. Since our goal is to compute shape interpolations without any supervision, we use the soft correspondences Π estimated by our model instead of ground-truth annotations.

As-rigid-as-possible loss. In general, there are infinitely many conceivable paths between a pair of shapes. In order to restrict our method to plausible sequences, we regularize the path using the theory of shape spaces [27, 62, 23]. As we work with discrete time, we approximate the 'distance' between shapes in the shape space manifold by means of the local distortion metric between two consecutive states $\mathbf{X}_k$ and $\mathbf{X}_{k+1}$. To that end, we choose the as-rigid-as-possible [51] metric:

$$E_{\mathrm{arap}}\big(\mathbf{X}_k, \mathbf{X}_{k+1}\big) := \\ \frac{1}{2} \min_{\substack{\mathbf{R}_i \in SO(3) \\ i=1,\dots,n}} \sum_{(i,j)\in\mathcal{E}} \big\|\mathbf{R}_i(\mathbf{X}_{k,j} - \mathbf{X}_{k,i}) - (\mathbf{X}_{k+1,j} - \mathbf{X}_{k+1,i})\big\|_2^2.$$

Intuitively, this functional rotates the local coordinate frame of each point in $\mathbf{X}_k$ to the corresponding deformed state $\mathbf{X}_{k+1}$ and penalizes deviations from locally rigid transformations. Moreover, the rotation matrices $\mathbf{R}_i$ can be computed in closed form which allows for an efficient optimization of E_{arap} (see [51] for more details). Finally, we can use this functional to construct the first component of our loss function for the whole sequence $(\mathbf{X}_k)_k$:

$$\ell_{\mathrm{arap}}\big(\mathbf{X}_0, \dots, \mathbf{X}_T\big) := \\ \sum_{k=0}^{T-1} E_{\mathrm{arap}}(\mathbf{X}_k, \mathbf{X}_{k+1}) + E_{\mathrm{arap}}(\mathbf{X}_{k+1}, \mathbf{X}_k). \tag{9}$$

		err.	p.p.	w/o p.p.
Axiom.	BCICP [44]	6.4	—	—
	ZoomOut [35]	6.1	—	—
	Smooth Shells [13]	2.5	—	—
Sup.	3D-CODED [18]	2.5	—	—
	FMNet [30]	5.9	PMF	11
	GeoFMNet [9]	1.9	ZO	3.1
Unsup.	SurFMNet [47]	7.4	ICP	15
	Unsup. FMNet [20]	5.7	PMF	10
	Weakly sup. FMNet [50]	1.9	ZO	3.3
	Deep shells [14]	1.7	—	—
	NeuroMorph (Ours)	**1.5**	SL	**2.3**

Table 1: **Unsupervised correspondences on FAUST [2] remeshed.** Mean geodesic error in % of the diameter on the test set. For methods that use an axiomatic technique for refinement (PMF [57], ZO [35], ICP [39] or SL [13]), we also show the result without.

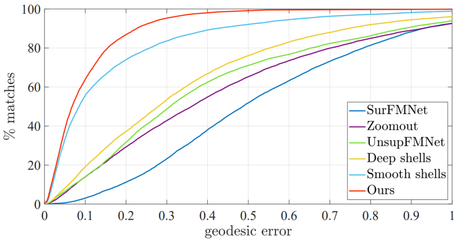


Figure 3: **Unsupervised correspondences on SHREC20 [10].** We only compare our method to other unsupervised methods here, since there are no dense ground-truth correspondences for this benchmark which is a requirement for most supervised approaches.

Geodesic distance preservation loss. The final component of our loss function in Equation (8) aims at preserving the pairwise geodesic distance matrices $\mathbf{D}_{\mathcal{X}}$ and $\mathbf{D}_{\mathcal{Y}}$ under the estimated mapping Π, this is given by $\ell_{\mathrm{geo}}\big(\Pi\big) := \|\Pi\mathbf{D}_{\mathcal{Y}}\Pi^{\top} - \mathbf{D}_{\mathcal{X}}\|_2^2$. Note that this energy only regularizes the estimation of the correspondences Π as $\mathcal{X}$ and $\mathcal{Y}$ are the (fixed) source and target shapes.

Intuitively, this objective promotes correspondences Π with bounded geodesic distortion. Variants of this objective are commonly used in classical shape matching [4, 59, 57] and have been also successfully integrated in a learning pipeline [20] in combination with functional maps [39].

3.4. Implementation details

During training, we sample a pair of input shapes from our training set, predict an interpolation and a set of dense point-to-point correspondences and optimize the model parameters according to our composite loss (8). All hyper-

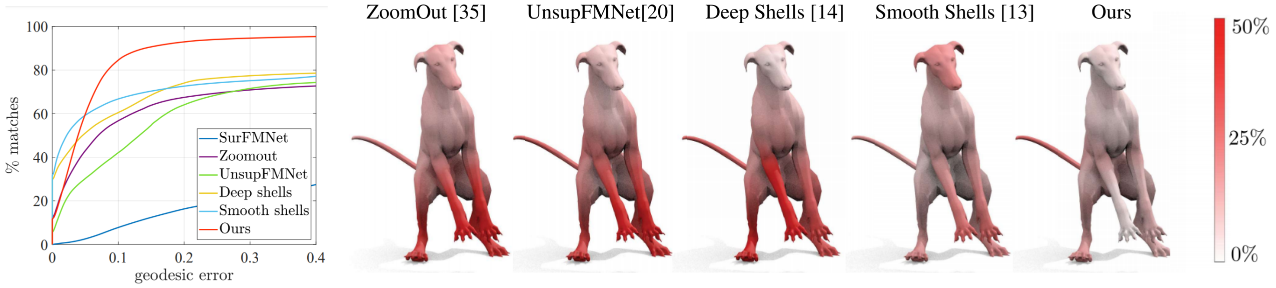


Figure 4: **Unsupervised correspondences on G-S-H.** We provide the cumulative geodesic error curves (in % of the diameter) of different approaches (left). For a detailed comparison, we display heat maps on one pose of the ’Galgo’ shape from our dataset (right). We color code the mean geodesic error for each point of the surface, averaged over all 1024 pairs from the test set. Our method is particularly good at discovering structural correspondences, i.e. matching extremities correctly.

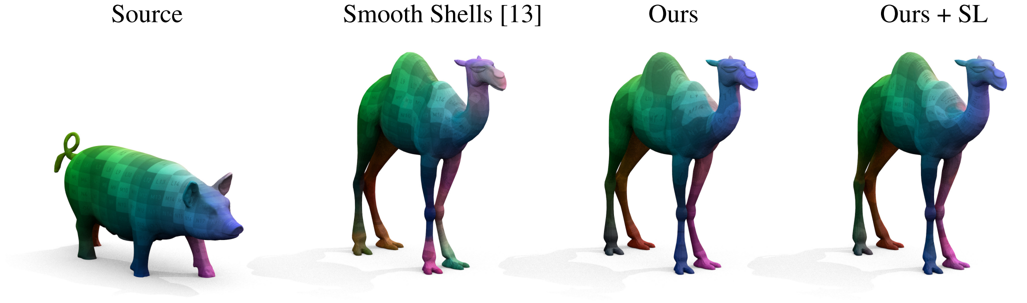


Figure 5: **Unsupervised correspondences on SHREC20.** A qualitative comparison on non-isometric pairs from SHREC20 [10]. While the correspondences predicted by our method are generally very accurate, the postprocessing still helps to remove local noise. The baseline [13] naturally produces smooth matches, but global parts of the geometry are sometimes mismatched (compare for instance the front legs and and head of the camel shown here).

parameters were selected on a validation set and the same configuration is used in all of our experiments.

Two parameters are varied during training: In the beginning, we set the number of discrete time steps to $T = 1$ and then increase it on a logarithmic scale. This multi-scale optimization strategy, which is motivated by classical non-learning interpolation algorithms [27, 23], leads to an overall faster and more robust convergence. The geodesic loss ℓ_{geo} initially helps to guide the optimization such that it converges to meaningful local minima. On the other hand, we found that it can actually be detrimental in the case of extremely non-isometric pairs (e.g. two different classes of animals). Therefore, we decay the weight $\lambda_{\text{geo}} = 0$ of this loss as a fine-tuning step during training after a fixed number of epochs.

As a form of data augmentation, we randomly subsample the triangulation of both input meshes separately and rotate the input pair along the azimuth axis in each iteration. This prevents our method from relying on pairs with compatible connectivity, since we ideally want our predictions to be independent from the discretization.

At test time, we simply query our model to obtain an interpolation of an input pair of shapes. The soft correspondences Π obtained with our method are generally very accurate, but the conversion to hard correspondences (i.e. point-to-point matches) via thresholding leads to a certain degree of local noise. To create more smooth correspondences, we additionally post-process our results with the multi-scale matching method smooth shells [13]. Post-processing is standard in unsupervised correspondence learning.

4. Experiments

We now evaluate the performance of NeuroMorph in terms of shape correspondence and interpolation (Sec. 4.1, 4.2), as well as for data augmentation in Sec. 4.3.

4.1. Shape correspondence

Datasets. We evaluate the matching accuracy of our method on two benchmarks. The first is FAUST [2], which contains 10 humans with 10 different poses each. We split it in a training and test set of 80 and 20 shapes respectively. Instead of the standard meshes, we use the more recent version of the benchmark [44] where each shape was re-meshed individually. This makes it challenging but also more realistic, since for real-world scans the sampling

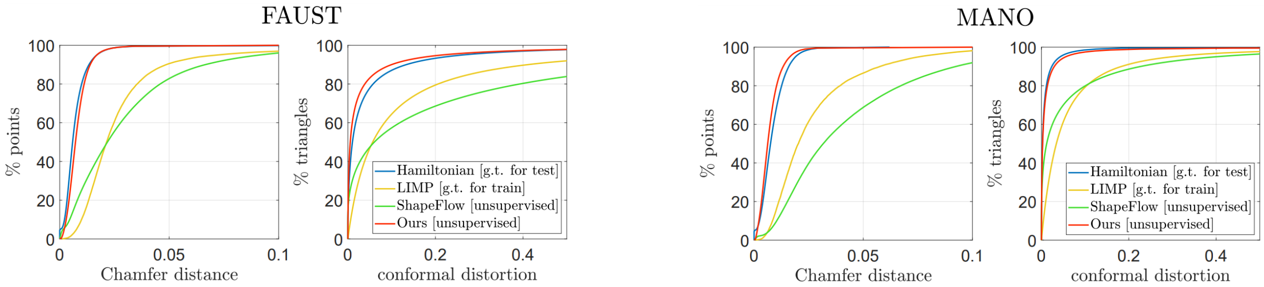


Figure 6: **Interpolation on FAUST [2] and MANO [46].** We show a quantitative comparison of interpolations obtained with Hamiltonian shape interpolation [11], LIMP [7], ShapeFlow [26] and our method. ShapeFlow [26] computes an extrinsic flow to interpolate a pair of objects in an unsupervised manner, but they do not model shape correspondences explicitly which is suboptimal for the large pose variations of deformable object categories. On both benchmarks, our method outperforms LIMP [7], despite the fact that the it uses g.t. correspondences for training. It is also on par with the axiomatic baseline Hamiltonian interpolation [11], which is remarkable since [11] requires dense correspondences even at test time.

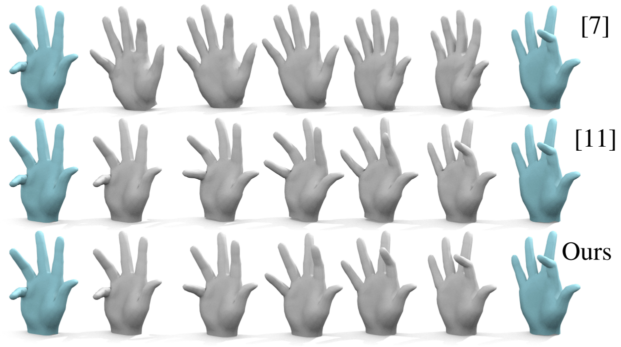


Figure 7: **Interpolation on MANO [46].** We show the interpolation sequence (gray) for a pair (blue) from the test set. LIMP [7], which requires ground-truth correspondences for training, explicitly reconstructs the geometry of intermediate shapes in a variational autoencoder architecture which limits the generalization to unseen poses. Hamiltonian shape interpolation [11] yields high quality results that are comparable to ours, but it is an axiomatic method that requires ground-truth correspondences at test time and multiple minutes of optimization per pair.

of surfaces is generally incompatible. The second benchmark we consider is the recent SHREC20 challenge [10] which focuses on non-isometric deformations. It contains 14 shapes of different animals, some of which are real scans with holes, topological changes and partial geometries. The ground truth for this dataset consists of sparse annotated keypoints which we use for evaluation. Since there are no dense annotated point-to-point correspondences, most existing supervised methods do not apply here. The final benchmark we show is G-S-H (Galgo, Sphynx, Human), for which we created our own dataset, see **??** for more details. It contains non-isometric pairs from three object categories (a dog 'Galgo', a cat 'Sphynx' and a human) with multiple challenging poses each, as well as dense ground truth matches.

Evaluation metrics. Following the Princeton benchmark protocol [28], the accuracy of a set of point-to-point correspondences is defined as the geodesic distance of the predicted and the ground-truth matches, normalized by the square root area of the mesh. For FAUST remeshed, we compute the distance for all points, whereas for SHREC20 this is done for all available sparse annotations.

Discussion. As shown in Table 1, Figure 3 and Figure 4, NeuroMorph obtains state-of-the-art results on FAUST remeshed, SHREC20 and our own benchmark G-S-H, respectively. The overall suboptimal performance of existing methods on the latter two benchmarks can be attributed to the fact that, expect for [13], most of them implicitly assume near-isometry or at least compatible local features. This, however, does not hold for most examples in SHREC20 (see Figure 5 for a qualitative comparison) and G-S-H (for instance, on SHREC20, NeuroMorph matches 92% of the vertices within 0.25 geodesic error vs. 79% of the second best, smooth shells). NeuroMorph is particularly good at discovering structural correspondences, which can then be further refined in post-processing.

4.2. Shape interpolation

Datasets. For shape interpolation, we report results on the FAUST [2] (see Section 4.1) and MANO [46] datasets. The latter consists of synthetic hands in various poses — we use 100 shapes for training and 20 different samples for testing.

Evaluation metrics. We use two metrics to quantify the precision of an interpolation. The conformal distortion metric signifies how much individual triangles of a mesh distort throughout an interpolation sequence, in comparison to the reference pose $\mathcal{X}$, see [25, Eq. (3)] for a definition. Less

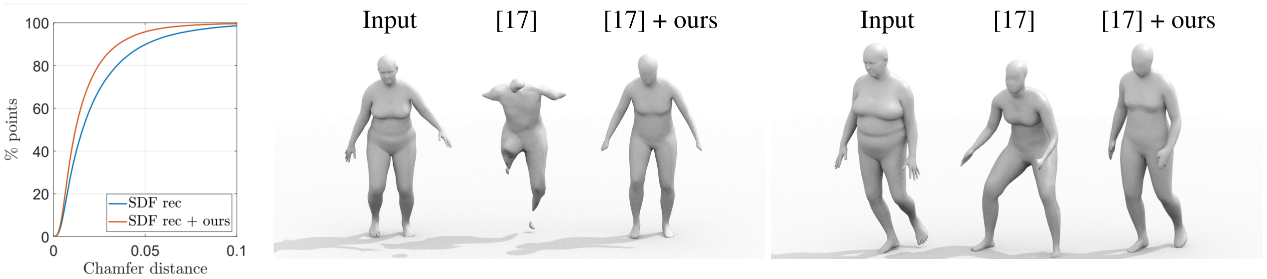


Figure 8: **Data augmentation for implicit surface reconstruction.** We show, as a proof of concept, that our method can be used to augment a small training set with additional poses. We verify this by comparing the reconstruction error of an implicit surface reconstruction method [17] with and without data augmentation. The general idea behind this is to construct a shape space that encodes arbitrary poses in a latent representation. As we show here, supplementing collections of sparse observations with intermediate poses constitutes a natural extension which helps to learn a meaningful shape distribution.

distortion corresponds to more realistic shapes. The other metric we consider measures the reconstruction error of the target shape $\mathcal{Y}$, defined as the Chamfer distance between $\mathcal{Y}$ and the deformed shape $\mathcal{X}(1)$. A good overlap at $t = 1$ is an important quality criterion because, while our interpolations exactly coincide with the first shape $\mathcal{X} = \mathcal{X}(0)$, they only approximately align with $\mathcal{Y} \approx \mathcal{X}(1)$. The same holds true for the three baselines [7, 26, 11] that we compare against.

Discussion. Results are shown in Figure 6. On both of these benchmarks, our method significantly outperforms the supervised baseline LIMP [7] which requires ground-truth correspondences for training. Similar to our approach, the unsupervised method ShapeFlow [26] continuously deforms a given input shape to obtain an interpolation. However, they do not estimate correspondences explicitly which limits the performance on deformable object categories like humans, animals or hands[3]. More surprisingly, our approach is even on par with the axiomatic, non-learning interpolation baseline [11] which requires to know dense correspondences at *test* time. See Figure 7 for a qualitative comparison on MANO.

4.3. Application: data augmentation

Our method is, to the best of our knowledge, the first one that jointly predicts correspondences and an interpolation of deformable objects in a single learning framework. As an application of unsupervised interpolation, we show how our method can be used to create additional training samples as a form of data augmentation. To that end, we train an implicit surface reconstruction method [17] on a small set of 20 SMPL shapes from the SURREAL dataset [56] and evaluate the obtained reconstructions on a separate test set of 100 shapes. Additionally, we use our method to create 3 additional, interpolated training poses for each pair in the training set and compare the results with the vanilla training. To measure the quality of the obtained reconstructions, we report the reconstruction error on the test set, defined as the Chamfer distance of the test shapes to the reconstructed surface, see Figure 8.

Overall, these results indicate that using our method to enlarge a training set of 3D shapes can be useful for downstream tasks, especially when training data is limited.

5. Conclusions

We presented a new framework for 3D shape understanding that simultaneously addresses the problems of shape correspondence and interpolation. The key insight we want to advocate is that these two goals mutually reinforce each other: Better correspondences yield more accurate interpolations and, vice versa, meaningful deformations of 3D surfaces act as a strong geometric prior for finding correspondences. In comparison to related existing approaches, our model can be trained in a fully unsupervised manner and generates correspondence and interpolation in a single pass. We show that our method produces stable results for a variety of correspondence and interpolation tasks, including challenging inter-class pairs with high degrees of non-isometric deformations. We expect that NeuroMorph will facilitate 3D shape analysis on large real-world datasets where obtaining exact ground-truth matches is prohibitively expensive.

Acknowledgements

We would like to thank Matan Atzmon and Aysim Toker for useful discussions and Roberto Dyke for quick help with the SHREC20 ground truth. This work was supported by the Munich Center for Machine Learning and by the ERC Advanced Grant SIMULACRON.

[3] These types of objects typically have a high pose variation with large degrees of non-rigid deformations, whereas ShapeFlow mainly specializes on man-made objects like chairs or cars. The few deformable examples that they show in [26, fig. 5] are intended as a proof of concept since they use ground-truth correspondences and overfit on a single pair of shapes.

References

[1] Yonathan Aflalo, Anastasia Dubrovina, and Ron Kimmel. Spectral generalized multi-dimensional scaling. *IJCV*, 118(3):380–392, 2016.

[2] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.

[3] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in neural information processing systems*, pages 3189–3197, 2016.

[4] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006.

[5] 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.

[6] 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.

[7] Luca Cosmo, Antonio Norelli, Oshri Halimi, Ron Kimmel, and Emanuele Rodolà. Limp: Learning latent shape representations with metric preservation priors. *arXiv preprint arXiv:2003.12283*, 2020.

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[9] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. *arXiv preprint arXiv:2003.14286*, 2020.

[10] 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.

[11] Marvin Eisenberger and Daniel Cremers. Hamiltonian dynamics for real-world shape interpolation. In *ECCV*, 2020.

[12] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38, pages 1–12. Wiley Online Library, 2019.

[13] 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.

[14] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixe, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *arXiv preprint*, 2020.

[15] 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.

[16] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019.

[17] 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.

[18] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mache approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.

[20] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[22] Behrend Heeren, Martin Rumpf, Peter Schröder, Max Wardetzky, and Benedikt Wirth. Splines in the space of shells. *Computer Graphics Forum*, 35(5):111–120, 2016.

[23] Behrend Heeren, Martin Rumpf, Max Wardetzky, and Benedikt Wirth. Time-discrete geodesics in the space of shells. In *Computer Graphics Forum*, volume 31, pages 1755–1764. Wiley Online Library, 2012.

[24] Behrend Heeren, Chao Zhang, Martin Rumpf, and William Smith. Principal geodesic analysis in the space of discrete shells. In *Computer Graphics Forum*, volume 37, pages 173–184. Wiley Online Library, 2018.

[25] Kai Hormann and Günther Greiner. Mips: An efficient global parametrization method. Technical report, Erlangen-Nuernberg University (Germany) Computer Graphics Group, 2000.

[26] Chiyu Jiang, Jingwei Huang, Andrea Tagliasacchi, Leonidas Guibas, et al. Shapeflow: Learnable deformations among 3d shapes. *arXiv preprint arXiv:2006.07982*, 2020.

[27] Martin Kilian, Niloy J Mitra, and Helmut Pottmann. Geometric modeling in shape space. In *ACM Transactions on Graphics (TOG)*, volume 26, page 64. ACM, 2007.

[28] Vladimir G Kim, Yaron Lipman, and Thomas A Funkhouser. Blended intrinsic maps. *Transactions on Graphics (TOG)*, 30(4), 2011.

[29] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[30] 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.

[31] Or Litany, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2):1681–1707, 2017.

[32] Or Litany, Emanuele Rodolà, Alex M Bronstein, Michael M Bronstein, and Daniel Cremers. Non-rigid puzzles. *Computer Graphics Forum (CGF), Proceedings of Symposium on Geometry Processing (SGP)*, 35(5), 2016.

[33] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. on Graphics (TOG)*, 2015.

[34] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015.

[35] 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.

[36] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019.

[37] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[38] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5379–5389, 2019.

[39] 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, 2012.

[40] 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 Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[41] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018.

[42] 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.

[43] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[44] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6):248:1–248:16, Dec. 2018.

[45] Emanuele Rodolà, Luca Cosmo, Michael Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum (CGF)*, 2016.

[46] 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), Nov. 2017.

[47] 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.

[48] Klaus Hildebrandt Ruben Wiersma, Elmar Eisemann. Cnns on surfaces using rotation-equivariant features. *Transactions on Graphics*, 39(4), July 2020.

[49] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, pages 1–17, 2019.

[50] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional map for shape matching. *arXiv preprint arXiv:2009.13339*, 2020.

[51] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007.

[52] Gary K. L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, July 2013.

[53] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019.

[54] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. *In Proceedings of European Conference on Computer Vision (ECCV)*, 16(9):356–369, 2010.

[55] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[56] Gul 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.

[57] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex M Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[58] Philipp von Radziewsky, Elmar Eisemann, Hans-Peter Seidel, and Klaus Hildebrandt. Optimized subspaces for deformation-based modeling and shape interpolation. *Computers & Graphics*, 58:128–138, 2016.

[59] Chaohui Wang, Michael M Bronstein, Alexander M Bronstein, and Nikos Paragios. Discrete minimum distortion correspondence problems for non-rigid shape matching. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 580–591. Springer, 2011.

[60] Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. 3dn: 3d deformation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1038–1046, 2019.

[61] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[62] Benedikt Wirth, Leah Bar, Martin Rumpf, and Guillermo Sapiro. A continuum mechanical approach to geodesics in shape space. *International Journal of Computer Vision*, 93(3):293–318, Jul 2011.

[63] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[64] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018.

[65] Chao Zhang, Behrend Heeren, Martin Rumpf, and William AP Smith. Shell pca: Statistical shape modelling in shell space. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1671–1679, 2015.

[66] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. *Proceedings of European Conference on Computer Vision (ECCV)*, 2016.

Chapter 9

G-MSM: Unsupervised Multi-Shape Matching with Graph-based Affinity Priors

The predominant paradigm of learning methods for shape correspondence, in the literature, is based on extracting local feature maps for pairs of input poses, which are then processed by a differentiable matching layer. On the other hand, in many real-world applications, we have a whole collection of query poses available. For most datasets, individual poses are correlated and the collection as a whole can provide relevant cues for pairwise matching. For instance, if there are several poses subject to topological merging from self-contact, then statistics from all input poses help convey the latent topology, *i.e.*, help distinguish between geometrically meaningful shape features and noise. Hence, we propose a learning approach for multi-shape matching, that jointly predicts correspondences between a set of $N > 2$ input poses. To this end, we introduce the notion of undirected, edge-weighted shape graphs $\mathcal{G}$. The weight between two poses encodes an affinity score, expressing the degree of geometric similarity. Since our approach is fully unsupervised, *i.e.*, does not depend on ground truth correspondences as inputs, we derive this similarity score from a pairwise correspondence heuristic. Specifically, this score is defined as the matching loss of the current pairwise correspondence estimate. Based on the shape graph $\mathcal{G}$, we then define a multi-shape correspondence map by concatenating pairwise maps along shortest paths in $\mathcal{G}$, and enforcing cycle-consistency with the pairwise map. Our overall network architecture consists of three modules, responsible for local feature extraction, pairwise matching, and multi-matching, respectively. We demonstrate, on a broad range of shape datasets, that the multi-shape correspondences are superior to pairwise approaches and multi-matching methods based on canonical embeddings. The improvement over pairwise baselines is especially significant for datasets with topological noise, and inter-class correspondences. We further compare our approach to several alternative, sparse graph heuristics, such as minimal spanning trees, traveling salesman graphs, and star graphs.

Individual contributions

Leading role in realizing the scientific project.

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

Copyright

Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers
G-MSM: Unsupervised Multi-Shape Matching with Graph-based Affinity Priors
2023 Conference on Computer Vision and Pattern Recognition (CVPR)
DOI: 10.1109/CVPR52729.2023.02180

We include the accepted version of the original publication [6].

RightsLink

Sign in/Register

G-MSM: Unsupervised Multi-Shape Matching with Graph-Based Affinity Priors

Requesting permission to reuse content from an IEEE publication

Conference Proceedings:
2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)

Author: [::Marvin::] [::Eisenberger::]; Aysim Toker; Laura Leal-Taixè; Daniel Cremers

Publisher: IEEE

Date: 17-24 June 2023

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

G-MSM: Unsupervised Multi-Shape Matching with Graph-based Affinity Priors

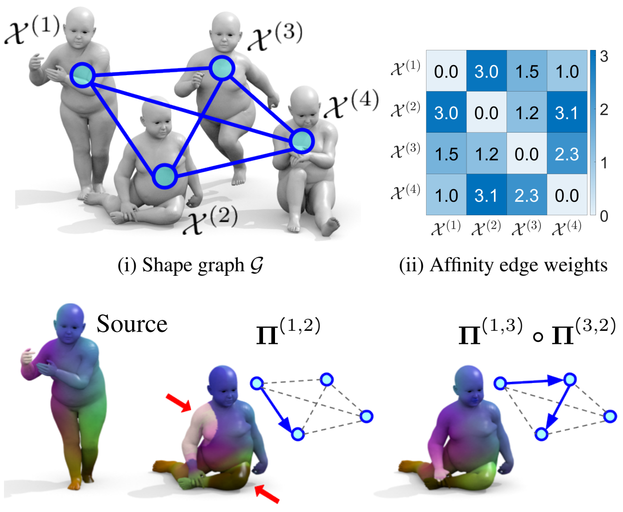
Marvin Eisenberger[1], Aysim Toker[1], Laura Leal-Taixé[1,†], Daniel Cremers[1,2,3]
[1]TU Munich, [2]Munich Center for Machine Learning, [3]University of Oxford

Abstract

*We present G-MSM (**G**raph-based **M**ulti-**S**hape **M**atching), a novel unsupervised learning approach for non-rigid shape correspondence. Rather than treating a collection of input poses as an unordered set of samples, we explicitly model the underlying shape data manifold. To this end, we propose an adaptive multi-shape matching architecture that constructs an affinity graph on a given set of training shapes in a self-supervised manner. The key idea is to combine putative, pairwise correspondences by propagating maps along shortest paths in the underlying shape graph. During training, we enforce cycle-consistency between such optimal paths and the pairwise matches which enables our model to learn topology-aware shape priors. We explore different classes of shape graphs and recover specific settings, like template-based matching (star graph) or learnable ranking/sorting (TSP graph), as special cases in our framework. Finally, we demonstrate state-of-the-art performance on several recent shape correspondence benchmarks, including real-world 3D scan meshes with topological noise and challenging inter-class pairs.*[1]

1. Introduction

Shape matching of non-rigid object categories is a central problem in 3D computer vision and graphics that has been studied extensively over the last few years. Especially in recent times, there is a growing demand for such algorithms as 3D reconstruction techniques and affordable scanning devices become increasingly powerful and broadly available. Classical shape correspondence approaches devise axiomatic algorithms that make specific assumptions about the resulting maps, such as near-isometry, area preservation, approximate rigidity, bounded distortion, or commutativity with the intrinsic Laplacian. In contrast, real-world scan meshes are often subject to various types of noise, including topological changes [16, 33], partial views [2], general non-isometric deformations [17, 65], objects in clutter [12], and varying data representations [57]. In this work, we address several of the aforementioned challenges and demonstrate that our proposed method achieves improved stability for a number of 3D scan mesh datasets.


$\mathcal{X}^{(1)}$
$\mathcal{X}^{(3)}$
$\mathcal{X}^{(4)}$
$\mathcal{X}^{(2)}$
$\mathcal{X}^{(1)}$ 0.0 3.0 1.5 1.0
$\mathcal{X}^{(2)}$ 3.0 0.0 1.2 3.1
$\mathcal{X}^{(3)}$ 1.5 1.2 0.0 2.3
$\mathcal{X}^{(4)}$ 1.0 3.1 2.3 0.0
$\mathcal{X}^{(1)}$ $\mathcal{X}^{(2)}$ $\mathcal{X}^{(3)}$ $\mathcal{X}^{(4)}$
3
2
1
0
Source
$\mathbf{\Pi}^{(1,2)}$
$\mathbf{\Pi}^{(1,3)} \circ \mathbf{\Pi}^{(3,2)}$

(i) Shape graph $\mathcal{G}$

(ii) Affinity edge weights

(iii) Putative correspondences $\mathbf{\Pi}^{(1,2)}$ vs multi-matching $\mathbf{\Pi}^{(1,3)} \circ \mathbf{\Pi}^{(3,2)}$

Figure 1. For a given collection of 3D meshes $\{\mathcal{X}^{(i)} | 1 \leq i \leq N\}$, (i) our method constructs, in a fully unsupervised manner, a shape graph $\mathcal{G}$ which approximates the underlying shape data manifold. (ii) Its edge weights (affinity scores) are derived from a putative pairwise correspondence loss signal. (iii) During training, we enforce cycle-consistency by propagating maps along shortest paths in the graph $\mathcal{G}$. As shown for the sample pair above $(\mathcal{X}^{(1)}, \mathcal{X}^{(2)})$, the resulting multi-matching $\mathbf{\Pi}^{(1,3)} \circ \mathbf{\Pi}^{(3,2)}$ is significantly more accurate than the pairwise map $\mathbf{\Pi}^{(1,2)}$.

The majority of existing deep learning methods for shape matching [2, 15, 20, 21, 24, 39, 51, 56] treat a given set of meshes as an unstructured collection of poses. During training, random pairs of shapes are sampled for which a neural network is queried and a pairwise matching loss is minimized. While this approach is straightforward, it often fails to recognize commonalities and context-dependent patterns

[†] Currently at NVIDIA

[1] Our implementation is available under the following link: https://github.com/marvin-eisenberger/gmsm-matching

which only emerge from analyzing the shape collection as a whole. *Not all samples of a shape collection are created equal.* In most cases, some pairs of poses are much closer than others. Maps between similar geometries are inherently correlated and convey relevant clues to one another. This is particularly relevant for challenging real-world scenarios, where such redundancies can help disambiguate noisy geometries, non-isometric deformations, and topological changes. The most common approach of existing multi-matching methods is to learn a canonical embedding per pose, either in the spatial [10] or Laplace-Beltrami frequency domain [26,28]. This incentivizes the resulting matches to be consistent under concatenation. However, such approaches are in practice still trained in a fully pairwise manner for ease of training. Furthermore, relying on canonical embeddings can lead to limited generalization for unseen test poses. Concrete approaches often assume a specific mesh resolution and nearly-isometric poses [10], or require an additional fine-tuning optimization at test time [26, Sec. 5].

Rather than interpreting a given training set as a random, unstructured collection of shapes, our approach explicitly models the underlying shape manifold. To this end, we define an affinity graph $\mathcal{G}$ on the set of input shapes whose edge weights (i.e. affinity scores) are informed by the outputs of a pairwise matching module. We then devise a novel adaptive multi-matching architecture that propagates matches along shortest paths in the underlying shape graph $\mathcal{G}$. The resulting maps are topology-aware, i.e., informed by geometries from the whole shape collection. An example is shown in Figure 1, where the multi-matching $\mathbf{\Pi}^{(1,3)} \circ \mathbf{\Pi}^{(3,2)}$ obtained by our approach is significantly more accurate than the naive, pairwise map $\mathbf{\Pi}^{(1,2)}$. During training, we promote cycle-consistency of shortest paths in the shape graph. In summary, our contributions are as follows:

1. Introduce the notion of an edge-weighted, undirected shape graph $\mathcal{G}$ to approximate the underlying data manifold for an unordered collection of 3D meshes.

2. Propose a novel, adaptive multi-shape matching approach that enforces cycle-consistency for optimal paths in the shape graph $\mathcal{G}$ in a self-supervised manner.

3. Demonstrate state-of-the-art performance for a range of challenging non-rigid matching tasks, including non-isometric matching due to topological noise [16, 33] and inter-class pairs [17,65].

2. Related work

Axiomatic correspondence methods Shape matching is an extensively studied topic with a variety of different approaches and methodologies. We summarize references relevant to our approach here and refer to recent surveys [54,60] for a more complete picture. Classical methods for non-rigid matching often devise optimization-based approaches that minimize some type of distortion metric [7,18,49,62,64]. A common prerequisite is the extraction of hand-crafted local descriptors that are approximately preserved under non-rigid shape deformations. Common definitions include histogram-based statistics [59] or fully intrinsic features based on the eigenfunctions of the Laplace-Beltrami operator [3,53,58]. Over the last few years, functional maps [44] have become a central paradigm in shape matching. The core idea is to reframe the pairwise matching task from functions (points to points) to functionals (functions to functions). There are several extensions of the original framework to allow for partial matching [35,36,50], orientation preservation [48], iterative map upsampling [19, 42] and conformal maps [14]. Our approach utilizes functional maps as a fundamental building block within the differentiable matching layer.

Learning-based methods More recently, several approaches emerged that aim at extending the power of deep feature learning to deformable 3D shapes. Many such methods fall under the umbrella term ‘geometric deep learning’ [9], with analogous applications on different classes of non-Euclidean data like graphs or general manifold data. One class of approaches are charting-based methods [6,40,43,46,52] which imitate convolutions in Euclidean space with parameterized, intrinsic patch operators. Likewise, [57] proposed a learnable feature refinement module based on intrinsic heat diffusion.

The pioneering work of [34] proposes a differentiable matching layer based on functional maps [44], in combination with a deep feature extractor with several consecutive ResNet layers [25]. Numerous extensions of this paradigm were proposed over the last few years to allow for unsupervised loss functions [24,51], learnable basis functions [39], point cloud feature extractors [15,26,56] or partial data [2]. Similarly, DeepShells [21] learns functional maps in an end-to-end trainable, hierarchical multi-scale pipeline. We adapt parts of its differentiable matching layer in our network. Other approaches [4, 23, 38] learn correspondences for a specific class of deformable objects by including additional domain knowledge, like a deformable human model [37]. Finally, [20] jointly learns to predict correspondences and a smooth interpolation between pairs of shapes.

Multi-shape matching Classical axiomatic multi-shape matching approaches devise optimization-based pipelines that enforce cycle-consistent maps. Specific solutions include semidefinite programming [27], convex relaxations of the corresponding quadratic assignment problem [30], graph cuts [55], as well as evolutionary game theory [11]. Such optimization-based approaches are computationally costly and therefore limited to matching sparse landmarks. Furthermore, there are a number of optimization frameworks that compute synchronized, cycle-consistent functional maps [22, 28, 29]. Notably, such approaches are often limited to nearly-isometric poses [22, 29] or require high-quality initializations [28]. More recent learning-based approaches promote cycle-consistency by predicting a canonical embedding for each observed pose [10, 26]. However, obtaining stable embeddings is often difficult when generalizing to unseen test poses. Moreover, such approaches assume a specific mesh resolution and nearly-isometric poses [10] or require an additional fine-tuning optimization at test time to obtain canonical embeddings [26, Sec. 5].

3. Method

3.1. Problem formulation

In the following, we consider a collection of 3D shapes $\mathcal{S} = \{\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(N)}\}$ from non-rigidly deformable shape categories. Each such shape $\mathcal{X}^{(i)}$ is a discretized approximation of a 2D Riemannian manifold, embedded in $\mathbb{R}^3$. Specifically, we define $\mathcal{X}^{(i)} = (\mathbf{V}^{(i)}, \mathbf{T}^{(i)})$, where $\mathbf{V}^{(i)} \in \mathbb{R}^{m\times 3}$ and $\mathbf{T}^{(i)} \subset \mathbf{V}^{(i)} \times \mathbf{V}^{(i)} \times \mathbf{V}^{(i)}$ are sets of vertices and triangular faces, respectively. The goal is then to construct an algorithm that computes dense correspondence mappings $\mathbf{\Pi}^{(i,j)}$ between any two surfaces $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$ from the shape collection $\mathcal{S}$. Specifically, such correspondences are represented by sparse assignment matrices $\mathbf{\Pi}^{(i,j)} \in \{0,1\}^{m\times n}$, where $\mathbf{\Pi}^{(i,j)}\mathbf{1}_n = \mathbf{1}_m$ and $\mathbf{\Pi}^{(i,j)}_{i',j'} = 1$ indicates a match between the i'-th vertex of $\mathcal{X}^{(i)}$ and the j'-th vertex of $\mathcal{X}^{(j)}$.

Scope Our method is unsupervised and thereby requires no additional inputs, like landmark annotations or ground-truth correspondences, beyond the raw input geometries $\mathcal{X}^{(i)}$. Following similar approaches in this line of work [4, 20, 39, 56], we assume that the shapes $\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(N)}$ have an approximately canonical orientation. In the literature, this setting is commonly referred to as 'weakly supervised', see [56] and later [20]. Existing approaches often make additional assumptions about the input data $\mathcal{S}$, focusing on nearly-isometric correspondences [34, 51], maps with bounded distortion [20, 21] or partial views of the same non-rigid object [2, 50]. Others specialize in distinct classes of shapes like deformable human bodies [4, 23, 38]. In contrast, we demonstrate in our experiments that our proposed multi-matching approach excels at a broad range of challenging settings, including non-isometric pairs, poses with topological noise from self-intersections, and inter-class matching.

3.2. Network architecture

We now define the neural network architecture that forms the basis of our proposed approach. It consists of three separate components **I-III**, see also Figure 2 for an overview. The first two modules are standard components found in most learning-based shape matching approaches [2, 15, 20, 21, 24, 39, 51, 56], namely a learnable feature backbone **I** and a differentiable, pairwise matching layer **II**. We briefly outline these here and provide additional details in Appendix A.2. The multi-matching architecture **III** is introduced in Section 3.3.

Feature extractor The first component **I** of our model is a standard, learnable feature extraction backbone for representation learning, defined as

$$\Phi_{\text{feat}} : \mathcal{X}^{(i)} \mapsto \mathbf{F}^{(i)} \in \mathbb{R}^{m\times l}. \tag{1}$$

For a given input shape $\mathcal{X}^{(i)} = (\mathbf{V}^{(i)}, \mathbf{T}^{(i)})$, the mapping Φ_{feat} produces an l-dimensional feature embedding $\mathbf{F}^{(i)}$ per vertex $\mathbf{V}^{(i)} \in \mathbb{R}^{m\times 3}$. While other choices are possible, we base Φ_{feat} on the off-the-shelf feature backbone DiffusionNet [57]. This network refines features via intrinsic heat-diffusion operators. Such operators are agnostic to the input discretization, thereby extremely robust to varying mesh resolutions and sampling densities. At the same time, it is computationally lightweight. For more details on the choice of backbone, see Appendix A.2.

Pairwise matching The second component **II** of our network is a differentiable, multi-scale matching scheme based on the recent pairwise shape matching method DeepShells [21]. The basis of this approach is the energy function

$$E_{\text{match}}(\mathbf{F}, \mathbf{G}; \tilde{\mathbf{\Pi}}) := \sum_{i'=1}^{m} \sum_{j'=1}^{n} \tilde{\mathbf{\Pi}}_{i',j'} \big\|\mathbf{F}_{i'} - \mathbf{G}_{j'}\big\|_2^2 \tag{2}$$

which has its roots in the theory of optimal transport. For a given transport plan $\tilde{\mathbf{\Pi}} \in [0,1]^{m\times n}$, the energy E_{match} specifies the distance between the discrete measures associated with two arbitrary l-dimensional feature embeddings $\mathbf{F} = (\mathbf{F}_1, \dots, \mathbf{F}_m) \in \mathbb{R}^{m\times l}$ and $\mathbf{G} = (\mathbf{G}_1, \dots, \mathbf{G}_n) \in \mathbb{R}^{n\times l}$. Taking the minimum over all possible transport plans $\arg\min_{\tilde{\mathbf{\Pi}}} E_{\text{match}}(\mathbf{F}, \mathbf{G}; \tilde{\mathbf{\Pi}})$ results in the Kantorovich formulation of optimal transport [45, 63]. Following the approach described in [21], we obtain a multi-scale shape matching scheme that minimizes Equation (2) in an iterative optimization. For a given pair of shapes $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$, this scheme defines a mapping

$$\Phi_{\text{match}} : \big(\mathbf{F}^{(i)}, \mathbf{F}^{(j)}\big) \mapsto \big(\mathbf{\Pi}^{(i,j)}, \mathbf{V}^{(i,j)}, \ell^{(i,j)}_{\text{match}}\big). \tag{3}$$

We provide further details on the exact update steps of the optimization scheme Φ_{match} in Appendix A.2. From a high-level perspective, Φ_{match} is defined as a deterministic, differentiable function that takes local feature encodings

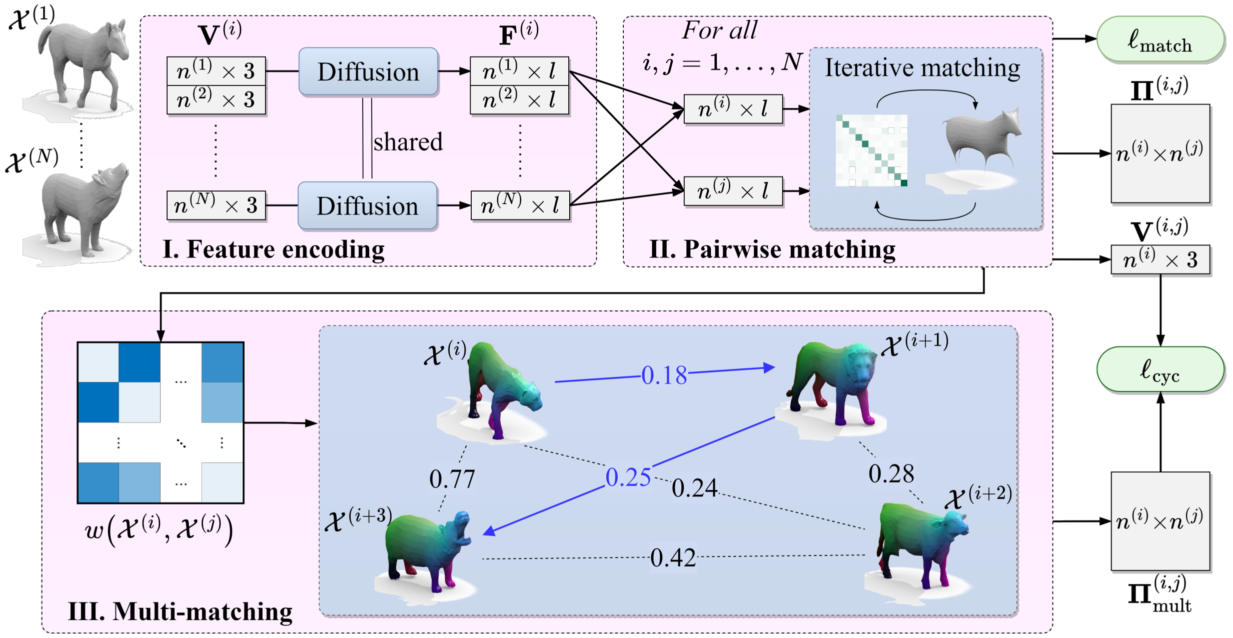


Figure 2. **Pipeline overview.** For a collection of shapes $\mathcal{S} = \{\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(N)}\}$, **I.** feature embeddings are extracted with DiffusionNet [57] and **II.** pairwise correspondences $\mathbf{\Pi}^{(i,j)}$ are predicted via an iterative, differentiable matching layer [21]. **III.** The pairwise matches are utilized to construct a shape graph $\mathcal{G} = (\mathcal{S}, w)$ with affinity edge weights $w(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) \geq 0$. During training, we minimize the pairwise matching loss $\ell^{(i,j)}_{\text{match}}$, as well as the cycle consistency loss $\ell^{(i,j)}_{\text{cyc}}$ between the pairwise registrations $\mathbf{V}^{(i,j)}$ and multi-matches $\mathbf{\Pi}^{(i,j)}_{\text{mult}}$.

$\mathbf{F}^{(i)} \in \mathbb{R}^{m \times l}$ and $\mathbf{F}^{(j)} \in \mathbb{R}^{n \times l}$ as input and predicts a set of correspondences $\mathbf{\Pi}^{(i,j)} \in \{0,1\}^{m \times n}$. Additionally, Φ_{match} outputs a deformed embedding $\mathbf{V}^{(i,j)} \in \mathbb{R}^{m \times 3}$ of the vertices of $\mathcal{X}^{(i)}$. These coordinates specify a registered version of the first input shape $\mathcal{X}^{(i)}$ that closely aligns with the pose of the second input shape $\mathcal{X}^{(j)}$. The third output $\ell^{(i,j)}_{\text{match}} > 0$ is a training loss signal.

3.3. Graph-based multi-shape matching

Shape graph We now provide details on our multi-shape matching architecture **III**. To this end, we start by defining an affinity graph

$$\mathcal{G} := (\mathcal{S}, w), \quad \text{with} \quad w : \mathcal{S} \times \mathcal{S} \to [0, \infty] \qquad (4)$$

on the set of training shapes $\mathcal{S} = \{\mathcal{X}^{(1)}, \dots, \mathcal{X}^{(N)}\}$, see **III** in Figure 2 for a visualization. W.l.o.g., we construct $\mathcal{G}$ as a complete graph (i.e. undirected, fully connected), where a missing edge between $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$ can be specified equivalently by setting the corresponding edge weight to $w(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) = \infty$.

We define the pairwise edge weights $w(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) \in [0, \infty]$ such that they represent affinity scores between pairs of shapes $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$. By convention, small values $w(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) \approx 0$ reflect that $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$ have a comparably similar geometric structure. Since our method is fully unsupervised, we have no a priori knowledge of such affinities and thereby have to infer them directly from the geometries $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$. To this effect, we propose a simple heuristic for a given pair of shapes $\mathcal{X}^{(i)} = (\mathbf{V}^{(i)}, \mathbf{T}^{(i)})$ and $\mathcal{X}^{(j)} = (\mathbf{V}^{(j)}, \mathbf{T}^{(j)})$ and define the (symmetric) affinity score w as

$$w(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}) := \min\Big\{E_{\text{match}}\big(\mathbf{V}^{(i,j)}, \mathbf{V}^{(j)}; \mathbf{\Pi}^{(i,j)}\big), \\ E_{\text{match}}\big(\mathbf{V}^{(j,i)}, \mathbf{V}^{(i)}; \mathbf{\Pi}^{(j,i)}\big)\Big\}. \qquad (5)$$

In this context, E_{match} is the (self-supervised) matching energy defined in Equation (2), while $\mathbf{\Pi}^{(i,j)}$ and $\mathbf{V}^{(i,j)}$ are the putative correspondences and registrations produced by Equation (3), respectively. The intuition behind this choice of edge weights w is that a small matching energy E_{match} implies a high correspondence accuracy, which is in turn indicative of a high geometric similarity between the input poses $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$.

Multi-matching Since we define the edge weights w according to the self-supervised matching score E_{match}, small weights $w(\mathcal{X}^{(i)}, \mathcal{X}^{(j)})$ generally correlate with a high correspondence accuracy of $\mathbf{\Pi}^{(i,j)}$. Based on this assumption, we obtain multi-shape matches from the putative correspondences $\mathbf{\Pi}^{(i,j)}$ via the following expression

$$\big(i, s_1, \dots, s_{M-1}, j\big) \;:=\; \mathrm{Dijkstra}\big(\mathcal{X}^{(i)}, \mathcal{X}^{(j)}; \mathcal{G}\big) \tag{6a}$$

$$\mathbf{\Pi}^{(i,j)}_{\mathrm{mult}} := \mathbf{\Pi}^{(i,s_1)} \circ \mathbf{\Pi}^{(s_1,s_2)} \circ \dots \circ \mathbf{\Pi}^{(s_{M-1},j)}. \tag{6b}$$

Rather than matching a pair of shapes $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$ directly, the multi-shape correspondence maps are passed along shortest paths $\mathcal{X}^{(i)}, \mathcal{X}^{(s_1)}, \dots, \mathcal{X}^{(s_{M-1})}, \mathcal{X}^{(j)}$ in the graph $\mathcal{G}$. The approach thereby favors edges with a close affinity, i.e., a small pairwise matching cost $w\big(\mathcal{X}^{(s_k)}, \mathcal{X}^{(s_{k+1})}\big)$. In our experiments, we demonstrate that this simple heuristic yields significant empirical improvements for a broad range of non-rigid matching tasks.

In practice, we utilize the multi-matching from Equation (6) for two distinct use-cases: For once, we can directly query the improved maps $\mathbf{\Pi}^{(i,j)}_{\mathrm{mult}}$ at test time. Additionally, we promote cycle-consistency during training via the following loss term

$$\ell^{(i,j)}_{\mathrm{cyc}} := E_{\mathrm{match}}\big(\mathbf{V}^{(i,j)}, \mathbf{V}^{(j)}; \mathbf{\Pi}^{(i,j)}_{\mathrm{mult}}\big). \tag{7}$$

This loss $\ell^{(i,j)}_{\mathrm{cyc}}$ imposes a soft penalty on inconsistencies between the registration $\mathbf{V}^{(i,j)}$ produced by the pairwise matching module Φ_{match}, and the multi-shape correspondences $\mathbf{\Pi}^{(i,j)}_{\mathrm{mult}}$ from Equation (6). As before, E_{match} is the matching energy defined in Equation (2).

3.4. Training protocol

The overall loss function that we minimize during training consists of two individual components

$$\ell := \mathbb{E}_{\mathcal{X}^{(i)}, \mathcal{X}^{(j)} \sim \mathcal{S}}\Big[\ell^{(i,j)}_{\mathrm{match}} + \lambda_{\mathrm{cyc}} \ell^{(i,j)}_{\mathrm{cyc}}\Big]. \tag{8}$$

Our complete pipeline is depicted in Figure 2. The whole network is trained end-to-end. In each training iteration, the backbone **I** and pairwise matching module **II** are queried in sequence to produce a pairwise matching for a pair of shapes $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$. The shape graph module **III** then produces the cycle-consistency loss $\ell^{(i,j)}_{\mathrm{cyc}}$. The shape graph $\mathcal{G}$ is updated regularly after a fixed number of epochs, taking into account the pairwise matches $\mathbf{\Pi}^{(i,j)}$ for all $i, j = 1, \dots, N$. For more details on the training schedule and choices of hyperparameters, see Appendix A.1.

4. Experiments

We provide various benchmark evaluations for non-rigid shape matching. We consider classical nearly-isometric datasets in Section 4.1, as well as more specialized benchmarks for matching with topological changes in Section 4.2 and inter-class pairs in Section 4.3. In Section 4.4, we compare different types of shape graph topologies. In Section 4.5, we provide an ablation study of our model, assessing the significance of individual network components.

Method	**FAUST**	**SCAPE**	**F** on **S**	**S** on **F**	**SUR**	**SH'19**
UFM [24]	5.7	10.0	12.0	9.3	9.2	15.5
SURFM [51]	7.4	6.1	19.0	23.0	38.9	37.7
WFM [56]	1.9	4.9	8.0	4.3	38.5	15.0
DiffNet [57]	1.9	2.6	2.7	1.9	8.8	11.0
DS [21]	1.7	2.5	5.4	2.7	2.7	12.1
NM [20]	**1.5**	4.0	6.7	2.0	9.7	2.8
IsoMuSh [22]	4.4	5.6	–	–	4.8	24.6
CZO [28]	2.2	2.5	–	–	2.2	6.3
UDM [10]	**1.5**	2.0	3.2	3.2	3.1	22.8
SyNoRiM [26]	7.9	9.5	21.9	24.6	12.7	7.5
Ours w/o **III**	1.7	3.3	4.2	1.7	8.1	6.2
Ours	**1.5**	**1.8**	**2.1**	**1.5**	**2.1**	**2.7**

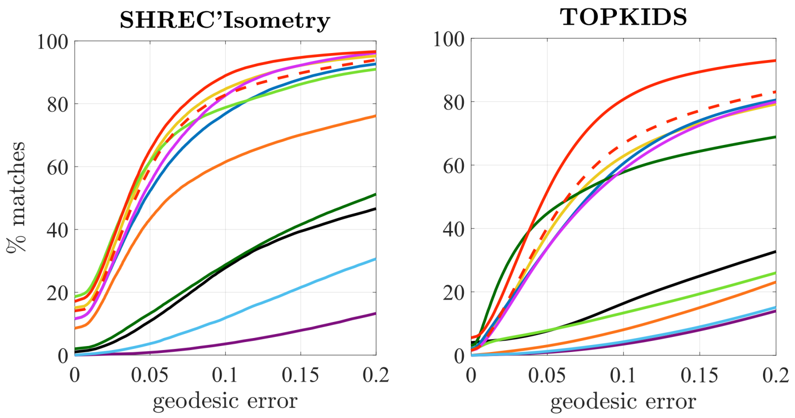
Table 1. **Nearly isometric matching.** A quantitative comparison on four nearly-isometric human shape benchmarks, FAUST [5], SCAPE [1], SURREAL [61] and SHREC'19 [41]. Following prior work [15, 56, 57], we additionally show generalization results when training on FAUST and testing on SCAPE (F on S), and vice versa. We consider both standard, pairwise baselines [20, 21, 24, 51, 56, 57] and multi-matching approaches [10, 22, 26, 28].

Baselines We compare G-MSM to existing deep learning approaches for unsupervised, deformable 3D shape correspondence. To this end, we consider both standard pairwise matching [20, 21, 24, 51, 56, 57] and multi-matching approaches [10, 26]. Since there are, to date, only very few learning-based multi-matching approaches, we additionally include Consistent ZoomOut [28] and IsoMuSh [22] as recent axiomatic multi-matching approaches.

Evaluation For each experimental setting, we report the mean geodesic correspondence error over all pairs of a given test set category. All evaluations are performed in accordance with the standard Princeton benchmark protocol [31].

4.1. Nearly isometric matching

Datasets We evaluate our method on four classical, nearly-isometric datasets. FAUST [5] contains 10 humans in 10 different poses each and SCAPE [1] contains 71 diverse poses of the same individual. We follow the standard benchmark protocol from existing work [15, 56, 57]. Specifically, we consider the more challenging remeshed geometries from [48] to avoid overfitting to a particular triangulation. SURREAL [61] consists of synthetic SMPL [37] meshes fitted to raw 3D motion capture data. The last benchmark, which is the most challenging among the four, is SHREC'19 Connectivity [41]. It contains human shapes in different poses with significantly varying sampling density and quality, as well as a small number of non-isometric poses.



	Method	SH'Iso	KIDS
🟧	UFM [24]	13.4	38.5
🟪	SURFM [51]	45.6	48.6
🟦	WFM [56]	38.0	47.9
⬛	DiffNet [57]	26.5	35.7
🟨	DS [21]	6.3	13.7
🟦	NM [20]	7.7	13.8
🟩	CZO [28]	7.6	39.3
🟩	UDM [10]	23.6	18.2
🟪	SyNoRiM [26]	6.2	13.8
🟥	Ours w/o **III**	6.3	12.0
🟥	Ours	**5.2**	**7.9**

Figure 3. **Matching with topological noise.** A summary of our quantitative comparisons on the topology benchmarks SHREC'Isometry [16] and TOPKIDS [33]. For both benchmarks, we show the cumulative error curves of our approach (red) and all considered baselines. Additionally, we provide the mean geodesic errors, averaged over all pairs of shapes, respectively (table, right).

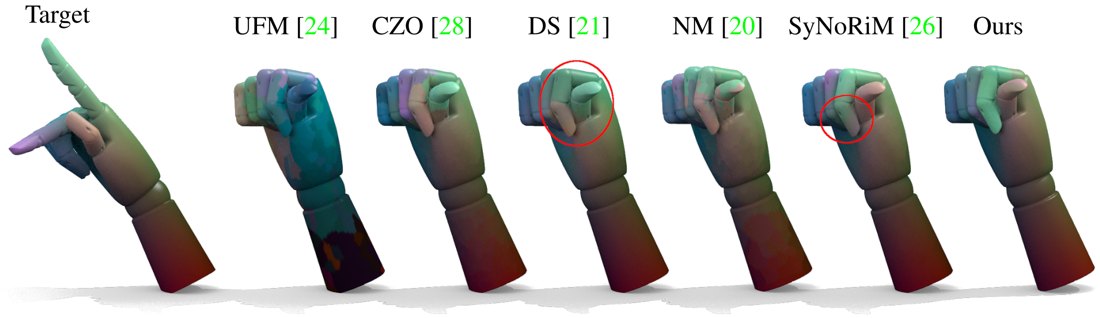


Figure 4. **Qualitative baseline comparison.** We consider a pair of real 3D scan meshes from SHREC'Iso [16], corresponding to the quantitative experiments shown in Figure 3. The two geometries are subject to topological merging due to self-contact of different fingers and parts of the hand. Correspondences are shown via a colormap for our method, as well as several baseline approaches. For this example, the best results are achieved by [21], [26] and our method. However, for both [21] and [26], the front part of the index and middle fingers are erroneous (tip of index finger should be bright green). See our supplementary material for additional qualitative examples.

Discussion The results on these four benchmarks are summarized in Table 1. Our method obtains state-of-the-art performance in all considered settings. Remarkably, these results were achieved directly through querying our network, whereas many baselines require correspondence postprocessing [10, 20, 24, 51, 56, 57]. Furthermore, the results underline that the shape graph module **III** plays a critical role in our pipeline for optimal performance.

4.2. Matching with topological changes

Datasets The benchmarks SHREC'Isometry [16] and TOPKIDS [33] focus on matching with topological noise. This is a common phenomenon when working with real scans, where the mesh topology is often corrupted by self-contact of separate parts of the scanned objects. Such topological merging severely affects the correspondence estimation since it distorts the intrinsic shape geometry non-isometrically. The first benchmark SHREC'Isometry [16] contains real scans of different humanoid puppets and hand models. A majority of poses in their 'heteromorphic' test set are subject to topological changes, see also Figure 4 for an example. The TOPKIDS [33] dataset contains synthetic shapes of human children where topological merging is emulated by computing the outer hull of intersecting geometries, see Figure 1 for an example.

Discussion Quantitative results are shown in Figure 3. We observe that topological merging commonly leads to unstable behavior for methods that rely on intrinsic priors like preservation of the Laplace-Beltrami operator [10, 28, 51, 56, 57] or pairwise geodesic distances [24]. It further inhibits approaches that learn to morph input geometries [20, 26] with explicit deformation priors, since merged regions tend to adhere to each other, see, e.g., the discussion on failure cases in [26, Sec. 7]. Our pipeline decreases the correspondence error by a decisive margin of 19% for SHREC'Iso and 73% for TOPKIDS. We provide a qualitative comparison in Figure 4, as well as additional examples in Appendix D.

	SH'20	SH'20 on SMAL	SH'20 on TOSCA					
			Cat	Centaur	Dog	Horse	Human	Wolf
UFM [24]	39.8	32.9	39.4	39.2	37.5	34.1	49.6	4.4
SURFM [51]	53.4	37.7	54.0	57.7	57.9	57.0	65.8	55.3
WFM [56]	31.4	20.2	20.6	21.9	16.7	22.4	38.1	5.7
DiffNet [57]	40.5	18.2	14.2	8.3	13.6	9.1	24.5	2.6
DS [21]	35.0	10.8	7.6	9.1	5.5	2.5	10.1	2.1
NM [20]	**10.0**	9.9	16.8	12.7	14.6	11.2	29.7	1.5
CZO [28]	21.7	–	–	–	–	–	–	–
UDM [10]	52.6	25.5	40.7	34.3	43.6	43.0	45.8	34.3
SyNoRiM [26]	10.4	5.7	12.8	11.6	10.6	7.1	28.2	2.0
Ours w/o **III**	11.1	3.4	6.3	6.0	4.9	2.6	20.1	2.2
Ours	10.6	**2.6**	**5.2**	**2.0**	**3.0**	**2.2**	**8.3**	**1.4**

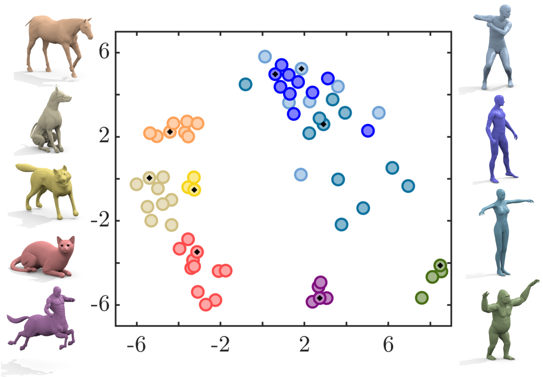


Figure 5. **Inter-class matching.** (left) A comparison of our approach to the considered baselines on SHREC'20 [17], as well as additional, synthetic test sets from SMAL [65] and TOSCA [8]. (right) Additionally, we visualize the shape graph node embeddings of our approach on TOSCA [8] through 2D multi-dimensional scaling. Since the learned edge weights express affinity scores, shapes with similar geometries tend to cluster together. Shapes with four legs (orange ≙ horse, red ≙ cat, . . .) and two legs (blue ≙ human, green ≙ gorilla) are linearly separable in the 2D MDS space. Interestingly, the centaur classes' embedding (purple) lies exactly between these two categories.

4.3. Inter-class matching

Datasets The SHREC'20 [17] challenge contains real scans of various four-legged animal models, including: elephant, giraffe, bear, and many more. These geometries were obtained from inhomogeneous acquisition sources, i.e., different types of scanners and 3D reconstruction pipelines. Sparse ground-truth correspondences were obtained through manual annotation. We further assess the generalization to additional test sets from the synthetic SMAL [65] and TOSCA [8] datasets. SMAL contains inter-class pairs between different animal classes, whereas TOSCA contains nearly-isometric pairs with both animal and human classes.

Discussion Our approach yields the most stable results overall, see Figure 5. Several baselines suffer from unstable behavior for animals from SHREC'20, because they depend on either noisy SHOT [59] input features [21], intrinsic priors that favor near isometries [28, 56, 57], or both [10, 24, 51]. While methods with an explicit deformation prior [20, 26] perform well on SHREC'20, they do not generalize well to unseen test poses from SMAL and TOSCA. Our method learns a topology-aware shape graph prior and thereby gets the best out of both worlds, i.e., robustness to inter-class pairs and strong generalization to unseen test pairs.

4.4. Sparse graph topologies

Throughout our experiments, we use complete shape graphs $\mathcal{G}$ with a full set of $\frac{N(N-1)}{2}$ edges, as specified in Section 3.3. Here, we explore a few alternative graph topologies with sparse connectivity patterns, i.e., $\mathcal{O}(N)$ edges: (ii) Minimal spanning trees (MST), (iii) minimal paths solving the traveling salesman problem (TSP) and (iv) star graphs, where all nodes are connected to one center node. For a detailed discussion and visualizations of these graph types, see Appendix B. We compare (ii)-(iv) to (i) the standard full graph and (v) our pipeline without the graph module **III**. A given graph type can be employed either for the cycle consistency loss in Equation (7) during training or for map refinement at test time. For a complete picture, we report results for all $5 \times 5 = 25$ possible combinations.

Our results in Table 2 indicate that, while the full graph is generally the most accurate, the sparse topologies often perform comparably, especially MST. This makes them a viable alternative to the full graph in certain scenarios with limited resources, both in terms of the required memory and query time. We provide a comprehensive cost analysis in Appendix C. In most cases, using the shape graph during training is beneficial, even when no graph is available at test time (Table 2, bottom row). This makes it relevant for online applications where not all test pairs are available at once. Regardless of the graph type, it is generally preferable to include some version of our graph module **III** rather than directly using the pairwise correspondences 'w/o'.

4.5. Ablation study

Our proposed architecture consists of several basic building blocks **I-III**, as defined in Section 3.2 and Section 3.3. While the shape-graph module **III** is unique to our approach, the feature backbone **I** and matching module **II** can, in principle, be replaced by any analogous off-the-shelf architectures. We compare several popular alternatives: For the feature backbone, we consider I.a the spectral convolution architecture from [21], I.b our network with SHOT [59] input features, I.c the ResNet architecture from [34], I.d PointNet [47], as well as I.e the message passing architecture from [20]. For the differentiable matching module, we compare II.a a functional map layer [44], II.b a single Sinkhorn

SH'Iso		*Train on*				
		Full	MST	TSP	Star	w/o
Test on	Full	5.16	**5.09**	5.20	5.90	5.17
	MST	5.68	5.49	5.50	6.36	5.49
	TSP	6.08	5.62	5.80	6.94	6.51
	Star	5.54	5.26	5.44	6.71	6.02
	w/o	5.32	5.27	5.42	6.33	6.27

TOPKIDS		*Train on*				
		Full	MST	TSP	Star	w/o
Test on	Full	**7.92**	8.13	8.44	11.03	9.13
	MST	8.56	8.62	9.39	10.57	9.98
	TSP	13.18	12.33	13.10	19.72	15.07
	Star	8.61	8.84	8.34	11.92	9.63
	w/o	10.62	10.64	11.61	13.62	12.02

Table 2. **Graph topology comparison.** We compare the quantitative performance of our model for different graph topologies $\mathcal{G}$. Specifically, we revisit the experiment from Figure 3 and report the mean geodesic error on SHREC'Iso [16] and TOPKIDS [33]. The standard 'full' graph is compared to three sparse topologies 'MST', 'TSP', 'star' graph, as well as the 'w/o **III**' variant of our pipeline.

		TOPKIDS		SHREC'20	
	with **III**	✓	✗	✓	✗
Feature I	**I.a** SpecConv [21]	8.53	13.68	28.54	34.99
	I.b SHOT [59]	7.93	13.66	22.74	29.81
	I.c ResNet [34]	7.94	13.14	39.69	40.66
	I.d PointNet [47]	8.78	14.10	11.01	11.54
	I.e GraphNN [20]	14.18	25.57	14.53	18.33
Match II	**II.a** FM [44]	39.12	40.66	50.58	51.37
	II.b Sinkhorn [13]	12.25	14.81	11.58	12.66
	II.c Softmax [20]	12.78	13.46	11.49	13.47
	G-MSM (ours)	**7.92**	**12.02**	**10.65**	**11.06**

Table 3. **Ablation network architecture.** We compare several off-the-shelf network architectures for the feature backbone **I** and matching module **II** to our full model, as defined in Section 3.2. For each setting, we contrast the results obtained with (✓) and without (✗) the graph-based multi-matching module **III**.

layer [13] and II.c a standard per-point Softmax [20].

We then replace either the feature backbone I.a-I.e or matching module II.a-II.c in our method and observe how it affects the accuracy on TOPKIDS [33] and SHREC'20 [17] from Section 4.2 and Section 4.3, respectively. The results are summarized in Table 3. Replacing either module **I** or **II** in our approach leads to a drop in performance. Moreover, we see that, regardless of the concrete architecture, our multi-matching approach **III** (✓in Table 3) improves the performance over the pairwise matches (✗).

5. Conclusion

We propose G-MSM, a novel multi-matching approach for non-rigid shape correspondence. For a given collection of 3D meshes, we define a shape graph $\mathcal{G}$ which approximates the underlying shape data manifold. Its edge weights w are extracted from putative pairwise correspondence signals in a self-supervised manner. Our network promotes cycle-consistency of optimal paths in $\mathcal{G}$. Thus, it produces context-aware multi-matches that are informed by commonalities and salient geometric features across all training poses. In our experiments, we demonstrate that this simple strategy yields significant improvements in correspondence accuracy on a wide range of challenging, real-world 3D mesh benchmarks.

Limitations & future work Our method can effectively learn the underlying canonical shape topology from a collection of 3D meshes. On the other hand, it relies on at least some of the poses to convey this latent topology. In the extreme case of $N = 2$ input poses, our multi-shape pipeline does not yield an improvement over the naive pairwise maps. The multi-matching **III** shows consistent improvements in our empirical evaluations. On the other hand, it is difficult to provide theoretical guarantees for our approach, since it is based on a self-supervised graph heuristic. We would like to further explore this direction in future work. While our approach learns correspondences in a self-supervised manner, it also assumes rigidly aligned input shapes which might be limiting for certain applications.

A promising avenue for future research is applying our multi-matching formulation to related but distinct settings to extend them beyond the pairwise training paradigm. One potential direction is extending our framework to allow for partial views, e.g., by leveraging recent advances on learnable partial functional maps [2].

Societal impact Advancing the robustness and accuracy of shape correspondence methods has the potential to open up new avenues for future applications based on 3D scan data. Our algorithm constitutes one small advancement in this effort of extending computer vision algorithms to the 3D domain. Since our algorithm is fully unsupervised, it can directly reduce deployment costs as no manual correspondence annotations are required to train our model. Shape correspondence is a fundamental building block at the heart of many 3D vision algorithms and we do not anticipate any immediate risk of misuse associated with this work.

Acknowledgements

We acknowledge support by the ERC Advanced Grant SIMULACRON, the Munich School for Data Science and the Munich Center for Machine Learning.

References

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416, 2005. 5

[2] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. Dpfm: Deep partial functional maps. In *2021 International Conference on 3D Vision (3DV)*, pages 175–185. IEEE, 2021. 1, 2, 3, 8

[3] Matthieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. *IEEE International Conference on Computer Vision (ICCV) - Workshop on Dynamic Shape Capture and Analysis*, 2011. 2

[4] 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*, 33:12909–12922, 2020. 2, 3, 14

[5] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE. 5

[6] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in neural information processing systems*, pages 3189–3197, 2016. 2

[7] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006. 2

[8] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. *Numerical geometry of non-rigid shapes*. Springer, 2008. http://tosca.cs.technion.ac.il/book/resources_data.html. 7

[9] 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. 2

[10] D. Cao and F. Bernard. Unsupervised deep multi-shape matching. In *European Conference on Computer Vision (ECCV)*, 2022. 2, 3, 5, 6, 7, 16

[11] Luca Cosmo, Emanuele Rodola, Andrea Albarelli, Facundo Mémoli, and Daniel Cremers. Consistent partial matching of shape collections via sparse modeling. In *Computer Graphics Forum*, volume 36, pages 209–221. Wiley Online Library, 2017. 2

[12] Luca Cosmo, Emanuele Rodola, 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. IEEE, 2016. 1

[13] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013. 8, 13

[14] Nicolas Donati, Etienne Corman, Simone Melzi, and Maks Ovsjanikov. Complex functional maps: a conformal link between tangent bundles. *arXiv preprint arXiv:2112.09546*, 2021. 2

[15] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. *arXiv preprint arXiv:2003.14286*, 2020. 1, 2, 3, 5

[16] Roberto Dyke, Caleb Stride, Yukun Lai, and Paul Rosin. Shrec-19: Shape correspondence with isometric and non-isometric deformations. *Eurographics Workshop on 3D Object Retrieval*, 2019. 1, 2, 6, 8

[17] 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. 1, 2, 7, 8

[18] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38, pages 1–12. Wiley Online Library, 2019. 2

[19] 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. 2, 13

[20] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021. 1, 2, 3, 5, 6, 7, 8, 16, 17

[21] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixe, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *arXiv preprint*, 2020. 1, 2, 3, 4, 5, 6, 7, 8, 12, 13, 15, 16, 17

[22] Maolin Gao, Zorah Lahner, Johan Thunberg, Daniel Cremers, and Florian Bernard. Isometric multi-shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193, 2021. 3, 5

[23] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 3, 14

[24] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019. 1, 2, 3, 5, 6, 7

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[26] Jiahui Huang, Tolga Birdal, Zan Gojcic, Leonidas J Guibas, and Shi-Min Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence, 2022. 2, 3, 5, 6, 7, 16, 17

[27] Qi-Xing Huang and Leonidas Guibas. Consistent shape maps via semidefinite programming. In *Computer graphics forum*, volume 32, pages 177–186. Wiley Online Library, 2013. 2

[28] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pages 265–278. Wiley Online Library, 2020. 2, 3, 5, 6, 7

[29] Faria Huq, Adrish Dey, Sahra Yusuf, Dena Bazazian, Tolga Birdal, and Nina Miolane. Riemannian functional map synchronization for probabilistic partial correspondence in shape networks. *arXiv preprint arXiv:2111.14762*, 2021. 3

[30] Itay Kezurer, Shahar Z Kovalsky, Ronen Basri, and Yaron Lipman. Tight relaxation of quadratic matching. In *Computer graphics forum*, volume 34, pages 115–128. Wiley Online Library, 2015. 2

[31] Vladimir G Kim, Yaron Lipman, and Thomas A Funkhouser. Blended intrinsic maps. *Transactions on Graphics (TOG)*, 30(4), 2011. 5

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 12

[33] Zorah Lähner, Emanuele Rodolà, Michael M Bronstein, Daniel Cremers, Oliver Burghard, Luca Cosmo, Andreas Dieckmann, Reinhard Klein, and Yusuf Sahillioglu. Shrec'16: Matching of deformable shapes with topological noise. *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2:11, 2016. 1, 2, 6, 8, 16

[34] 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. 2, 3, 7, 8

[35] Or Litany, Emanuele Rodolà, Alex Bronstein, and Michael Bronstein. Fully spectral partial shape matching. *Computer Graphics Forum*, 36(2):1681–1707, 2017. 2

[36] Or Litany, Emanuele Rodolà, Alex M Bronstein, Michael M Bronstein, and Daniel Cremers. Non-rigid puzzles. *Computer Graphics Forum (CGF), Proceedings of Symposium on Geometry Processing (SGP)*, 35(5), 2016. 2

[37] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 2, 5, 14

[38] Riccardo Marin, Simone Melzi, Emanuele Rodolà, and Umberto Castellani. FARM: functional automatic registration method for 3d human bodies. *CoRR*, abs/1807.10517, 2018. 2, 3

[39] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *Advances in Neural Information Processing Systems*, 33:1608–1620, 2020. 1, 2, 3

[40] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015. 2

[41] Simone Melzi, Riccardo Marin, Emanuele Rodolà, and Umberto Castellani. Matching humans with different connectivity. *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2019. 5

[42] 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. 2

[43] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017. 2

[44] 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, 2012. 2, 7, 8, 13

[45] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019. 3

[46] Adrien Poulenard and Maks Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018. 2

[47] 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. 7, 8

[48] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Trans. Graph.*, 37(6):248:1–248:16, Dec. 2018. 2, 5

[49] Emanuele Rodola, Samuel Rota Bulo, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4177–4184, 2014. 2

[50] Emanuele Rodolà, Luca Cosmo, Michael Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Computer Graphics Forum (CGF)*, 2016. 2, 3

[51] 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. 1, 2, 3, 5, 6, 7

[52] Klaus Hildebrandt Ruben Wiersma, Elmar Eisemann. Cnns on surfaces using rotation-equivariant features. *Transactions on Graphics*, 39(4), July 2020. 2

[53] Raif M Rustamov et al. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Symposium on geometry processing*, volume 257, pages 225–233, 2007. 2

[54] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, pages 1–17, 2019. 2

[55] Frank R Schmidt, Eno Töppe, Daniel Cremers, and Yuri Boykov. Intrinsic mean for semi-metrical shape retrieval via graph cuts. In *Joint Pattern Recognition Symposium*, pages 446–455. Springer, 2007. 2

[56] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional map for shape matching. *arXiv preprint arXiv:2009.13339*, 2020. 1, 2, 3, 5, 6, 7

[57] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022. 1, 2, 3, 4, 5, 6, 7, 12, 13

[58] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009. 2

[59] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. *In Proceedings of European Conference on Computer Vision (ECCV)*, 16(9):356–369, 2010. 2, 7, 8

[60] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011. 2

[61] Gul 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. 5, 14, 15

[62] Matthias Vestner, Zorah Lähner, Amit Boyarski, Or Litany, Ron Slossberg, Tal Remez, Emanuele Rodolà, Alex M. Bronstein, Michael M. Bronstein, Ron Kimmel, and Daniel Cremers. Efficient deformable shape correspondence via kernel matching. In *International Conference on 3D Vision (3DV)*, October 2017. 2

[63] Cédric Villani. *Topics in optimal transportation*. American Mathematical Soc., 2003. 3

[64] Thomas Windheuser, Ulrich Schlickwei, Frank R Schimdt, and Daniel Cremers. Large-scale integer linear programming for orientation preserving 3d shape matching. In *Computer Graphics Forum*, volume 30, pages 1471–1480. Wiley Online Library, 2011. 2

[65] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017. 1, 2, 7, 17

Part III

Conclusion and Outlook

Chapter 10

Summary

In this thesis, we summarize several of our novel contributions towards robust correspondence and interpolation of non-rigid 3D shapes. A primary motivation is extending such techniques to noisy real-world data. To this end, we introduce various tools and specific approaches that address shape analysis tasks for a broad range of deformable 3D object categories, such as human bodies, animals, or hands. We primarily focus on surface-based shape representations, specifically 3D triangular meshes. In the following, we discuss several key aspects of our contributions.

Intrinsic/Extrinsic Shape Embeddings In many of our proposed methods, we demonstrate that combining extrinsic shape alignment and intrinsic matching in holistic frameworks yields substantial mutual benefits. While intrinsic shape embeddings are robust to nearly-isometric, non-rigid pose deformations, they are often not sufficiently expressive to capture relevant shape features. Common well-known failure modes include ambiguous maps in the case of self-similarities and intrinsic symmetries. Moreover, intrinsic representations are agnostic to useful geometric features of an observed object, such as its outer normals, or the local curvature at each point. Thus, we find that, in practice, incorporating both types of shape representations in a joint embedding space produces more robust predictions.

Volumetric Fields vs. Surface-based Deformation An important consideration in the context of extrinsic shape alignment is the appropriate representation of deformation models. We mainly distinguish between Eulerian and Lagrangian flow fields. In the Eulerian perspective, the flow of a unit of mass (*i.e.*, a point on a 3D surface), is specified in terms of its location in the surrounding 3D space. In contrast, in the Lagrangian viewpoint we directly model the flow at each point on the surface. Our work comprises instances of both volumetric [1, 3] and surface-based [2, 4–6] deformation fields. On the one hand, volumetric fields allow for reasoning about interactions between points that are in close proximity. Hence, they can be constrained to avoid self-intersections of the deformed surface, and to specify conservation laws of physical properties such as volume-preservation. Moreover, they allow for applying existing flow fields to different poses, and to extrapolate from a sparse set of aligned points to the full resolution [3]. On the other hand, surface-based deformation fields yield a more compact representation, reducing the associated computation cost. They further simplify modeling important properties, such as translation equivariance, and enable reasoning about intrinsic shape distortion (*cf*. Sec. 3.3.4).

Multi-Scale Matching Multi-scale methods for shape correspondence are characterized by iterative refinement and gradual upsampling of the resolution. We introduce a specific approach in [2], which jointly predicts an extrinsic shape alignment and an intrinsic functional map between pairs of input surfaces. The level of detail, in this case, is indicated by the number of retained Laplace-Beltrami eigenfunctions in both the smooth shell low-pass filtering and the spectral embedding. Initially, the optimization yields a coarse alignment of the input pairs, whereas subsequent iterations focus on matching fine-scale details. The same methodology is further extended and combined with geometric deep learning in several follow-up works [5, 6].

Alignment-based Shape Interpolation A common assumption in classical shape interpolation approaches is that both input meshes $\mathcal{X}$ and $\mathcal{Y}$ have the same number of vertices with an identical triangulation (*cf*. Sec. 3.3.3). A subtle difficulty in extending this formulation to noisy real-world meshes are minor inconsistencies due to the incompatible discretization. In our contributions [1, 3, 4], we often define interpolation by continuously morphing the first input shape $\mathcal{X}$ such that the final frame of the obtained sequence overlaps closely with the second reference pose $\mathcal{Y}$. In practice, however, the predicted overlap is only approximate. One alternative are generative shape space models (*cf*. Sec. 3.3.5), which reconstruct intermediate sequences through latent space interpolation. On the other hand, we assert in [4] that such approaches often yield inferior predictions on unseen test poses compared to the alignment-based formulation. Moreover, the discretization of the generated intermediate frames is again incompatible with the input poses $\mathcal{X}$ and $\mathcal{Y}$.

Optimization vs. Deep Learning While a majority of classical shape analysis works obtain solutions in optimization-based frameworks, many recent methods leverage geometric deep learning. Whereas optimization approaches often rely on generic handcrafted shape descriptors (*cf*. Sec. 3.2.3), deep learning approaches leverage representation learning (*cf*. Sec. 3.2.4) to identify relevant geometric features in the training set distribution. On the other hand, the resulting feature maps are more specialized, with a limited generalization to samples outside the training distribution. Assumptions about solutions of an optimization problem are often incorporated in a transparent manner, and specific formulations yield optimality guarantees and other explicit mathematical properties of solutions. Conversely, many non-convex optimization formulations are actually more prone to inferior local minima and, hence, reliant on high-quality initializations. While learning approaches generally have a lower deployment cost of making predictions for unseen test pairs, this is contrasted with a much larger, up-front training cost. Nevertheless, with geometric data becoming more accessible and wide-spread, we expect that that impact of geometric deep learning for shape analysis tasks will further increase in the future. The contributions discussed in this thesis include examples of both optimization [1–3] and learning [4–6] approaches.

Self-Supervised Learning Despite the significant progress of geometric deep learning for shape analysis tasks, a major limitation of state-of-the-art supervised approaches remains the need for correspondence labels during training. While raw geometric data is widely available, obtaining ground-truth maps requires extensive manual annotation by human experts. As a result, existing real-world benchmarks for shape correspondence are often limited to sparse, annotated landmarks [45]. Some datasets [22, 82] leverage parametric deformable models [78, 79] to generate ground-truth maps, but this is only feasible for specific shape categories, such as human bodies. Overall, most annotated benchmarks are limited to a comparably small number of poses (≈ 100). While synthetic datasets are often orders of magnitude larger [125], the significant domain gap between real-world scans and synthetic meshes limits their practical utility. Hence, in our contributions we address the classical data annotation bottleneck through self-supervised learning. Specially, we devise several models [4–6] based on geometric priors, where the ground-truth correspondences are replaced with extrinsic alignment loss functions.

Chapter 11

Future Research

Classical shape analysis benchmarks often comprise nearly-isometric, synthetic shapes with a consistent triangulation [15, 22, 28]. Since these assumptions are too restrictive for many relevant real-world applications, we consider real scan datasets in several of our proposed approaches [1, 2, 4–6]. Nevertheless, even to date a significant domain gap remains, inhibiting shape analysis algorithms from being applied to raw sensory observations. Addressing such challenging scenarios requires approaches that are robust to severe degrees of scanning noise, outliers, disconnected components, topological changes, partial views, and self-occlusions. While there are distinct works that consider meshes subject to topological noise [71], varying discretization densities [82], and partial matching [7, 48, 99], to date there is still a lack of large-scale benchmarks and methods that address several of these challenges at once.

Another potential avenue for future work involves algorithms capable of processing different data representations, such as point clouds, depth maps, voxel grids, signed distance fields, or neural implicit surfaces. While for many of the shape analysis tools and concepts introduced in Chapter 3 there exist analogous definitions for different formats [30, 109, 134], there is still a significant demand for specialized benchmarks and approaches, as well as hybrid frameworks between multiple representations.

Finally, there is a significant potential for future approaches that combine generative machine learning models with classical shape analysis techniques [52, 121, 134]. In this manner, specific approaches can help improve 3D synthesis through geometrically meaningful priors, leverage object generation for improving shape analysis tasks, or devise hybrid frameworks that jointly address shape generation, correspondence, and interpolation.

Part IV

Appendix

Appendix A

Supplementary Material – Smooth Shells: Multi-Scale Shape Registration with Functional Maps

A. MCMC - pseudo code

In Section 5 we already gave a rough description of our MCMC initialization algorithm. Here, we provide a more detailed pseudo code:

Algorithm 3. *(MCMC)*

```
1. Initialize $\tau_{\text{best}} := 0, \mathbf{X}^*_{\text{best}} := \mathbf{X}$.
2. For $i = 1, ..., N_{\text{prop}}$:
    2.1 Sample new proposal $\tau_{\text{prop}} \sim \mathcal{N}(0, \mathbf{I})$.
    2.2 Compute the current alignment $\mathbf{X}^*_{\text{prop}}$ by making
        a surrogate run with the initial guess $\tau_{\text{prop}}$.
    2.3 Compute the acceptance probability $\alpha := \exp\Bigl(-\frac{1}{2\sigma^2_{\text{match}}}\bigl(E(\mathbf{X}^*_{\text{prop}}) - E(\mathbf{X}^*_{\text{best}})\bigr)\Bigr)$ using
        the energy $E$ from Eq. (12).
    2.4 Sample $u \sim \text{Unif}(0,1)$ and either accept or reject the new sample $\tau_{\text{prop}} \in \mathbb{R}^{K_{\text{init}} \times 3}$:
        $(\tau_{\text{best}}, \mathbf{X}^*_{\text{best}}) := \begin{cases} (\tau_{\text{prop}}, \mathbf{X}^*_{\text{prop}}), & u \leq \alpha \textit{ (accept)} \\ (\tau_{\text{best}}, \mathbf{X}^*_{\text{best}}), & u > \alpha \textit{ (reject)} \end{cases}$
```

We usually set the number of surrogates to $N_{\text{prop}} := 100$. In the majority of cases in our experiments this is more than sufficient. Furthermore, we usually choose a small objective variance $\sigma^2_{\text{match}} := 0.001$ to get a sharp distribution and therefore more accurate samples τ.

Remarks One aspect of our method that we did not talk about yet is how to compute a good initial rigid pose. For most datasets in our experiments this is a requirement, e.g. SHREC'19 [34] connectivity has random rigid poses for all inputs. In theory, our MCMC algorithm can account for rigidly displaced inputs $\mathcal{X}$ and $\mathcal{Y}$ but in practice our $N_{\text{prop}} = 100$ surrogates are not enough for extreme cases. Therefore, we initially apply another surrogate based method that initializes with different rigid poses and determines the best one according to the objective E from Eq. (12). A thorough description of this is beyond the scope of this paper, but all the details can be found in our implementation.

B. Proof of Theorem 1

Theorem 1 gives an upper bound on how much the geometry of our smooth shells can change between two states K and $K+1$. For spectral reconstruction, a projection on a new eigenfunction is added in each iteration. Depending on the magnitude of the new projection $\bigl(\phi_{K+1} \otimes \phi_{K+1}\bigr)$, this can lead to arbitrarily high changes :

$$\|\mathcal{T}_{K+1}(X) - \mathcal{T}_K(X)\|_{L^2} = \|\bigl(\phi_{K+1} \otimes \phi_{K+1}\bigr)X\|_{L^2}. \quad (14)$$

In comparison, Theorem 1 states that the change from $\mathcal{S}_K(X)$ to $\mathcal{S}_{K+1}(X)$ can be bounded by choosing a small upsampling variance σ.

Proof. We will proof the statement for scalar functions $X \in L^2(\mathcal{X})$. The extension to vector valued functions $L^2(\mathcal{X}, \mathbb{R}^3)$ is trivial – we just need to apply the identity to each component at a time. Now let $K > 0$ and $\sigma > 0$. For brevity we will denote the sigmoid weights with $s_k^K := \frac{1}{1+\exp\bigl(\sigma(k-K)\bigr)}$. Using the spectral decomposition of operators, we can deduce:

$$\begin{aligned}
&\bigl\|\mathcal{S}_{K+1}(X) - \mathcal{S}_K(X)\bigr\|^2_{L^2} = \\
&\sum_{k=1}^{\infty}\bigl|\bigl(s_k^{K+1} - s_k^K\bigr)\bigl\langle\phi_k, X\bigr\rangle_{L^2}\bigr|^2 = \\
&\sum_{k=1}^{\infty}\Bigl|\Bigl(\bigl(s_k^K\bigr)^{-1} - \bigl(s_k^{K+1}\bigr)^{-1}\Bigr)s_k^{K+1}s_k^K\bigl\langle\phi_k, X\bigr\rangle_{L^2}\Bigr|^2 = \\
&\sum_{k=1}^{\infty}\bigl|(1-e^{-\sigma})\exp\bigl(\sigma(k-K)\bigr)s_k^{K+1}s_k^K\bigl\langle\phi_k, X\bigr\rangle_{L^2}\bigr|^2 \leq \\
&\sum_{k=1}^{\infty}\Bigl|(1-e^{-\sigma})\Bigl(1+\exp\bigl(\sigma(k-K)\bigr)\Bigr)s_k^{K+1}s_k^K\bigl\langle\phi_k, X\bigr\rangle_{L^2}\Bigr|^2 = \\
&\sum_{k=1}^{\infty}\bigl|(1-e^{-\sigma})s_k^{K+1}\bigl\langle\phi_k, X\bigr\rangle_{L^2}\bigr|^2 = \\
&\bigl|(1-e^{-\sigma})\bigr|^2\bigl\|\mathcal{S}_{K+1}(X)\bigr\|^2_{L^2}.
\end{aligned}$$

Taking the square root on both sides then yields the desired identity. □

Remarkably, this bound is independent of the index K. Small eigenfunctions ϕ_k typically represent coarse structures like limbs. Therefore, in particular the first iterations using spectral reconstruction lead to big changes in the geometry, see Eq. (14).

C. Runtime Analysis

We analyze the time complexity of our method in comparison to other popular matching methods in Figure 9. In particular, we compare the runtime of the whole pipelines for instances of the same pair of Michael shapes from the TOSCA dataset that was remeshed to different resolutions between 500 and $50k$ vertices.

D. Additional Qualitative Evaluations

We provide some additional qualitative evaluations and comparisons of our pipeline in order to give the reader a better understanding about the merits of our method, see Figure 10. Additionally, we provide a failure case in Figure 11. Our method is deformation based with an as-rigid-as-possible assumption. This means that in places of

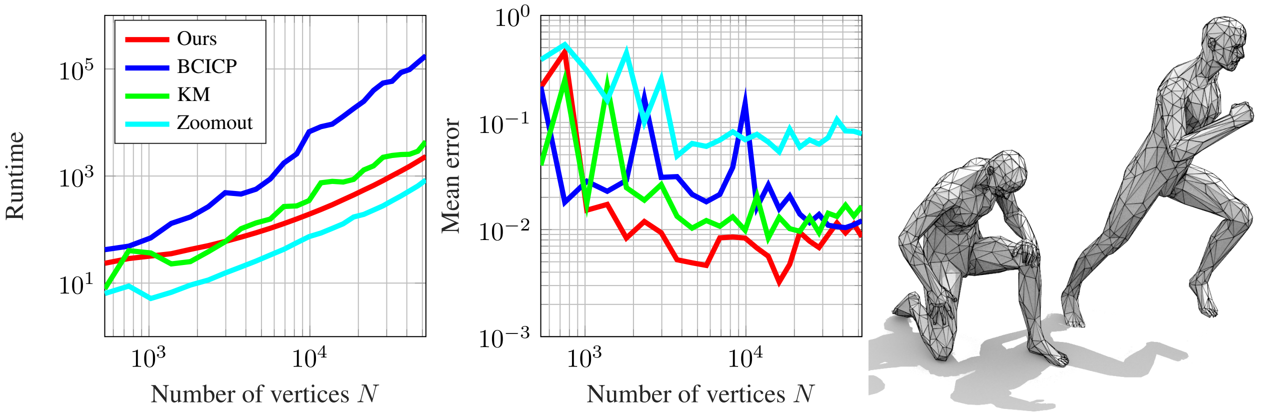


Figure 9: We compare the runtimes of our method with BCICP [42], Kernel matching [56] and Zoomout [35]. To this end, we remesh the Michael shape from TOSCA to different resolutions, on the right side we display the pair for $N = 1000$. Besides the runtime we also compare the matching accuracies of all methods. Here, our method is the most accurate one and stable across resolutions, whereas our runtime is the second best after Zoomout.

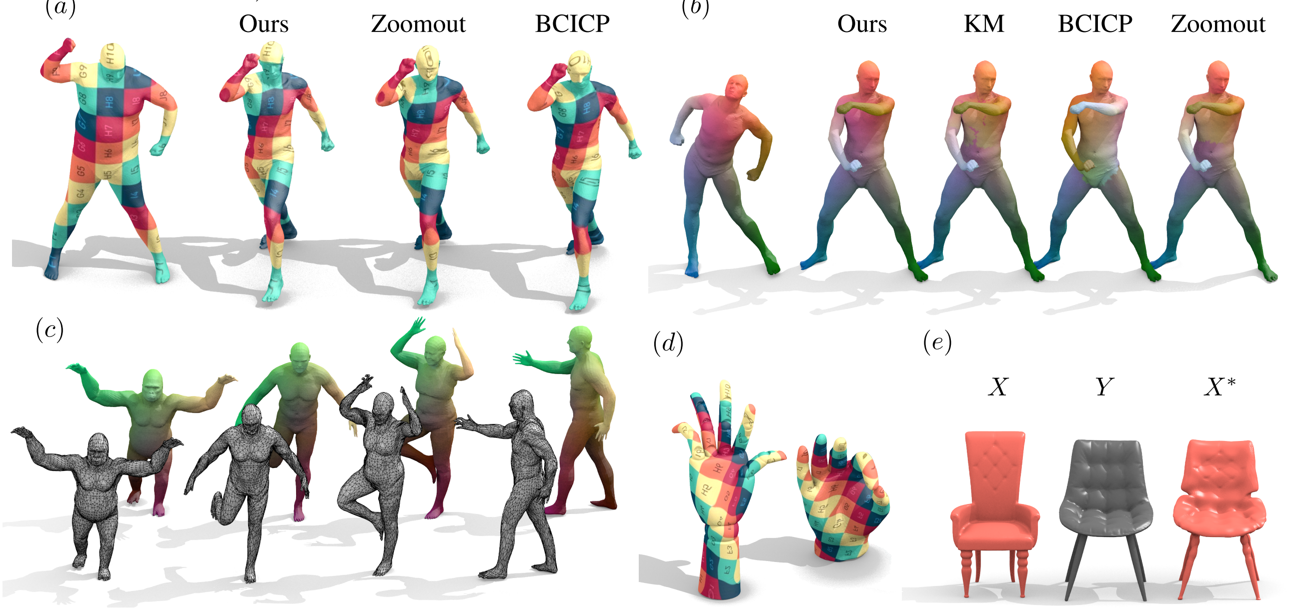


Figure 10: Here, we show additional qualitative evaluations of our method. (a) and (b) are comparisons for a pair from the SHREC'19 and the SCAPE dataset respectively. (a) is challenging due to an incompatible meshing, (b) has equivalent meshing but is still susceptible to mismatches due to self-similarities (left-right: BCICP, front-back: KM, Zoomout). (c) shows how our method can be used to smoothly transfer meshings for interclass pairs, here for a gorilla from TOSCA to humans from FAUST. The maps are smooth in the sense that local structures are preserved and deformations only occur in the form of uniform, global stretching of parts. I.e. the face still looks like a gorilla after deformation although the rest of the body adapts to the human form. (d) shows a texture transfer from a template hand (right) to a scanned hand of a puppet (left). The latter is a scan of a real world object from [14], obtained with the handheld Space Spider scanner from Artec. This is a challenging example due to different resultions of the inputs, different small scale features and a different size of the residual part at the bottom. (e) shows how our method can be applied to deform an object (red chair) and align it with a reference shape (black chair) to create a new object. The deformed red chair X^* has the global structure of Y and the fine scale details of X.

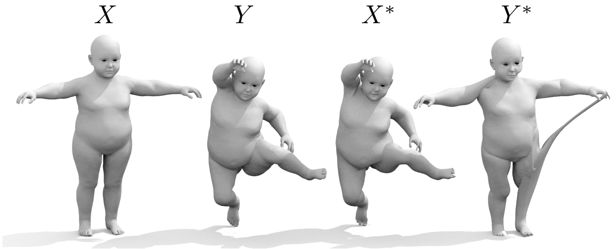


Figure 11: A failure case of our method for a pair of shapes from TOPKIDS. If we align the the canonical pose X with the reference Y we get a meaningful alignment X^* and high quality correspondences. However, if we try to apply our method the other way around, we get undesirable "cheese pull" effects at the left hand of the deformed kid Y^*. The reason for that is that in the pose Y the fingers touch the left knee and the meshing connects. In order to avoid this effect, we either need a mesh separation policy or use an intermediate template where the original topology is known. We prefer the latter approach in our quantitative evaluations on FAUST and TOPKIDS because finding a meaningful topological cut is a complicated problem on its own.

topological changes the meshing cannot be separated. Our method still tries to align the shape as good as possible with the reference which invariably leads to a "cheese pull" effect. This is also the main reason why we use an intermediate template to match the FAUST and TOPKIDS shapes in our quantitative evaluations.

Appendix B

Supplementary Material – Hamiltonian Dynamics for Real-World Shape Interpolation

A Proof of Theorem 1

We now provide a proof of Theorem 1 which states that the quality of the approximation of $v^{(t+2)}$ is improved by one error order when the extrapolation step (11d) is added after the velocity update step.

Proof. If we remove the extrapolation step (11d) from Algorithm 1, Taylor's Theorem implies that $v^{(t+1)}$ yields an estimator of error order $\mathcal{O}(\tau)$ for the velocity in the next timestep $v^{(t+2)}$:

$$v^{(t+2)} = v^{(t+1)} + \mathcal{O}(\tau). \tag{14}$$

The standard backward distance approximation provides an estimation of $\dot{v}^{(t+1)}$:

$$\dot{v}^{(t+1)} = \frac{v^{(t+1)} - v^{(t)}}{\tau} + \mathcal{O}(\tau). \tag{15}$$

Combining this with a Taylor expansion of $v^{(t+2)}$ then yields the statement from the Theorem:

$$v^{(t+2)} = v^{(t+1)} + \tau\dot{v}^{(t+1)} + \mathcal{O}(\tau^2) = v^{(t+1)} + \tau\frac{v^{(t+1)} - v^{(t)}}{\tau} + \mathcal{O}(\tau^2) = \tag{16}$$

$$2v^{(t+1)} - v^{(t)} + \mathcal{O}(\tau^2) = \bar{v}^{(t+1)} + \mathcal{O}(\tau^2). \tag{17}$$

B Runtime analysis

We compare the runtime of our method to other popular shape interpolation methods based on our experiments on TOSCA in Figure 11. Only divergence-free interpolation [19] is faster than our method. Most importantly, for our approach and [19] the runtime is essentially independent of the resolution because the optimization is done on a fixed resolution of 2k vertices. Only the last forward pass on the whole input shape $p \in \mathbb{R}^{n\times 3}$ depends on the resolution n but this step is cheap in comparison to the optimization.

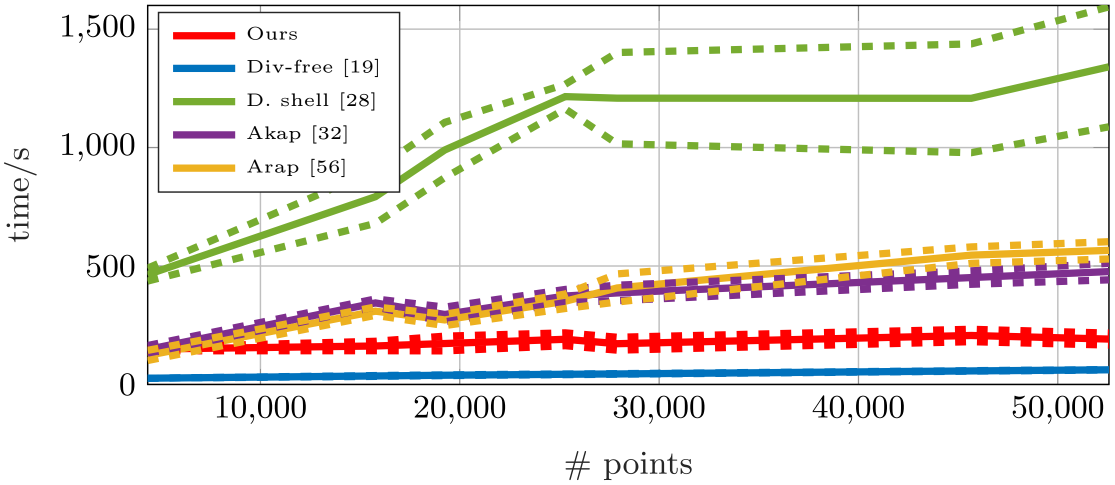


Fig. 11. A runtime analysis of different interpolation methods on all pairs in TOSCA. We plot the mean computation time (solid line) and two lines corresponding to one standard deviation (dashed).

C Additional qualitative evaluations

To give a more complete picture, we show additional examples of interpolations with our method on the two datasets with reals scans, SHREC'19 Isometry and FAUST in Figure 12 and Figure 13. Finally, we show a failure case under topological changes on a scanned hand from SHREC'19 in Figure 14. Although our method is not able to separate the touching parts for these cases, our method still produces a more plausible result than other classical interpolation methods. In our case, the fingers appear to be glued together whereas [28] produces undesirable artifacts.

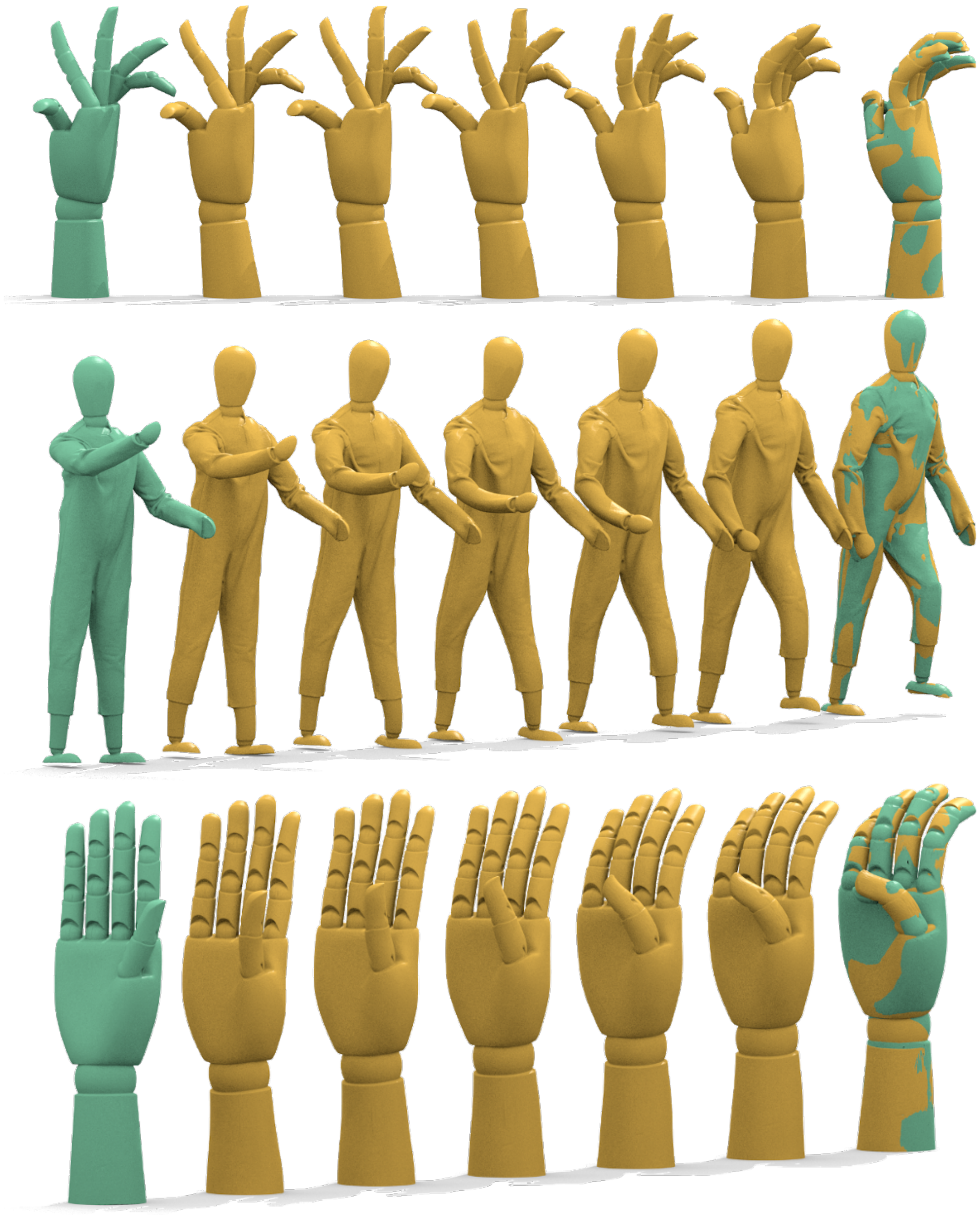


Fig. 12. Additional examples of interpolations with our method on SHREC'19.

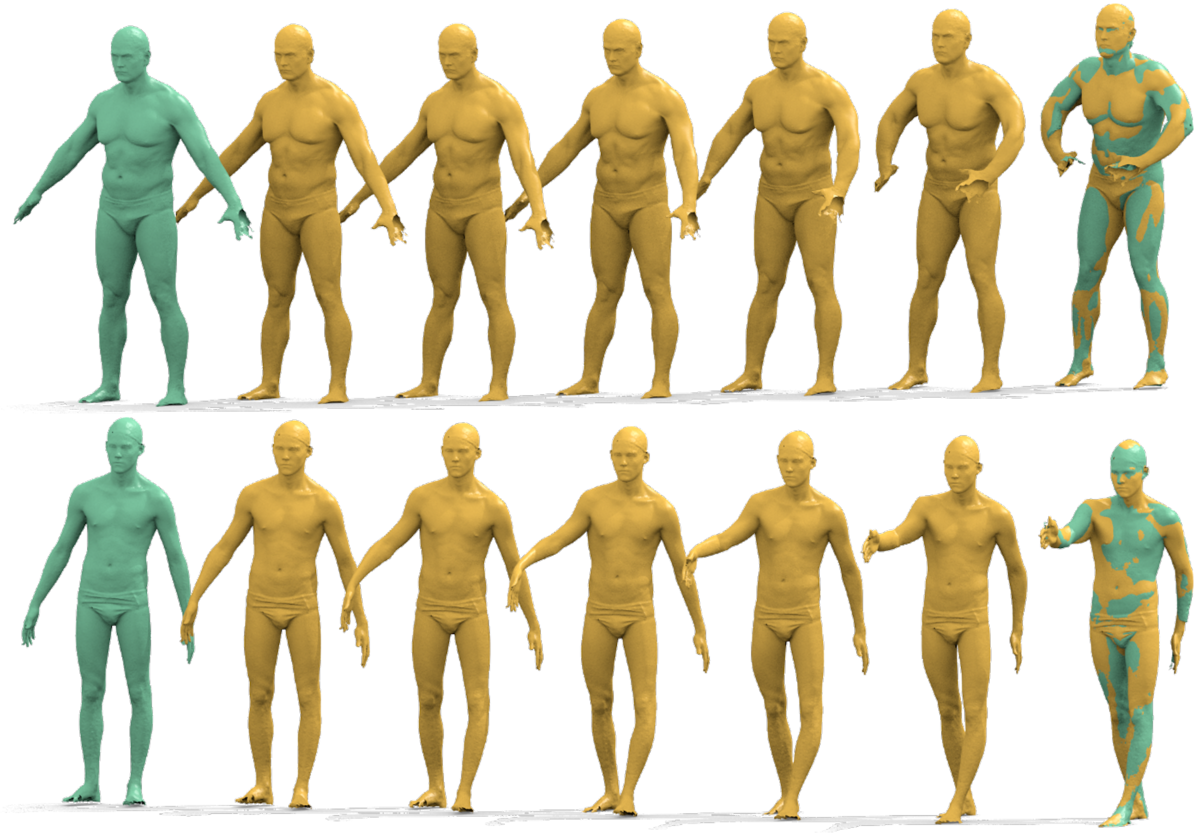


Fig. 13. Additional examples of interpolations with our method on FAUST.

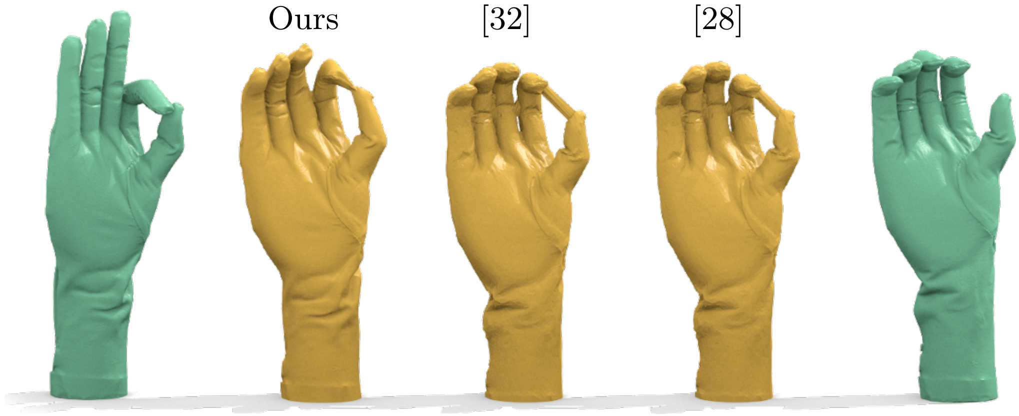


Fig. 14. An example of a scanned hand from SHREC'19 Isometry [17] where we show the second to last frame from our method, [28] and [32] respectively. In SHREC'19 there are various pairs with topological changes. In the case presented here, the meshing connects between the index finger and the thumb. This makes our method fail because the input matching from [20] is not able to separate the two fingers entirely.

Appendix C

Supplementary Material – Deep Shells: Unsupervised Shape Correspondence with Optimal Transport

Deep Shells: Unsupervised Shape Correspondence with Optimal Transport Supplementary Material

Marvin Eisenberger
Technical University of Munich
marvin.eisenberger@in.tum.de

Aysim Toker
Technical University of Munich
aysim.toker@in.tum.de

Laura Leal-Taixé
Technical University of Munich
leal.taixe@tum.de

Daniel Cremers
Technical University of Munich
cremers@tum.de

A Overview

In our experiments, we showed that our method outperforms prior approaches in terms of accuracy, map smoothness, runtime, and the ability to generalize to unseen data. Here, we provide further insights into the proposed architecture and additional qualitative results to give a more complete picture of Deep Shells.

Sec. B presents an ablation study that assesses the role of the different components of our method. In Sec. C we show additional qualitative comparisons. In Sec. D, we provide the cumulative matching curves corresponding to the numbers in Table 1 of the main paper. Finally, in Sec. E, we give more details on the extrinsic-intrinsic product space embedding inspired by smooth shells [4].

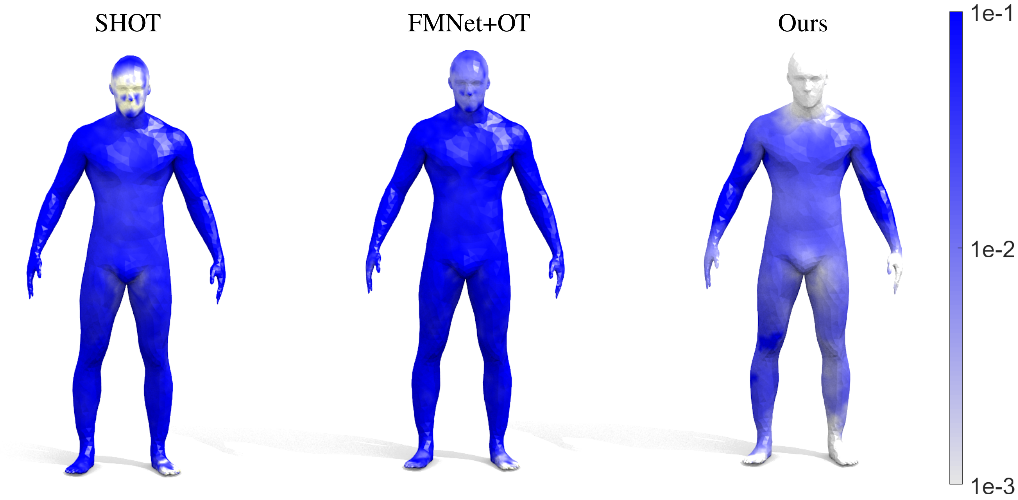


Figure 1: A comparison of the quality of different local features. Here, we show the assignment error averaged over the test set of FAUST remeshed. Besides being more accurate in general, our learned features are in particular able to faithfully distinguish between extremities like hands and feet.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

	Ours	i. SHOT+OT	ii. FMNet+OT	iii. SpecConv+ZO	iv. SpecConv+PMF
FAUST	1.7	4.6	2.0	3.8	6.6
SCAPE	2.5	7.8	4.0	4.4	8.8

Table 1: The results of our ablation study, see Appendix B for more details.

B Ablation study

Deep Shells is comprised of a spectral CNN backbone and a differentiable optimal transport matching layer. Here, we want to assess how these different components contribute to our results in order to get a better understanding of our method. To that end, we replace different parts of our pipeline and report how it affects the geodesic error on FAUST [2] remeshed and SCAPE [1] remeshed, see Table 1 for a summary of the resulting accuracies. In particular, we perform the following ablations:

i. Using SHOT descriptors [10] as inputs to our OT layer instead of learned features.

ii. Replace the spectral convolution backbone with the 7 layer ResNet architecture from FMNet [6] and follow-up work [5, 9], train + evaluate this modified network from scratch.

iii. Postprocessing our deep features with Zoomout [7] ...

iv. ... and with PMF [11] instead of passing them to our OT matching layer.

Additionally, we compare the quality of our obtained features with SHOT (i) and FMNet+OT (ii). To that end, we directly compute nearest-neighbor correspondences from the raw features and show the average error on the first shape of the FAUST remeshed test set, see Figure 1. Finally, we explore how the number of eigenfunctions used during inference time affects the accuracies on FAUST remeshed reported in the main paper, see Fig. 2. On one hand, these results show that our method still achieves state-of-the-art performance for as few as 100 eigenfunctions. Nevertheless, using more high frequency information improves the results, in particular in terms of the local error, quantified by the conformal distortion of individual triangles.

Overall, our ablation study suggests that there is an intricate interplay between our feature extractor and the matching layer. The spectral convolution backbone quantifiably improves over the standard FMNet architecture which computes features independently per vertex. On the other hand, the OT matching layer proved to be an integral part of our pipeline: Even without the spectral CNN backbone, our results are still on par with prior work and postprocessing our deep features with different axiomatic methods impairs the results. Moreover, our OT layer yields the most regular correspondences with the least amount of local distortions, see Figure 3 in the main paper.

C Additional qualitative examples

To give a more complete picture, we show additional qualitative comparisons: For once, we show generalization results from FAUST to the KIDS [8] benchmark in Fig. 3. Although the local features

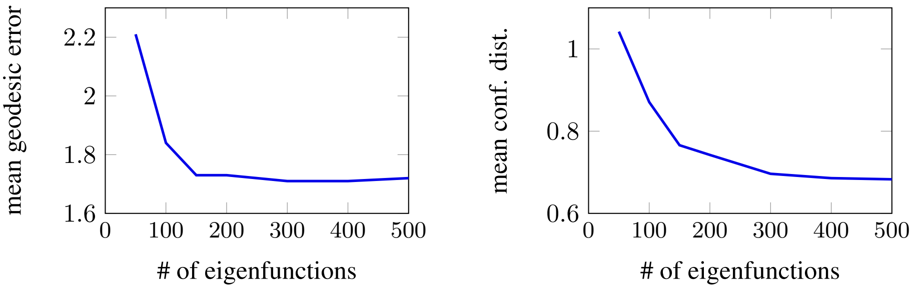


Figure 2: The accuracy of our method using a varying numbers of eigenfunctions during inference time. In particular, we compare the mean geodesic error in % of the shape diameter (left) and the mean conformal distortion (right) averaged over all pairs of the test set of FAUST remeshed.

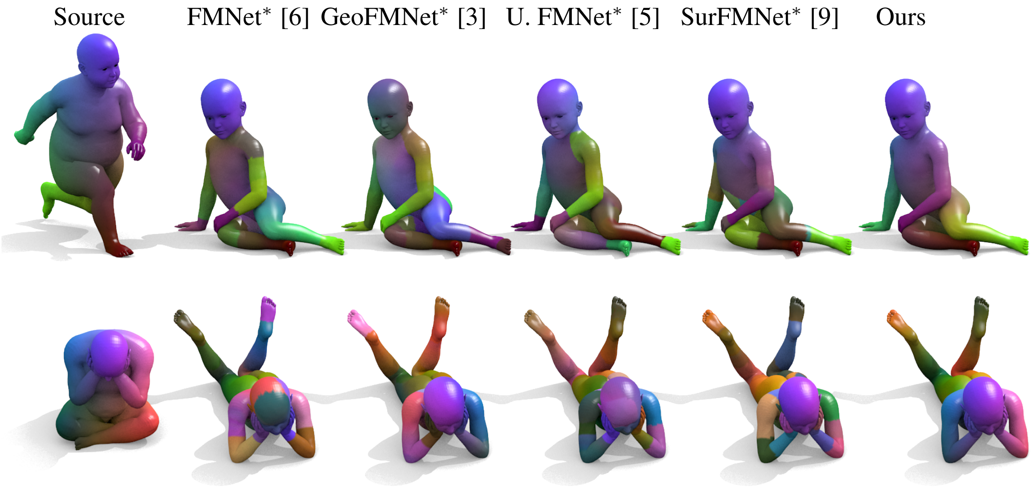


Figure 3: A qualitative comparison with other learning based methods for examples from the KIDS [8] dataset. Methods with a star (*) require postprocessing. For all approaches, we use the weights trained on FAUST from the first column of Table 1 in the main paper.

of these two datasets are similar, KIDS has several poses with self-intersections which leads to noisy SHOT descriptors. Among the methods considered here, only GeoFMNet [3] does not rely on SHOT features as inputs but their point cloud feature extractor is not rotation-invariant. Consequently, it does not generalize well from the shapes in FAUST with mostly standing humans to the examples from KIDS which have a broader variety of poses. Moreover, in Figure 4, we present three more FAUST to SCAPE pairs analogously to the results in Figure 2 of the main paper.

D Matching curves

Here, we provide the cumulative matching curves corresponding to our quantitative results from Table 1 in the main paper, see Figure 5. These curves show the percentage of points below a certain error threshold, the ground truth correspondences would lead to a constant curve at 1.

E More details on smooth shells

In Eq. (3) of the main paper, we defined the $K+6$ dimensional shape embedding which is the basis of our OT matching layer. Here, we provide additional details on this extrinsic-intrinsic embedding. For further details, we refer the interested reader to [4].

The product space embedding $\mathbf{X}_k$, defined in Eq. (3) of the paper, consists of three terms: The intrinsic features Φ_k, the smoothed extrinsic coordinates X_k and the outer normals $\mathbf{n}_k^{\mathcal{X}}$ of X_k. In particular, the intrinsic features are the first k Laplace-Beltrami eigenfunctions on the surface $\mathcal{X}$. The smoothed extrinsic coordinates, on the other hand, are defined as follows:

$$X_k := \sum_{i=1}^{\infty} s_\sigma(k-i)\langle X, \phi_i\rangle_{L_2}\phi_i. \tag{1}$$

Here, $X : \mathcal{X} \to \mathbb{R}^3$ is the original extrinsic embedding function and s_σ is the sigmoid function with a rescaling of the inputs by a scalar $\sigma > 0$. These embeddings $X_k : \mathcal{X} \to \mathbb{R}^3$ now constitute a family of approximations of the input geometry X where k controls the level of detail. For once, small values of k yield a coarse approximation of X and on the other hand $\lim_{k\to\infty} X_k = X$, see Figure 6. This property is now useful for our hierarchical matching layer: We can start with an alignment of coarse approximations of both input surfaces $\mathcal{X}$ and $\mathcal{Y}$ and then gradually increase the level of detail in our alternating optimization scheme, see Figure 1 of the main paper.

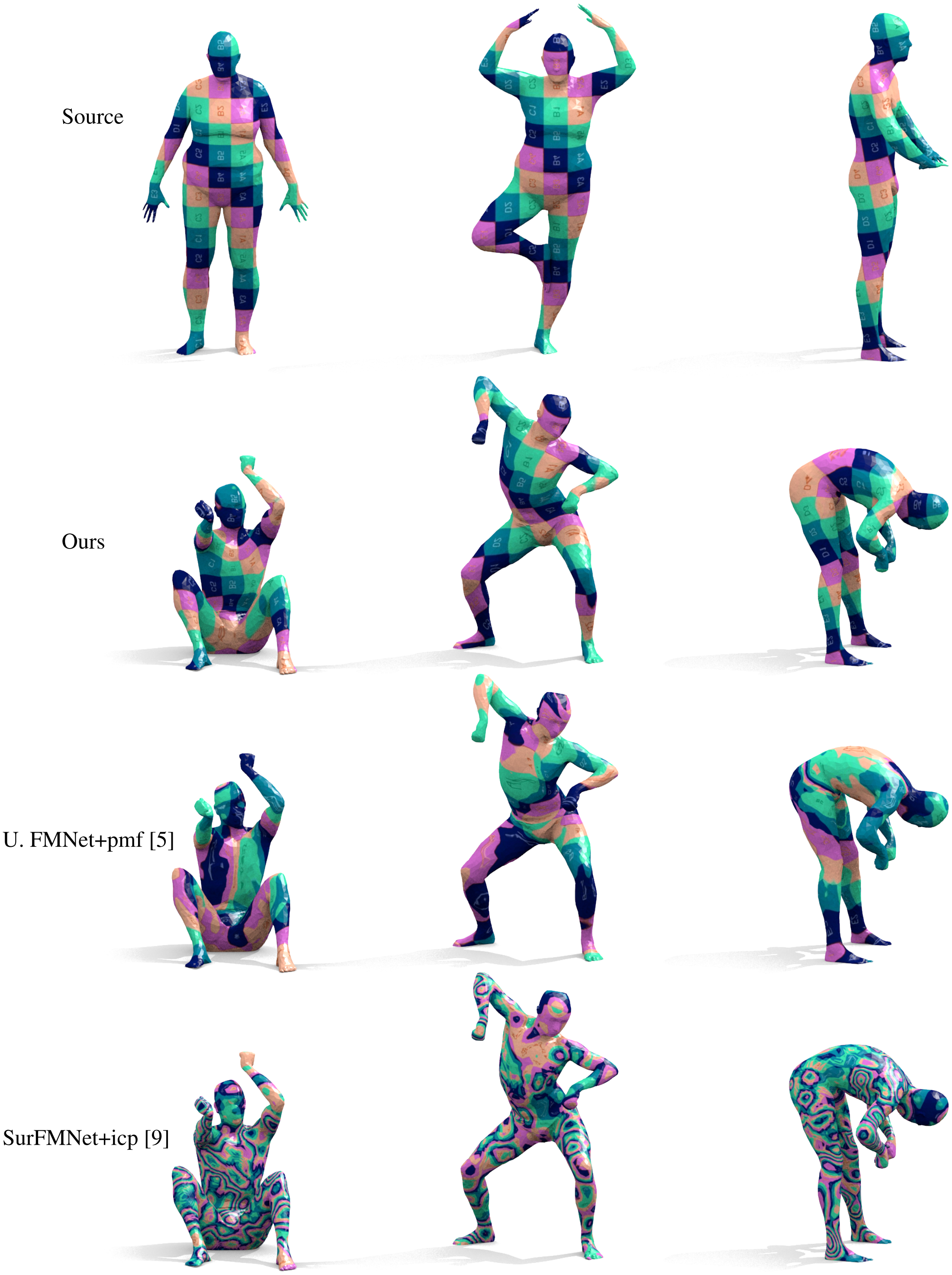


Figure 4: Additional qualitative comparisons corresponding to our inter-dataset experiments in the last two columns of Table 1 in the main paper. In particular, we compute a texturemap from three different FAUST shapes to target shapes from SCAPE and compare our results to [5] and [9].

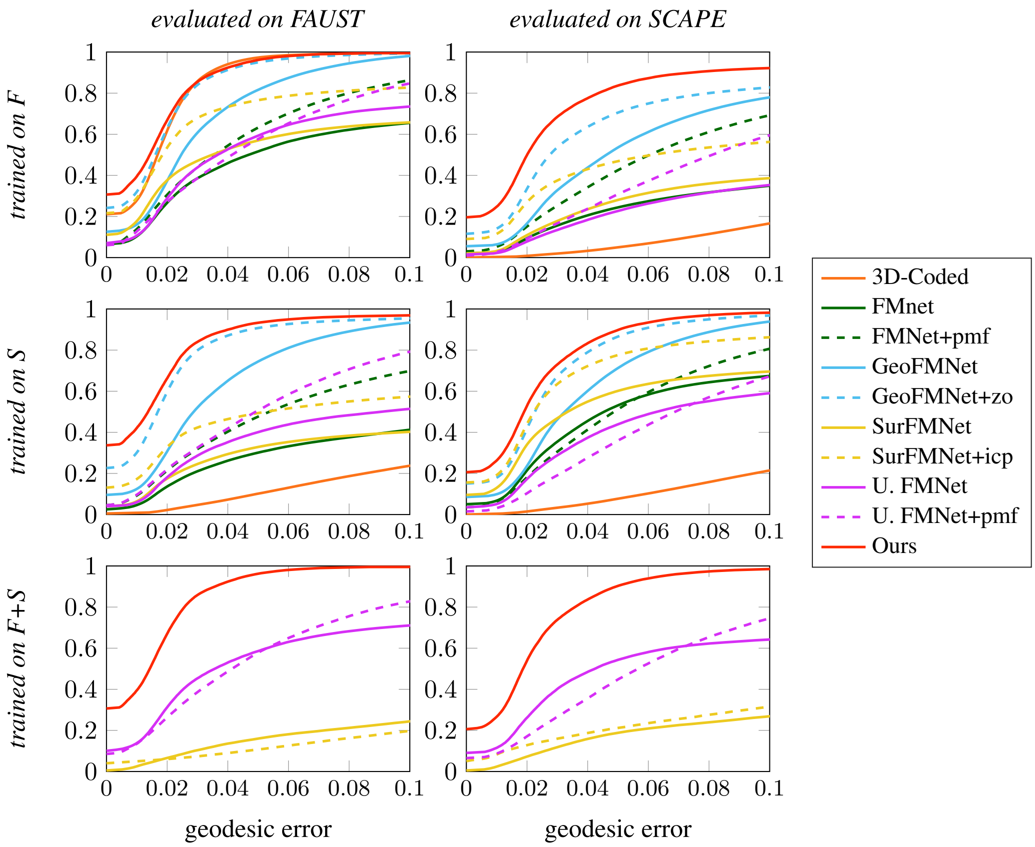


Figure 5: The cumulative geodesic error curves corresponding to the comparison in Table 1 in the paper. The inter-dataset experiments in the last row are only feasible for unsupervised methods because there are no ground-truth labels between FAUST and SCAPE.

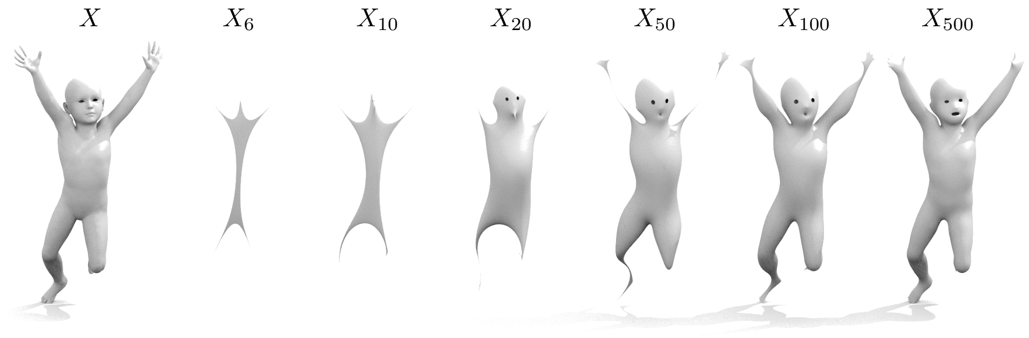


Figure 6: Examples of extrinsic shape approximations X_k for different levels of detail k.

References

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM transactions on graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[2] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA, June 2014. IEEE.

[3] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. *arXiv preprint arXiv:2003.14286*, 2020.

[4] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. *arXiv preprint arXiv:1905.12512*, 2019.

[5] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019.

[6] 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.

[7] 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.

[8] Emanuele Rodolà, Samuel Rota Bulo, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[9] 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.

[10] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. *In Proceedings of European Conference on Computer Vision (ECCV)*, 16(9):356–369, 2010.

[11] Matthias Vestner, Roee Litman, Emanuele Rodolà, Alex M Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Appendix D

Supplementary Material – NeuroMorph: Unsupervised Shape Interpolation and Correspondence in One Go

A. Our dataset G-S-H

In our experiments, we showed quantitative evaluations on multiple benchmarks, including FAUST [2], FAUST remeshed [44], MANO [46], SURREAL [56] and the SHREC20 challenge [10]. These five, as well as many more existing 3D shape datasets, can be roughly classified in two classes: (i) Synthetic datasets with dense ground-truth, near-isometries or a compatible meshing and (ii) real datasets with non-isometric pairs and sparse annotated ground-truth correspondences[4]. In many cases, for (i) the objects are within the same class and therefore have a similar intrinsic geometry, but they undergo challenging, extrinsic deformations with large, non-rigid pose discrepancies. For (ii), the topological proportions of a pair of objects can be quite different, but the poses are less challenging than (i).

To address this disconnect between non-isometries and large non-rigid deformations, in existing benchmarks we create our own dataset, where the goal is to jointly address all the challenges mentioned above: Our benchmark has non-isometric pairs of objects from different classes, large-scale non-rigid poses and dense annotated ground-truth correspondences for evaluation.

A.1. The dataset

We created objects of 3 different classes for our dataset with the tool ZBrush: A dog (Galgo), a cat (Sphynx) and a human. In modeling these shapes, we took great care to obtain generic but anatomically correct instances of these distinct species, see Figure 9 for example shapes from all three classes. We furthermore endowed all objects with a UV-map parameterization, as well as a wireframe acting as a deformation cage. Moreover, the range of motions of one object is specified by a hierarchical set of joints that is consistent for all objects in the dataset. We then animate the different objects by specifying different configurations in terms of deformation handles and applying the deformation to the full shapes with a skinning technique. The UV parameterizations were defined in a way that they are consistent across all considered classes, as a patchwork of smaller components/regions of all objects.

A.2. Experiments

We performed a number of experiments on our new benchmark. For evaluation, we select a number of $\sim$ 120 uniformly sampled keyframes for training and define 32 different poses as our test set. In the main paper, the matching accuracy for our method, as well as other unsupervised matching approaches are compared for this setup. Specifically, we followed the same evaluation protocol that we mentioned earlier in Section 4.1 for the results in Figure 4. Since we have dense ground-truth correspondences that are consistent across all surfaces, we can also display the mean geodesic error at each individual point of the objects. Specifically, we take the UV-map parameterization on one pose of the 'Galgo' shape from our dataset and display the mean matching error of all pairs in the test set. Furthermore, we show qualitative examples of interpolations obtained with our method in Figure 10.

B. Ablation study

We now provide an ablation study where we examine how certain parts of our method contribute to our empirical results. Specifically, we perform the following ablations:

(i) Remove the auxiliary correspondence loss ℓ_{geo}.

(ii) Train for correspondences directly without the interpolator module from our architecture (see Figure 2). This means that we only use ℓ_{geo} and ignore the other two loss components.

(iii) Remove the max-pooling layers in Equation (4) from our architecture.

(iv) Replace the EdgeConv layer in Equation (3) with a standard PointNet [42] layer.

(v) Replace our feature extractor with KPConv[5] [53].

We then report how these changes affect the geodesic error and the mean conformal distortion (interpolation error) on FAUST remeshed, corresponding to the results in Table 1 and Figure 6. Specifically, we compare the results without post-processing:

		Geo. err.	Conf. dist
Ours		**2.3**	**0.10**
(i)	No ℓ_{geo}	13.0	0.13
(ii)	No interp.	4.7	–
(iii)	No maxpool	2.5	0.14
(iv)	EdgeConv	10.6	0.25
(v)	Use KPConv	4.2	0.28

Table 2: Ablations.

These experiments indicate that both the interpolator and the feature extractor are crucial for obtaining high quality results: Modifying technical details of our feature extractor leads to suboptimal results (iii)-(v). The difference is particularly large when EdgeConv is replaced by PointNet

[4] Of course not all existing benchmarks fall under one of these two categories. Some notable exceptions are datasets that specify on a certain class of objects (like humans) [?, 2] or a specific type of input noise (like partiality or topological changes) [45, ?]

[5] KPConv is a state-of-the-art architecture for point cloud learning, but its main emphasis is on tasks like object classification and segmentation. It was, however, used in a matching pipeline before in prior work [9].

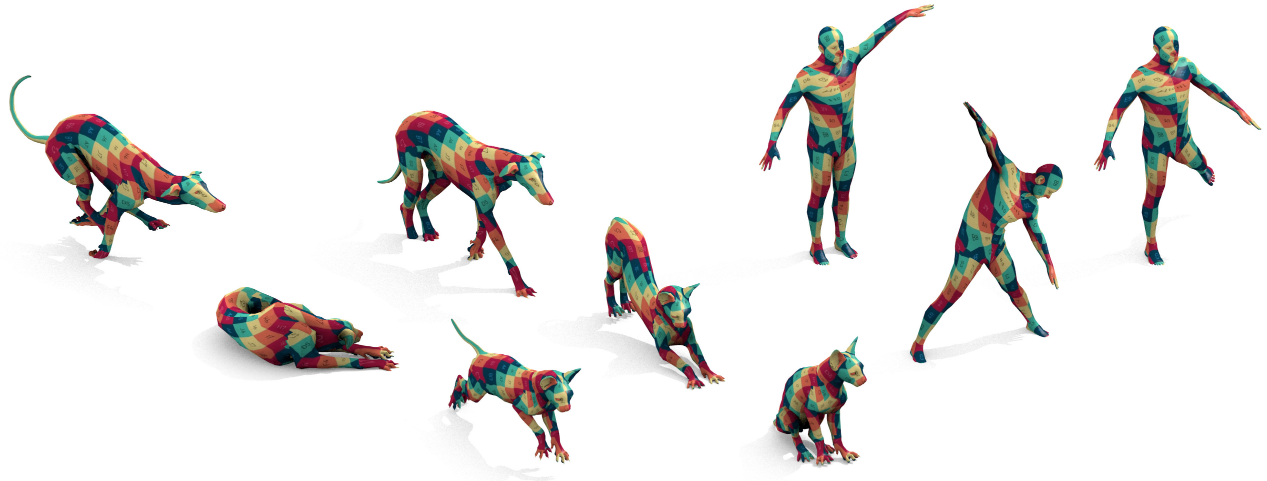


Figure 9: **The G-S-H dataset.** We show 3 examples each for the 3 classes in our G-S-H dataset. Note, that all 3 classes share the same parameterization, despite the varying body proportions. In particular this means that we can obtain dense ground truth correspondences between all pairs of shapes, which we indicate here with a texture map.

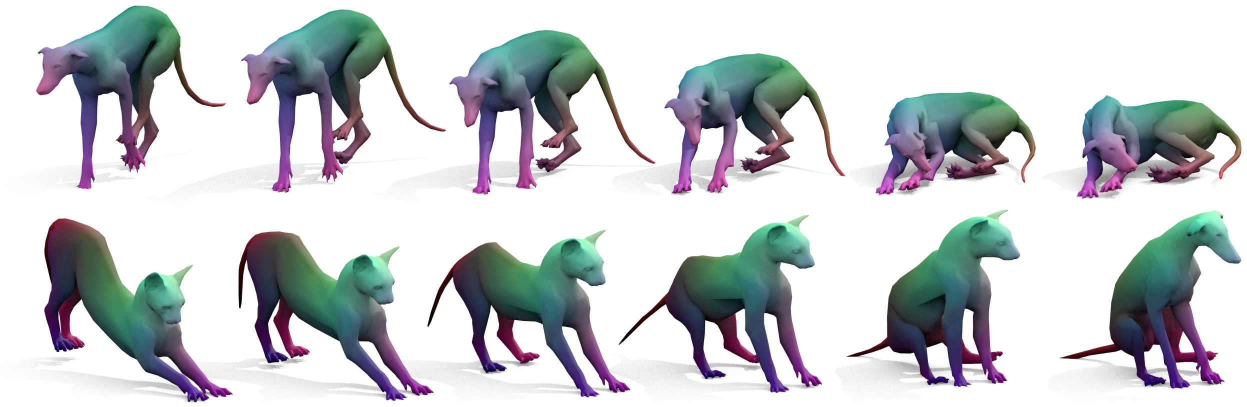


Figure 10: **Interpolation on G-S-H.** Two interpolation sequences on our own benchmark G-S-H obtained with NeuroMorph. This shows clearly that, while our method contains interesting non-isometric pairs, the non-rigid pose variety is still significant.

(iv). Similarly, without the interpolator module, the correspondence estimation is less accurate (ii), since they are not based on an explicit notion of extrinsic deformation. Finally, without the geodesic loss ℓ_{geo}, the matching accuracy deteriorates significantly (i). This can be attributed to the fact that, without a notion of intrinsic geometry, our method is prone to run into unmeaningful local minima.

C. Additional qualitative examples

Finally, we show a few more qualitative results from the SHREC20 benchmark. Specifically, we display examples of non-isometric interpolations in Figure 11 and a qualitative comparison of correspondences obtained with our method and smooth shells [13] in Figure 12.

D. Digital puppeteering

One interesting property of our method is that it is able to learn geometrically plausible pose priors for any shape $\mathcal{X}$. Given any target pose $\mathcal{Y}$, we generally obtain a meaningful new pose of the input object $\mathcal{X}$ as the last pose of the interpolation sequence $t = 1$. Consequently, by considering a distribution of target poses $\mathcal{Y}$, we automatically obtain a shape space of admissible poses with the object identity $\mathcal{X}$. This allows for digital puppeteering as an application of our method. To that end, we jointly train NeuroMorph for a set of poses from the TOSCA dataset of animals and humans, as well as the SURREAL dataset which consists of a large collection of SMPL shapes. As a proof of concept,

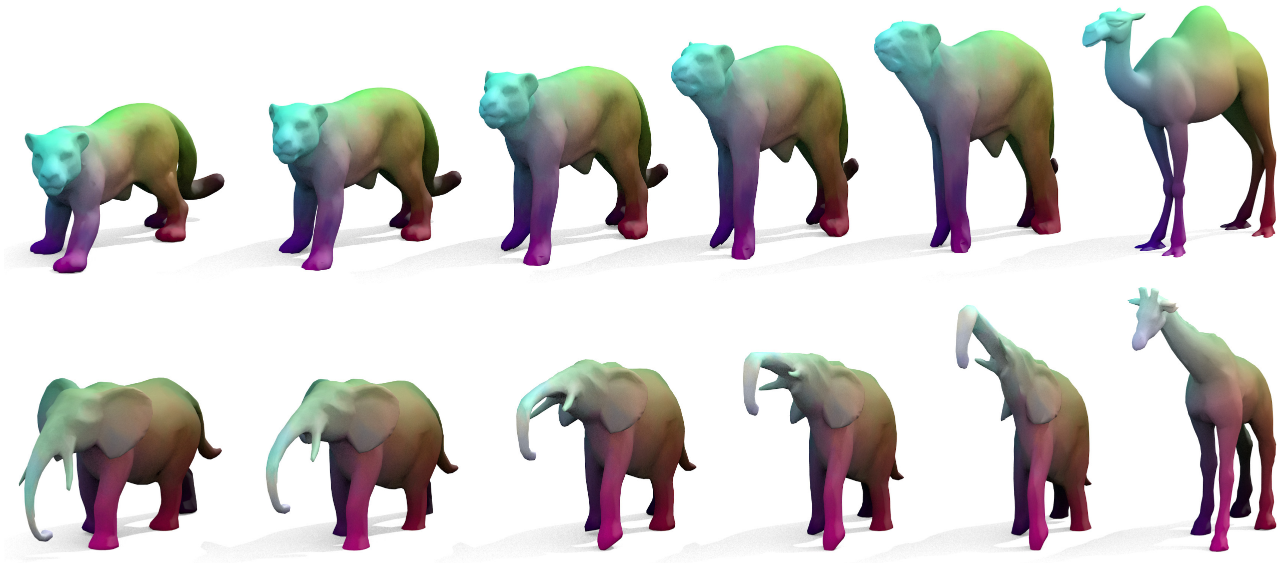


Figure 11: **Interpolation on SHREC20.** We show two additional examples of interpolation sequences obtained with our method for pairs of shapes from the SHREC20 [10] dataset. For each input pair, our method acts on the pose of the first input objects (left) while mostly preserving its identity. The elephant uses its trunk to imitate the shape of the giraffe's head. While this can be considered meaningful from a geometric perspective, it also reveals a limitation of our approach. The fully unsupervised setup occasionally fails to find semantically exact correspondences, if the geometric features have a very different appearance.

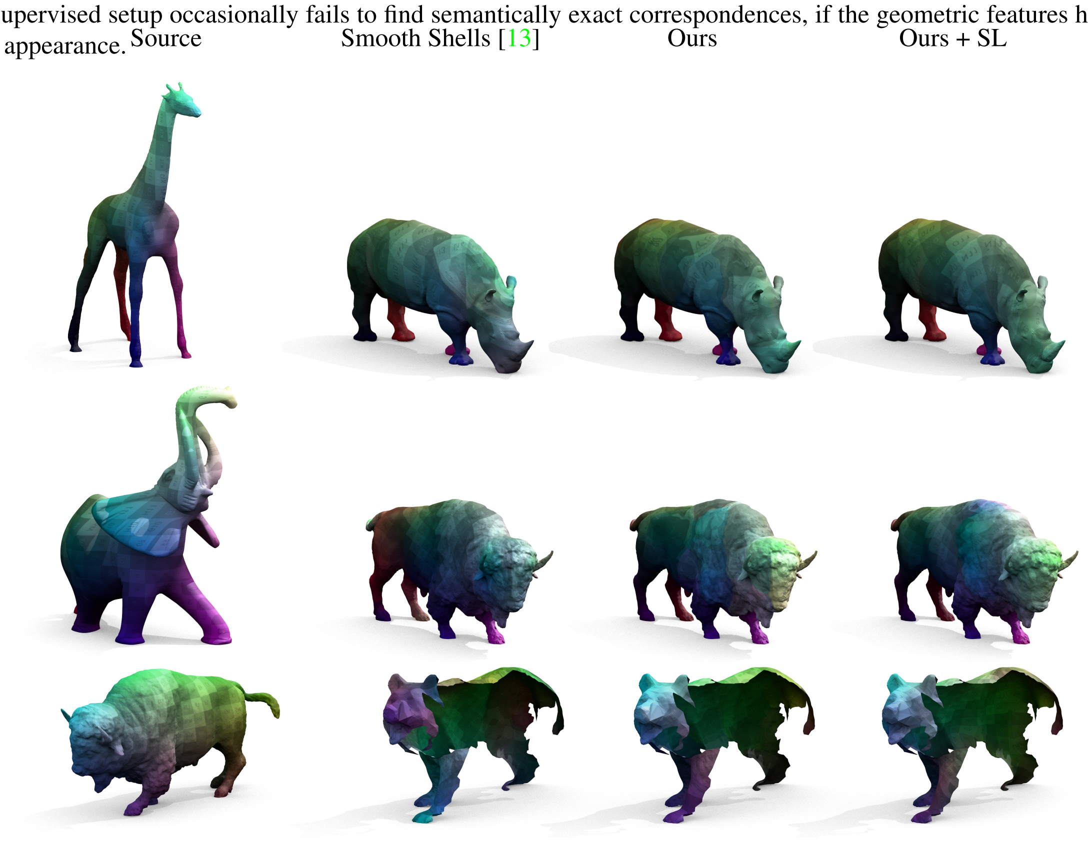


Figure 12: **Unsupervised correspondences on SHREC20.** We show two more qualitative comparisons of correspondences obtained with different methods on the SHREC20 benchmark.

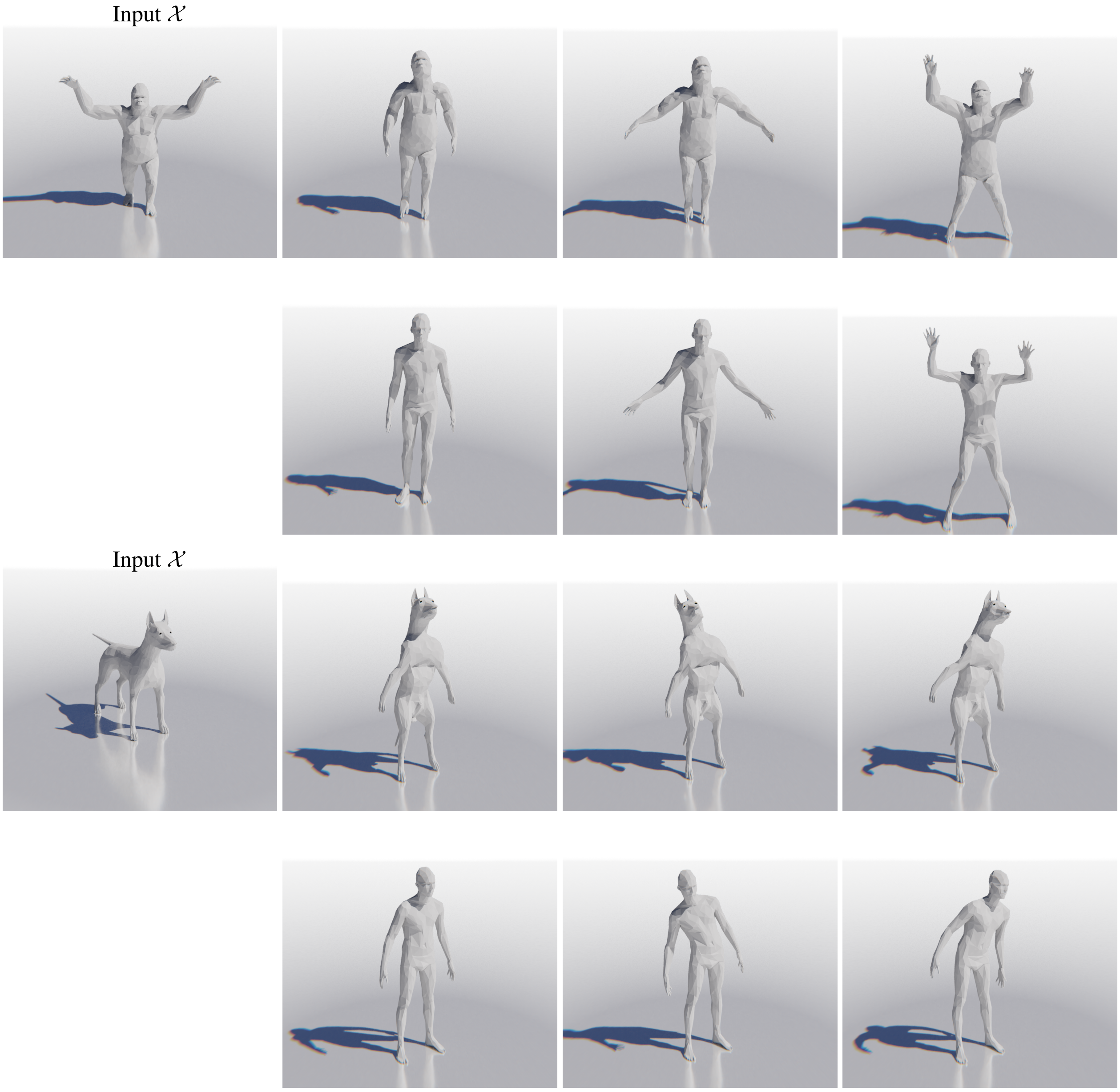


Figure 13: **Digital puppeteering.** We train NeuroMorph jointly for a collection of animal and human (SMPL) shapes. In that manner, we effectively learn a pose prior for the animal shapes which allows us to animate them according to a reference sequence of SMPL shapes from DFAUST. See also our attached videos for the full, animated versions of the two sequences shown here.

we then query our network for a time-continuous sequence of SMPL shapes from the DFAUST dataset and animate the sequence by replacing the human shape with different animals, see Figure 13 and also see our attached videos in the supplementary material.

E. Additional quantitative comparisons

For the sake of completeness, we also provide quantitative comparisons on the SHREC19 [**?**] benchmark, see Figure 14. Note that, like for FAUST, we again use the more recent remeshed version of the dataset, first introduced in [9].

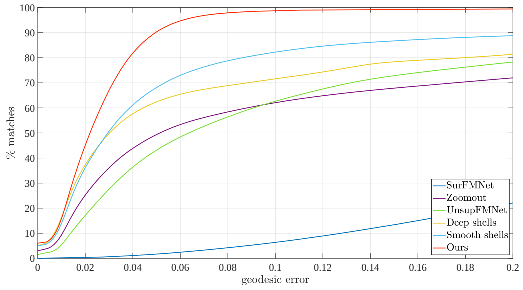


Figure 14: **Unsupervised correspondences on SHREC19 remeshed.** A comparison of unsupervised approaches, showing the cumulative geodesic error curves on the 430 challenge test pairs.

Appendix E

Supplementary Material – G-MSM: Unsupervised Multi-Shape Matching with Graph-based Affinity Priors

A. Implementation details

A.1. Training details

In the following, we provide additional details on our training protocol and choice of parameters. Throughout our experiments, our model was trained on a single NVIDIA Quadro RTX 8000 graphics card with 48GB VRAM.

Data preprocessing For a given shape collection $\mathcal{S}$, we apply a few data standardization steps to ensure training stability. For each $\mathcal{X}^{(i)}$, we normalize the scale of the shape by setting the approximate geodesic diameter to a constant value $\sqrt{\text{area}(\mathcal{X}^{(i)})} = \frac{2}{3}$. The pose is further centered around the origin by setting the mean vertex position to $0 \in \mathbb{R}^3$. The eigenvalues and eigenvectors required for our method are precomputed prior to training our model. Otherwise, our method is directly applicable to any collection of shapes that fulfill the weak pose alignment as discussed in Section 3.1.

Training scheduling Our general training protocol is outlined in Section 3.4 and illustrated visually in Figure 2 of the main paper.

In each forward pass, we first query the DiffusionNet backbone Φ_{feat} to obtain sets of local features $\mathbf{F}^{(i)} := \Phi_{\text{feat}}(\mathcal{X}^{(i)})$ and $\mathbf{F}^{(j)} := \Phi_{\text{feat}}(\mathcal{X}^{(j)})$ for a pair of input shapes $\mathcal{X}^{(i)}$ and $\mathcal{X}^{(j)}$. In the second step, the matching module $\big(\mathbf{\Pi}^{(i,j)}, \mathbf{V}^{(i,j)}, \ell_{\text{match}}^{(i,j)}\big) := \Phi_{\text{match}}\big(\mathbf{F}^{(i)}, \mathbf{F}^{(j)}\big)$ computes a set of putative correspondences $\mathbf{\Pi}^{(i,j)}$, registered vertices $\mathbf{V}^{(i,j)}$ and the corresponding matching loss $\ell_{\text{match}}^{(i,j)}$. Finally, Equation (6) produces the multi-shape correspondences $\mathbf{\Pi}_{\text{mult}}^{(i,j)}$, which then allows us to compute the loss $\ell_{\text{cyc}}^{(i,j)}$ through Equation (7).

As stated in Section 3.4, the shape graph $\mathcal{G}$ is updated regularly after a fixed number of epochs. This interval is chosen in dependence of the number of training shapes as a round figure that results in around $10k - 15k$ training iterations per update. To reduce the computational load, the pairwise correspondences between all pairs of training shapes $\mathbf{\Pi}_{\text{mult}}^{(i,j)}$ are precomputed and stored each time the shape graph is constructed. Additionally, we wait for 5 shape graph update cycles before activating the cycle-consistency loss. This burn-in period allows the feature extractor and putative correspondence modules to converge to a certain degree which facilitates a stable training and reduces stochasticity.

Learning Our model is trained in an end-to-end manner with the Adam optimizer [32], using standard parameters. All the learnable weights are contained in the DiffusionNet backbone **I**. The DeepShells pairwise matching module **II** and multi-matching module **III** convert the learned features into correspondences, but these maps themselves are fully deterministic. The backward pass updates the DiffusionNet weights both in terms of the pairwise alignment loss $\ell_{\text{match}}^{(i,j)}$ and the cycle-consistency loss $\ell_{\text{cyc}}^{(i,j)}$. The former stems from Equation (3) and the latter term is defined in Equation (7). The cycle-consistent correspondences themselves $\mathbf{\Pi}_{\text{mult}}^{(i,j)}$ are obtained from non-differentiable operations, since they are the quantized outputs from DeepShells concatenated via Dijkstra's algorithm in Equation (6a). Instead, the gradients from $\ell_{\text{cyc}}^{(i,j)}$ pass information back to the DeepShells layer **II** through the registrations $\mathbf{V}^{(i,j)}$.

Hyperparameters We set the cycle-consistency loss weight to $\lambda_{\text{cyc}} = 0.5$. The number of latent dimensions of the DiffusionNet encodings is chosen as $l = 128$ and the architecture comprises 4 consecutive DiffusionNet blocks. For the DeepShells matching layer, we directly use the hyperparameters specified by the original publication and the corresponding source code [21]. Specifically, the number of eigenfunctions used to compute the smooth shell product space poses (see Appendix A.2.2) is upsampled on a log-scale between $k_{\min} = 6$ and $k_{\max} = 21$.

A.2. Architecture details

A.2.1 Feature backbone

Our network proposed in Section 3.2 leverages the recent DiffusionNet [57] backbone for local feature extraction. We outline the basic architecture of this module here and refer to [57, Sec. 3] for further technical details.

The core motivation is to model feature propagation of signals $f(x,t)$ on the surface of a shape $\mathcal{X}^{(i)}$ as heat diffusion, governed by the standard heat equation

$$\frac{\partial}{\partial t} f(x,t) = \Delta f(x,t). \tag{9}$$

In this context, Δ is the intrinsic Laplace-Beltrami operator on the surface $\mathcal{X}^{(i)}$. In the discrete case, a common approximation is the cotangent Laplacian $\mathbf{L} := -\mathbf{M}^{-1}\mathbf{S} \in \mathbb{R}^{m \times m}$, where $\mathbf{M}$ and $\mathbf{S}$ are the mass matrix and stiffness matrix, respectively. We further consider the truncated basis of eigenfunctions $\mathbf{\Psi} \in \mathbb{R}^{m \times k}$ and corresponding diagonal matrix of eigenvalues $\mathbf{\Lambda} \in \mathbb{R}^{k \times k}$ of the discretized Laplacian. For a given signal $\mathbf{f} \in \mathbb{R}^m$ on $\mathcal{X}^{(i)}$, the heat propagation for a time-interval $t > 0$ according to Equation (9) then results in the approximate solution

$$H_t(\mathbf{f}) := \mathbf{\Psi} \exp(t\mathbf{\Lambda})\mathbf{\Psi}^{\dagger}\mathbf{f}. \tag{10}$$

For a given feature matrix $\mathbf{F} \in \mathbb{R}^{m \times l}$, an individual operator H_t is applied separately to each channel with different learnable time step weights t. A key benefit of such propagation operators is that they are indifferent to the sampling density and therefore robust to remeshing and local noise. On the other hand, pure heat diffusion is spatially isotropic and therefore not sufficiently expressive. To break radial symmetry, DiffusionNet additionally leverages a gradient-based feature refinement layer. At every point on the surface, it computes inner products between spatial gradients of the (scalar) feature signals on the tangent plane.

Putting everything together, an individual DiffusionNet block takes a set of features $\mathbf{F}$, propagates information both via a spatial diffusion and a spatial gradient layer, and feeds them to a per-point multilayer perceptron (MLP). The first layer is initialized by the input features, defined as the vertex coordinates $\mathbf{V}^{(i)}$. For further technical details, we refer the interested reader to the original publication [57].

A.2.2 Hierarchical pairwise matching

In Section 3.2, we further introduced our differentiable matching layer Φ_{match} based on DeepShells [21]. The final map is fully specified by Equation (3). In the following, we provide additional technical details required to derive the exact optimization steps and compute Φ_{match} in practice.

Following [21, Sec. 3], we first introduce the following latent feature representation of a shape $\mathcal{X}^{(i)}$ that is used within each DeepShells layer

$$\mathbf{F}^{(k)}_{\mathcal{X}^{(i)}} := \Big(\boldsymbol{\Psi}^{(k)}\,,\ S^{(k)}\big(\mathbf{V}^{(i)}\big)\,,\ \mathbf{N}^{(k)}\Big) \in \mathbb{R}^{m\times(k+6)} \tag{11}$$

where $\boldsymbol{\Psi}^{(k)}$ are the first k eigenfunctions of the intrinsic Laplace-Beltrami operator on $\mathcal{X}^{(i)}$, corresponding to the smallest eigenvalues. The operator $S^{(k)}$ is the smoothing map initially proposed in [19, Eq. (8)]. The matrix $\mathbf{N}^{(k)}$ denotes the outer normals of the (smoothed) input geometry.

Overall, the resulting feature tensor $\mathbf{F}^{(k)}_{\mathcal{X}^{(i)}}$ yields a $(k+6)$-dimensional embedding per vertex $\mathbf{V}^{(i)} \in \mathbb{R}^{m\times3}$, depending on the number of eigenfunctions k. In order to align two shapes in this embedding space, an affine transformation is proposed

$$\hat{\mathbf{F}}^{(k)}_{\mathcal{X}^{(i)}}\big(\mathbf{C}^{(k)},\tau^{(k)}\big) := \Big(\boldsymbol{\Psi}^{(k)}\mathbf{C}^{(k)^{\dagger}}\,,\ S^{(k)}\big(\mathbf{V}^{(i)}\big)+ \boldsymbol{\Psi}^{(k)}\tau^{(k)}\,,\ \hat{\mathbf{N}}^{(k)}\Big) \in \mathbb{R}^{m\times(k+6)}, \tag{12}$$

that deforms the input shape $\mathcal{X}^{(i)}$ in the $(k+6)$-dimensional embedding space. This deformation is parameterized with a functional map $\mathbf{C}^{(k)} \in \mathbb{R}^{k\times k}$ [44] and displacement coefficients $\tau^{(k)} \in \mathbb{R}^{k\times3}$. The outer normals of the deformed pose are denoted as $\hat{\mathbf{N}}^{(k)}$.

As we discussed in Section 3.2, the correspondence task in our framework is fully specified by the optimal transport energy Equation (2). To make the resulting update steps differentiable, an additional entropy regularization term is added to the energy

$$E_{\text{match,reg}}\big(\mathbf{F},\mathbf{G};\tilde{\boldsymbol{\Pi}}\big) = E_{\text{match}}\big(\mathbf{F},\mathbf{G};\tilde{\boldsymbol{\Pi}}\big) + \lambda_{\text{ent}}\sum_{i',j'}\tilde{\boldsymbol{\Pi}}_{i',j'}\log\tilde{\boldsymbol{\Pi}}_{i',j'}. \tag{13}$$

This is a common approach that was initially proposed by the seminal work of Cuturi et al. [13]. One compelling implication is that $E_{\text{match,reg}}$ can be minimized efficiently with respect to the transport plans $\tilde{\boldsymbol{\Pi}} \in [0,1]^{m\times n}$ through Sinkhorn's algorithm. Moreover, each individual update step of this algorithm is differentiable which makes it viable for standard gradient-based optimization. The resulting map Φ_{match} is specified by the following alternating scheme of optimization steps

$$\big(\mathbf{C}^{(k)},\tau^{(k)}\big) \mapsto \underset{\substack{\tilde{\boldsymbol{\Pi}}^{(k)}\in\\ \mathcal{T}(\mathcal{X}^{(i)},\mathcal{X}^{(j)})}}{\arg\min}\ E_{\text{match,reg}}\Big(\hat{\mathbf{F}}^{(k)}_{\mathcal{X}^{(i)}},\mathbf{F}^{(k)}_{\mathcal{X}^{(j)}};\tilde{\boldsymbol{\Pi}}^{(k)}\Big), \tag{14a}$$

$$\tilde{\boldsymbol{\Pi}}^{(k)} \mapsto \underset{\mathbf{C}^{(k)},\tau^{(k)}}{\arg\min}\ E_{\text{match,reg}}\Big(\hat{\mathbf{F}}^{(k)}_{\mathcal{X}^{(i)}},\mathbf{F}^{(k)}_{\mathcal{X}^{(j)}};\tilde{\boldsymbol{\Pi}}^{(k)}\Big), \tag{14b}$$

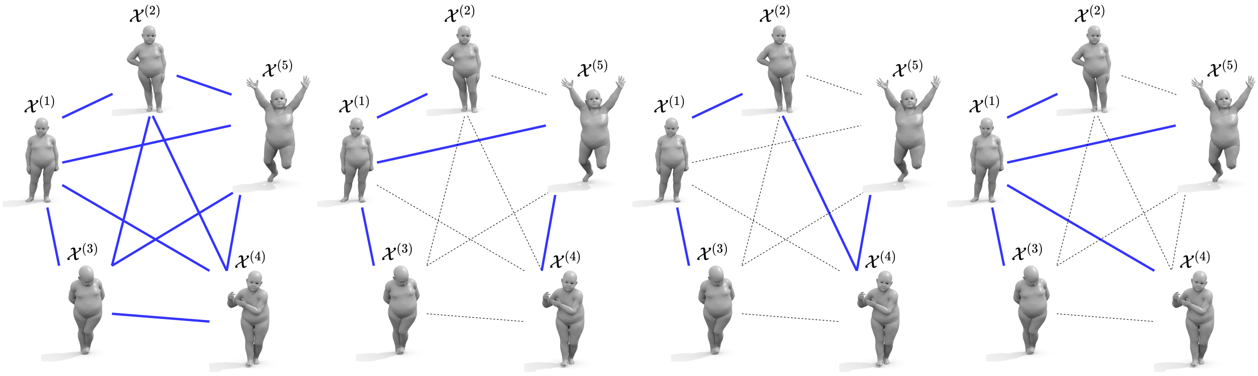
Through this scheme, the minimization of the energy $E_{\text{match,reg}}$ is decoupled into two separate update steps, each of which can be solved efficiently in closed form. The first expression is minimized via Sinkhorn's algorithm to obtain an optimal transport plan $\tilde{\boldsymbol{\Pi}}^{(k)}$ from the transportation polytope $\mathcal{T}(\mathcal{X}^{(i)},\mathcal{X}^{(j)})$. The second update results in a standard linear least squares problem, see [21, Sec. 3] for additional details. For the initial step, we replace the $(k+6)$-dimensional feature embeddings with the learned features $\mathbf{F}^{(i)}$ and $\mathbf{F}^{(j)}$ produced by the DiffusionNet backbone. The map Φ_{match} then alternates between the minimization steps Equation (14a) and Equation (14b) while increasing the number of eigenfunctions k after each step. The final outputs are defined as

$$\boldsymbol{\Pi}^{(i,j)} := \underset{\boldsymbol{\Pi}}{\arg\min}\ E_{\text{match}}\Big(\hat{\mathbf{F}}^{(k_{\max})}_{\mathcal{X}^{(i)}},\mathbf{F}^{(k_{\max})}_{\mathcal{X}^{(j)}}\ ;\ \boldsymbol{\Pi}\Big) \tag{15a}$$

$$\mathbf{V}^{(i,j)} := \mathbf{V}^{(i)} + \boldsymbol{\Psi}^{(k_{\max})}\tau^{(k_{\max})}, \tag{15b}$$

$$\ell^{(i,j)}_{\text{match}} := \sum_{k} E_{\text{match}}\Big(\hat{\mathbf{F}}^{(k)}_{\mathcal{X}^{(i)}},\mathbf{F}^{(k)}_{\mathcal{X}^{(j)}}\ ;\ \tilde{\boldsymbol{\Pi}}^{(k)}\Big). \tag{15c}$$

The matches $\boldsymbol{\Pi}^{(i,j)} \in \{0,1\}^{m\times n}$, with $\boldsymbol{\Pi}^{(i,j)}\mathbf{1}_n = \mathbf{1}_m$, produced by Φ_{match} are thereby the outputs of the final optimization layer $k_{\max}$. In practice, they are obtained as the hard nearest-neighbor assignment between the final obtained shape embeddings $\hat{\mathbf{F}}^{(k_{\max})}_{\mathcal{X}^{(i)}}$ and $\mathbf{F}^{(k_{\max})}_{\mathcal{X}^{(j)}}$.



(i) Full graph.

(ii) MST graph.

(iii) TSP graph.

(iv) Star graph.

Figure 6. An overview of different shape graph topologies $\mathcal{G}$. We consider graphs that are (i) fully connected, (ii) minimal spanning trees, (iii) minimal Hamiltonian paths, specified by the traveling salesman problem and (iv) star graphs centered around one canonical pose. We provide a detailed discussion in Appendix B. Unless stated otherwise, the full graph (i) is the default for our approach.

B. Shape graph topology

We explore the following classes of graph topologies for the shape graph $\mathcal{G}$, see also Figure 6 for a visualization:

i. **Full graph**. The default setting for our method is the fully connected graph with the edge weights defined in Equation (5).

ii. **MST graph**. We consider the minimal spanning tree corresponding to the full graph $\mathcal{G}$. This graph topology is a minimal choice, in the sense that it has the smallest total edge weight among all subgraphs of $\mathcal{G}$ that span the set of nodes $\mathcal{S}$.

iii. **TSP graph**. Based on the traveling salesman problem, we predict a Hamiltonian path of minimal total edge weight. This effectively defines an optimal ordering of the input set $\mathcal{S}$.

iv. **Star graph**. We define a complete, bipartite graph which connects one specific center node with all $N-1$ remaining nodes. The idea is to imitate template-based shape matching methods such as [4, 23] where all training poses are matched to a canonical shape.

Unless stated otherwise, we use the fully connected graph (i) by default in our experiments in Section 4. Choosing the number of retained edges is generally subject to a trade-off between accuracy and efficiency. Using (i) all $\frac{N(N-1)}{2}$ edges often yields the most accurate matching $\mathbf{\Pi}_{\text{mult}}^{(i,j)}$, since this leads to the shortest possible path lengths in Equation (6a). Nevertheless, the sparse graph topologies (ii)-(iv) might be preferable for specific applications. All three definitions (ii)-(iv) specify variants of spanning trees with exactly $N-1$ edges. This means, that the memory complexity for storing the graph, as well as the full query runtime cost is in $\mathcal{O}(N)$, see Appendix C for a cost analysis. Thus, they are more suitable for very large training sets or in scenarios where computational resources are scarce. See Section 4.4 in the main paper for an empirical comparison of the different topologies (i)-(iv).

Discussion Each of the variants (ii)-(iv) has interesting properties that might give rise to potential new avenues of applications in future work: (ii) Removing the (k-1) largest edges in the MST graph yields a subdivision of $\mathcal{G}$ into k optimal clusters (MST clustering). (iii) The TSP graph orders the input shapes into a sequence, i.e., predicts a canonical ordering. (iv) Choosing an optimal star graph automatically selects one of the shapes in the collection $\mathcal{S}$ as the canonical pose. By comparing the set of all possible star graphs, we can in principle rank all input poses in terms of how representative they are of the underlying shape manifold.

C. Empirical computation cost

Training We empirically measure the computation cost of our full pipeline. To this end, we choose a training set of $\{10^2, 20^2, 50^2, 100^2\}$ pairs from the SURREAL [61] dataset with a fixed mesh resolution of 6890 vertices, which is common for SMPL [37] meshes. The resulting training runtime and memory costs are summarized in Table 4. Averaged over all samples, our model takes around $\approx 0.4s$ per training pair. The majority of the cost for constructing the graph $\mathcal{G}$ results from querying all sample pairs, which is equivalent to the epoch training cost minus the backward pass. The remaining cost stems from precomputing the concatenated, pairwise matches as discussed in Appendix A.1.

	#pairs = 10^2	20^2	50^2	100^2
Epoch training time (s)	37.85 ± 2.48	174.82 ± 8.31	921.23 ± 25.56	4192.91 ± 142.03
Graph construction (s)	36.04 ± 1.95	157.30 ± 7.04	893.23 ± 9.05	3814.96 ± 44.48
Required RAM (GB)	3.57 ± 0.02	3.96 ± 0.04	5.81 ± 0.28	9.02 ± 1.73

Table 4. **Empirical training cost.** We quantify the computation cost of our pipeline for different training set sizes. For a given number of shapes $N = |\mathcal{S}|$, one epoch consists of #pairs $= N^2 \in \{10^2, \dots, 100^2\}$ optimization steps that each match a pair of shapes $\mathcal{X}^{(i)}, \mathcal{X}^{(j)} \in \mathcal{S}$.

		#pairs = 10^2	20^2	50^2	100^2
Total query time (s)	Full graph	41.00 ± 3.32	166.91 ± 9.54	1077.06 ± 28.35	4370.36 ± 115.00
	MST graph	3.62 ± 0.29	8.10 ± 0.36	22.48 ± 0.90	50.98 ± 2.57
Graph storage (MB)	Full graph	10.66 ± 0.00	42.64 ± 0.01	266.47 ± 0.08	1065.89 ± 0.33
	MST graph	0.96 ± 0.00	2.03 ± 0.00	5.22 ± 0.00	10.55 ± 0.00

Table 5. **Total query cost.** We report the computation cost of our pipeline at test time. Note, that these results only apply to the specific setting where all pairs of a given set of shapes are queried and the graph $\mathcal{G}$ is *precomputed*. Under these circumstances, the MST graph proves to be superior as its computation cost increases linearly $\mathcal{O}(N)$ in the number of shapes N. For pairwise matching at test time, or when the graph $\mathcal{G}$ needs to be extracted on the fly, the advantages of MST are less prominent.

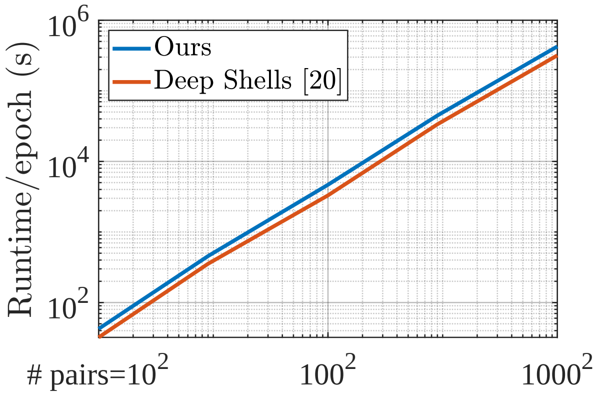


Figure 7. **Total training time** We compare the total training time per epoch of our approach to Deep Shells [21] for different training set sizes on the SURREAL [61] dataset.

Query time We additionally compare the required cost for querying our model. Aside from our main pipeline, we also consider the sparse 'MST' graph type introduced in Appendix B. The resulting costs are summarized in Table 5. For a given set of query shapes, one has to distinguish whether the graph $\mathcal{G}$ is precomputed or needs to be predicted on the fly. In the latter case, an additional cost of constructing the graph is added, see the second row of Table 4. Notably, this cost does not depend on the test graph topology. This also means that the main computational advantages of the MST graph are less prevalent when no offline graph precomputation is possible, e.g., for a pair of unseen test poses. When the unseen pose is supposed to be added to an existing graph in an online fashing, only N pairs between the old training set and the new pose need to be computed. This, however, again entails the same cost for either the full graph or MST. On the other hand, MST is much faster for precomputed graphs. Also, the storage cost of the MST graph is always more efficient than the dense 'full' setting. This makes sparse graph topologies relevant when memory is limited or for very large training sets, since the required memory of the full graph grows quadratically $\mathcal{O}(N^2)$ with the number of training shapes $N = |\mathcal{S}|$.

Comparison to [21] We show a comparison of the total training time per epoch to the baseline approach Deep Shells [21] in Figure 7. The runtime of our approach is on par with [21], while leading to significantly more accurate correspondences, see Table 1. The comparison further demonstrates that, in practice, the training time of both approaches increases quadratically in the number of training shapes. Thus, they feasibly scale to a large training set of 1000^2 training pairs from the SURREAL [61] dataset.

D. Qualitative results

For a more complete picture, we provide several additional qualitative comparisons. Figure 8 and Figure 9 show results corresponding to the benchmark comparisons from Section 4.2 and Section 4.3 of the main paper.

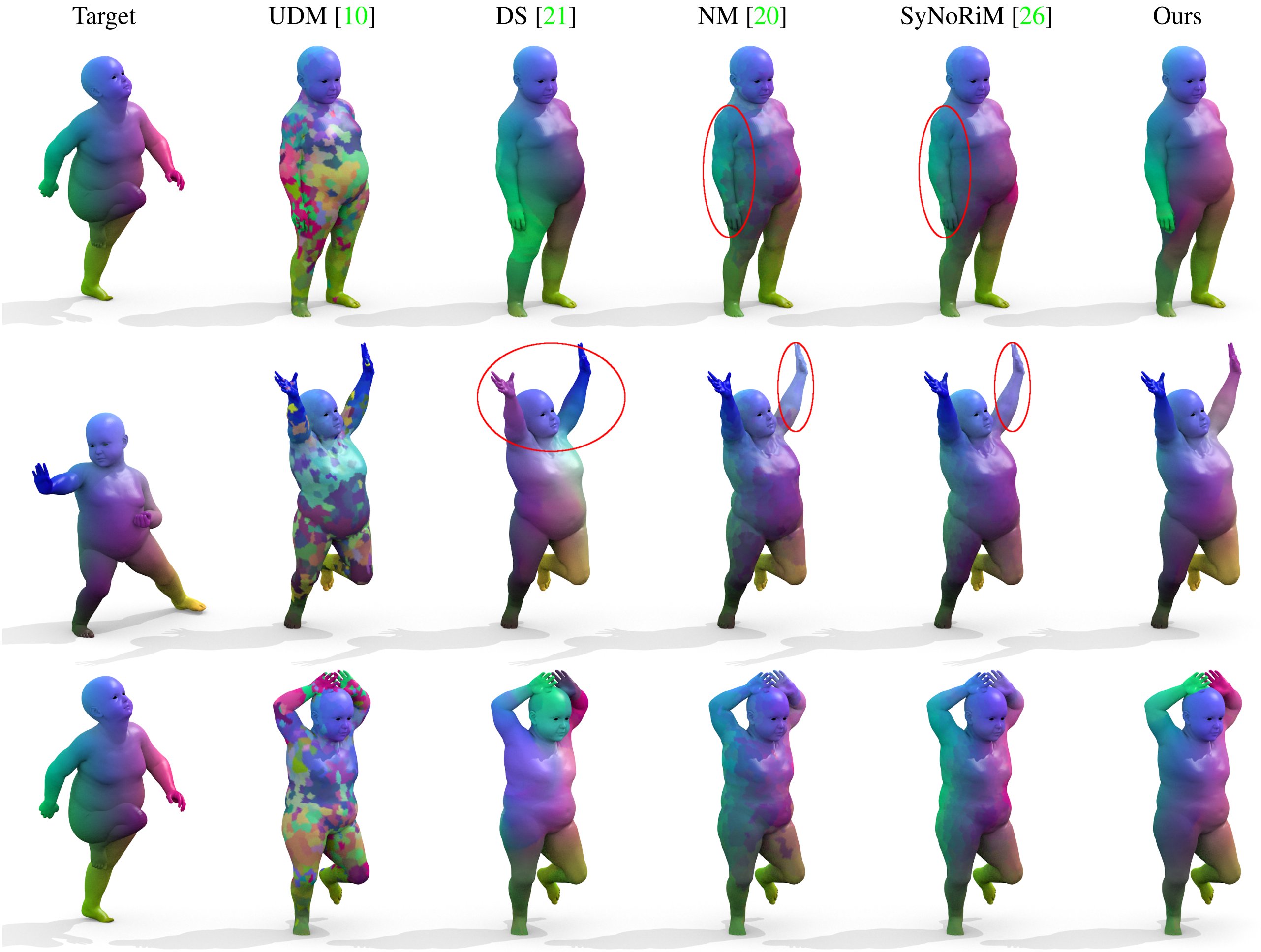


Figure 8. **Topological noise qualitative.** We compare the quality of the predicted correspondences on three pairs from TOPKIDS [33], corresponding to the quantitative results from Figure 3. All three pairs are corrupted by topological noise in places of self-contact, e.g., where the child's arms touch its upper body or head.

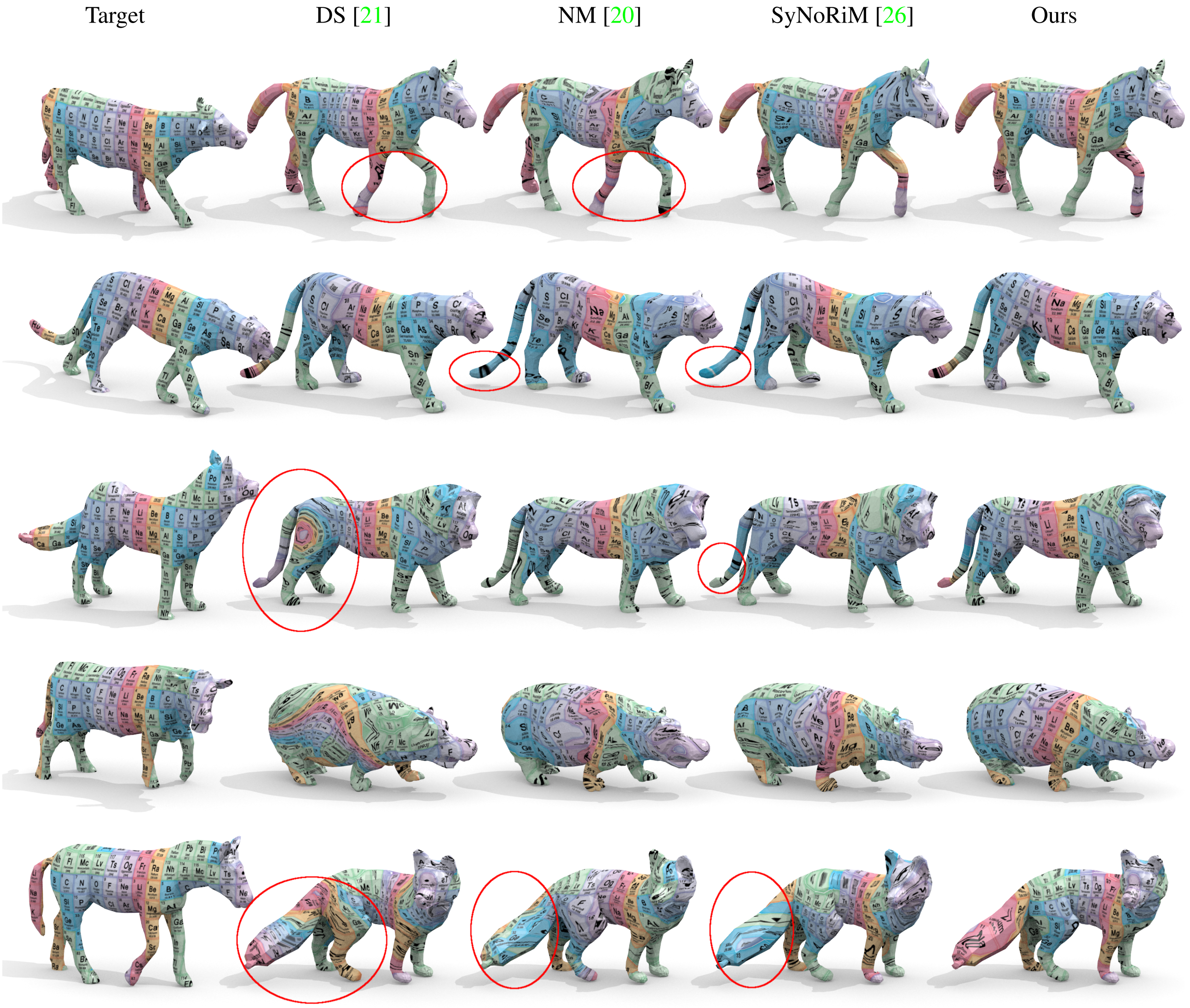


Figure 9. **Inter-class texture transfers.** We assess the map smoothness of several baseline approaches, in comparison to our proposed method. The five sample pairs are taken from the SMAL [65] test set corresponding to our benchmark comparison in Figure 5. In each case, the obtained matches are visualized via a texture map.

List of Figures

1 Introduction

3 Theoretical Background

Core Publications

[1] M. Eisenberger and D. Cremers. Hamiltonian dynamics for real-world shape interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 179–196. Springer International Publishing, 2020 (cited on pp. 6, 9–12, 145–147, 149).

[2] M. Eisenberger, Z. Lahner, and D. 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 (cited on pp. 6, 9–14, 60, 99, 145–147, 149).

[3] M. Eisenberger, Z. Lähner, and D. Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38 of number 5, pages 1–12, 2019 (cited on pp. 6, 9–12, 14, 73, 115, 145–147).

[4] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi. Neuromorph: unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021 (cited on pp. 6, 9–12, 14, 116, 145–147, 149).

[5] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. Deep shells: unsupervised shape correspondence with optimal transport. *Advances in Neural information processing systems*, 33:10491–10502, 2020 (cited on pp. 6, 9–14, 145–147, 149).

[6] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. G-msm: unsupervised multi-shape matching with graph-based affinity priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22762–22772, 2023 (cited on pp. 6, 9–12, 14, 130, 145–147, 149).

Own Publications

[7] V. Ehm, P. Roetzer, M. Eisenberger, M. Gao, F. Bernard, and D. Cremers. Geometrically consistent partial shape matching. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024 (cited on pp. 9, 10, 149).

[1] M. Eisenberger and D. Cremers. Hamiltonian dynamics for real-world shape interpolation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 179–196. Springer International Publishing, 2020 (cited on pp. 6, 9–12, 145–147, 149).

[2] M. Eisenberger, Z. Lahner, and D. 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 (cited on pp. 6, 9–14, 60, 99, 145–147, 149).

[3] M. Eisenberger, Z. Lähner, and D. Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38 of number 5, pages 1–12, 2019 (cited on pp. 6, 9–12, 14, 73, 115, 145–147).

[4] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi. Neuromorph: unsupervised shape interpolation and correspondence in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7473–7483, 2021 (cited on pp. 6, 9–12, 14, 116, 145–147, 149).

[8] M. Eisenberger, A. Toker, L. Leal-Taixé, F. Bernard, and D. Cremers. A unified framework for implicit sinkhorn differentiation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 509–518, 2022 (cited on pp. 9, 10).

[5] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. Deep shells: unsupervised shape correspondence with optimal transport. *Advances in Neural information processing systems*, 33:10491–10502, 2020 (cited on pp. 6, 9–14, 145–147, 149).

[6] M. Eisenberger, A. Toker, L. Leal-Taixé, and D. Cremers. G-msm: unsupervised multi-shape matching with graph-based affinity priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22762–22772, 2023 (cited on pp. 6, 9–12, 14, 130, 145–147, 149).

[9] M. Gao, P. Roetzer, M. Eisenberger, Z. Lähner, M. Moeller, D. Cremers, and F. Bernard. Sigma: scale-invariant global sparse shape matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 645–654, 2023 (cited on pp. 9, 10).

[10] A. Toker, L. Kondmann, M. Weber, M. Eisenberger, A. Camero, J. Hu, A. P. Hoderlein, Ç. Şenaras, T. Davis, D. Cremers, et al. Dynamicearthnet: daily multi-spectral satellite dataset for semantic change segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21158–21167, 2022 (cited on pp. 9, 10).

Bibliography

[11] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018 (cited on p. 31).

[12] R. P. Adams and R. S. Zemel. Ranking via Sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011 (cited on p. 35).

[13] Y. Aflalo, A. Bronstein, and R. Kimmel. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences*, 112(10):2942–2947, 2015 (cited on p. 5).

[14] M. Agueh and G. Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011 (cited on p. 34).

[15] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, pages 408–416, 2005 (cited on pp. 6, 149).

[16] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: a quantum mechanical approach to shape analysis. *IEEE International Conference on Computer Vision (ICCV) - Workshop on Dynamic Shape Capture and Analysis*, 2011 (cited on pp. 17, 24).

[17] S.-Y. Baek, J. Lim, and K. Lee. Isometric shape interpolation. *Computers & Graphics*, 46:257–263, 2015 (cited on p. 16).

[18] F. Bernard, J. Thunberg, P. Swoboda, and C. Theobalt. Hippi: higher-order projected power iterations for scalable multi-matching. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 10284–10293, 2019 (cited on pp. 27, 28).

[19] D. P. Bertsekas. A distributed algorithm for the assignment problem. *Lab. for Information and Decision Systems Working Paper, MIT*, 1979 (cited on p. 34).

[20] P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992 (cited on pp. 20, 23).

[21] B. L. Bhatnagar, C. Sminchisescu, C. Theobalt, and G. Pons-Moll. Loopreg: self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. *Advances in Neural Information Processing Systems*, 33:12909–12922, 2020 (cited on p. 5).

[22] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: dataset and evaluation for 3D mesh registration. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Piscataway, NJ, USA. IEEE, June 2014 (cited on pp. 6, 147, 149).

[23] D. Boscaini, J. Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *Advances in neural information processing systems*, pages 3189–3197, 2016 (cited on pp. 25, 26).

[24] M. Botsch, L. Kobbelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon mesh processing*. CRC press, 2010 (cited on pp. 18, 19).

[25] M. Botsch, M. Pauly, M. H. Gross, and L. Kobbelt. Primo: coupled prisms for intuitive surface modeling. In *Symposium on Geometry Processing*, number CONF, pages 11–20, 2006 (cited on pp. 29, 30).

[26] C. Brandt, C. von Tycowicz, and K. Hildebrandt. Geometric flows of curves in shape space for processing motion of deformable objects. In *Computer Graphics Forum*, volume 35 of number 2, pages 295–305. Wiley Online Library, 2016 (cited on p. 4).

[27] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS*, 103(5):1168–1172, 2006 (cited on p. 16).

[28] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes*. Springer, 2008. http://tosca.cs.technion.ac.il/book/resources_-data.html (cited on pp. 6, 18, 149).

[29] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017 (cited on pp. 5, 16, 24).

[30] T. Caissard, D. Coeurjolly, J.-O. Lachaud, and T. Roussillon. Laplace–beltrami operator on digital surfaces. *Journal of Mathematical Imaging and Vision*, 61:359–379, 2019 (cited on p. 149).

[31] D. Cao and F. Bernard. Unsupervised deep multi-shape matching. In *European Conference on Computer Vision (ECCV)*, 2022 (cited on pp. 27, 28).

[32] D. Cao, P. Roetzer, and F. Bernard. Unsupervised learning of robust spectral shape matching. *arXiv preprint arXiv:2304.14419*, 2023 (cited on p. 27).

[33] I. Chavel. *Eigenvalues in Riemannian geometry*. Academic press, 1984 (cited on pp. 17, 18).

[34] Q. Chen and V. Koltun. Robust nonrigid registration by convex optimization. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2039–2047, 2015 (cited on p. 5).

[35] A. J. Chorin and J. E. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Springer, 1993 (cited on p. 29).

[36] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. 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 (cited on p. 31).

[37] L. Cosmo, A. Norelli, O. Halimi, R. Kimmel, and E. Rodola. Limp: learning latent shape representations with metric preservation priors. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 19–35. Springer, 2020 (cited on pp. 4, 31).

[38] K. Crane, C. Weischedel, and M. Wardetzky. The heat method for distance computation. *Communications of the ACM*, 60(11):90–99, 2017 (cited on p. 17).

[39] M. Cuturi. Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013 (cited on p. 35).

[40] M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In *International conference on machine learning*, pages 685–693. PMLR, 2014 (cited on pp. 32, 34).

[41] M. Cuturi, O. Teboul, and J.-P. Vert. Differentiable ranking and sorting using optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019 (cited on p. 35).

[42] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 317–324, USA. ACM Press/Addison-Wesley Publishing Co., 1999. ISBN: 0201485605. DOI: 10.1145/311535.311576 (cited on p. 17).

[43] M. P. Do Carmo and J. Flaherty Francis. *Riemannian geometry*, volume 6. Springer, 1992 (cited on pp. 15, 16, 18).

[44] N. Donati, A. Sharma, and M. Ovsjanikov. Deep geometric functional maps: robust feature learning for shape correspondence. *arXiv preprint arXiv:2003.14286*, 2020 (cited on pp. 24, 26, 27).

[45] R. M. Dyke, Y.-K. Lai, P. L. Rosin, S. Zappalà, S. Dykes, D. Guo, K. Li, R. Marin, S. Melzi, and J. Yang. Shrec'20: shape correspondence with non-isometric deformations. *Computers & Graphics*, 92:28–43, 2020 (cited on p. 147).

[46] P. Emami and S. Ranka. Learning permutations with Sinkhorn policy gradient. *arXiv preprint arXiv:1805.07010*, 2018 (cited on p. 35).

[47] H. Fan, H. Su, and L. 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 (cited on p. 31).

[48] M. Gao, Z. Lahner, J. Thunberg, D. Cremers, and F. Bernard. Isometric multi-shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14183–14193, 2021 (cited on pp. 16, 27, 28, 149).

[49] D. Griffiths and D. J. Higham. *Numerical Methods for Ordinary Differential Equations. Initial Value Problems.* Springer, 2010 (cited on p. 29).

[50] E. Grinspun, A. N. Hirani, M. Desbrun, and P. Schröder. Discrete shells. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 62–67. Eurographics Association, 2003 (cited on pp. 29, 30, 35).

[51] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning*, pages 3789–3799. PMLR, 2020 (cited on p. 31).

[52] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. A papier-mache approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018 (cited on p. 149).

[53] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. 3d-coded: 3d correspondences by deep deformation. In *The European Conference on Computer Vision (ECCV)*, Sept. 2018 (cited on pp. 5, 26, 31).

[54] O. Halimi, O. Litany, E. Rodola, A. M. Bronstein, and R. Kimmel. Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4370–4379, 2019 (cited on pp. 24, 25, 27).

[55] C. Häne, S. Tulsiani, and J. Malik. Hierarchical surface prediction for 3d object reconstruction. In *2017 International Conference on 3D Vision (3DV)*, pages 412–420. IEEE, 2017 (cited on p. 31).

[56] T. Hawkins. Cauchy and the spectral theory of matrices. *Historia mathematica*, 2(1):1–29, 1975 (cited on p. 18).

[57] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016 (cited on p. 25).

[58] B. Heeren, M. Rumpf, P. Schröder, M. Wardetzky, and B. Wirth. Exploring the geometry of the space of shells. In *Computer Graphics Forum*, volume 33 of number 5, pages 247–256. Wiley Online Library, 2014 (cited on pp. 4, 29).

[59] B. Heeren, M. Rumpf, P. Schröder, M. Wardetzky, and B. Wirth. Splines in the space of shells. *Computer Graphics Forum*, 35(5):111–120, 2016 (cited on pp. 4, 29).

[60] B. Heeren, M. Rumpf, M. Wardetzky, and B. Wirth. Time-discrete geodesics in the space of shells. In *Computer Graphics Forum*, volume 31 of number 5, pages 1755–1764. Wiley Online Library, 2012 (cited on pp. 4, 5, 28, 30).

[61] B. Heeren, C. Zhang, M. Rumpf, and W. Smith. Principal geodesic analysis in the space of discrete shells. In *Computer Graphics Forum*, volume 37 of number 5, pages 173–184. Wiley Online Library, 2018 (cited on p. 29).

[62] J. Huang, T. Birdal, Z. Gojcic, L. J. Guibas, and S.-M. Hu. Multiway non-rigid point cloud registration via learned functional map synchronization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022 (cited on pp. 26–28).

[63] R. Huang, J. Ren, P. Wonka, and M. Ovsjanikov. Consistent zoomout: efficient spectral map synchronization. In *Computer Graphics Forum*, volume 39, pages 265–278. Wiley Online Library, 2020 (cited on pp. 27, 28).

[64] Q.-X. Huang and L. Guibas. Consistent shape maps via semidefinite programming. In *Computer graphics forum*, volume 32 of number 5, pages 177–186. Wiley Online Library, 2013 (cited on p. 28).

[65] P. Huber, R. Perl, and M. Rumpf. Smooth interpolation of key frames in a riemannian shell space. *Computer Aided Geometric Design*, 52:313–328, 2017 (cited on p. 29).

[66] C. Jiang, J. Huang, A. Tagliasacchi, L. Guibas, et al. Shapeflow: learnable deformations among 3d shapes. *arXiv preprint arXiv:2006.07982*, 2020 (cited on pp. 29, 31).

[67] I. Kezurer, S. Z. Kovalsky, R. Basri, and Y. Lipman. Tight relaxation of quadratic matching. In *Computer graphics forum*, volume 34 of number 5, pages 115–128. Wiley Online Library, 2015 (cited on p. 5).

[68] M. Kilian, N. J. Mitra, and H. Pottmann. Geometric modeling in shape space. In *ACM Transactions on Graphics (TOG)*, volume 26 of number 3, page 64. ACM, 2007 (cited on pp. 4, 5, 29, 30).

[69] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013 (cited on pp. 5, 31).

[70] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955 (cited on pp. 5, 34).

[71] Z. Lähner, E. Rodolà, M. M. Bronstein, D. Cremers, O. Burghard, L. Cosmo, A. Dieckmann, R. Klein, and Y. Sahillioglu. Shrec'16: matching of deformable shapes with topological noise. *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2:11, 2016 (cited on p. 149).

[72] E. L. Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963 (cited on pp. 4, 5).

[73] Y. LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998 (cited on p. 34).

[74] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. Learning a model of facial shape and expression from 4D scans. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6):194:1–194:17, 2017 (cited on pp. 5, 31).

[75] O. Litany, T. Remez, E. Rodolà, A. Bronstein, and M. Bronstein. Deep functional maps: structured prediction for dense shape correspondence. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5659–5667, 2017 (cited on pp. 24–27).

[76] R. Litman and A. M. Bronstein. Learning spectral descriptors for deformable shape correspondence. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):171–180, 2013 (cited on p. 25).

[77] L. Liu, D. Campbell, H. Li, D. Zhou, X. Song, and R. Yang. Learning 2d-3d correspondences to solve the blind perspective-n-point problem. *arXiv preprint arXiv:2003.06752*, 2020 (cited on p. 35).

[78] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015 (cited on pp. 5, 31, 147).

[79] R. Marin, S. Melzi, E. Rodolà, and U. Castellani. FARM: functional automatic registration method for 3d human bodies. *CoRR*, abs/1807.10517, 2018 (cited on pp. 5, 147).

[80] R. Marin, M.-J. Rakotosaona, S. Melzi, and M. Ovsjanikov. Correspondence learning via linearly-invariant embedding. *Advances in Neural Information Processing Systems*, 33:1608–1620, 2020 (cited on p. 26).

[81] J. Masci, D. Boscaini, M. Bronstein, and P. Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 37–45, 2015 (cited on pp. 24–26).

[82] S. Melzi, R. Marin, E. Rodolà, and U. Castellani. Matching humans with different connectivity. *Proceedings of Eurographics Workshop on 3D Object Retrieval (3DOR)*, 2019 (cited on pp. 25, 147, 149).

[83] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov. Zoomout: spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155, 2019 (cited on p. 23).

[84] G. Mena, D. Belanger, S. Linderman, and J. Snoek. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018 (cited on p. 35).

[85] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: learning 3d reconstruction in function space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019 (cited on p. 31).

[86] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017 (cited on pp. 25, 26).

[87] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5379–5389, 2019 (cited on p. 29).

[88] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30, 2012 (cited on pp. 5, 16, 17, 21–23).

[89] M. Ovsjanikov, E. Corman, M. Bronstein, E. Rodolà, M. Ben-Chen, L. Guibas, F. Chazal, and A. Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH ASIA 2016 Courses*, pages 1–60. 2016 (cited on p. 21).

[90] G. Pai, J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov. Fast sinkhorn filters: using matrix scaling for non-rigid shape correspondence with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 384–393, 2021 (cited on pp. 32, 35).

[91] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove. Deepsdf: learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019 (cited on p. 31).

[92] B. N. Parlett. *The symmetric eigenvalue problem*, volume 20. siam, 1998 (cited on pp. 18, 22).

[93] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics*, 2(1):15–36, 1993 (cited on p. 19).

[94] A. Poulenard and M. Ovsjanikov. Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018 (cited on p. 25).

[95] C. R. Qi, H. Su, K. Mo, and L. 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 (cited on p. 26).

[96] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017 (cited on p. 26).

[97] J. Ren, M. Panine, P. Wonka, and M. Ovsjanikov. Structured regularization of functional map computations. In *Computer Graphics Forum*, volume 38 of number 5, pages 39–53. Wiley Online Library, 2019 (cited on p. 23).

[98] E. Rodola, A. M. Bronstein, A. Albarelli, F. Bergamasco, and A. Torsello. A game-theoretic approach to deformable shape matching. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 182–189. IEEE, 2012 (cited on p. 5).

[99] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial functional correspondence. In *Computer graphics forum*, volume 36 of number 1, pages 222–236. Wiley Online Library, 2017 (cited on pp. 23, 149).

[100] P. Roetzer, P. Swoboda, D. Cremers, and F. Bernard. A scalable combinatorial solver for elastic geometrically consistent 3d shape matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 428–438, 2022 (cited on p. 20).

[101] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*. 245:1–245:17, 36(6), Nov. 2017 (cited on pp. 30, 31).

[102] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov. Unsupervised deep learning for structured shape matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1627, 2019 (cited on pp. 25, 27).

[103] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009 (cited on p. 24).

[104] Y. Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*:1–17, 2019 (cited on p. 4).

[105] R. Santa Cruz, B. Fernando, A. Cherian, and S. Gould. Deeppermnet: visual permutation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3949–3957, 2017 (cited on p. 35).

[106] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. Superglue: learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020 (cited on p. 35).

[107] A. Sharma and M. Ovsjanikov. Weakly supervised deep functional map for shape matching. *arXiv preprint arXiv:2009.13339*, 2020 (cited on p. 26).

[108] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. Diffusionnet: discretization agnostic learning on surfaces. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022 (cited on pp. 24, 26).

[109] N. Sharp and K. Crane. A laplacian for nonmanifold triangle meshes. In *Computer Graphics Forum*, volume 39 of number 5, pages 69–80. Wiley Online Library, 2020 (cited on pp. 26, 149).

[110] Z. Shen, J. Feydy, P. Liu, A. H. Curiale, R. San Jose Estepar, R. San Jose Estepar, and M. Niethammer. Accurate point cloud registration with robust optimal transport. *Advances in Neural Information Processing Systems*, 34:5373–5389, 2021 (cited on p. 32).

[111] B. R. Sheth and R. Young. Two visual pathways in primates based on sampling of space: exploitation and exploration of visual information. *Frontiers in integrative neuroscience*, 10:37, 2016 (cited on p. 3).

[112] J. Solomon. Optimal transport on discrete domains. *AMS Short Course on Discrete Differential Geometry*, 2018 (cited on p. 33).

[113] J. Solomon, M. Ben-Chen, A. Butscher, and L. Guibas. As-killing-as-possible vector fields for planar deformation. In *Computer Graphics Forum*, volume 30 of number 5, pages 1543–1552. Wiley Online Library, 2011 (cited on pp. 29, 30).

[114] J. Solomon, K. Crane, and E. Vouga. Laplace-beltrami: the swiss army knife of geometry processing. In *Symposium on Geometry Processing Graduate School (Cardiff, UK, 2014)*, volume 2, 2014 (cited on p. 17).

[115] J. Solomon, F. De Goes, G. Peyré, M. Cuturi, A. Butscher, A. Nguyen, T. Du, and L. Guibas. Convolutional wasserstein distances: efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (ToG)*, 34(4):1–11, 2015 (cited on pp. 32, 34, 35).

[116] O. Sorkine. Laplacian mesh processing. *Eurographics (State of the Art Reports)*, 4(4), 2005 (cited on p. 17).

[117] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry processing*, volume 4, pages 109–116, 2007 (cited on pp. 29, 30).

[118] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184, 2004 (cited on p. 17).

[119] M. D. Spivak. A comprehensive introduction to differential geometry. *(No Title)*, 1999 (cited on p. 29).

[120] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multiscale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009 (cited on pp. 17, 24).

[121] R. Sundararaman, G. Pai, and M. Ovsjanikov. Implicit field supervision for robust non-rigid shape matching. In *European Conference on Computer Vision*, pages 344–362. Springer, 2022 (cited on p. 149).

[122] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. Kpconv: flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019 (cited on p. 26).

[123] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. *In Proceedings of European Conference on Computer Vision (ECCV)*, 16(9):356–369, 2010 (cited on pp. 24, 25).

[124] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011 (cited on p. 4).

[125] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid. Learning from synthetic humans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2017 (cited on p. 147).

[126] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017 (cited on p. 31).

[127] M. Vestner, R. Litman, E. Rodolà, A. M. Bronstein, and D. Cremers. Product manifold filter: non-rigid shape correspondence via kernel density estimation in the product space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017 (cited on pp. 5, 24).

[128] C. Villani. *Topics in optimal transportation*. American Mathematical Soc., 2003 (cited on p. 32).

[129] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018 (cited on p. 31).

[130] X. Wang. *Learning and reasoning with visual correspondence in time*. PhD thesis, Carnegie Mellon University, 2019 (cited on p. 4).

[131] T. Windheuser, U. Schlickwei, F. R. Schimdt, and D. Cremers. Large-scale integer linear programming for orientation preserving 3d shape matching. In *Computer Graphics Forum*, volume 30, pages 1471–1480. Wiley Online Library, 2011 (cited on p. 20).

[132] B. Wirth, L. Bar, M. Rumpf, and G. Sapiro. A continuum mechanical approach to geodesics in shape space. *International Journal of Computer Vision*, 93:293–318, 2011 (cited on pp. 4, 5).

[133] D. Xu, H. Zhang, Q. Wang, and H. Bao. Poisson shape interpolation. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling*, pages 267–274, 2005 (cited on p. 17).

[134] G. Yang, S. Belongie, B. Hariharan, and V. Koltun. Geometry processing with neural fields. *Advances in Neural Information Processing Systems*, 34:22483–22497, 2021 (cited on p. 149).

[135] L. Yang, W. Liu, Z. Cui, N. Chen, and W. Wang. Mapping in a cycle: Sinkhorn regularized unsupervised learning for point cloud shapes. In *European Conference on Computer Vision*, pages 455–472. Springer, 2020 (cited on p. 35).

[136] Y. Yang, C. Feng, Y. Shen, and D. Tian. Foldingnet: point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–215, 2018 (cited on p. 31).

[137] Z. J. Yew and G. H. Lee. Rpm-net: robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11824–11833, 2020 (cited on pp. 32, 35).

[138] L. Younes. *Shapes and diffeomorphisms*, volume 171. Springer, 2010 (cited on p. 4).

[139] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black. 3d menagerie: modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017 (cited on pp. 5, 31).