

Thèse de doctorat

NNT : 2024IPPA043

INSTITUT
POLYTECHNIQUE
DE PARIS



Robust Spectral Methods for Shape Analysis and Deformation Assessment

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à École polytechnique

École doctorale n°626 École doctorale de l'Institut Polytechnique de Paris (EDIPP)
Spécialité de doctorat : Informatique, Données et Intelligence Artificielle

Thèse présentée et soutenue à Palaiseau, le 4 Juillet 2024, par

ROBIN MAGNET

Composition du Jury :

| | |
|---|------------------------|
| Mohamed Daoudi Professeur, Université Lille (CRIStAL) | President [Rapporteur] |
| Florian Bernard Professeur, Universität Bonn | Rapporteur |
| Mario Botsch Professeur, TU Dortmund | Examinateur |
| Omri Azencot Assistant Professor, Ben-Gurion University of the Negev | Examinateur |
| Maks Ovsjanikov Professeur, Ecole Polytechnique (LIX) | Directeur de thèse |

Abstract

Automatically processing and analyzing 3D shapes is an active area in modern research with implications in various fields. A key challenge in shape analysis lies in efficiently comparing shapes, for example to detect abnormalities in scans of organs, which often requires automatically deforming one shape into another, or establishing correspondences between surfaces. In this context, the functional map framework, based on spectral shape analysis, offers a flexible approach to representing and computing these correspondences, serving as a foundation for subsequent analysis. Unfortunately, despite its flexibility, this framework has seen limited applications outside of computer graphics, as some properties such as the smoothness of correspondences or the scalability of the algorithms to real-world shapes have not been explicitly tackled.

This thesis seeks to address the limitations of existing spectral methods, with the ultimate goal to achieve robust and efficient shape comparisons applicable to real-world data. In the first part, we concentrate on assessing deformations between shapes effectively. Leveraging recent advancements in functional maps, we introduce a descriptor of differences between shapes, capturing information about the distortion around each point, without explicitly deforming the shape. Next, we apply similar tools on a set of skull scans for craniofacial disease detection, highlighting the specific requirements of shape matching practitioners. Notably, we underscore the significance of correspondence smoothness and scalability to dense meshes, often overlooked in shape correspondence research.

In the second part, we address these needs by extending existing functional map methods. Firstly, we introduce a novel shape correspondence pipeline, which explicitly promotes smoothness of computed correspondences, alongside a new challenging shape matching dataset. Secondly, we focus on enhancing the scalability of functional map pipelines to handle real-world dense meshes. For this, we present an approximation of the functional map, enabling the computation of correspondences on meshes with hundreds of thousands of vertices in a fraction of the processing time required by standard algorithms. Finally, we introduce a new learning-based approach, by modifying existing techniques for functional map computations, eliminating the need for large dense matrix storage in GPU memory, thereby improving scalability and numerical stability.

Overall, our work contributes efficient tools for analyzing differences between shapes and provides general methods to simplify and accelerate correspondence computations, facilitating downstream applications.

Remerciements

J'aimerais tout d'abord remercier mes relecteurs Mohamed Daoudi et Florian Bernard pour avoir accepté de relire méticuleusement ce manuscrit, ainsi que Mario Botsch et Omri Azencot pour avoir accepté de faire parti de mon jury. Je remercie particulièrement mon directeur de thèse, Maks Ovsjanikov, pour son aide et ses conseils constamment pertinents depuis maintenant quatre ans, qui date le début de mon stage de fin d'études. Merci également à tous les chercheurs et chercheuses avec lesquels j'ai pu collaborer au cours de cette thèse : Jing Ren, Olga Sorkine-Hornung, Kevin Bloch, Maxime Taverne, Roman Hossein Khonsari, Simone Melzi et Maya Geoffroy. Merci à Mazdak Abulnaga, pour le projet en cours qui n'apparaît pas dans ce manuscrit, ainsi que pour ce mois passé à Paris. Je souhaite aussi remercier Jean Feydy et Etienne Corman avec qui les discussions m'ont permis de découvrir d'autres facettes de la recherche autour de la vision 3D.

Après cinq années sur le plateau de Saclay, entre études et thèse, me voilà enfin prêt à partir. Malgré ma frilosité à faire le trajet, je remercie tous ceux qui ont rendu toutes mes visites au LIX très agréables, et qui m'ont permis de rester motivé. Je ne remercie donc pas le RER B ou le 91-06. En revanche, un merci particulier aux FMappers Ramana et Maxime, avec qui j'ai partagé des bureaux, et Nicolas et Souhaib pour ces longues discussions sur la recherche. Merci aux thésards plus récents, Souhail, Tim, Leopold et Diego pour la nouvelle dynamique que vous apportez et votre bonne humeur, ainsi qu'aux postdocs Gautam, Lei, Roman, Maysam et Emery. Un merci particulier à Vincent pour ta motivation et les discussions autour de mon futur. Au cours de ma thèse, j'ai passé quelques mois en stage à Pittsburgh, et je remercie toutes les personnes rencontrées là-bas, en particulier Fabian Prada qui m'a contacté pour m'offrir cette opportunité. Thanks to Shoe-Yang, Mia, Cartem, Shahrukh, Chao and all other oompa loompas for all these times, from Costco tourism to dying from pizza. A very particular thanks for my Modern American Family, I had unbelievable fun with y'all, I hope this friendship can last long.

Merci évidemment à tous ceux qui ont été là au cours de ces trois ans. À Martin, pour ... Euh ... Tout ? Les discussions, les bières, les soirées, la colloc, la motivation, le sport, enfin vraiment tout ce qui m'a permis d'être bien. Merci infiniment à Julie, sûrement la seule personne à avoir dû faire face à ma fatigue et mon stress et qui m'a constamment soutenu, depuis le tout premier jour de stage. Merci aux C, disons à la Colloc, Agathe, le Tids, Gab, la Tong, le Moums et Thomas (ordre alphabétique, pas de jaloux), même si on arrive à se réunir qu'une fois par an. Merci à Jean et

Anouk (ou Anouk et Jean), à Khalil, à Arya et à Élise présents depuis la prépa, à Hugo et Tom depuis le collège. Il est difficile de citer tout le monde, mais je remercie aussi la Clermonterie, Florian et les bobos pour ces soirées partagées. Finalement, merci à ma famille, à mes frères Théo et Loris, à mes parents, qui ont rendu chaque période de ma vie si simple à traverser. Vous m'avez transmis peut-être plus que vous ne l'imaginez.

Au cas où certains d'entre vous n'auraient pas bien compris le message, je vous l'écris une dernière fois : Merci.

Contents

| | |
|---|-----------|
| Remerciements | ii |
| 1 Introduction | 1 |
| 1.1 Context | 3 |
| 1.1.1 Shape Analysis and Shape Matching | 3 |
| 1.1.2 Shape Matching Approaches | 5 |
| 1.1.3 Limitations of existing approaches | 7 |
| 1.2 Contributions | 8 |
| 1.3 Publication Record | 9 |
| 2 Introduction en français | 11 |
| 2.1 Contexte | 13 |
| 2.1.1 Analyse de Formes et Mise en Correspondance | 13 |
| 2.1.2 Méthodes de Mise en Correspondence | 16 |
| 2.1.3 Limites des Méthodes Actuelles | 17 |
| 2.2 Contributions | 18 |
| 2.3 Publications | 20 |
| 3 DWKS: A Local Descriptor of Deformations Between Meshes and Point Clouds | 21 |
| 3.1 Introduction | 21 |
| 3.2 Related Work | 23 |
| 3.3 Background | 24 |
| 3.3.1 Functional Maps | 24 |
| 3.3.2 Shape Matching | 25 |
| 3.3.3 Shape Difference Operators | 25 |
| 3.3.4 Matching with Shape Difference Operators | 26 |
| 3.4 Our approach – DWKS | 27 |
| 3.4.1 Motivation and Overview | 27 |
| 3.4.2 Definition | 29 |
| 3.4.3 Stability of descriptors | 31 |
| 3.4.4 Matching Pipeline | 31 |
| 3.5 Experiments | 32 |
| 3.6 Conclusion and future work | 34 |

| | |
|---|-----------|
| 4 Assessing Craniofacial Growth and Form Without Landmarks: A New Automatic Approach Based on Spectral Methods | 37 |
| 4.1 Introduction | 37 |
| 4.2 Materials and Methods | 39 |
| 4.2.1 Overview | 39 |
| 4.2.2 Data | 40 |
| 4.2.3 Functional Maps | 42 |
| 4.2.4 Computing correspondences | 43 |
| 4.2.5 Deformation Analysis | 44 |
| 4.3 Results | 47 |
| 4.3.1 Functional Map Network | 47 |
| 4.3.2 Extrinsic analysis | 47 |
| 4.4 Discussion | 49 |
| 4.5 Conclusion | 51 |
| 5 Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization | 55 |
| 5.1 Introduction | 55 |
| 5.2 Related Work | 58 |
| 5.3 Notation & Background | 59 |
| 5.4 Discrete Solver for Dirichlet Energy | 61 |
| 5.4.1 Problem Formulation | 62 |
| 5.4.2 Smoothness-promoting Discrete Solver | 62 |
| 5.5 Smoothness Analysis in Unified Framework | 64 |
| 5.6 Experiments | 65 |
| 5.6.1 DEFORMTHINGS4D-MATCHING Dataset | 65 |
| 5.6.2 Comparison on Smoothness Formulation | 66 |
| 5.6.3 Implementation & Parameters | 69 |
| 5.7 Conclusion, Limitations & Future Work | 69 |
| 6 Scalable and Efficient Functional Map Computations on Dense Meshes | 71 |
| 6.1 Introduction | 71 |
| 6.2 Related Works | 73 |
| 6.3 Method Overview | 75 |
| 6.4 Notations & Background | 76 |
| 6.4.1 Notations | 76 |
| 6.4.2 Functional Maps and the ZoomOut algorithm | 76 |
| 6.4.3 Eigenbasis approximation | 77 |
| 6.5 Our approach | 78 |
| 6.5.1 Approximate Functional Map | 78 |
| 6.5.2 Approximation Errors | 79 |
| 6.5.3 Improved Eigenbasis Approximation | 81 |
| 6.5.4 Scalable ZoomOut | 83 |
| 6.5.5 Implementation | 84 |
| 6.6 Results | 84 |

| | | |
|----------|---|------------|
| 6.6.1 | Timings | 85 |
| 6.6.2 | Evaluation | 86 |
| 6.7 | Conclusion, Limitations, and Future Work | 88 |
| 7 | Memory-Scalable and Simplified Functional Map Learning | 91 |
| 7.1 | Introduction | 91 |
| 7.2 | Related Works | 93 |
| 7.3 | Background & Motivation | 94 |
| 7.4 | Method | 97 |
| 7.4.1 | Scalable Dense Maps | 97 |
| 7.4.2 | Scalable Dense Maps | 97 |
| 7.4.3 | Differentiable ZoomOut | 99 |
| 7.4.4 | Overall Pipeline and Implementation | 99 |
| 7.4.5 | Properties of learned features | 100 |
| 7.5 | Results | 101 |
| 7.5.1 | Datasets | 101 |
| 7.5.2 | Shape Matching Results | 101 |
| 7.5.3 | Scalability to Dense Meshes | 102 |
| 7.5.4 | Learned Features | 104 |
| 7.6 | Conclusion, Limitations & Future Work | 105 |
| 8 | Conclusion | 107 |
| 8.1 | Evolution of the Field and Impact of Our Work | 107 |
| 8.2 | Limitations and Future Work | 108 |
| A | DWKS: A Local Descriptor of Deformations Between Meshes and Point Clouds: Additional Results | 113 |
| A.1 | Spectral Properties of SDO | 113 |
| A.1.1 | Theoretical properties | 113 |
| A.1.2 | Practical Computation | 114 |
| A.1.3 | Algebraic structure | 115 |
| A.2 | Parameters | 115 |
| A.2.1 | Optimization Objective | 115 |
| A.2.2 | Hyperparameters | 116 |
| A.3 | Synthetic faces dataset | 117 |
| A.4 | Comparison with partial spectral matching | 117 |
| B | Assessing Craniofacial Growth and Form Without Landmarks: A New Automatic Approach Based on Spectral Methods | 125 |
| B.1 | Age Distribution | 125 |
| B.2 | Using age as a feature | 125 |
| C | Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization: Additional Results | 127 |
| C.1 | Smoothness Reformulation | 127 |
| C.1.1 | Non-Rigid ICP | 127 |

| | | |
|---------------------|--|------------|
| C.1.2 | As-Rigid-As-Possible | 127 |
| C.1.3 | Smooth Shells | 128 |
| C.1.4 | Reversible Harmonic Maps | 128 |
| C.2 | DEFORMTHINGS4D-MATCHING Dataset | 129 |
| C.3 | FAUST dataset | 130 |
| C.4 | Additional Results | 130 |
| C.5 | Parameters | 132 |
| C.6 | Initialization | 133 |
| C.7 | Discrete Optimization | 133 |
| D | Scalable and Efficient Functional Map Computations on Dense Meshes: Additional Results | 135 |
| D.1 | Function χ | 135 |
| D.2 | Coefficient weighting | 135 |
| D.3 | Proof of Proposition 6.1 | 136 |
| D.4 | Proof of Proposition 6.2 | 137 |
| D.5 | Values of theoretical quantities | 139 |
| D.6 | Functional Map approximation | 140 |
| D.7 | Implementation details | 140 |
| D.8 | Texture transfer | 141 |
| E | Memory-Scalable and Efficient Functional Map Computations on Dense Meshes: Additional Results | 143 |
| E.1 | Implementation Details | 143 |
| E.2 | More Baselines & Ablation | 145 |
| E.3 | ZoomOut Algorithm | 147 |
| E.4 | Adapting Scalable ZoomOut | 147 |
| E.5 | Dense Meshes | 147 |
| Bibliography | | 149 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Example existing 3D data. Left: 3D scan of an indoor scene from the ScanNet dataset [56], Middle: Hand-made animation of a raccoon from the DEFORMINGTHINGS4D dataset [124], Right: Slices of a single volumetric MRI scan, from the OASIS dataset [137]. | 2 |
| 1.2 | Shape matching seeks to find corresponding points (x and y) on similar shapes M and N , here extracted from the dataset of [176]. Finding correspondences for all points on one shape amounts to computing a function T , that maps points on shape M to points on shape N | 3 |
| 2.1 | Exemple de données 3D existantes. À gauche : scan 3D d'une scène d'intérieur provenant du jeu de données ScanNet [56], au milieu : animation manuelle d'un raton laveur provenant du jeu de données DEFORMINGTHINGS4D [124], à droite : coupes d'un scan IRM volumétrique unique, provenant du jeu de données OASIS [137] : Tranches d'une IRM volumétrique unique, provenant de l'ensemble de données OASIS [137]. | 12 |
| 2.2 | La correspondance de formes cherche à trouver des points correspondants (x et y) sur des formes similaires M et N , ici extraites du jeu de données de [176]. Trouver des correspondances pour tous les points d'une même forme revient à calculer une fonction T , qui donne, pour tous les points sur M , un point sur N | 13 |
| 3.1 | Our method uses two collections of noisy point-clouds with roughly similar deformations (left) and outputs a point-wise inter-collection map (right). | 22 |
| 3.2 | Examples of DWKS descriptors for meshes. The left part displays the source meshes and their deformed version. The right part displays for each mesh the DWKS descriptors at 3 fixed energy levels shown at the bottom (seen as a function of the mesh). Notice that descriptors remain somewhat consistent even in the case of partiality. | 26 |
| 3.3 | Our pipeline takes two collections as input, with given base shapes. DWKS descriptors for each deformation are aggregated to smooth out the noise, and are then used for point-wise map computation. | 29 |

| | | |
|------|---|----|
| 3.4 | Visualization of the fitting pipeline. Starting from descriptors, a first point-wise map is computed, which is then projected into a low-dimension functional map ignoring some outlier vertices. This functional map is then refined using ZoomOut algorithm [145]. | 30 |
| 3.5 | Results on the Sumner dataset using the complete shapes. We show our method obtains similar results as [48] in this case. | 33 |
| 3.6 | Results on the Sumner dataset. While our method doesn't achieve a visually perfect result due to the absence of tailored refinement, it outperforms usual methods. Digits on the legend describe the average geodesic error for each method. | 34 |
| 3.7 | Results on the DFAust dataset. We match two collections of 6 meshes representing humans in jumping motion. The bottom row represents accuracy curves of pointwise maps in the complete (left) and partial (right) cases. Numbers in the legend give the average accuracy multiplied by 10^3 | 35 |
| 4.1 | Visualization of the inner and outer surfaces of a mesh in horizontal section. | 40 |
| 4.2 | The dataset included highly non-isometric deformations (top and middle row) as well as topological dissimilarities around the orbital cavity (bottom row). | 41 |
| 4.3 | Comparison of point-based and functional correspondences. Top: A point on one shape was transferred to another shape using a vertex-to-vertex map T . Bottom: A function was transferred using a functional map \mathbf{C} . The image below the arrow is a representation of the entries of the functional map matrix. Note that these two transfers do not require T or \mathbf{C} to be ground truth maps. | 43 |
| 4.4 | Examples of low frequency (<i>i.e.</i> slowly varying) eigenfunctions of the Laplace-Beltrami operator. | 44 |
| 4.5 | Local significance of the normal consistency scheme for inner and outer surfaces. | 45 |
| 4.6 | Visualization of computed correspondences between two random pairs of shapes in the collection. Top : correspondence between two nearly isometric scans. bottom : correspondences between highly non-isometric scans. Vertices on the left are given RGB colors depending on their XYZ coordinates, and corresponding vertices between left and right are given the same color. | 46 |
| 4.7 | Top : Surrogate template chosen as the closest to the limit shape. Bottom : Corrected template by applying the average deformation to the template | 47 |
| 4.8 | Example of the computed deformation between the template and a shape from the dataset. | 48 |
| 4.9 | Cumulative explained variance ratio of the PCA. | 48 |
| 4.10 | Projection of the dataset on the fourth and seventh principal components. | 50 |

| | |
|--|----|
| 4.11 Visualization of the first principal deformations D_j for $j = 1, 2$. For each component, the leftmost and rightmost shapes displayed the two opposite deformations. The color indicated the norm of the displacement D_j at each vertex. | 51 |
| 4.12 ‘Typical’ skull for each class, as detected by the logistic regression. The deformation was exaggerated for visualization purposes. Note that these skulls don’t represent an actual patient, but the deformations on which each embedding is projected. | 52 |
| 4.13 Example of misclassification due to a large missing part. This patient was predicted to be a normal skull, while it belongs to the metopic group. | 52 |
| | |
| 5.1 Our method can deal with noisy inputs and produce high-quality and smooth pointwise maps for non-isometric shape pairs. As a comparison, ZoomOut [145], the current state-of-the-art refinement method, cannot explicitly control the map smoothness and can have large discontinuous patches in the obtained maps. We report the smoothness metric E_D for each map. | 56 |
| 5.2 DeformThings4D-Matching Dataset. We construct a new dataset for non-isometric shape matching based on the DEFORMTHINGS4D [124]. We show some example humanoid shapes and visualize the cross/inter-category correspondences via color transfer. Note that the shapes in the same category are remeshed independently (zoom in to see the mesh wireframes). | 57 |
| 5.3 Previous methods focus on improving map accuracy and do not have explicit control over the map smoothness. Here we show an example of a non-isometric pair. We report the Dirichlet energy (E_D) of maps after refinement by different methods. | 61 |
| 5.4 Qualitative evaluation on two pairs from DEFORMTHINGS4D-MATCHING. For a near-isometric shape pair shown on the <i>top</i> , all methods achieve smooth maps. For a shape pair that is far from isometry shown on <i>bottom</i> , nICP, ARAP, RHM, and Shells achieve relatively smooth maps but contain large patch of back-to-front ambiguity. The maps obtained by ZoomOut and Discrete Solver are locally smooth due to their spectral representation, but fail to maintain global smoothness. As a comparison, our methods can be robustly generalized to non-isometric shape maps and achieve globally smooth maps. | 65 |
| 5.5 Starting from a poor initial map, our method can produce a more smooth and accurate map compared to the baseline methods. | 68 |

| | | |
|-----|---|----|
| 6.1 | Our method produces point-to-point correspondences between dense meshes <i>efficiently</i> , using values only located at sparse samples, displayed in white. The source and target shapes from the DEFORMINGTHINGS4D dataset [124] are composed of roughly 100 000 vertices, and correspondences are displayed using texture transfer. The map computation (including all preprocessing) took 60 seconds on a standard machine. | 72 |
| 6.2 | Overall pipeline of our method, using real data from [176]. Given two dense input shapes, we first generate an approximate eigenbasis computation by using a modified version of the approach introduced in [148] (Sec. 6.5.3). We then propose a new, scalable version of ZoomOut (Sec. 6.5.4), which exploits our functional map approximation (Sec. 6.5.1) and comes with theoretical approximation bounds. Ultimately, this leads to dense pointwise correspondences between the two input shapes visualized here via color transfer. | 73 |
| 6.3 | Example of a local function u_j (red color) centered on v_j (red vertex), visualized without (Left) and with (Right) our adaptive radius strategy. Other samples v_k are displayed in black. | 81 |
| 6.4 | Effect of the adaptive radius on functional map approximation. Top row displays a pointwise map T from the right mesh to the left mesh using color transfer. Bottom row displays $\bar{\mathbf{C}}$ (Left), $\hat{\mathbf{C}}$ when using the pipeline from [149] (Middle) and our functional map $\tilde{\mathbf{C}}$ (Right). . . | 82 |
| 6.5 | Qualitative results on the SHREC19 dataset. Although processing time differ heavily, there is no significant difference between our method and results from ZoomOut. However, remeshing the surface before ZoomOut results in locally constant correspondences. | 85 |
| 6.6 | Texture transfer using our scalable version of ZoomOut. Samples used in the pipeline are shown as white dots. | 87 |
| 6.7 | Accuracy curves for different methods presented in Table 6.2. Numbers in the legend provide the average geodesic error ($\times 10^3$). | 88 |
| 6.8 | Texture transfer using our scalable version of ZoomOut on a pair of the SHREC19 dataset. Samples used in the pipeline are shown as white dots. | 89 |
| 7.1 | Our method takes a set of point features as input, which can be learned and uses a differentiable version of the ZoomOut algorithm to produce correspondences. Due to its light memory cost and attractive processing time this can be used while training a network, or when running the network on very dense meshes. | 92 |
| 7.2 | Our pipeline takes as input two shapes and use a feature extractor network to obtain pointwise features. These features are used to compute an initial pointwise map and then fed to our Differentiable ZoomOut block. All the pointwise maps Π are our scalable dense maps, which are fast and memory efficient. | 95 |

| | | |
|-----|--|-----|
| 7.3 | Our scalable dense maps relies on the underlying structure of Eq. (7.5), where the sum is computed for each contiguous memory block highlighted in the image. The entries are evaluated on the fly while performing summation, and results from each block are then accumulated to obtain the final per-rows values. The implementation is provided by the Keops package [44]. | 98 |
| 7.4 | GPU memory usage when processing a single pair of shapes, depending on their vertex count. Note, e.g., that AttentiveFMaps [123] runs out of 24GB memory after 11k vertices. | 103 |
| 7.5 | Example of feature functions learned by our model, with or without consistency loss. As noted by [12], smoother features are generally preferred for generalization purposes. | 105 |
| A.1 | Results on the cats and lions dataset. Left mesh shows the base shape of the collection of complete cats. Right meshes display the computed pointwise maps in the case of a collection of partial cats (top row) and partial lions (bottom row). | 116 |
| A.2 | Visualization of the Synthetic Faces dataset. | 117 |
| A.3 | Results on the Synthetic Faces dataset. Top row shows pointwise maps after refinement when ignoring the area shape difference operators. Bottom row show the accuracy curves for this setting (left) and when using both type shape difference operators (right). Notice the stability of our method. | 118 |
| A.4 | Visualization of the Cats and Lions dataset. The two bottom shapes are matched together using the deformations shown above them. | 119 |
| A.5 | Quantitative results associated to results on Figure A.1. Top graph displays accuracy results for the experiment matching a partial cat to a complete cat, bottom one those for the experiment marching a partial lion to a complete cat. | 121 |
| A.6 | Visualization of the three collections corresponding to the jumping motion. The biggest shape represents the base shape of the collection and the smaller ones the used deformations. | 122 |
| A.7 | Examples of results on the DFaust dataset for the jumping motion. Left mesh shows the base shape of the first individual. Right meshes display the computed pointwise maps in the case of a collection of partial cats (top row) and partial lions (bottom row). | 123 |
| A.8 | Results on the DFaust dataset. Each line show results for a given motion, and each column results for a given individual. | 123 |
| B.1 | Age Distribution of the three groups in the dataset. | 126 |
| C.1 | Example of wrapping a DRAKE shape to a CRYPTO shape to establish cross-category correspondences. | 129 |

| | | |
|-----|--|-----|
| C.2 | We show two non-isometric shape pairs from TOSCA dataset can compare pointwise maps obtained from different methods via color transfer. Note that TOSCA non-isometric dataset only provide <i>sparse</i> ground-truth correspondences. We therefore color the vertices that do not have GT correspondences in <i>black</i> | 133 |
| C.3 | Example of correspondences without (enter) and with (right) the conformal term of Discrete Optimization. While some parts are smoother, the overall effect is marginal. | 134 |
| D.1 | Possible choices for function χ | 135 |
| D.2 | Ground truth functional map $\bar{\mathbf{C}}$ using the functional space $\bar{\mathcal{F}}$ without (Left) and with (Right) adaptive radius. Notice that up to a change of sign, both functional maps look similar | 140 |
| D.3 | Texture transfer using our scalable version of ZoomOut on a pair of the SHREC19 dataset. Samples used in the pipeline are shown as white dots. | 141 |
| E.1 | PCK curves on the SMAL dataset | 146 |
| E.2 | We leverage on the capacity of DiffusionNet [198] to perform on various discretization of the same shape. Left and right are two shapes from the SHREC19 dataset [144]. We show on each shape features obtained on the remeshed and original version of the dataset. | 148 |
| E.3 | Example of texture transfer of our method on the SHREC19 [144] dataset using our pipeline. | 148 |

List of Tables

| | | |
|-----|---|-----|
| 4.1 | Average True Positive Rates (TPR) after cross-validation for multiple logistic regression objectives, differentiating Control against Trigonocephaly (C-T), Control and Metopic Crests together against Trigonocephaly ((C+M)-T), and eventually Control against Metopic Crests against Trigonocephaly (C-M-T). | 49 |
| 5.1 | DEFORMTHINGS4D-MATCHING Dataset: Summary over 433 shape pairs. We highlight the best two in blue, except those of Shells and RHM (see text for details). | 67 |
| 5.2 | Results on a random subset of 200 pairs of the FAUST dataset. We highlight the best two in blue. | 67 |
| 5.3 | TOSCA Non-Isometric Dataset: Summary over 95 shape pairs. We highlight the best two in blue, except those of Shells and RHM (see text for details). | 68 |
| 6.1 | Timing in seconds for different methods when processing a pair with 50K and 200K vertices and applying ZoomOut from spectral size 20 to 100 | 85 |
| 6.2 | Evaluation of different methods on the complete SHREC19 dataset. Blue highlights the best two methods. | 86 |
| 6.3 | Norm of the estimation error Δ with and without adaptive radius on the SHREC19 dataset | 88 |
| 7.1 | Mean geodesic errors ($\times 100$) when training and testing on the Faust, Scape and Shrec19 datasets. Best result is shown in bold. | 100 |
| 7.2 | Mean geodesic errors ($\times 100$) on the DeformingThing4D dataset subset from [123] (DT4D-H). Best results are highlighted in bold. | 102 |
| 7.3 | Average processing time in seconds, between CPU, naïve and scalable GPU implementations of ZoomOut and Differentiable ZoomOut | 104 |
| C.1 | Accuracy on DEFORMTHINGS4D-MATCHING | 131 |
| C.2 | Bijectivity on DEFORMTHINGS4D-MATCHING | 131 |
| C.3 | Coverage on DEFORMTHINGS4D-MATCHING | 132 |
| C.4 | Smoothness on DEFORMTHINGS4D-MATCHING, via Conformal Distortion | 132 |
| C.5 | Results on TOSCA nonIsometric using WKS initialization | 133 |

| | | | |
|-----|---|-----------|-----|
| D.1 | Evaluation of the reweighting scheme on the SHREC19 dataset. | | 136 |
| E.1 | Mean geodesic errors ($\times 100$) when training and testing on the Faust, Scape and Shrec19 datasets. Due to the fine-tuning strategy on ULRSSM (w/ fine-tune), we do not highlight its results. See text for details. | | 144 |
| E.2 | Mean geodesic errors ($\times 100$) when training and testing on the SMAL dataset. | | 145 |

Introduction

In recent years, there has been a notable surge in the accessibility of 3D data across diverse domains, encompassing fields such as medical imaging [168, 218], autonomous vehicles [85, 20], computer graphics [29, 124], and scientific investigations at the microscopic and molecular levels [106, 41]. This data, as depicted in Figure 1.1, is remarkably diverse, ranging from individual subjects to intricate real-world environments. While some datasets, such as ShapeNet objects [42] or real scans in ScanNet [56], can be straightforward for analysis when rendered as an image, others pose notable challenges, particularly volumetric data like MRI scans [137] or intricate protein structures [106]. The manual processing of such complex data places a considerable burden, highlighting the demand for efficient tools to facilitate rapid analysis in 3D.

In response to these demands, the field of 3D data analysis has undergone rapid development, achieving notable breakthroughs, particularly in areas such as protein folding [106] and scene representation from images [108]. Geometry Processing [33], the subdomain of 3D data analysis focusing on curved surfaces, has also had considerable success in tasks such as rendering, animating, or deforming shapes. Unlike general unstructured 3D data such as point clouds, shapes are often represented as surfaces embedded in 3D, which offers rich additional information compared to mere points in space. Leveraging decades of mathematical research in differential geometry studying such objects [82, 228], the field of geometry processing has developed powerful tools dedicated to shape analysis [33]. However, in practice, many of these tools still struggle to perform consistently across all types of shapes in a fully automatic manner. For example, tasks like comparing shapes to detect differences, such as diseases in organs, without human annotation, present significant challenges to the research community.

In practice, 3D shapes often belong to coherent collections, either tied to specific acquisition devices [206, 218] or belonging to particular groups like anatomical structures [137, 218] or animal models [247, 124]. Analyzing such collections automatically and quickly, even in the presence of a limited number of shapes, is a fundamental research direction crucial for providing meaningful tools to practitioners.

Recent advances in Artificial Intelligence (AI), particularly in domains like im-

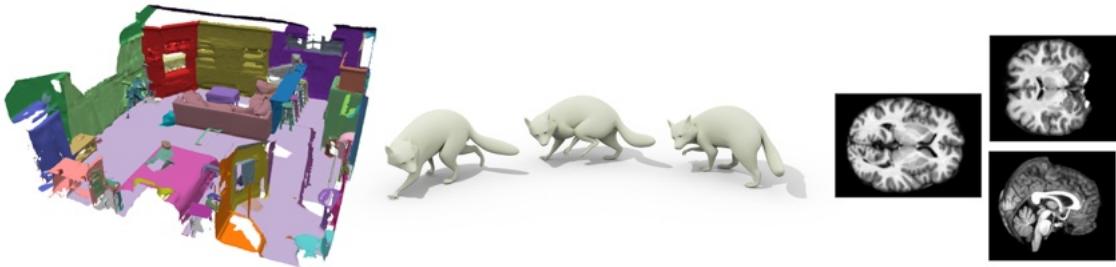


Figure 1.1: Example existing 3D data. Left: 3D scan of an indoor scene from the ScanNet dataset [56], Middle: Hand-made animation of a raccoon from the DEFORMINGTHINGS4D dataset [124], Right: Slices of a single volumetric MRI scan, from the OASIS dataset [137].

age and text generation, have been notable, with the emergence of powerful tools such as StableDiffusion [183], BLIP2 [122], and large language models [221, 104] that are now widely accessible. Given this progress, it is natural to contemplate the potential application of such methods to 3D shapes. However, AI has so far found success in only specific areas of shape analysis. This limitation can be attributed, in part, to the fact that 3D shapes typically lack the regular and structured representation found in images and text, where pixels form a grid and words follow a sequential order. In contrast, shapes often have a surface structure, represented, in practice, as triangle meshes. This makes direct adaptation of text and image processing tools inefficient. As a result, specialized AI tools are being developed to leverage this unique structure [43, 171, 232, 205, 198], but are yet to reach the level of performance of image analysis networks. A second limitation of the success of AI in 3D lies in the slow, complex, and costly nature of 3D data acquisition, leading to limited data availability compared to image and text data. Modern networks therefore seek to explicitly enforce geometric constraints to cater for this potential lack of data [198, 205]. Consequently, the field of artificial intelligence is still rapidly evolving to address the unique problems posed by surfaces.

In this thesis, we aim to contribute to the development of efficient tools, whether axiomatic or AI-powered, for shape analysis within the context of relatively small collections of shapes. Moreover, the focus will extend to ensuring their scalability to real-world objects that can have high levels of detail and contain hundreds of thousands to millions of individual primitives (points, triangles, etc.).

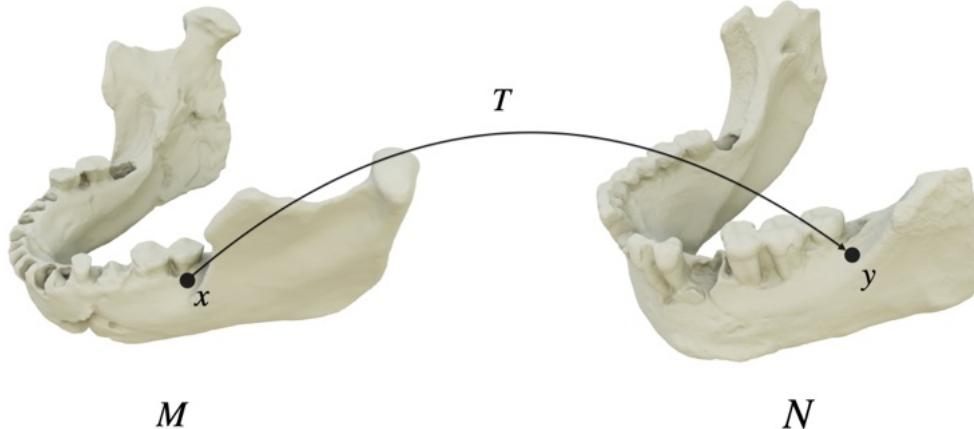


Figure 1.2: Shape matching seeks to find corresponding points (x and y) on similar shapes M and N , here extracted from the dataset of [176]. Finding correspondences for all points on one shape amounts to computing a function T , that maps points on shape M to points on shape N .

1.1 Context

1.1.1 Shape Analysis and Shape Matching

Shape analysis is a central problem in geometry processing [33]. It focuses specifically on the automated processing and modification of 3D shapes, defined as curved surfaces lying in the ambient space.

Historically, geometry processing has been associated with (or emerging from) with computer graphics, a domain focused on the display and manipulation of images. However, with the recent proliferation of 3D scanning devices [206, 163], its influence has spread across diverse disciplines, notably in the medical domain, where anatomical structures are digitized into 3D scans for subsequent comparative analysis [218, 90]. Another noteworthy field embracing 3D shape analysis is biology, where cellular structures are now subjected to 3D scanning [170, 41], occasionally in sequential fashion, resulting in sets of 4D videos [102].

In these different fields, 3D shape analysis serves multiple purposes. This can take place in two different scenarios: single shape analysis and multi-shape analysis. While these scenarios are inherently related, as analysis of a single shape is usually necessary for comparison with others, their distinct focus merits separate exploration.

Single-shape analysis usually leverages the surface structure of a shape to synthesize information or simplify its manipulation [33]. For instance, surface parameterization [67, 121, 207], a significant challenge in this domain, seeks to map a shape

onto a plane, which often is used for its visualization or the manipulation of its texture. Given a parametrization, users can, for example, apply or modify textures on the surface using simple images [91] instead of directly manipulating the shape itself [6]. Another line of research focuses on shape modification, where surfaces can be smoothed [215, 210, 101], and anomalies addressed, such as closing holes [194, 14]. The domain of shape modification is furthermore closely related to shape deformation and animation, where transformations respecting certain constraints, like fixed positions, are sought [9, 209]. Without robust automated deformation models, manual redesign of surfaces post-transformation would be required, making such tasks highly time-consuming. At their core, shape processing methods heavily rely on efficient computations of geometric quantities [33, 167, 146] such as vertex normals, curvature, or geodesic distances [192, 53], which additionally provide meaningful insights into the shapes, including their total area, number of components, and even the number of holes. While still an active field of research, satisfying automated approaches have been developed for many of those tasks, enabling them to be used to analyze groups of shapes in a faster manner.

Multi-shape analysis entails a comprehensive comparison of shapes. This can involve both a simple pair or a larger collection. Analyzing a collection often aims to extract meaningful signals from shapes, enabling, for instance, subgroup classification or detection [203]. For example, in biological science, tasks such as disease detection or species clustering based on phenotype could be performed automatically [89, 217]. This typically first requires establishing relations between shapes, such as describing differences between pairs of shapes in the collection. Employing statistical analysis of these relations can then provide broader insights into the collection as a whole. However, comparing pairwise shapes can have multiple meanings, and the problem and is often approximated depending on the downstream task [186, 89]. Moreover, even using approximations, obtaining these pairwise relations automatically poses significant challenges. Such relations involve, for instance, detecting similar zones on two shapes or performing segmentation on surfaces to extract shared parts [198, 169, 21]. Some approaches also involve detecting points of interest, or landmarks, rather than entire regions [244]. Detecting these similar points aids in guiding deformation from one shape to another, or in automatically comparing distances between landmarks, which can be cumbersome when done manually on real objects. Another useful tool when working with collections of shapes is shape registration to a template [22, 28, 138], which involves relating all shapes in the collection to a simplified and known shape, called a template, often using deformations. This is particularly beneficial when the same shape has been independently scanned multiple times using the same or different scanning devices, as when filming a 3D video [29, 124]. Using a template usually facilitates downstream analysis or animation, with simplified statistics or visualization.

A notable subject of research in shape analysis focuses on shape correspondence [224, 59], particularly dense shape correspondence, commonly known as shape matching. This process aims to establish, for every point on one shape, a correspond-

ing point on another shape, as presented in Figure 1.2. While shape correspondence itself may not be directly usable for statistical analysis, it serves as a fundamental component in most pairwise shape analysis methods, as it forms the basis of shape comparison by establishing relations between local points. For instance some problems, such as landmarks detection or segmentation, can be naturally derived from shape correspondence [198], many other tasks easily leverage such results. Notably, having dense correspondence between two shapes simplifies the process of deforming one shape into another, as it can be formulated as finding the optimal deformation aligned with the computed correspondences [9, 209]. Conversely, computing the deformation between two shapes can be used to obtain correspondences between overlapping points [69], highlighting the inherent connection between these two problems. Most shape deformation methods, therefore, explicitly or implicitly rely on shape matching computation at their core. Other problems can also be framed as shape correspondence, for instance, landmarks tracking in a 3D video can be described as obtaining correspondences between successive frames of the video. Deformation is however not always necessary to obtain correspondences [109, 159], and some works have devised methods to characterize geometric differences between shapes directly based on correspondences [186, 98, 76], without relying on any deformation models.

While useful in many downstream applications, computing correspondences can be challenging. First, obtaining shape correspondence through human annotation is prohibitively costly, leading to a growing interest in the automatic derivation of correspondences. However, deriving high-quality shape correspondence introduces conceptual challenges, necessitating the development of several priors and approximations [109, 5, 159]. Notably, the definition of shape matching can vary depending on the context, imposing diverse criteria on its quality across different applications. For instance, in tracking scenarios, emphasis is placed on the smoothness of correspondences to ensure a seamless progression of points over time. Texture transfer requires high-quality, angle-preserving, and reversible maps for aesthetically pleasing results [77, 76, 190, 191]. In the case of shape deformation, precision in correspondences may have reduced significance, provided there are no discontinuous patches that could significantly impact the final outcomes. As a result, distinct approaches have emerged to address these challenges, reflecting the nuanced requirements of different applications.

1.1.2 Shape Matching Approaches

Numerous approaches have been employed for shape matching across various applications. Traditionally, these methods fall into two main categories: extrinsic matching, primarily relying on the 3D coordinates of the shape, and intrinsic matching, which instead aims to leverage local information on the surface while remaining agnostic to the embedding..

Extrinsic matching typically involves obtaining correspondences by computing

a reasonable deformation between multiple shapes. Various models have been developed for this purpose, ranging from simple rigid transformations [22] to more involved deformations [9, 209, 236]. Volumetric deformation models, which deform the ambient space rather than the surface itself [64, 57, 69], have found substantial application in the medical imaging field, particularly relying on discrete diffeomorphism [16, 17, 57, 58]. Earlier works tried to transport one point cloud into another, using optimal transport [54] or Gaussian mixture models [147], but usually ignoring the surface structure of the data. While extrinsic matching algorithms demonstrate significant power and intuitive interpretation, they exhibit sensitivity to initialization and can be computationally slow. Notably, in the absence of hand-placed landmarks, the pre-alignment of shapes, a non-trivial task without human intervention, is often necessary for optimal performance.

In contrast, intrinsic matching relies on local surface information that usually remains invariant under isometry, *i.e.* transformations preserving distances between points (*e.g.*, rotations or translations). Several works have attempted to compute these intrinsic maps, often relying on initial signals [109] such as local surface descriptors [211, 219, 15], while minimizing distortion metrics like area distortion or the intrinsic Dirichlet energy [77, 76]. Another approach attempts to preserve pairwise geodesic distances between pairs of points before and after matching [227, 208], but suffers from slow algorithms heavily reliant on good initialization. To achieve precise and visually coherent texture transfer, novel formulations of discrete diffeomorphism [190, 191, 213] have been developed, as well as formulations for geometrically consistent shape matching [239, 238, 182]. However, these methods are only applicable to shapes with a limited number of vertices and can only handle shapes with the same topology, defined by the number of holes and handles in the surfaces. In comparison, the functional map framework [159, 153, 154, 145, 177, 162], a significant focus of this thesis, relaxes the problem of shape correspondence and instead aims to efficiently transfer functions between shapes while advocating several geometric constraints such as area preservation. This relaxed formulation can easily accommodate shapes with different topologies and can be adapted to the challenging setting of partial shape matching [181, 128, 13]. Moreover, recent advancements, leveraging powerful neural networks [198, 60, 13, 40, 212], have obtained state-of-the-art results on standard shape matching benchmarks, bringing this flexible framework into the spotlight. However, despite their inherent invariance under rigid transformations, intrinsic matching methods remain highly sensitive to intrinsic symmetries of shapes, missing parts, and sometimes topology changes such as holes. It is worth noting that some intrinsic matching approaches have attempted to leverage extrinsic matching models to enhance their performance [70, 71, 72], but these remain unstable without additional initial information such as pre-alignment.

1.1.3 Limitations of existing approaches

Despite their considerable achievements, shape matching methods encounter limitations, not only in terms of performance but also in their scope of applications.

Extrinsic approaches, primarily deformation-based, have had success in the medical imaging community [64]. However, intrinsic shape matching methods, typically associated with computer graphics, have seen limited application beyond the evaluation of shape correspondence and texture transfer. This emphasis on pure shape correspondence often neglects essential properties crucial for downstream applications, such as the smoothness of correspondences, which can significantly impact the statistical analysis of results, although it might degrade the precision of the correspondences. Addressing and promoting such properties presents ongoing challenges for the shape matching community.

The methodologies employed often depend on the datasets used for evaluation. Current benchmarks primarily focus on shape matching tasks [28, 29, 10, 247, 144, 124], featuring human or animal shapes derived from either registered real scans [28] or purely synthetic shapes [124]. However, these datasets mostly consist of very similar shapes, which lack the intricate details found in real-world data and typically comprise shapes with a low number of vertices, sometimes further downsampled when evaluated [174]. Moreover, while some datasets incorporate shapes with holes and missing parts, they do not capture the complexities of acquisition noise or occlusion encountered in real scans. Assessing the performance of current shape matching methods on more diverse data divergent from existing benchmarks, potentially real-world data, holds significant promise for advancing the field.

Despite achieving impressive performance, existing methods still have room for improvement when dealing with highly non-isometric shapes, particularly in scenarios lacking prior knowledge such as rigid alignment or landmarks. Furthermore, the synergy between the intrinsic and extrinsic approaches in shape matching has been largely unexplored [70, 71]. Leveraging artificial intelligence tools, particularly recent deep neural networks [232, 198, 60], shows promise in addressing these challenges. However, such learning-based approaches still suffer from poor generalization performance on novel surfaces [184, 60, 40, 212]. Additionally, while these methods have shown efficacy on small-scale data, they often struggle with computational efficiency when applied to dense real 3D scans [123, 212]. Thus, beyond enhancing performance, adapting these approaches to real data remains a crucial challenge.

In this thesis, our aim is to augment the stability and applicability of shape matching methods, with a specific emphasis on enhancing their scalability to complex shapes and suitability for real-world scenarios.

1.2 Contributions

This thesis addresses multiple challenges in shape comparison, leveraging and significantly extending modern spectral shape matching methods. Our contributions are guided by the requirements of underlying downstream tasks utilizing shape matching, particularly in scenarios where high-resolution meshes are needed. The organization of the thesis is as follows:

First Part: Local Analysis of Differences. The initial chapter focuses on locally characterizing the differences between two shapes. Obtaining meaningful signals about parts of the surface that undergo significant variations between two shapes is a very challenging problem, key to tasks like anomaly detection or disease detection. Specifically, in Chapter 3, we provide an extensive analysis of Shape Difference Operators [186, 160], which use functional maps to describe all intrinsic deformations between shapes. We derive a local descriptor for deformations between shapes, and demonstrate its consistency across collections undergoing similar deformations, enabling shape matching between these collections.

Second Part: Analysis of a Cohort of Patients. Chapter 4, a collaborative work with surgeons from Necker Hospital in Paris, uses modern functional map methods to analyze a collection of skulls in young children. The primary goal is to automatically detect trigonocephaly, a disease partly characterized by a triangular forehead. While successful, this project highlights areas for improvement in functional map methods, particularly emphasizing the importance of smooth correspondences over precise accuracy. Furthermore, the data consisted of high resolution meshes, highlighting the speed bottlenecks of spectral shape matching methods. The following chapters thus will tackle such issues.

Third Part: Enforcing Smoothness. Chapter 5 aims at deriving a novel shape matching pipeline, based on the Discrete Optimization framework [177], by explicitly enforcing the smoothness of correspondences. Existing benchmarks for shape correspondence perform well under the assumption of near-isometry, but we introduce a more challenging non-isometric dataset, a remeshed version of the DEFORMINGTHINGS4D dataset [124], where most baselines struggle without an explicit smoothness constraint, while highlighting the significant improvement brought by our approach.

Fourth Part: Functional Maps on Dense Meshes. In Chapter 6, we address the computational inefficiency of functional map methods on dense meshes. We therefore develop an efficient estimator of the functional map using only a sparse set of points on the shape, and provide a theoretical upper bound on the estimation error. This approach yields comparable results to classical methods, with sub-sample accuracy, and in a fraction of the processing time, without the need for explicit resolution downscaling.

Fifth Part: Scalable Functional Map Learning. The final section, covered in Chapter 7, focuses on novel shape matching methods employing deep neural networks and functional maps. Our approach employs efficient GPU methods for kernel computations [44], eliminating the need to store dense quadratic matrices in memory. Additionally, we implement GPU porting of an existing refinement algorithm, removing the necessity for unstable differentiation through a linear system. This implementation allows for the use of very dense shapes during training or testing, overcoming the scalability challenges associated with current methods.

Sixth Part: Conclusion. Chapter 8 concludes the manuscript, providing first an overview of the evolution of the field of geometry processing and the impact of our work during the time of this thesis. We then discuss the current existing challenges in the field as well as promising direction for future works.

Despite the apparent simplicity of spectral shape matching methods, they often present challenges in implementation and the complexity of their use. To address this, all our implementations are available online, and we have developed a Python package named PYFMAPS with minimal requirements, providing the implementation of numerous approaches. This package, available at <https://github.com/RobinMagnet/pyFM>, has garnered reasonable success, with several parts of the code being utilized in other works [123, 92]. Notably, all our contributions leverage this package.

1.3 Publication Record

The content of our work relies on the following publications:

- R. Magnet and M. Ovsjanikov. DWKS : A Local Descriptor of Deformations Between Meshes and Point Clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021 [130]. Presented in Chapter 3.
- R. Magnet, K. Bloch, M. Taverne, S. Melzi, M. Geoffroy, R. H. Khonsari, and M. Ovsjanikov. Assessing craniofacial growth and form without landmarks: A new automatic approach based on spectral methods. *Journal of Morphology*, 2023 [134]. Presented in Chapter 4.
- R. Magnet, J. Ren, O. Sorkine-Hornung, and M. Ovsjanikov. Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization. In *2022 International Conference on 3D Vision (3DV)*, 2022 [133]. Won the best paper award, and presented in Chapter 5.
- R. Magnet and M. Ovsjanikov. Scalable and Efficient Functional Map Computations on Dense Meshes. *Computer Graphics Forum*, 2023 [131]. Presented in Chapter 6.

- R. Magnet and M. Ovsjanikov. Memory-Scalable and Simplified Functional Map Learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024 [132]. Presented in Chapter 7.

CHAPITRE 2

Introduction en français

Ces dernières années, les données 3D sont de plus en plus accessibles dans de nombreux domaines, notamment pour l'imagerie médicale [168, 218], les véhicules autonomes [85, 20], l'infographie [29, 124], et les analyses scientifiques à l'échelle microscopique et moléculaire [106, 41]. Ces données, dont certains exemples sont présentés sur la Figure 2.1, sont d'une remarquable diversité, présentant tout autant des sujets individuels créés informatiquement que des environnements complexes du monde réel. Si certaines données, telles que les objets de ShapeNet [42] ou les captures de scènes d'intérieur de ScanNet [56], peuvent être facilement analysés lorsqu'ils sont observés sur une image, d'autres sont beaucoup plus complexes à appréhender de cette façon, par exemple les données IRM [137] ou les structures de protéines [106]. Le traitement et l'analyse manuelle de ces données 3D représentent une charge de travail considérable, ce qui souligne la nécessité de disposer d'outils efficaces pour faciliter leur analyse.

Via cette accessibilité accrue, le domaine de l'analyse des données 3D a connu un développement rapide, réalisant des percées notables, en particulier dans des domaines tels que le repliement des protéines [106] et la représentation de scènes à partir d'images [108]. Le traitement automatique de la géométrie [33], le sous-domaine de l'analyse des données 3D axé sur les surfaces courbes, a connu un succès considérable dans des tâches telles que le rendu, l'animation ou la déformation des formes. Contrairement aux données 3D générales non structurées telles que les nuages de points, les formes sont représentées par des surfaces, qui offrent de nombreuses informations supplémentaires par rapport à de simples points dans l'espace. S'appuyant sur des décennies de recherche mathématique en géométrie différentielle qui étudie ces objets [82, 228], de puissants outils dédiés à l'analyse des formes ont été développés [33]. Cependant, dans la pratique, beaucoup de ces outils ont encore du mal à fonctionner de manière cohérente sur toutes les variétés de formes, de manière automatique. Par exemple, des tâches telles que la comparaison de formes, sans annotation humaine, pour la détection de différences, comme une maladie dans un organe, représentent encore des défis importants pour le milieu.

Dans la pratique, les formes 3D se présentent souvent au sein de collections cohérentes, soit liées à du matériel d'acquisition spécifique [206, 218] ou appartenant à des groupes sémantiques particuliers comme des structures anatomiques [137, 218] ou des modèles animaux [247, 124]. L'analyse automatique et rapide de ces collec-

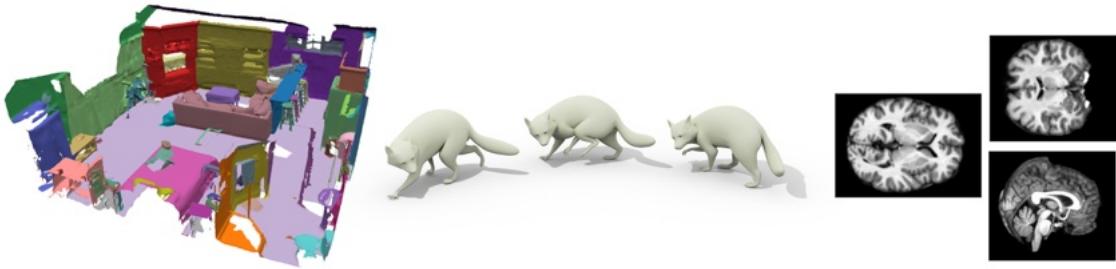


FIGURE 2.1 : Exemple de données 3D existantes. À gauche : scan 3D d'une scène d'intérieur provenant du jeu de données ScanNet [56], au milieu : animation manuelle d'un raton laveur provenant du jeu de données DEFORMINGTHINGS4D [124], à droite : coupes d'un scan IRM volumétrique unique, provenant du jeu de données OASIS [137] : Tranches d'une IRM volumétrique unique, provenant de l'ensemble de données OASIS [137].

tions, même en présence d'un nombre limité d'éléments, est un axe de recherche fondamental afin de fournir des outils pratiques aux praticiens.

Les progrès récents de l'intelligence artificielle (IA) sont impressionnants, en particulier dans des domaines tels que la génération d'images et de textes, avec l'émergence d'outils puissants tels que StableDiffusion [183], BLIP2 [122], et les grands modèles de langage [221, 104] qui sont aujourd'hui facilement accessibles. Compte tenu de ces progrès, il est naturel d'envisager l'application potentielle de ces méthodes aux formes 3D. Cependant, jusqu'à présent, l'IA n'a réussi à s'imposer que dans des domaines spécifiques de l'analyse des formes. Cette limitation peut être attribuée, en partie, au fait que les formes 3D ne présentent généralement pas la représentation régulière et structurée que l'on trouve dans les images et les textes, où les pixels forment une grille et les mots suivent un ordre séquentiel. En revanche, les formes présentent une structure de surface, dont la représentation, généralement sous forme d'ensembles de triangles, rend inefficace l'adaptation directe des outils de traitement de texte et d'image. Par conséquent, des outils d'IA spécialisés sont en cours de développement pour tirer parti de cette structure unique [43, 171, 232, 205, 198], et n'ont pas encore atteint le niveau de performance des réseaux d'analyse d'images. Une deuxième limite au succès de l'IA en 3D réside dans la nature lente, complexe et coûteuse de l'acquisition de données 3D, ce qui conduit à une disponibilité limitée par rapport aux données d'image et de texte. Les réseaux modernes cherchent donc à appliquer explicitement des contraintes géométriques pour pallier ce manque potentiel de données [198, 205]. Par conséquent, le domaine de l'intelligence artificielle continue d'évoluer rapidement pour répondre aux problèmes uniques posés par les surfaces.

Dans cette thèse, nous visons à contribuer au développement d'outils efficaces, qu'ils soient axiomatiques ou alimentés par l'intelligence artificielle, pour l'analyse

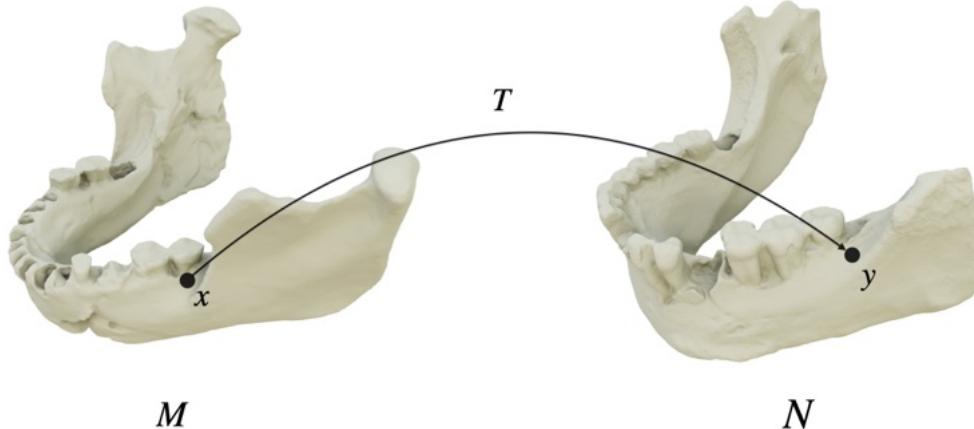


FIGURE 2.2 : La correspondance de formes cherche à trouver des points correspondants (x et y) sur des formes similaires M et N , ici extraites du jeu de données de [176]. Trouver des correspondances pour tous les points d'une même forme revient à calculer une fonction T , qui donne, pour tous les points sur M , un point sur N .

des formes dans le contexte de collections de formes relativement petites. En outre, l'accent sera mis sur l'extensibilité de ces outils aux formes du monde réel qui présentent des niveaux de détail élevés.

2.1 Contexte

2.1.1 Analyse de Formes et Mise en Correspondance

L'analyse des formes est un problème central du domaine du traitement automatique de la géométrie [33]. Ce problème se concentre spécifiquement sur le traitement et la modification automatisés des formes 3D, définies comme des surfaces courbes situées dans l'espace ambiant.

Historiquement, le traitement automatique de la géométrie est lié à l'infographie, un domaine centré sur l'affichage et la manipulation d'images numériques. Toutefois, avec la récente prolifération des dispositifs de numérisation 3D [206, 163], son influence s'est étendue à de multiples disciplines, notamment dans le domaine médical, où les structures anatomiques font l'objet de numérisations en vue d'une analyse comparative ultérieure [218, 90]. En biologie, l'analyse des formes en 3D gagne également en popularité : des structures cellulaires sont désormais soumises à des captures 3D [170, 41], parfois de manière séquentielle, ce qui produit des vidéos 3D, appelées captures 4D [102].

Dans ces différents domaines, l'analyse des formes en 3D a de multiples objectifs, qui se divisent essentiellement deux catégories : l'analyse d'une unique forme et

l’analyse d’un ensemble de formes. Bien que ces catégories soient intrinsèquement liées, puisque analyser une forme seule est généralement nécessaire pour la comparer à d’autres, leurs objectifs restent distincts et méritent d’être explorés séparément.

L’analyse d’une forme unique exploite généralement la structure de surface de la forme afin d’obtenir des informations ou pour simplifier sa manipulation [33]. Par exemple, la paramétrisation de la surface [67, 121, 207], un défi important dans ce domaine, cherche à cartographier une forme sur un plan 2D, ce qui facilite souvent sa visualisation ou la manipulation de sa texture. Grâce à cette paramétrisation, les utilisateurs peuvent, par exemple, appliquer ou modifier des textures sur la surface en utilisant de simples images [91] au lieu de manipuler directement la forme elle-même [6]. Une autre ligne de recherche se concentre sur la modification de la forme elle-même, où les surfaces peuvent être lissées [215, 210, 101], ou les anomalies traitées, par exemple en fermant des trous [194, 14]. Le domaine de la modification des formes est en outre étroitement lié à la déformation et à l’animation de celles-ci, où l’on recherche des transformations respectant certaines contraintes, comme des positions attitrées pour certaines parties de la surface [9, 209]. En l’absence de modèles de déformation automatisés et robustes, il faudrait redessiner manuellement les surfaces après la transformation, ce qui prendrait beaucoup de temps. Les méthodes d’analyse de forme reposent essentiellement sur des calculs efficaces de quantités géométriques [33, 167, 146] telles que les normales des sommets, la courbure ou les distances géodésiques [192, 53], qui fournissent en outre des informations importantes sur les formes, notamment leur surface totale, le nombre de composants et même le nombre de trous. Bien qu’il s’agisse encore d’un domaine de recherche actif, des approches automatisées satisfaisantes ont été développées pour bon nombre de ces tâches, ce qui leur permet d’être utilisées pour analyser des groupes de formes de manière plus rapide.

L’analyse de groupes de formes implique donc une comparaison complète des formes, qu’il s’agisse d’une simple paire ou d’une collection plus large. L’analyse d’une collection vise souvent à extraire des signaux significatifs des formes, permettant par exemple la classification ou la détection de sous-groupes [203]. En sciences biologiques, des tâches telles que la détection de maladies ou le regroupement d’espèces sur la base du phénotype peuvent alors être effectuées automatiquement [89, 217]. En règle générale, il s’agit d’abord d’établir des relations entre les formes, par exemple en décrivant les différences entre les paires de formes de la collection. L’analyse statistique de ces relations peut ensuite fournir des informations plus générales sur la collection dans son ensemble. Cependant, la comparaison des formes par paire n’a pas de formulation mathématique unifiée et fait souvent l’objet d’approximations, en fonction de la tâche en aval [186, 89]. En outre, même en utilisant des approximations, l’obtention automatique de ces relations par paires pose des défis importants. Ces relations consistent, par exemple, en la détection de zones similaires sur deux formes ou en la segmentation commune des surfaces pour extraire les parties partagées [198, 169, 21]. Certaines approches impliquent également la détection de points d’intérêt, ou points de repères, plutôt que de régions

entières [244]. La détection de ces points similaires permet par exemple de guider la transformation d'une forme vers une autre, ou de comparer automatiquement les distances entre les points de repère, ce qui peut s'avérer fastidieux lorsqu'on le fait manuellement sur des objets réels. Un autre outil utile pour travailler avec des collections de formes est l'enregistrement des formes sur un modèle [22, 28, 138], qui consiste à relier toutes les formes de la collection à un modèle 3D simplifié et connu, souvent à l'aide de déformations. Cette méthode est particulièrement utile lorsque la même forme a été scannée indépendamment plusieurs fois à l'aide du même dispositif d'acquisition ou de dispositifs différents, comme lors du tournage d'une vidéo en 3D [29, 124]. L'utilisation d'un modèle 3D facilite généralement l'analyse ou l'animation en aval en proposant des statistiques et des visualisations simplifiées.

L'un des principaux sujets de recherche dans le domaine de l'analyse des formes est la mise correspondance des formes [224, 59], en particulier la mise correspondance dense de formes. Ce processus vise à établir, pour chaque point d'une forme, son point correspondant sur une autre forme, comme présenté dans la Figure 2.2. Bien que la correspondance des formes ne soit pas directement utilisable pour l'analyse statistique, elle constitue un élément fondamental de la plupart des méthodes d'analyse des formes par paire, car elle est la base de la comparaison des formes en établissant des relations entre les points. Par exemple, certains problèmes, tels que la détection de points de repère ou la segmentation, sont naturellement dérivés de la correspondance des formes, et de nombreuses autres tâches tirent facilement parti de ces résultats. Une correspondance dense entre deux formes simplifie notamment le processus de déformation d'une forme en une autre, puisqu'il s'agit essentiellement de trouver la déformation optimale alignée sur les correspondances calculées [9, 209]. Inversement, le calcul de la déformation entre deux formes fournit finalement des correspondances entre des points qui se chevauchent [69], ce qui met en évidence le lien inhérent entre ces deux problèmes. La plupart des méthodes de déformation de formes reposent donc explicitement ou implicitement sur le calcul de correspondances. D'autres problèmes peuvent également être définis comme des correspondances de formes. Par exemple, le suivi de points de repère dans une vidéo 3D peut être décrit comme l'obtention de correspondances entre des images successives de la vidéo. La déformation n'est cependant pas toujours nécessaire pour obtenir des correspondances [109, 159], et certains travaux ont conçu des méthodes pour caractériser les différences géométriques entre les formes directement basées sur les correspondances [186, 98, 76], sans s'appuyer sur des modèles de déformation.

Bien que particulièrement puissante, l'obtention de correspondances de formes par l'annotation humaine est d'un coût prohibitif, ce qui a conduit à un intérêt croissant pour leur dérivation automatique. Cependant, l'obtention d'une correspondance de forme de haute qualité présente des défis conceptuels, nécessitant le développement de plusieurs approximations [109, 5, 159]. Notamment, la définition de la correspondance des formes peut varier en fonction du contexte, imposant divers critères de qualité pour différentes applications. Par exemple, dans les scénarios de suivi de points de repère, l'accent est mis sur la continuité des correspondances

afin de garantir une progression continue des points dans le temps. Le transfert de textures nécessite des correspondances de haute qualité, inversibles, préservant les angles afin d'obtenir des résultats esthétiques. Dans le cas de la déformation des formes, la précision des correspondances a une importance réduite, à condition qu'il n'y ait pas de zones discontinues susceptibles d'avoir un impact significatif sur les résultats finaux. En conséquence, des approches distinctes sont apparues pour automatiser ce problème, reflétant les exigences nuancées des différentes applications.

2.1.2 Méthodes de Mise en Correspondence

Différentes approches ont été employées pour la mise en correspondance des formes dans diverses applications. Traditionnellement, ces méthodes se répartissent en deux grandes catégories : l'appariement extrinsèque, qui s'appuie principalement sur la position 3D de la forme dans l'espace et l'appariement intrinsèque, qui vise plutôt à exploiter les informations locales tout en restant indépendant de ces coordonnées.

L'appariement extrinsèque consiste généralement à obtenir des correspondances en calculant une déformation raisonnable entre plusieurs formes. Divers modèles ont été développés à cette fin, allant de simples transformations rigides [22] à des déformations plus complexes [9, 209, 236]. Les modèles de déformation volumétrique, qui déforment l'espace ambiant plutôt que la surface elle-même [64, 57, 69], sont également largement utilisés dans le domaine de l'imagerie médicale, en s'appuyant notamment sur le formalisme des difféomorphismes discrets [16, 17, 57, 58]. Des travaux antérieurs ont tenté de transporter un nuage de points vers un autre, en utilisant le transport optimal [54] ou des modèles de mélange gaussien [147], mais en ignorant généralement la structure surfacique des données. Bien que les algorithmes d'appariement extrinsèque soient très puissants et intuitifs, car ils déforment visuellement la surface, ils restent sensibles à l'initialisation et peuvent être très lents. Notamment, en l'absence de points de repère placés à la main, le pré-alignement des formes, une tâche non triviale sans intervention humaine, est souvent nécessaire pour obtenir des performances optimales.

En revanche, l'appariement intrinsèque s'appuie sur des informations de surface locales qui restent généralement invariantes sous isométrie, c'est-à-dire des transformations préservant les distances entre les points (par exemple, des rotations ou des translations). Plusieurs travaux ont tenté de calculer ces correspondances intrinsèques, en s'appuyant souvent sur des signaux initiaux [109] tels que des descripteurs de surface locaux [211, 219, 15], tout en minimisant les mesures de distorsion telles que la distorsion d'aire ou l'énergie de Dirichlet intrinsèque [77, 76]. Une autre approche tente de préserver les distances géodésiques entre les paires de points avant et après l'appariement [227, 208], mais produit des algorithmes très lents, qui dépendent fortement de l'initialisation. Afin d'obtenir un transfert de texture précis et visuellement cohérent, de nouvelles formulations de difféomorphismes discrets [190, 191, 213] ont été développées, ainsi que des formulations pour la mise en correspondance géométriquement cohérente [239, 238, 182]. Cependant, ces méthodes ne sont appli-

cables qu’aux formes ayant un nombre limité de points et ne peuvent traiter que les formes ayant la même topologie, définie par le nombre de trous et de poignées dans les surfaces. En comparaison, les cartes fonctionnelles [159, 153, 154, 145, 177, 162], un sujet essentiel dans cette thèse, simplifie le problème de la correspondance des formes et vise plutôt à transférer efficacement les fonctions entre les formes tout en préconisant plusieurs contraintes géométriques telles que la préservation de l’aire. Cette formulation assouplie s’adapte facilement à des formes ayant des topologies différentes et peut être adaptée au sujet difficile de la mise en correspondance partielle des formes [181, 128, 13]. En outre, des avancées récentes, tirant parti de puissants réseaux neuronaux [198, 60, 13, 40, 212], ont permis d’obtenir des résultats de pointe sur des benchmarks de correspondance de formes standard, plaçant cette méthode comme l’approche privilégiée pour résoudre ces problèmes. Cependant, malgré leur invariance aux transformations rigides, les méthodes de mise en correspondance intrinsèque restent très sensibles aux symétries intrinsèques des formes, aux parties manquantes et parfois aux changements de topologie tels que les trous. Il convient de noter que certaines approches d’appariement intrinsèque ont tenté d’exploiter des modèles d’appariement extrinsèques pour améliorer leurs performances [70, 71, 72], mais elles restent instables sans informations initiales supplémentaires, telles que le pré-alignement.

2.1.3 Limites des Méthodes Actuelles

Malgré leur succès considérable, les méthodes de mise en correspondance de formes rencontrent des limites, non seulement en termes de performances, mais aussi en termes de champ d’application.

Les approches extrinsèques, principalement basées sur la déformation, ont eu beaucoup de succès dans le domaine de l’imagerie médicale [64]. Cependant, les méthodes intrinsèques de mise en correspondance des formes, typiquement associées à l’infographie, ont vu leur application limitée en dehors de l’évaluation de performance pure ou du transfert de texture. L’accent mis sur la correspondance de forme pure néglige souvent des propriétés, comme la continuité des correspondances, essentielles pour les applications en aval et ayant un impact significatif sur l’analyse statistique des résultats, bien qu’elle puisse dégrader la précision des correspondances. La prise en compte et la promotion de ces propriétés constituent des défis importants pour la communauté de la mise en correspondance des formes.

Les méthodes employées sont souvent spécifiquement développées pour les ensembles de données utilisés en évaluation. Les tests de référence actuels se concentrent principalement sur les tâches de correspondance de formes [28, 29, 10, 247, 144, 124], présentant des formes humaines ou animales dérivées de scans réels enregistrés [28] ou de formes purement synthétiques [124]. Cependant, ces ensembles de données sont principalement constitués de formes très similaires, qui ne possèdent pas le niveau de détails que l’on trouve dans les données du monde réel, et ne comprennent généralement que des formes avec un faible nombre de sommets, parfois encore sous-

échantillonnées lors de l'évaluation [174]. En outre, bien que certains ensembles de données intègrent des formes avec des trous et des parties manquantes, ils ne capturent pas les complexités du bruit d'acquisition ou de l'occlusion rencontrées dans les scans réels. L'évaluation des performances des méthodes actuelles de correspondance de formes sur des données plus variées, différentes des références existantes, et potentiellement provenant du monde réel, est une évolution nécessaire du domaine.

Malgré des performances impressionnantes, les méthodes existantes peuvent encore largement être améliorées lors du traitement des formes fortement non isométriques, en particulier dans les scénarios dans lesquels peu d'informations préalables sont données, telles que l'alignement rigide ou des points de repère. En outre, la co-existence des domaines intrinsèque et extrinsèque dans l'appariement des formes a été largement sous-exploité [70, 71], entravant potentiellement les possibilités d'améliorer les résultats en tirant parti des points forts de chaque domaine. L'utilisation d'outils d'intelligence artificielle, en particulier les récents réseaux neuronaux profonds [232, 198, 60], s'avère prometteuse pour relever ces défis. Cependant, ces approches "profondes" souffrent encore d'une faible performance de généralisation sur de nouvelles surfaces [184, 60, 40, 212]. En outre, si ces méthodes se sont révélées efficaces sur des données à petite échelle, elles se heurtent souvent à des problèmes d'efficacité de calcul lorsqu'elles sont appliquées à des scans 3D réels de grande taille [123, 212]. Ainsi, au-delà de l'amélioration des performances, l'adaptation de ces approches à l'extensibilité des données réelles reste un défi crucial.

Dans cette thèse, notre objectif est d'accroître la stabilité et l'applicabilité des méthodes de correspondance des formes, en mettant l'accent sur l'amélioration de leur extensibilité et de leur adéquation avec les scénarios du monde réel.

2.2 Contributions

Cette thèse aborde plusieurs défis en matière de comparaison de formes, en utilisant et améliorant des méthodes modernes de comparaison spectrale de formes. Nos contributions sont guidées par les exigences des tâches en aval utilisant la comparaison de formes, en particulier dans les scénarios dans lesquels des maillages de haute résolution sont utilisés. L'organisation de la thèse est la suivante :

Première partie : Analyse Locale des Différences. Le premier chapitre se concentre sur la caractérisation locale des différences entre deux formes. L'obtention d'information sur les zones de différences entre deux surfaces est un problème très difficile, essentiel pour des sujets tels que la détection d'anomalies ou la détection de maladies.

Deuxième partie : Analyse d'une Cohorte de Patients. Le Chapitre 4 présente un travail collaboratif avec des chirurgiens de l'hôpital Necker à Paris, où nous utilisons des méthodes modernes de cartes fonctionnelles afin d'analyser une collec-

tion de crânes de jeunes enfants. L'objectif principal est de détecter automatiquement la trigonocéphalie, une maladie caractérisée en partie par un front triangulaire. Malgré la réussite du projet, ce chapitre met en évidence des domaines dans lesquels les méthodes de cartographie fonctionnelle peuvent être améliorées, notamment en soulignant l'importance des correspondances continues par rapport à la précision. En outre, les données étaient constituées de maillages à haute résolution, ce qui a souligné les goulots d'étranglement en matière de vitesse des méthodes de correspondance spectrale des formes. Les chapitres suivants cherchent à améliorer ces sujets.

Troisième partie : Renforcement de la Continuité. Dans le Chapitre 5, nous présentons un nouvel algorithme de mise en correspondance de formes, fondé sur le cadre de l'optimisation discrète [177], où nous renforçons explicitement la continuité des correspondances. Les méthodes existantes pour la correspondance des formes fonctionnant déjà bien dans le cadre de surfaces isométriques, nous introduisons un ensemble de données non isométriques plus complexe, qui est une version retravaillée de l'ensemble de données DEFORMINGTHINGS4D [124], où la plupart des algorithmes de références ne produisent pas de résultats satisfaisants en l'absence de contrainte sur la continuité des correspondances.

Quatrième partie : Cartes Fonctionnelles sur Maillage Dense . Dans le Chapitre 6, nous nous penchons sur l'inefficacité calculatoire des méthodes de cartes fonctionnelles sur des maillages denses. Nous développons alors un estimateur efficace de la carte fonctionnelle en utilisant seulement un ensemble épars de points sur la surface, et nous fournissons une limite supérieure théorique sur l'erreur d'estimation. Cette approche donne des résultats comparables aux méthodes classiques, avec une précision identique, et ce dans une fraction du temps de traitement standard, sans avoir besoin de ré-échantillonnage explicite de la surface.

Cinquième partie : Apprentissage de Cartes Fonctionnelles. La dernière section, couverte par le Chapitre 7, se concentre sur de nouvelles méthodes de correspondance de formes utilisant des réseaux neuronaux profonds et des cartes fonctionnelles. Notre approche utilise des méthodes GPU efficaces, développées pour le calcul des méthodes à noyaux [44], et qui éliminent le besoin de stocker d'énormes matrices denses en mémoire. En outre, proposons le portage GPU d'un algorithme de raffinement existant, en supprimant la nécessité d'une différenciation instable à travers un système linéaire. Cette implémentation permet d'utiliser des formes de très haute résolution, pendant l'entraînement ou le test, ce qui permet de surmonter les problèmes d'applicabilité des méthodes actuelles.

Sixième partie : Conclusion. Le chapitre 8 conclut le manuscrit, proposant tout d'abord un aperçu de l'évolution du domaine du traitement de la géométrie et de l'impact de notre travail pendant la durée de cette thèse. Nous discutons ensuite des défis actuels dans le domaine ainsi que des directions prometteuses pour les travaux

futurs.

Malgré leur simplicité apparente, les méthodes d'analyse spectrale de forme posent souvent des problèmes d'implémentation. Pour y remédier, toutes nos implémentations sont disponibles en ligne, et nous avons développé un paquetage Python nommé PYFMAPS, facilitant l'utilisation de nombreuses méthodes. Ce paquetage, disponible à l'adresse <https://github.com/RobinMagnet/pyFM>, a connu un succès décent, avec plusieurs parties du code utilisées dans d'autres travaux [123, 92]. Notamment, toutes nos contributions s'appuient sur cette implémentation.

2.3 Publications

Le contenu de cette thèse s'appuie sur les publications suivantes :

- R. Magnet and M. Ovsjanikov. DWKS : A Local Descriptor of Deformations Between Meshes and Point Clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021 [130]. Présenté dans le Chapitre 3.
- R. Magnet, K. Bloch, M. Taverne, S. Melzi, M. Geoffroy, R. H. Khonsari, and M. Ovsjanikov. Assessing craniofacial growth and form without landmarks : A new automatic approach based on spectral methods. *Journal of Morphology*, 2023 [134]. Présenté dans le Chapitre 4.
- R. Magnet, J. Ren, O. Sorkine-Hornung, and M. Ovsjanikov. Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization. In *2022 International Conference on 3D Vision (3DV)*, 2022 [133]. Cet article a gagné le prix du meilleur article de la conférence 3DV. Présenté dans le Chapitre 5.
- R. Magnet and M. Ovsjanikov. Scalable and Efficient Functional Map Computations on Dense Meshes. *Computer Graphics Forum*, 2023 [131]. Présenté dans le Chapitre 6.
- R. Magnet and M. Ovsjanikov. Memory-Scalable and Simplified Functional Map Learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024 [132]. Présenté dans le Chapitre 7.

DWKS: A Local Descriptor of Deformations Between Meshes and Point Clouds

We propose a novel pointwise descriptor, called DWKS, aimed at finding correspondences across two deformable shape collections. Unlike the majority of existing descriptors, rather than capturing local geometry, DWKS captures the *deformation* around a point within a collection in a multi-scale and informative manner. This, in turn, allows computing inter-collection correspondences without using landmarks. To this end, we build upon the successful spectral WKS descriptors, but rather than using the Laplace-Beltrami operator, show that a similar construction can be performed on shape difference operators, that capture differences or distortion within a collection. By leveraging the collection information, our descriptor facilitates difficult non-rigid shape matching tasks, even in the presence of strong partiality and significant deformations. We demonstrate the utility of our approach across a range of challenging matching problems on both meshes and point clouds. The code for this chapter can be found at <https://github.com/RobinMagnet/DWKS>.

3.1 Introduction

Shape matching is a ubiquitous problem in 3D computer vision, with various applications like texture and deformation transfer. Numerous methods have therefore been developed during the last decade to compute correspondences between surfaces, relying on simple rigid deformations to more recent learning-based models [224, 188].

While these methods have shown impressive results on several datasets, there remain some very challenging scenarios, especially when dealing with symmetries and non-isometric shapes. The existence of intrinsic symmetries in non-rigid shapes (*e.g.* left-right symmetry in a human shape) can be handled in multiple ways, using an orientation preserving constraint on shape descriptors [174], exploring the space of maps [175], or adding priors through learning-based methods [127, 184, 60]. These methods respectively require the existence of precise shape descriptors, an automatic selection of symmetric and non-symmetric map, and large datasets for neural network training. The case of non-isometric shapes is typically addressed

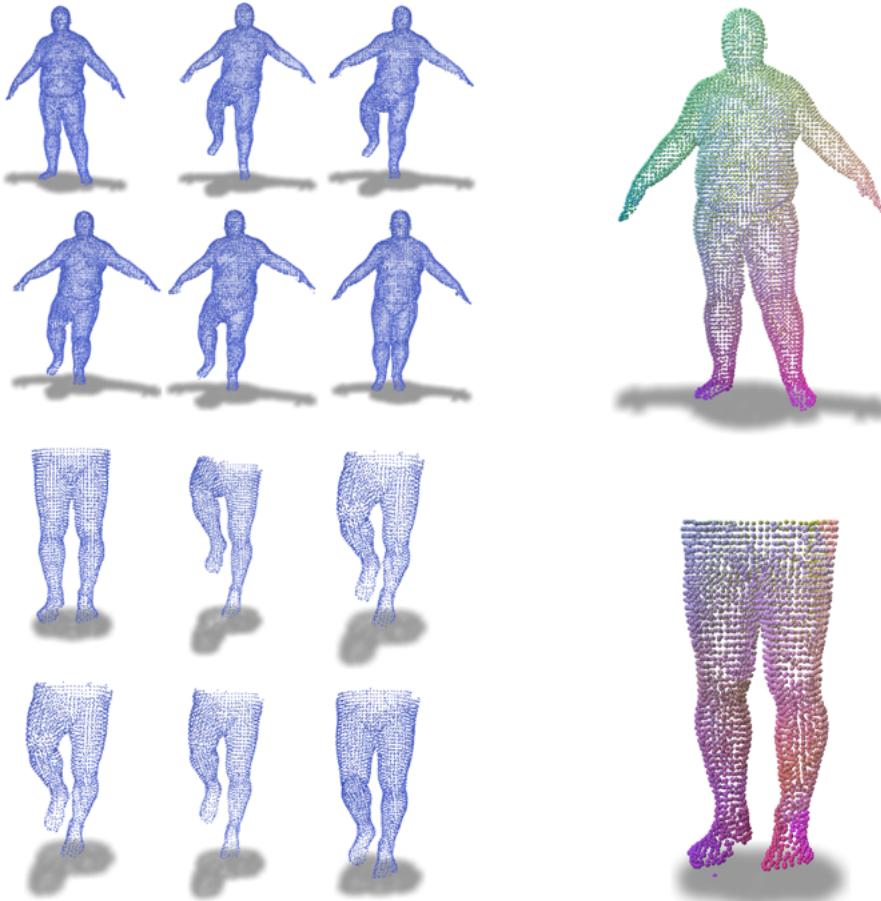


Figure 3.1: Our method uses two collections of noisy point-clouds with roughly similar deformations (left) and outputs a point-wise inter-collection map (right).

either by requiring user-specified landmarks [7, 77, 190] or, again, through extensive neural network training [87], among many other approaches. In the specific case of *partial* non-rigid shape matching, several methods have been developed [181, 128], relying on theoretical properties of the changes to the Laplace-Beltrami operator under partiality.

Remarkably, while 3D shapes often come in the context of a collection, very few methods [193, 48] have tried to leverage the commonality of *the deformations* that exist within the collections to facilitate matching across them, *e.g.* to disambiguate such symmetries or address partiality. Notably, while matching two human shapes in resting pose can suffer from their intrinsic symmetry ambiguity, using information from the deformed version of these shapes with *e.g.* their left knee up could help disambiguate their symmetries. Figure 3.1 exhibits how local deformations of the knees when jumping on one leg allow to compute correspondences between two human shapes, even when the upper-half of the body is missing.

In this work, we propose to develop a local descriptor of differences between surfaces and point clouds. Our method is inspired by successful spectral point-

based descriptors extracted from the Laplace-Beltrami operator [211, 15]. Our main insight is that a similar construction can be performed on other functional operators, leading to informative descriptors that capture different properties of shapes and collections. In our work, we use shape difference operators [186] that have been used for both analyzing deformations within a collection [98] and even shape synthesis [99]. In the context of cross-collection mapping, shape difference operators have been used in [193, 48] as global objectives within the functional map framework, which can limit their utility to *achieve* shape matching. Instead, we demonstrate that pointwise spectral descriptors can successfully be extracted from shape difference operators. Our descriptor, termed DWKS, thus combines the power and flexibility of local descriptors with the information of shape distortion present in shape difference operators, which, as we demonstrate below, makes it applicable in partial cross-collection matching scenarios. We also exploit recent advances in constructing robust operators [197] to enable accurate and efficient matching across shapes represented as both meshes and point clouds.

Our main contributions can be summarized as follows: 1. we introduce a novel pointwise descriptor that reflects deformation around a point within a collection, 2. we demonstrate how spectral methods, and specifically the WKS descriptors can be extended beyond the Laplacian to shape difference operators, and 3. we demonstrate how difficult matching scenarios with partiality and symmetry ambiguity, on both point clouds and meshes, can strongly benefit from our descriptor without landmarks or neural network training.

3.2 Related Work

The shape matching literature is very vast, and we will only highlight existing methods that are most relevant to our setting. We refer the reader to a recent survey [188] on the subject for more information. We base our method on the functional map framework defined in [159] which seeks to match functional spaces on the shapes instead of the shapes themselves, and has led to impressive results in the last decade. Several follow-up works [153, 230, 174, 95, 145, 175] have brought substantial improvements on the original pipeline, and all heavily rely on the existence of consistent descriptor functions of shapes which are functions supposed to be preserved by the mapping, based either on local descriptors [211, 15, 51] or landmarks. Generating informative and robust descriptors in a fully automatic way remains a very challenging problem, and often requires near-isometric shapes without symmetries. To alleviate this issue, recent works have sought to learn descriptors using neural networks, either from usual descriptors [127, 184] or directly from raw data [60, 196]. This allows to incorporate prior information into descriptors, and possibly disambiguate symmetries like left and right for a human shape.

A more demanding setting lies in partial shape matching, which is a simple case of non-isometric shape matching. A remarkable adaptation of the original framework was introduced in [181, 128], based on the theoretical insights of the relation between the Laplace-Beltrami operators of a shape and a compact subset of it.

Methods computing correspondences using functional maps can typically produce somewhat noisy correspondences, which then serve as initialization for refinement algorithms. The original refinement technique uses a variant the ICP algorithm [159] which supposes shapes to be isometric, but new more general approaches have then been developed [75, 174, 77, 145, 100]. The ZoomOut algorithm [145] is of particular interest as it starts from very rough correspondences to obtain high quality maps through spectral up-sampling. While it also relies on a strong near-isometry assumption, the theoretical background developed in [181] can be used to adapt the algorithm to partial shapes in practice.

More related to our contribution are multiple works on shape collections. Several methods have been developed to refine intra-collection correspondences using cycle consistency constraints, *e.g.* [150, 93, 199, 100, 83] among many others. These methods also leverage information within shape collections, but are typically not aimed at computing cross-collection maps and often still rely on pairwise map estimation as a building block. Extracting information about the variability of shapes within a collection was brought about by the introduction of shape difference operators [186], which summarize intrinsic distortion between a pair or a collection of shapes as two functional operators using simply rough correspondences between two shapes represented as functional maps. These two “difference” operators, together with a source shape, have been shown to be sufficient to reconstruct the deformed version up to isometric deformations [49], and up to rigid motion using additional extrinsic shape differences [49, 99]. Shape difference operators provide a powerful tool for summarizing the variability within shape collections, which has motivated their use in computing cross-collection shape correspondences. Our work is directly inspired by the excellent results shown in [193, 48] where corresponding shape difference operators are matched together to compute cross-collection functional maps. The solving procedure, however, relies on SVD, which suffers both from sign ambiguity and possible instability. Furthermore, the method supposes the global deformations to be matched to correspond, which breaks down in the case of partiality.

In this work we focus on computing local or pointwise descriptors from shape differences, which can be used both within the functional maps pipeline and beyond [15, 159]. We show that while shape difference operators [186] capture the global difference between shapes, their properties allow pointwise information to be extracted in the form of vertex-wise descriptors. This information can be used either in conjunction with the pipeline of [193, 48] in the case of complete shapes, or even directly, in the case of partial shapes.

3.3 Background

3.3.1 Functional Maps

Our work falls within the functional map framework originally introduced in [159] and that we review below briefly for completeness. Given two surfaces \mathcal{M} and \mathcal{N} , a point-wise correspondence $T : \mathcal{N} \rightarrow \mathcal{M}$ can be equivalently represented as a linear (functional) map $F : L^2(\mathcal{M}) \rightarrow L^2(\mathcal{N})$ between the space of squared integrable

functions on each shape.

Using an appropriate basis for the two functional spaces, the functional map F can be represented as a possibly infinite matrix. Specifically, eigenfunctions of the Laplace-Beltrami operator of each shape have had a lot of success in spectral shape analysis [120, 179, 159] and can be interpreted as Fourier basis for functions on surfaces, and enable basis truncation due to their multiscale nature.

3.3.2 Shape Matching

The standard functional correspondence pipeline [162] between shapes \mathcal{M} and \mathcal{N} looks for a functional map $\mathbf{C} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}}$ from $L^2(\mathcal{M})$ to $L^2(\mathcal{N})$, where $k_{\mathcal{M}}$ and $k_{\mathcal{N}}$ represent the size of the corresponding (truncated) basis.

Given a set of descriptor functions on each shape $\{(f_i, g_i)\}_{i=1}^p$ with $f_i \in L^2(\mathcal{M})$ and $g_i \in L^2(\mathcal{N})$, expected to be preserved under the functional map, we encode them in their respective basis as two matrices $\mathbf{A} \in \mathbb{R}^{k_{\mathcal{M}} \times p}$ and $\mathbf{B} \in \mathbb{R}^{k_{\mathcal{N}} \times p}$. Standard choices are HKS [211] or WKS [15] descriptors.

Denoting $\Delta^{\mathcal{M}}$ (resp. $\Delta^{\mathcal{N}}$) the Laplace-Beltrami operator on shape \mathcal{M} (resp. \mathcal{N}), expressed in their respective basis as diagonal matrices, the functional correspondence problem is written as:

$$\arg \min_{\mathbf{C} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}}} \|\mathbf{CA} - \mathbf{B}\|_F^2 + \mu_l \|\mathbf{C}\Delta^{\mathcal{M}} - \Delta^{\mathcal{N}}\mathbf{C}\|_F^2 \quad (3.1)$$

with $\|\cdot\|_F$ the Frobenius norm. Here the first term ensures descriptor preservation, while the second one favors isometric maps, and $\mu_l \in \mathbb{R}$ is a manually set scaling factor.

Among many extensions to this basic pipeline, *e.g.* [181, 117, 230, 84, 174] a notable one, introduced in [153] and that we use below, proposed a term promoting the functional maps to arise from pointwise correspondences. For this, a functional operator is associated to each input descriptor $\Gamma_{f_i}, \Gamma_{g_i}$, that acts on other functions through multiplication. These operators are then introduced into the optimization objective from Equation (3.1), by promoting commutativity with them, namely $\mu_{dc} \sum_i \|\mathbf{C}\Gamma_{f_i} - \Gamma_{g_i}\mathbf{C}\|_F^2$. While this pipeline can produce accurate correspondences given appropriate descriptors, it suffers from multiple issues. Namely, it does not allow disambiguating symmetries, requires specific adaptation for partial matching and more broadly does not take into account information about *collections* that shapes often naturally are part of.

3.3.3 Shape Difference Operators

Our work also heavily relies on shape difference operators introduced in [186], that intuitively capture differences or distortion across a pair or within a collection of shapes. Specifically given shapes \mathcal{M}_1 and \mathcal{M}_2 with known correspondences encoded as a functional map F between them, and inner products $\langle \cdot, \cdot \rangle_{\mathcal{M}_1}$ and $\langle \cdot, \cdot \rangle_{\mathcal{M}_2}$ on each shape, the associated *shape difference operator* is defined as the unique linear operator D acting on $L^2(\mathcal{M}_1)$ so that

$$\langle f, Dg \rangle_{\mathcal{M}_1} = \langle F(f), F(g) \rangle_{\mathcal{M}_2} \quad \forall f, g \in L^2(\mathcal{M}_1) \quad (3.2)$$

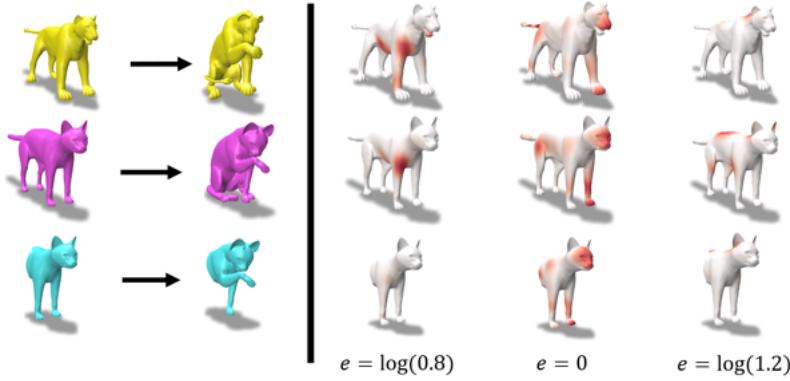


Figure 3.2: Examples of DWKS descriptors for meshes. The left part displays the source meshes and their deformed version. The right part displays for each mesh the DWKS descriptors at 3 fixed energy levels shown at the bottom (seen as a function of the mesh). Notice that descriptors remain somewhat consistent even in the case of partiality.

This operator can be seen as compensating the distortion induced by F with respect to the given inner products. Note that shape difference operators sharing both a common source shape and inner product can all be compared as they all act on the same functional space.

The original work [186] introduced two shape difference operators, which capture the complete intrinsic distortion across shapes. The first one $V_{\mathcal{M}_1, \mathcal{M}_2}$, is associated to the standard L^2 inner product on both shapes $\langle f, g \rangle_{L^2(\mathcal{S})} = \int_{\mathcal{S}} f(x)g(x)d\mu^{\mathcal{S}}$ on a shape \mathcal{S} . The second one, denoted $R_{\mathcal{M}_1, \mathcal{M}_2}$, is associated to the H_0^1 inner product $\langle f, g \rangle_{H_0^1(\mathcal{S})} = \langle \nabla f, \nabla g \rangle_{L^2(\mathcal{S})}$.

The two operators $V_{\mathcal{M}_1, \mathcal{M}_2}$ and $R_{\mathcal{M}_1, \mathcal{M}_2}$ are called respectively area-based and conformal shape differences, since they equal identity if the underlying maps are respectively area-preserving and conformal [186].

Using the spectral basis of size k_1 and k_2 to encode the F into $\mathbf{C} \in \mathbb{R}^{k_2 \times k_1}$, the shape difference operators can be computed directly as $k_1 \times k_1$ matrices

$$\mathbf{V}_{\mathcal{M}, \mathcal{N}} = \mathbf{C}^\top \mathbf{C} \quad (3.3)$$

$$\mathbf{R}_{\mathcal{M}, \mathcal{N}} = (\Delta^{\mathcal{M}})^\dagger \mathbf{C}^\top \Delta^{\mathcal{N}} \mathbf{C} \quad (3.4)$$

with † denoting the Moore-Penrose pseudo inverse.

3.3.4 Matching with Shape Difference Operators

Although originally shape difference operators were introduced for shape analysis, they also have been used for solving cross-collection shape correspondence problems [193, 48]. Specifically, given two shape collections $\{\mathcal{M}_i\}_{i=0}^n$ and $\{\mathcal{N}_i\}_{i=0}^n$, where deformation between \mathcal{M}_0 and \mathcal{M}_i are similar to deformation between \mathcal{N}_0 and \mathcal{N}_i for any $i \in \{1, \dots, n\}$, the goal is to compute a cross-collection map between \mathcal{M}_0 and \mathcal{N}_0 which we denote as \mathcal{M} and \mathcal{N} .

With similar differences between these pairs of shapes, we expect their associated shape difference operators $\mathbf{D}_i^{\mathcal{M}}$ and $\mathbf{D}_i^{\mathcal{N}}$ to act similarly, where \mathbf{D} denotes any of the area or conformal shape difference operator and the index indicates the operator is associated to the deformation between shape 0 and shape i . In the functional map framework, this is equivalent to expecting the sought functional map to commute with these operators, that is $\mathbf{C}\mathbf{D}_i^{\mathcal{M}} \simeq \mathbf{D}_i^{\mathcal{N}}\mathbf{C}$. This leads to the optimization problem solved in the recent approach of [48] (which extends the method in [193]):

$$\arg \min_{\substack{\mathbf{C} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}} \\ \|\mathbf{C}\|_F = 1}} \sum_{i=1}^n E_i(\mathbf{C}) + \alpha \|\mathbf{C}\Delta^{\mathcal{M}} - \Delta^{\mathcal{N}}\mathbf{C}\|_F^2 \quad (3.5)$$

where $\alpha \in \mathbb{R}$ is a scaling factor and

$$E_i(\mathbf{C}) = \|\mathbf{C}\mathbf{R}_i^{\mathcal{M}} - \mathbf{R}_i^{\mathcal{N}}\mathbf{C}\|_F^2 + \|\mathbf{C}\mathbf{V}_i^{\mathcal{M}} - \mathbf{V}_i^{\mathcal{N}}\mathbf{C}\|_F^2 \quad (3.6)$$

Note that unlike the standard functional matching pipeline (3.1), the optimization objective (3.5) does not rely on the existence of coherent descriptors. Moreover, without the constraint $\|\mathbf{C}\|_F = 1$ the trivial solution $\mathbf{C} = 0$ would give zero error. The authors of [193, 48] solve the problem in Equation (3.5) using SVD, which results both in sign ambiguity for the solution and instability in practice. Moreover, the second term of the objective (3.5) acts as a powerful regularizer in the case of near-isometric shapes but fails in more challenging settings, including partiality.

In this work, we build on this pipeline and use local pointwise descriptors extracted from the shape difference operators. This allows both to use standard optimization techniques, thus avoiding the costly SVD associated with $\|\mathbf{C}\|_F = 1$ regularization, and to remove the need for near-isometric regularization. Ultimately, our framework is both more efficient and leads to significant improvements, especially in the case of partial shapes.

3.4 Our approach – DWKS

Algorithm 1 Computing DWKS descriptors

Require: Shape Difference Operator \mathbf{D} in the reduced basis, eigenvectors Φ on source shape, energy values $(e_j)_j$, scale parameter σ

- 1: Compute the eigenvectors \mathbf{U} and the eigenvalues $(\lambda_i)_i$ of \mathbf{D}
 - 2: Compute $\Psi = \Phi\mathbf{U}$ the eigenvectors of \mathbf{D} in the canonical basis.
 - 3: Use Ψ , $(\lambda_i)_i$ and σ to compute the DWKS descriptor for each e_j using Eq. (3.7).
 - 4: **Return** One DWKS descriptors for each energy value.
-

3.4.1 Motivation and Overview

The standard functional correspondence pipeline described in Section 3.3.2 relies on both commutativity with the Laplacian operators and alignment of local descriptors. Interestingly, spectral descriptors such as HKS or WKS [211, 36, 15] are

Algorithm 2 Aggregate DWKS descriptors for a collection

Require: Functional maps $\{\mathbf{C}_i\}_{i=0}^n$ between the base shape and each shape i , eigenvectors Φ on source shape, diagonal matrices $\{\Delta_i\}$ of eigenvalues of the Laplacian for each shape, energy values $(e_j)_j$, scale parameter σ

- 1: **for** $i = 1$ to n **do**
 - 2: Compute \mathbf{V}_i and \mathbf{R}_i using Eqs. (3.3) and (3.4) with \mathbf{C}_i , Δ_0 and Δ_i .
 - 3: Compute DWKS of \mathbf{V}_i and \mathbf{R}_i using Algo. 1 with σ , Φ and $(e_j)_{j=1}^p$
 - 4: **end for**
 - 5: **Return** DWKS descriptors for the complete collection
-

extracted from the same Laplacian operators. Nevertheless, their use in the optimization problem of Equation (3.1) both helps to prevent trivial solutions and injects local information into the process. Our main goal is to mimic this construction for cross-collection matching, but using shape difference operators. Interestingly, commutativity with shape differences has already been advocated in [193, 48]. We seek to extend this construction by also extracting pointwise descriptors from shape difference operators, similarly to the way WKS is extracted from the Laplacian.

Unfortunately, such an adaptation is not straightforward primarily because unlike the Laplace-Beltrami operators whose spectral properties are well-understood and have intuitive physical interpretations, shape difference operators are much less studied, and it is therefore not clear whether pointwise spectral descriptors can be extracted in the same manner. We thus start with the following key observation (with proof given in the supplementary materials):

Theorem 3.1. *Given a non-degenerate functional map F , both the area-based and conformal shape difference operators are positive (semi)-definite, provided that the area and stiffness matrices of the Laplacian are positive (semi)-definite.*

This theorem, which interestingly was not demonstrated in the original shape difference work [186], provides the first insight into the possibility of applying spectral approaches to shape difference operators, since, similarly to the Laplacian, their *eigenvalues are guaranteed to be non-negative*. Moreover, we remark that shape difference operators enjoy both locality and composition or functoriality properties (see, respectively, propositions 4.2.3 and 4.2.4 in [50]). The former remark resonates with the more general property of shape difference operators in [186], which states these two operators act on functions depending on the *local distortion* induced by the underlying correspondence map.

While the eigenfunctions of the Laplace-Beltrami operator capture the “smootherest” possible functions on the surface, the eigenfunctions of the shape difference operators, intuitively, capture areas of distortion between the shapes (see [186, 96] for a discussion of this property). Moreover, functions that are *preserved* by the shape differences $Df = f$ (and thus correspond to eigenvalue 1) correspond to areas of *no* distortion (see Theorem 4.2.1 in [50]). One can draw a parallel with the constant function, corresponding to the zero eigenvalue of the Laplacian. Moreover, shape differences naturally enjoy the multiplicative algebra [186] (see also Proposition 4.2.4 in

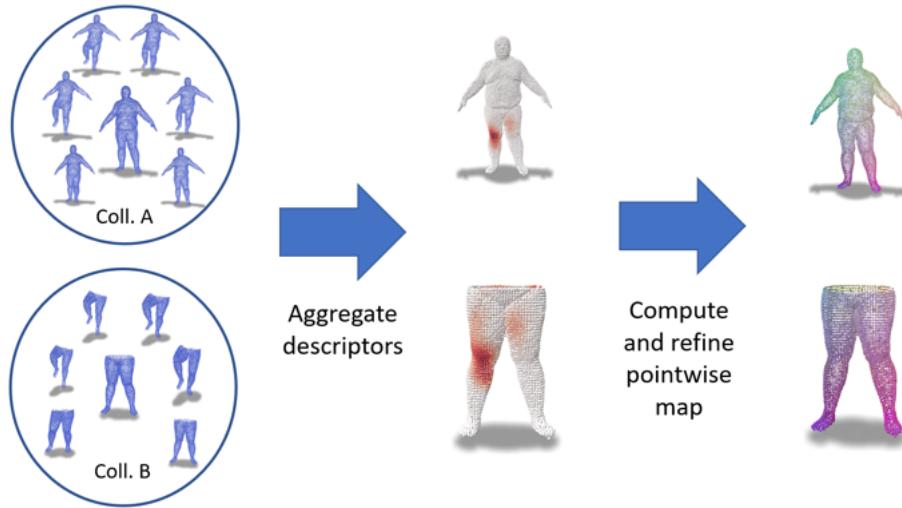


Figure 3.3: Our pipeline takes two collections as input, with given base shapes. DWKS descriptors for each deformation are aggregated to smooth out the noise, and are then used for point-wise map computation.

[50]), which means, for example, that $D_{\mathcal{N}, \mathcal{M}} = (D_{\mathcal{M}, \mathcal{N}})^{-1}$ and $D_{\mathcal{M}, \mathcal{N}} = D_{\mathcal{M}, \mathcal{P}} D_{\mathcal{P}, \mathcal{N}}$ for any shape difference operator D and shapes \mathcal{M}, \mathcal{N} and \mathcal{P} (up to the appropriate change of basis).

The two observations above suggest that the spectrum of shape difference operators is more naturally expressed using the *log-scale*. This way, the undeformed regions correspond to $\log(1) = 0$ log-eigenvalues. Moreover, the log-eigenvalues of the operator that captures the inverse deformation are simply negative of that of the direct deformation. Finally, in some cases (*e.g.*, when deformations commute) the composition of difference operators leads to log-eigenvalues being *sums* of individual difference operators. We expand upon these observations and provide a more formal treatment in the supplementary materials.

A final but essential remark is that the shape difference operators we use in practice are all expressed using the truncated basis of Laplace-Beltrami eigenfunctions using Equations (3.3) and (3.4). The eigenfunctions of shape differences can therefore only represent very smooth functions and in particular cannot represent a Dirac delta function on the mesh but rather a heat kernel centered around a point.
s

3.4.2 Definition

Using these remarks and inspired by the definition of WKS descriptors [15], we define the DWKS descriptor of a given shape difference operator D acting on shape \mathcal{M} as

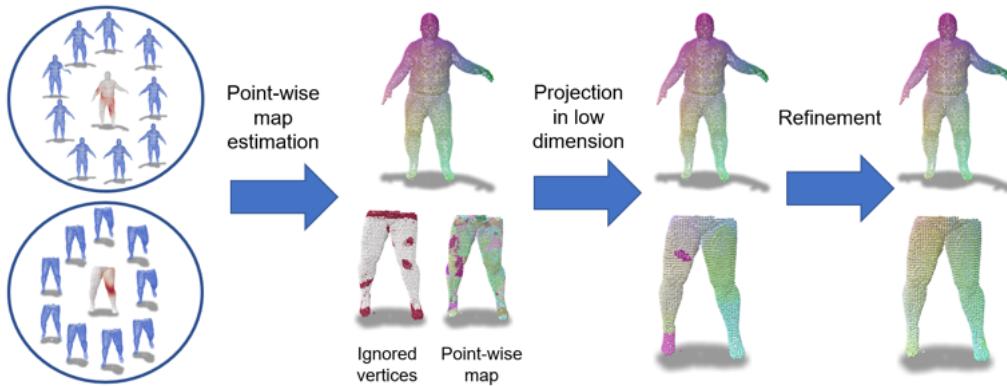


Figure 3.4: Visualization of the fitting pipeline. Starting from descriptors, a first point-wise map is computed, which is then projected into a low-dimension functional map ignoring some outlier vertices. This functional map is then refined using ZoomOut algorithm [145].

$$\begin{aligned} \text{DWKS}(D) : \mathcal{M} \times \mathbb{R} &\rightarrow \mathbb{R} \\ (x, e) &\mapsto C \sum_{\substack{k=1 \\ \lambda_k > 0}}^{k_{\mathcal{M}}} e^{-\frac{(e - \log(\lambda_k))^2}{2\sigma^2}} \psi_k(x)^2 \end{aligned} \quad (3.7)$$

with $(\lambda_k)_k$ and $(\psi_k)_k$ respectively the eigenvalues and eigenvectors of the operator D , σ a manually set parameter, and C ensures that $\int_{\mathbb{R}} \|\text{DWKS}(D)(\cdot, e)\|_{\mathcal{M}} de = 1$.

DWKS can be interpreted as a Gaussian blur of the spectrum of the operator, where the parameter σ defines the spread of eigenvectors on the log scale. Remarks from Section 3.4.1 motivate the choice of a constant σ across all energy levels.

Note that in order to compute a DWKS descriptor we assume to be given either a pair or a collection of shapes with functional maps between them. The functional maps are represented in the truncated Laplacian eigenbasis, which leads to small-size shape difference operator matrices. Note that DWKS also produces a separate pointwise descriptor for each (area-based and conformal) shape difference operator, and can be extended to any shape difference operator by applying the construction described in Algorithm 1.

Examples of DWKS descriptors of the area shape difference operator, seen as functions of the shape at a given energy level, are displayed on Figure 3.2. Each line displays descriptors for similar deformations of a cat and a lion, which do not share either similar geometry or number of vertices. Note that the descriptors seem quite similar up to some noise, as with $e = \log 1.2$. The bottom line demonstrates that the descriptors remain stable even in the case of partiality, as they capture local information.

3.4.3 Stability of descriptors

As seen from Figure 3.2, DWKS descriptors seem to remain stable even in the case of partial shapes, but the reason might be unclear.

When comparing DWKS descriptors between two complete and two partial shapes, two phenomena occur. On the one hand, eigenfunctions of shape difference operators and therefore the associated DWKS descriptors are more localized in the partial case since the spectral basis can represent more precise elements, as shown in [181]. On the other hand, only a fraction of the less localized eigenfunctions in the complete case are located on the zone represented by the partial shape. Eventually, we observe in practice that these two effects get averaged out by the Gaussian blur.

In practice, DWKS descriptors provide partial information on shape deformation and therefore use additional regularization to obtain more meaningful point-to-point correspondences. In the following we present one possible pipeline, illustrated on Figure 3.3 to obtain point-wise maps from DWKS descriptors.

3.4.4 Matching Pipeline

We suppose being given two similar collections of shapes $(\mathcal{M}_i)_{i=0}^n$ and $(\mathcal{N}_i)_{i=0}^n$, aligned in the sense that the deformation between \mathcal{M}_0 and \mathcal{M}_i is similar to the one between \mathcal{N}_0 and \mathcal{N}_i for all i . Note that this information can be automatically retrieved from unaligned collections of different size using the pipeline from [48]. We also assume to have access to approximate intra-collection maps, which can be computed using known near-isometric shape matching technique.

For simplicity, we equivalently write \mathcal{M} (resp. \mathcal{N}) or \mathcal{M}_1 (resp. \mathcal{N}_1). Our matching pipeline proceeds in four steps, shown in Figure 3.4:

1. Compute shape difference operators of dimension $k_{\mathcal{M}}$ and $k_{\mathcal{N}}$ for each collection, and aggregate DWKS descriptors for each of them in matrices $A \in \mathbb{R}^{n_{\mathcal{M}} \times np}$ and $B \in \mathbb{R}^{n_{\mathcal{N}} \times np}$.
2. Compute an approximate point-wise map using DWKS descriptors.
3. Project the point-wise map into a low dimension functional map, using only a subset of the vertices.
4. Refine the functional map using *e.g.* the ZoomOut [145] algorithm.

In the first step, we use p evenly spaced energy values (e_1, \dots, e_p) and compute descriptors using Algorithm 2.

In the second step, we firstly combine the standard functional map pipeline described in Section 3.3.2 with the commutativity terms introduced in [193, 48]:

$$\begin{aligned} \mathbf{C}^* = \arg \min_{\mathbf{C} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}}} & E_d(\mathbf{C}) + \mu_{dc} E_{dc}(\mathbf{C}) + \mu_l E_l(\mathbf{C}) \\ & + \mu_c E_c(\mathbf{C}) + \mu_a E_a(\mathbf{C}) \end{aligned} \quad (3.8)$$

with $E_d(\mathbf{C}) = \|\mathbf{CA} - \mathbf{B}\|_F^2$ the descriptor preservation term where \mathbf{A} and \mathbf{B} are matrix A and B projected in the spectral basis, $E_{dc}(\mathbf{C})$ promotes commutativity with operators built from individual descriptors described in Section 3.3.2, $E_l(\mathbf{C})$ the standard commutativity with the Laplace-Beltrami Operator $\|\mathbf{C}\Delta^{\mathcal{M}} - \Delta^{\mathcal{N}}\mathbf{C}\|_F^2$, E_c and E_a respectively enforcing commutativity with the conformal and area-based shape difference operators *i.e.* $\sum_i \|\mathbf{CR}_i^{\mathcal{M}} - \mathbf{R}_i^{\mathcal{N}}\mathbf{C}\|_F^2$ and $\sum_i \|\mathbf{CV}_i^{\mathcal{M}} - \mathbf{V}_i^{\mathcal{N}}\mathbf{C}\|_F^2$ where \mathbf{R}_i is the i -th conformal shape difference operator and \mathbf{V}_i the i -th area one. The result \mathbf{C}^* from problem (3.8) is then transformed into a point-wise map $T^F : \mathcal{N} \rightarrow \mathcal{M}$ using the standard method from [159].

In Step 3., we seek to project the point-wise map T into a low-dimensional functional map. To do so, we first discard the fraction α of vertices of \mathcal{N} with the largest descriptor distance defined for vertex j as $d(j) = \|l_{T^F(j)}(A) - l_j(B)\|^2$ where l_m denotes the m -th line of a matrix. This usually ignores vertices near cuts and holes where descriptors are less precise, as seen on Figure 3.4.

In step 4., we refine the low-dimensional functional map using the ZoomOut algorithm [145]. Note that the absence of a refinement algorithm tailored for partial matching makes results particularly sensitive to the refinement parameters. During the first iterations, we ignore vertices of \mathcal{N} belonging to the previous subsample and use the complete set of vertices for the last few iterations.

3.5 Experiments

Parameters. Unless stated otherwise, the parameters for DWKS are fixed across all experiments. The energy values are set to 200 linearly-spaced values between $-\log 3$ and $\log 3$. The standard deviation parameter σ is set to 1.2% of the total range. The size of the computed shape difference operators is set to $k_{\mathcal{M}} = k_{\mathcal{N}} = 50$ and the functional map used to compute them are of size $3k_{\mathcal{M}} \times k_{\mathcal{M}}$ as advocated in [48]. Parameters for optimization problem (3.8) are $\mu_{dc} = 10$, $\mu_l = 0$, and $\mu_a = \mu_c = 10^{-4}$. All the terms of Equation (3.8) have been introduced separately in previous works [159, 153, 48], and we refer the reader to these articles or to the supplementary material for a more in depth discussion on their effect. The low-dimension functional map is a 15×15 matrix for complete shape, and a 15×15 matrix for partial shape with λ the estimated slope of the slanted diagonal of the functional map as described in [181]. We set $\alpha = 20\%$ in the case of partial shapes, and $\alpha = 5\%$ in the case of complete shapes, where the amount of noise is reduced. More details about the parameters values can be found in the supplementary material. The implementations of our method and of the baselines are available at <https://github.com/RobinMagnet/DWKS>.

Cats and Lions. This first experiment uses synthetic data to evaluate the stability of our method in the standard settings used in [48], and show how our pipeline can handle partiality where the matching technique from [48] might struggle. The two collections consist of 10 similar versions of a cat and a lion meshes as those displayed on Figure 3.2. We also manually create a collection of lions cut in half, as seen in

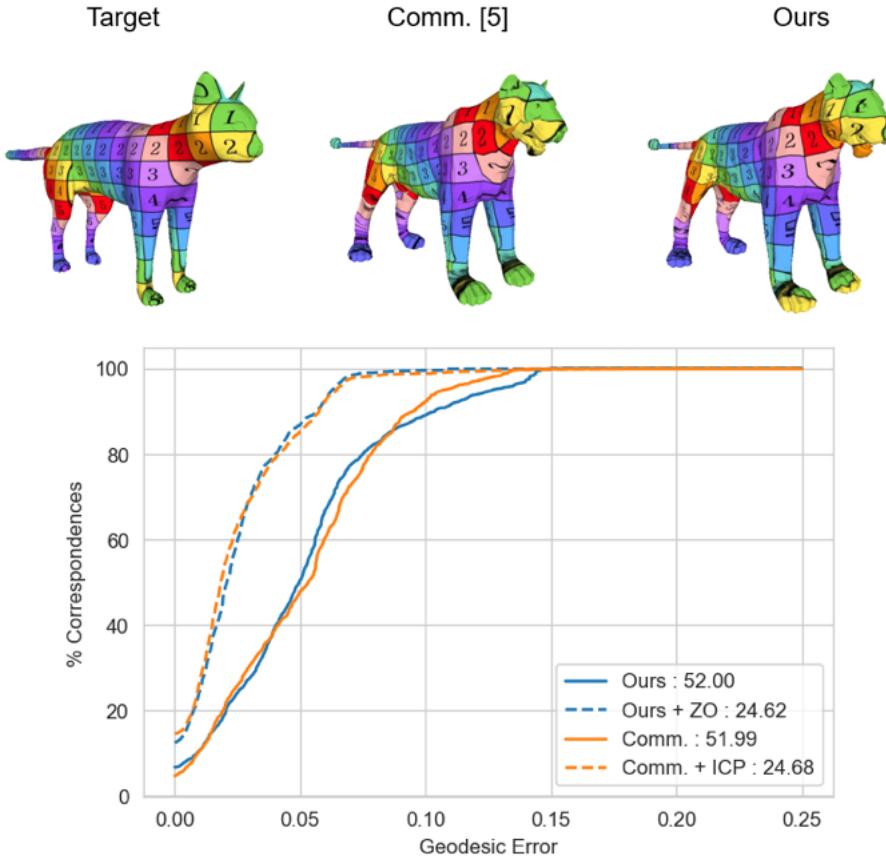


Figure 3.5: Results on the Sumner dataset using the complete shapes. We show our method obtains similar results as [48] in this case.

Figure 3.6. Using standard parameters and subsampling 1 out of 3 descriptors for faster computation, Figure 3.5 shows our method achieves similar accuracy than [48] on complete shapes without the need for a costly SVD solver. Figure 3.6 displays our results in the case of partial before and after the refinement step, compared to those from [48] where we set $\mu_l = 0$ in their objective (3.5) for fairness since the near-isometry assumption fails. We additionally show results obtained by the standard functional map pipeline [153] using WKS descriptors, described in Section 3.3.2.

Synthetic face dataset. We use a similar setting on another synthetic dataset [186] consisting of two collections with 10 faces with multiple expressions. As we wish to focus on real noisy scans in the following experiment, we refer the reader to the supplementary material for illustrations of results on this dataset.

DFaust. We finally tested our pipeline on the DFAust dataset [29], which consists of multiple similar collections of real scans of human shapes, which we see as point clouds. This dataset is especially challenging since real data contains notable holes and outlier vertices, which forces us to use approximate intra-collection maps.

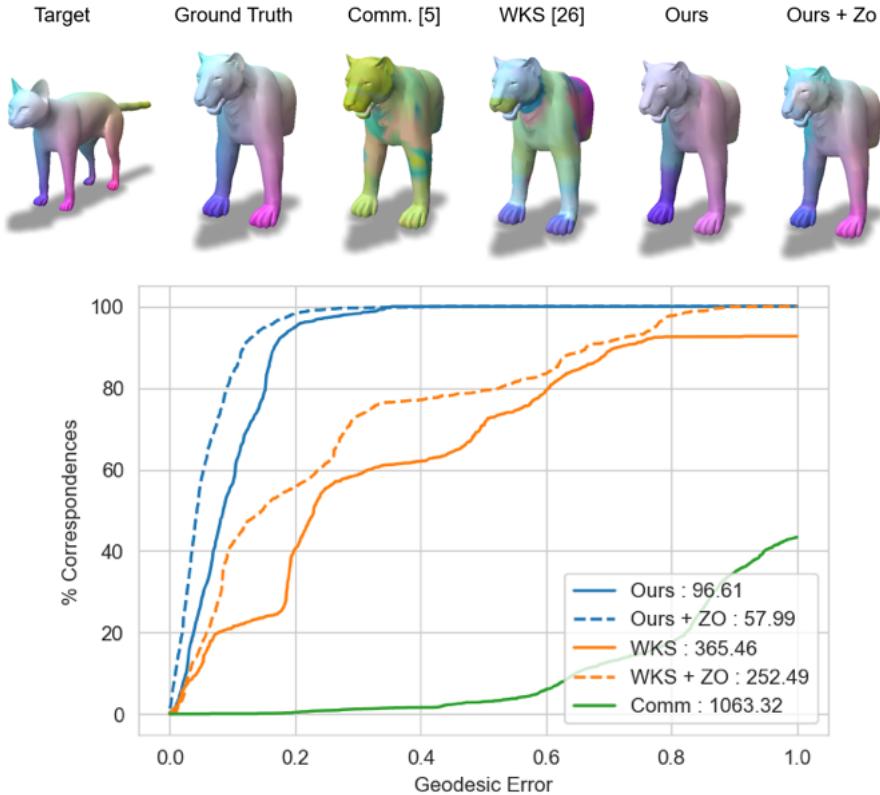


Figure 3.6: Results on the Sumner dataset. While our method doesn't achieve a visually perfect result due to the absence of tailored refinement, it outperforms usual methods. Digits on the legend describe the average geodesic error for each method.

Using a recent formulation of a Laplacian for point clouds [197], we apply our complete pipeline to collections of complete and partial shapes. Note that the method from [48] can be similarly adapted to work with point clouds and still serves as a baseline. In the case of partial shapes, we again do not apply ICP refinement to results from [48] for fairness. A pointwise map obtained when matching the two collections of humans in jumping motion are shown in Figure 3.1. In Figure 3.7, we provide both a qualitative and quantitative evaluation. Our pipeline brings significant improvement to [48] both in the complete and partial setting even without the refinement step, which demonstrates its robustness to noise and applicability to real scenarios. Results from [48], by contrast, do not achieve satisfying results even in the isometric case. Additional quantitative and qualitative results on this dataset as well as comparisons to other baselines are available in the supplementary material.

3.6 Conclusion and future work

In this chapter, we introduced a pointwise descriptor of deformation between surfaces, able to efficiently encode information about local distortion within a collec-

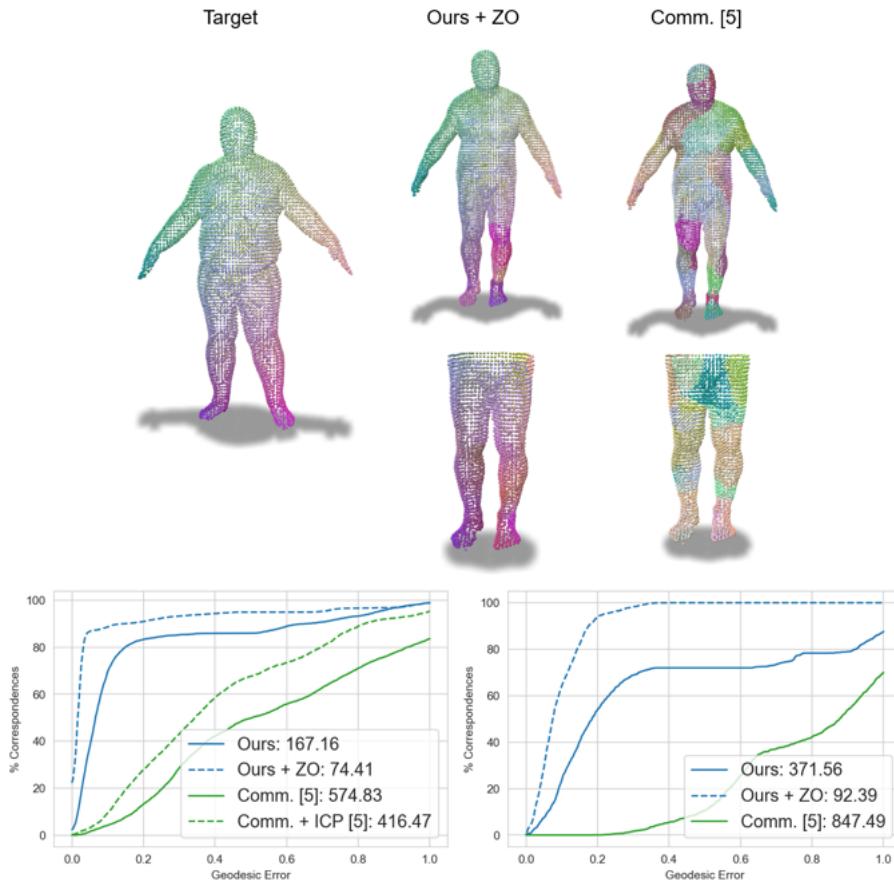


Figure 3.7: Results on the DFAust dataset. We match two collections of 6 meshes representing humans in jumping motion. The bottom row represents accuracy curves of pointwise maps in the complete (left) and partial (right) cases. Numbers in the legend give the average accuracy multiplied by 10^3 .

tion at a vertex-level. Our pipeline enables to leverage the common deformations of meshes and point clouds to compute maps in challenging scenarios, including symmetry and partiality.

Our approach however suffers from some limitations, as it only focuses on intrinsic deformations of shapes. Furthermore, the absence of robust refinement algorithms in the case of partial shapes makes our method very sensitive to the parameters of these algorithms. Finally, while the parameters were set as constants across our experiments, they still might have to be manually set by the user.

In the future, it will be interesting to exploit meaningful *extrinsic* shape difference operators in the vein of [99], and to potentially overcome the choice of a base shape through the introduction of consistent latent spaces [98].

Acknowledgments The authors thank the anonymous reviewers for their valuable comments and suggestions. Parts of this work were supported by the ERC Starting Grant No. 758800 (EXPROTEA) and the ANR AI Chair AIGRETTE.

Assessing Craniofacial Growth and Form Without Landmarks: A New Automatic Approach Based on Spectral Methods

Here we present a novel method for the morphometric analysis of series of 3D shapes, and demonstrate its relevance for the detection and quantification of two craniofacial anomalies: trigonocephaly and metopic ridges, using CT-scans of young children. Our approach is fully automatic, and does not rely on manual landmark placement and annotations. Our approach furthermore allows to differentiate shape classes, enabling successful differential diagnosis between trigonocephaly and metopic ridges, two related conditions characterized by triangular foreheads. These results were obtained using recent developments in automatic non-rigid 3D shape correspondence methods and specifically spectral approaches based on the functional map framework. Our method can capture local changes in geometric structure, in contrast to methods based, for instance, on global shape descriptors. As such, our approach allows performing automatic shape classification and provides visual feedback on shape regions associated with different classes of deformations. The flexibility and generality of our approach paves the way for the application of spectral methods in quantitative medicine.

4.1 Introduction

Trigonocephaly is a puzzling congenital craniofacial malformation secondary to a premature fusion of the metopic suture, located between the two frontal bones. Trigonocephaly is characterized by a triangular, keel-shaped forehead, biparietal widening and hypotelorism [116]. Interestingly, the usual theories on craniofacial growth are of little help to understand the mechanisms leading to this malformation, and there is currently no clear explanation accounting for the striking increase in its prevalence in the last three decades [223]. Trigonocephaly requires surgical correction before one year of age for aesthetic and functional imperatives. In this context, early and reliable diagnosis is key in the management of trigonocephaly [141].

Numerous anthropometric measurements have been proposed in the literature

to help diagnose this condition [107, 38], and also to differentiate trigonocephaly from metopic ridges, a minor anomaly due to an excessive ossification of the metopic suture, without orbital deformation or biparietal widening, that does not generally require surgical correction. Quantifying morphological differences between trigonocephaly and metopic ridges can be hard to perform reliably. Instead, distinguishing between these two conditions is typically performed using purely subjective, qualitative analysis. Furthermore, even though the differences between these two conditions are three-dimensional, most of the metrics proposed in the literature to differentiate them are two-dimensional [26, 25], and few are three-dimensional [47].

The quantification of three-dimensional shapes requires the use of specific statistical approaches such as geometric morphometrics [31, 114, 88]. These approaches are now commonly used in biomedical sciences, to better describe phenotypes, study growth, and evaluate treatment outcomes [187]. Usual morphometric approaches rely on the placement of landmarks, and subsequent analysis of their variability using rigid and non-rigid registration [30, 31, 115, 32, 204, 235, 3]. This procedure presents many indisputable advantages including the preservation of homology of anatomical regions throughout datasets encompassing great inter-specific variation [135, 86, 165], or along different developmental stages [155, 234, 78]. Although intra-operator biases related to the manual placement of reference points exist, they have been extensively studied [11, 229, 55, 233], allowing for their detailed quantification. However, inter-operator biases are still more difficult to assess, and this could be important in the era of big data. Also, classical geometric morphometrics may face some other limitation like missing data and/or topological discrepancies. Several attempts of applying landmark-free approaches to shape assessment have been recently proposed in the literature [24, 188, 220]. Among all 3D shape assessment methods with or without landmarks, spectral techniques [159, 162], originally developed for computer graphics, have rarely been applied to the medical field to date.

Unlike standard geometric approaches, spectral shape analysis goes beyond the 3D geometry of a surface and analyzes *functions* defined on the shapes, which comes with a rich and flexible mathematical framework. In particular, functional maps [159] and their numerous extensions [145, 100] allow to compute highly accurate dense point-to-point correspondences, and can be adapted to a wide range of settings and deformation models [177].

Notable previous works [151, 113, 112, 201, 152] have used partly outdated spectral methods for medical data analysis, and especially the so-called shape DNA descriptors [179]. These methods typically relied on restricted deformation models such as near-isometries, and, importantly, only enabled *global* shape comparison, by associating a single descriptor vector to each shape. Unfortunately, as we demonstrate below, this is insufficient to reliably distinguish subtle differences that might exist across different shape categories.

In contrast, functional maps provide a general and flexible framework that has proven capable of efficiently obtaining accurate fully-automatic *landmark-free local* correspondences between shapes, leading to state-of-the-art results even in challenging settings like partiality [181, 13] or non-isometry [174, 65]. Unlike global shape

embeddings, local correspondences enable more accurate analysis of shape collections, highlighting precise *regions* associated with shape changes, and even building task-specific deformation models. Furthermore, spectral methods can be used to refine noisy initial correspondences obtained through rigid alignment, as we demonstrate in this chapter.

In this work we leveraged these recent state-of-the-art fully automatic shape correspondence methods, which we adapted to fit within a morphometric analysis pipeline to study morphological differences between trigonocephaly, metopic ridges and controls. Our results pave the way for the use of this new set of methods in quantitative medicine.

4.2 Materials and Methods

4.2.1 Overview

Algorithm 3 General pipeline for our algorithm

Require: **Input:** Collection of shapes

- 1: Compute initial correspondences (Sec. 2.4)
 - 2: Refine correspondences (Sec 2.4)
 - 3: Extract a template and deformations (Sec 2.5)
 - 4: Perform deformation analysis (Sec 2.5)
-

We used a template-based morphometric analysis framework, where deformations between the template and each shape were defined using dense vertex-to-vertex correspondences - that is, assigning a point on the target shape for each point on the template. Both the correspondences and the template were obtained automatically using a landmarks-free approach by adapting recent developments in the functional map framework [145, 100]. The proposed pipeline is fast, fully automatic, flexible as it can handle different types of deformations, and provides visual feedback on zones undergoing the most important shape modifications. In the following, Section 4.2.2 presents the dataset, Section 4.2.3 provides some background on functional maps, while Sections 4.2.4 and 4.2.5 describe our extension of spectral methods to morphometrics.

Our work, briefly described on Algorithm 3, builds on a variety of recent works [98, 100], for which we here only provide a detailed description of the necessary modifications. We refer the interested reader to the original publications for a complete description of the methods. A complete implementation of our approach, that can be used by other researchers is available at https://github.com/RobinMagnet/Morpho_FMaps.



Figure 4.1: Visualization of the inner and outer surfaces of a mesh in horizontal section.

4.2.2 Data

This study relies on the reuse of retrospective data acquired during standard patient care and complies to MR-004 (CNIL1818709X, 2018-155, #110, 03.05.2018). All patients were informed of the reuse of their personal data for this specific study. Our dataset included all patients with non-syndromic trigonocephaly that benefited from fronto-orbital advancement at Necker - Enfants Malades Hospital (Assistance Publique - Hôpitaux de Paris), at the National Reference Center for Craniosynostoses and Craniofacial Malformations (CRMR CRANIOST, Filière Maladies Rares TeteCou) from 2004 to 2019 with an available digital pre-operative CT-scan. In addition, all patients diagnosed with metopic ridges managed in the same center during the same period, with an available digital CT-scan, were also included. For metopic ridges, all scans were performed for diagnostic purposes before patients were sent to our center, radiological examination being of little use in this condition in the vast majority of cases. This cohort of patients with metopic ridges thus corresponded to a specific subset of patients that raised diagnostic questions with trigonocephaly, and was interesting in the context of the assessment of a classification method. The final differential diagnosis between trigonocephaly and metopic ridges was based on the expert opinion of the craniofacial surgeons of the National Reference Center. Control age-matched patients were included, with available digital CT-scans performed in the emergency department of the same hospital (for acute headache, soft-tissue infections, epilepsy, or trauma). All control CT-scans were assessed by two independent reviewers (craniofacial surgeon and pediatric radiologist), to ensure that only scans without skull fractures, craniofacial anomalies, structural abnormalities, and of sufficient quality for 3D reconstruction were included. Age and gender were noted for all patients. The dataset consisted in $N = 155$ CT-scans of skulls including $n_t = 85$ patients with trigonocephaly (mean age 219.3 ± 81.4 days), $n_m = 27$

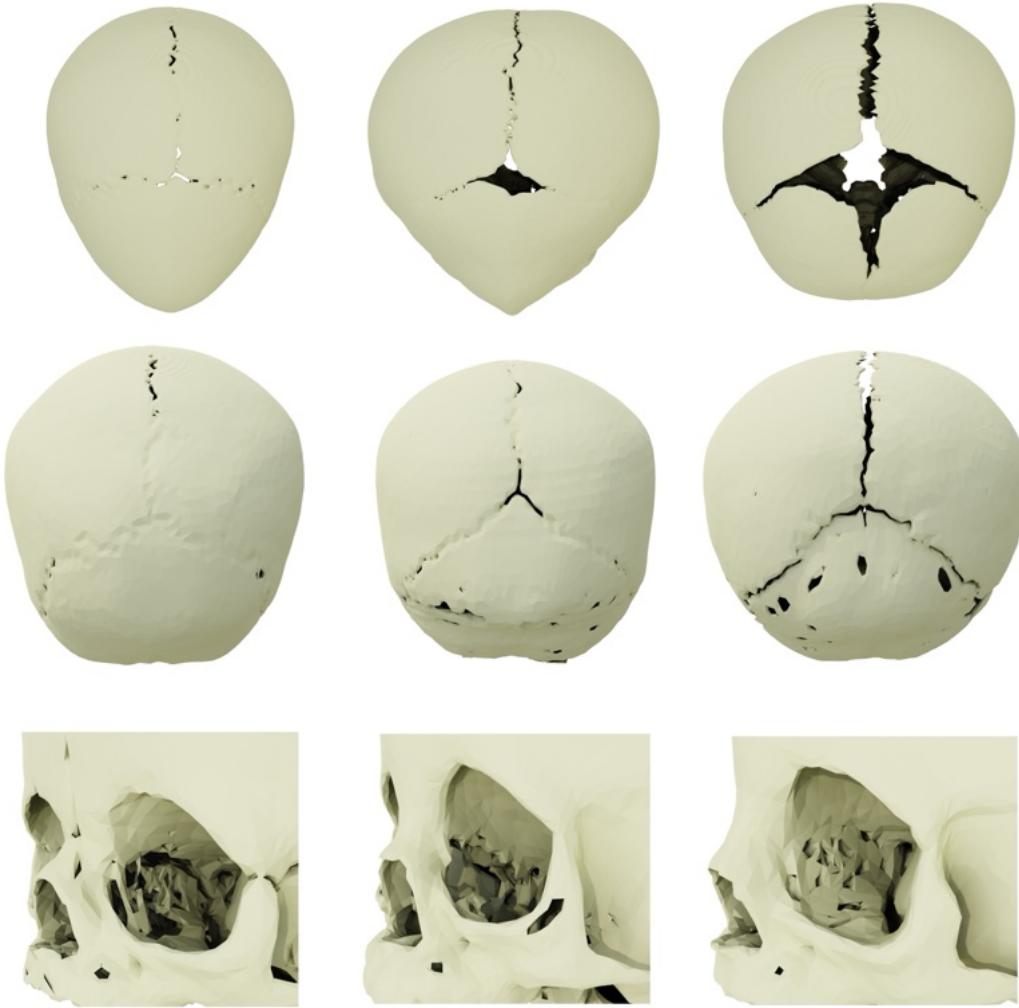


Figure 4.2: The dataset included highly non-isometric deformations (top and middle row) as well as topological dissimilarities around the orbital cavity (bottom row).

patients with a metopic ridge (mean age 379.25 ± 224.7 days), and $n_c = 43$ control patients (mean age 218.7 ± 107.8 days). The detailed age distribution for each group is provided in the supplementary material. The scans were segmented using 3D Slicer [79], and skulls were exported as 3D surface objects. The surface objects generated from segmentation masks consisted in watertight meshes composed of an inner and an outer surface (Figure 4.1). Due to the segmentation process which forced watertightness, scan quality, and variation in ossification across patients, several skulls presented multiple either natural or artificial surface interruptions (especially around the orbital cavity and on the anterior skull base) connecting the two layers of the surface at unusual places. This generated both severe topological dissimilarities between the surfaces due to the addition of several holes, and highly non-isometric transformations that distorted geodesic distances between inner and outer surfaces (Figure 4.2). While addressing these changes can pose challenges for certain shape analysis approaches, using the functional map framework proved

effective in mitigating this issue. In particular, while Section 4.2.4 showed a small adaptation to double-surface for initialization, we did not adapt the method to avoid or explicitly handle topological challenges present in the data.

Each scan in the collection was remeshed to reach roughly 40 000 triangles, then normalized and centered, and isolated components were automatically removed. The eigenfunctions of the Laplace-Beltrami Operator, introduced in Section 4.2.3, were precomputed separately for each shape.

4.2.3 Functional Maps

We based our approach on the Consistent ZoomOut algorithm [100], a landmark-free method, which allows refining noisy initial correspondences in a collection of shapes using functional maps.

Functional maps were introduced in [159] as an efficient and adjustable way to perform shape correspondence. While standard shape matching methods directly optimized for point-wise correspondences between two surfaces, functional maps considered instead transferring *real-valued functions* across shape pairs (Figure 4.3).

More precisely, the key idea of functional maps consisted in encoding functions using a basis on each shape and then representing correspondences as linear operators across basis functions of different shapes. These linear operators could be encoded as compact matrices (functional maps), that intuitively allowed to “translate” across different bases. Crucially the size of these functional maps was independent of the number of vertices on each shape, and only depended on the chosen basis size. In practice, the standard choice was to use the eigenfunctions of the Laplace-Beltrami operator as the functional basis on each shape (Figure 4.4), as they presented a natural generalization of Fourier analysis to general domains. In practice, this meant that the standard functional maps pipeline [162] consisted in first computing $K \approx 50 - 100$ basis functions on each shape, optimizing for a $K \times K$ functional map matrix, and then converting this matrix to a dense point-to-point correspondence. As greater values of K corresponded to more precise point-wise maps, most recent algorithms [145, 100, 177] iteratively refined an initial small-sized (blurry) functional map into a bigger one.

Functional maps have also been used to establish correspondences within *shape collections* [100], which can lead to more accurate results, by exploiting collection-wise consistency constraints. In particular, functional map networks (FMN) consider collections of shapes $(\mathcal{S}_i)_{i=1}^N$, related to each other with functional correspondences, encoded as $K \times K$ matrices, which together define a *graph* of correspondences [150, 98, 100]. Using an initial functional map network, the Consistent ZoomOut algorithm [100] allows establishing *consistent* maps across any shape pair, by constructing a virtual template referred to as a “limit shape”. The limit shape is a purely algebraic construct (intuitively it corresponds to a space where functions from all shapes get averaged out) and does not have a concrete geometric structure. However, Consistent ZoomOut allows building functional maps between this limit shape and every shape in the collection, leading to accurate correspondences between arbitrary shape pairs, even those not present in the original functional map

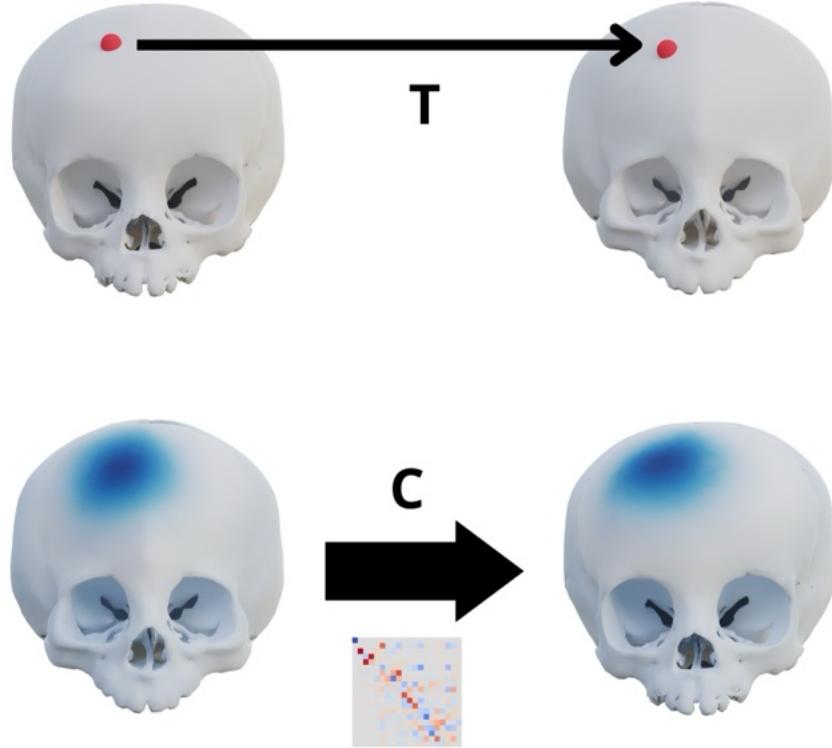


Figure 4.3: Comparison of point-based and functional correspondences. Top: A point on one shape was transferred to another shape using a vertex-to-vertex map T . Bottom: A function was transferred using a functional map C . The image below the arrow is a representation of the entries of the functional map matrix. Note that these two transfers do not require T or C to be ground truth maps.

network.

4.2.4 Computing correspondences

Consistent ZoomOut [100] requires initial correspondences to build the network. These initial correspondences can be very approximate as they are later refined by the algorithm. In this work, we relied on a rigid alignment of the surfaces, which was made possible as all skulls were roughly facing the same direction due to the acquisition process. Note that our method tolerated noisy initial maps and no manually placed anatomical landmarks were required.

Specifically, initial maps between a pair of shapes \mathcal{S}_1 and \mathcal{S}_2 were computed as follows. We first applied an Iterative Closest Point (ICP) algorithm [22] for approximate rigid alignment. Using this initialization, each vertex $y \in \mathcal{S}_2$ was then associated to the nearest vertex $x = T(y) \in \mathcal{S}_1$ with a coherent normal, i.e., $y \rightarrow x = \arg \min_{x_i \in \mathcal{S}_1, \langle n_{x_i}, n_y \rangle > 0} \|x_i - y\|^2$. This procedure ensured that even if the alignment was not clinically relevant, the inner and outer surfaces (described in Section 4.2.2)

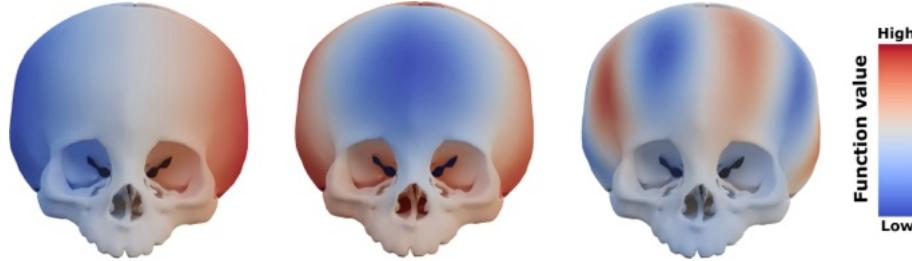


Figure 4.4: Examples of low frequency (*i.e.* slowly varying) eigenfunctions of the Laplace-Beltrami operator.

were differentiated as schematically explained in Figure 4.5. In Figure 4.5, without normal consistency, point x_1 , which lied on the outer surface of \mathcal{S}_1 , was mapped to point y_2 on the inner surface of \mathcal{S}_2 , and x_2 was also mapped to y_2 . The obtained correspondences T could then be transformed into a functional map \mathbf{C} of size $K \times K$. Note that these initial maps were of poor quality as they relied on an ICP alignment, which may have poorly aligned clinically relevant areas of the skull. Furthermore, the normal consistency procedure could have failed in zones with high curvature. We thus refined them using Consistent ZoomOut before performing downstream analysis tasks.

We have significantly improved the speed of both the initialization and refinement steps by using only a small subset of 2000 randomly chosen pairs out of the approximately 12000 possibilities. The initial functional correspondences were of size $K \times K$, with $K = 50$, and using a spectral upsampling step of $k_{step} = 5$, these maps were refined until dimension 100 was reached. The refined network then provided correspondences between any two shapes in the collection, even if the two shapes were not directly connected by an edge inside the network. These correspondences were of high quality and significantly improved those obtained by simple ICP alignment, and were then used for our subsequent morphometric analysis.

4.2.5 Deformation Analysis

Unsupervised Analysis

As we sought to perform template-based morphometric analysis but only had access to a “limit shape” [98] that did not correspond to any known geometry, we designed a method to extract an actual shape from the collection that was the most intrinsically similar to the limit shape. We defined the intrinsic dissimilarity between shape i and the limit shape as $\|D_c^{(i)} - I\|_{H_0^1}^2$ where $D_c^{(i)}$ was the conformal characteristic shape difference [186, 98], a $K \times K$ matrix which captures the differences between the limit shape and shape i , I was the identity matrix, and $\|\cdot\|_{H_0^1}$ the norm in the Sobolev space [186]. The shape that minimized this dissimilarity metric was called \mathcal{T} , and can be conceived as a median shape of the collection. Note that correspondences between \mathcal{T} and each shape in the collection were then available thanks to the functional map network, mentioned above.

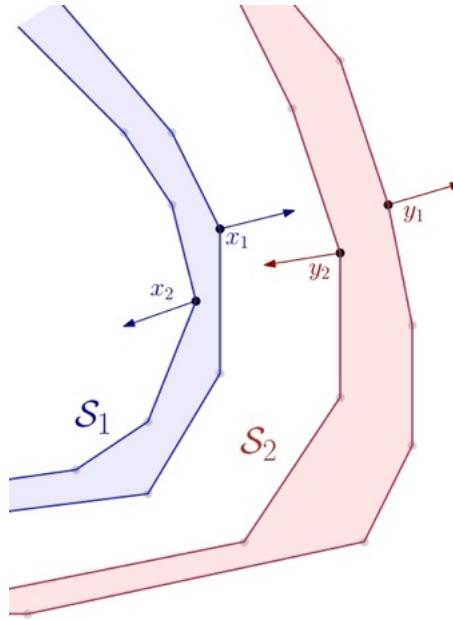


Figure 4.5: Local significance of the normal consistency scheme for inner and outer surfaces.

Using shape \mathcal{T} , acting as a template, we generated for each shape \mathcal{S}_i a vertex-wise deformation field $d^{(i)}$ deforming \mathcal{T} into \mathcal{S}_i . This deformation was defined by first rigidly aligning the two shapes using the computed correspondences, and then using the vertex-wise displacement in 3D provided by the same correspondences. We further smoothed the deformation fields following the approach from [70, 71], projecting $d^{(i)}$ into the truncated Laplacian basis using K_d eigenvectors, with $K_d = 100$

Using deformation fields as embeddings for shapes, we performed a Principal Component Analysis (PCA) to extract principal components $(D_j)_{j=1}^{p_{\text{pca}}}$, which we here called principal deformations. Note that projecting the deformation fields into these components provided a new reduced embedding $\tilde{d}^{(i)} \in \mathbb{R}^{p_{\text{pca}}}$ for each shape \mathcal{S}_i . Principal deformations can be thought of as deformations of the template \mathcal{T} to \mathbb{R}^3 , and could be visualized by applying deformation tD_j to the template \mathcal{T} for various values of $t \in \mathbb{R}$. Note that we informally referred to as a positive (resp. negative) deformation the visualization of tD_j with $t > 0$ (resp. $t < 0$). This visualization provided a qualitative feedback in addition to the standard projection on the first components.

Supervised Analysis

The procedure up to the current point was fully unsupervised in the sense that no manual intervention or labels were used. In order to validate the previous analysis, we trained a logistic regression model to predict to which group a skull belonged, using only $(\tilde{d}^{(i)})_{i=1}^N$ as input. This step demonstrated that the fully unsupervised analysis generated a meaningful representation for each shape.



Figure 4.6: Visualization of computed correspondences between two random pairs of shapes in the collection. Top : correspondence between two nearly isometric scans. bottom : correspondences between highly non-isometric scans. Vertices on the left are given RGB colors depending on their XYZ coordinates, and corresponding vertices between left and right are given the same color.

For each class, the logistic regression learned a vector β so that a high value of $\langle \tilde{d}^{(i)}, \beta \rangle_2$ meant that $\tilde{d}^{(i)}$ was predicted to belong to the given class. For visualization purposes, we built the reference deformation $D(\beta) = \sum_{j=1}^{p_{\text{pca}}} \beta_j D_j$ which corresponded to the “canonical” deformation of the template associated with the given class. That is, deformations of the template which aligned the most with $D(\beta)$ were predicted to be part of the given class, and thus the deformation $D(\beta)$ provided a visual representation of how the classification decision for each class is made.

We performed a PCA and computed logistic regression models using the first $p_{\text{pca}} = 10$ components. We fitted 5 types of logistic regressions : (1) trigonocephaly vs controls (**C-T**), (2) (controls + metopic ridges) vs trigonocephaly (**(C+M)-T**), (3) controls vs metopic ridges vs trigonocephaly (**C-M-T**) and (4) metopic ridges vs trigonocephaly (**M-T**) (5) metopic ridges vs controls (**C-M**). We evaluate these regressions using a 5-fold cross validation, weighting samples in order to compensate for the non-uniform distribution of labels.

We compared our results to another fully unsupervised baseline, by applying a similar logistic regression to the standard Shape-DNA descriptor [179]. This global shape descriptor embeds the shape as the list of the first K_{dna} normalized eigenvalue of its Laplace-Beltrami operator. We tested several values of K_{dna} , and obtained the best results using 100 eigenvalues.

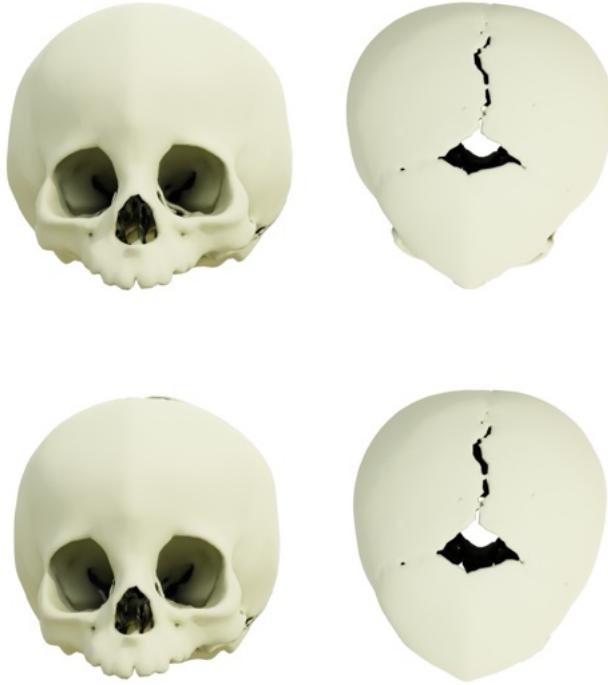


Figure 4.7: Top : Surrogate template chosen as the closest to the limit shape. Bottom : Corrected template by applying the average deformation to the template

4.3 Results

4.3.1 Functional Map Network

The refined networks allowed us to obtain per-vertex correspondences between any pair of shapes (Figure 4.6) within the collection, which improved upon those obtained by the initial ICP alignment. In Figure 4.6, we visualized the computed point-to-point correspondences. Specifically, we first associated color (r, g, b) values based on the (x, y, z) coordinates of each point on the leftmost shape. We then transferred these color values using the computed correspondences onto the rightmost shape. Thus, points with same (r, g, b) values are seen as corresponding. Note the overall smoothness of the transferred colors. This was a non-trivial task, and obtaining visually satisfying correspondences was a necessary condition for quality.

4.3.2 Extrinsic analysis

Unsupervised Analysis.

To perform our analysis, we first extracted the implicit template \mathcal{T} (Figure 4.7, top), which can be deformed smoothly into any shape in the collection using the network (Figure 4.8). Finally, adding the average deformation to the template

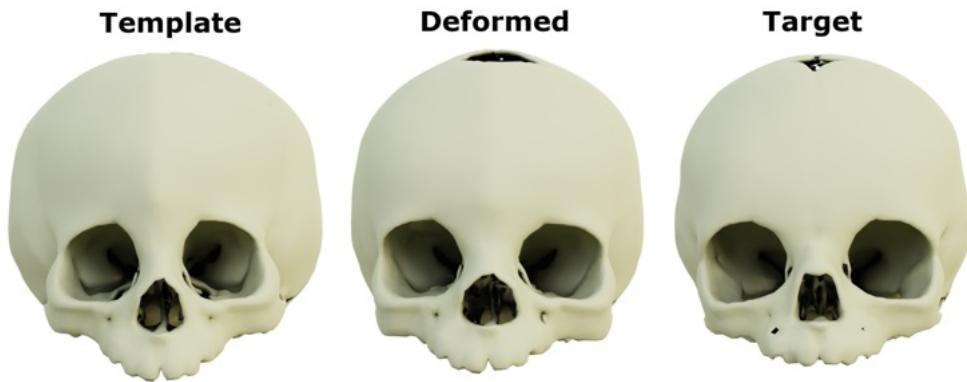


Figure 4.8: Example of the computed deformation between the template and a shape from the dataset.

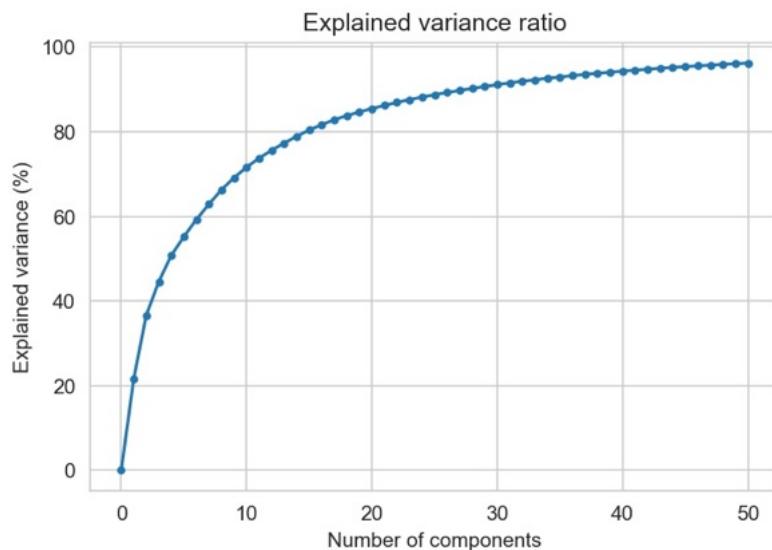


Figure 4.9: Cumulative explained variance ratio of the PCA.

allowed the correction of its geometry (Figure 4.7, bottom). This correction step was performed for visualization purposes only, as the PCA used in our quantitative analysis automatically factored this average deformation out.

Logistic Regressions.

We provide average True Positive Rates (TPR) of the 5 types of logistic regressions on the validation set in 4.1. For comparison, we added results of performing similar regressions using the standard Shape-DNA descriptor [179] as input.

The principal components of the PCA carried deformations of the template and were referred to as principal deformations (Section 4.2.5). The cumulative variances explained by the principal components were displayed in Figure 4.9, while Figure 4.10 showed the projection of the dataset on the fourth and seventh components, which displayed more separation than the first two. The theoretical de-

Table 4.1: Average True Positive Rates (TPR) after cross-validation for multiple logistic regression objectives, differentiating Control against Trigonocephaly (C-T), Control and Metopic Crests together against Trigonocephaly ((C+M)-T), and eventually Control against Metopic Crests against Trigonocephaly (C-M-T).

| Methods | C-T | (C+M)-T | C-M-T | M-T | C-M |
|-----------|--------------|--------------|--------------|--------------|--------------|
| Shape-DNA | 66.0% | 64.4% | 46.5% | 69.2% | 53.3% |
| Ours | 92.9% | 91.1% | 76.4% | 85.3% | 72.2% |

formation associated with the positive and negative values of each of the first two principal components were displayed in Figure 4.11. As explained in 4.2.5, the logistic regression learned for each class a representative deformation of the template in order to classify each skull which enables us to generate a “typical” skull for each cohort, built by exaggerating the representative deformation. In 4.12, we displayed such “typical” skulls for each class next to the template.

Computational efficiency

After pre-processing, refining the functional map network made of 155 scans took 40 minutes, the main bottleneck lying in the eigendecomposition of a large sparse matrix. Note that this can be improved by reducing the number of provided initial correspondences. The downstream analysis takes a few minutes to compute.

4.4 Discussion

We showed (Table 4.1) that a landmark-free approach, based on state-of-the-art spectral analysis tools, can reliably distinguish trigonocephaly from normal skulls and skulls with metopic ridges. The specific diagnosis of metopic ridges was not straightforward, especially compared to the control cohort, but the clinical relevance of our results consisted in the fact that ridges were separated from trigonocephaly, which is the main practical issue when managing patients with “triangular foreheads”. In contrast, the Shape-DNA based approach failed to make reliable predictions on the validation set. This is because the distinctions between different classes (and, in particular, trigonocephaly and metopic ridges) pertains to local geometrical characteristics. Thus, a method, based on a global shape descriptor, such as the Shape-DNA, is unable to reliably extract the signal necessary for such a distinction. We identified two primary factors contributing to incorrect classification in our experimental setup. Firstly, cases where the surfaces exhibited significant issues, such as substantial missing or additional parts, were found to lead to misclassification (Figure 4.13). Secondly, incorrect correspondences between surfaces resulted in outliers in the deformations, and this issue was observed to be influenced by mesh quality, particularly in the orbital cavity region.

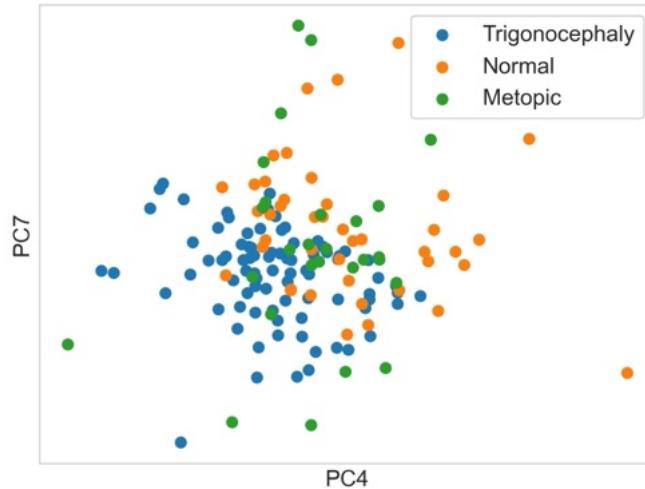


Figure 4.10: Projection of the dataset on the fourth and seventh principal components.

We also showed that our method could produce a clear representation of the classification outcomes (4.12). Notably, we observed that trigonocephaly was detected when associated to pronounced deformations in both the forehead and the orbit regions. In contrast, the deformation associated with metopic ridges lied between the one for the control group and for the trigonocephaly group, with a relatively minimal impact on orbital morphology, consistent with expectations from usual clinical observations. Nevertheless, it is important to be cautious when analyzing such visualizations, as they only indicated the overall direction in which each deformation were projected. Indeed, following 4.12, the model seemed to heavily rely on age as a distinguishing factor. However, we demonstrated in the supplementary material that this feature did not offer a reliable signal.

We also highlight that our method directly ran on raw surfaces without specific manual processing such as closing holes or smoothing, and did not require any manual intervention such as annotated regions or landmarks. While this results in several challenges for many approaches, the efficiency of our method in this scenario hints at its potential generality. These lower requirements, associated to a rather simple underlying statistical model justified that the true positive rates for our classifications were lower than similar works using this additional information.

While spectral methods have already been applied to biomedical data [151, 113, 112, 201, 152, 143], the pipeline proposed here is the first attempt to leverage the recent functional maps-based methods for fully automatic dense point-to-point correspondence computation, to a clinical question. Landmark-free approaches in general are highly relevant in morphometrics, especially in a clinical context where simple and objective tools have to be provided to medical practitioners. Spectral approaches seem promising in designing shape-based diagnostic tools for craniofacial surgeons, with the possibility to easily extend the dataset without major computational costs

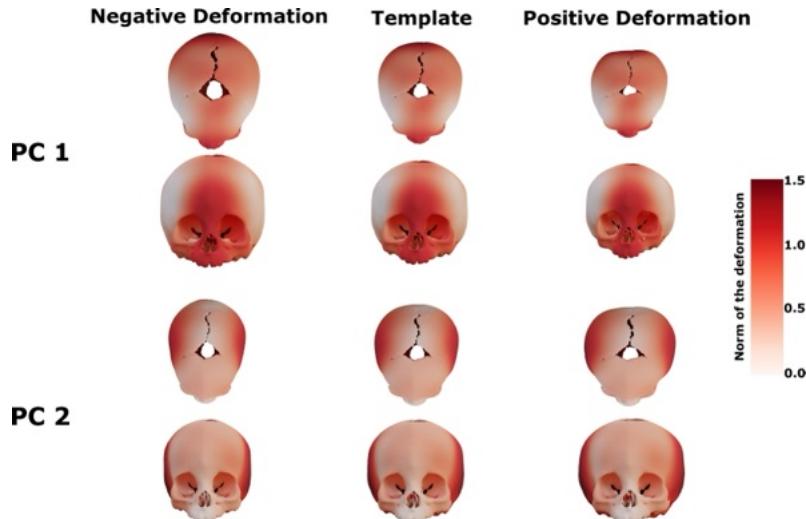


Figure 4.11: Visualization of the first principal deformations D_j for $j = 1, 2$. For each component, the leftmost and rightmost shapes displayed the two opposite deformations. The color indicated the norm of the displacement D_j at each vertex.

and to include multiple diagnostic categories, as illustrated here. Regarding the diagnosis of trigonocephaly, methods based on 3D data proposed in the literature are still scarce [27, 189, 23], and spectral approaches could be an interesting alternative for designing diagnostic tools, assessing growth in temporal series, and for evaluating results of medical or surgical treatments.

More broadly, this pipeline also appears interesting as it provides a very general framework bringing together functional maps and morphometrics. It leverages first the flexibility and generality of spectral methods, allowing to work at high calculation speeds and processing entire collections efficiently, on datasets involving a great variety of shapes undergoing potentially major deformations. Our approach furthermore offers a clear visual feedback of the results, highlighting the main zones of deformations across cohorts and enabling direct classification of diagnostic, overcoming the burden of designing a template or manually specifying landmarks or keypoints. Finally, as very minimal information about skulls themselves was used, namely the rough pre-alignment and the normal consistency scheme, the complete pipeline can easily be adapted to new collections of shapes with little adjustment.

4.5 Conclusion

In this work we applied recent spectral shape matching developments to the detection of trigonocephaly in young children. Our method is fully automatic, and especially does not require manual placement on landmarks on CT scans, and can be generalized to other diseases. Results clearly outperform standard spectral methods as they detect local deformations of the skulls, which motivates the use of such new methods to morphometric analysis.

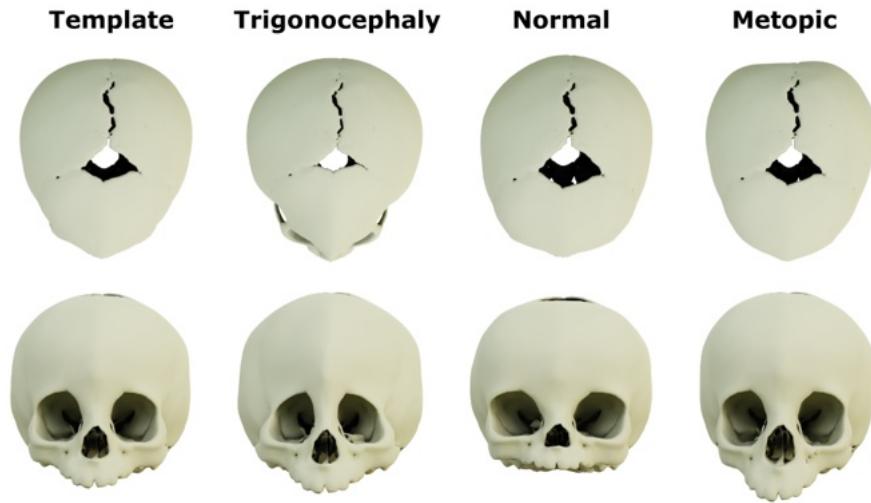


Figure 4.12: ‘Typical’ skull for each class, as detected by the logistic regression. The deformation was exaggerated for visualization purposes. Note that these skulls don’t represent an actual patient, but the deformations on which each embedding is projected.

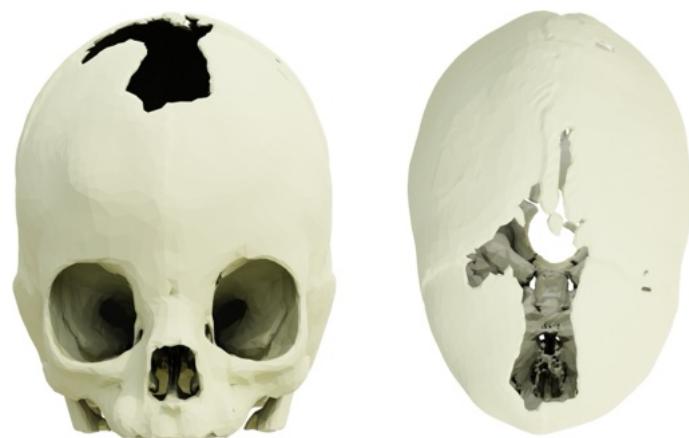


Figure 4.13: Example of misclassification due to a large missing part. This patient was predicted to be a normal skull, while it belongs to the metopic group.

Acknowledgements

Parts of this work were supported by the ERC Starting Grant No. 758800 (EXPROTEA) and the ANR AI Chair AIGRETTE. We acknowledge Dr Giovanna Paternoster, Dr Éric Arnaud and Dr Syril James (craniofacial surgeons), Jean Feydi (theoretical morphometrics), Pr. Nathalie Boddaert (medical imaging), Pr. Sébastien Laporte and Pr. Mehran Moazen (biomechanics), Dr Lara van de Lande (maxillofacial surgery) for their valuable help.

Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization

We introduce pointwise map smoothness via the Dirichlet energy into the functional map pipeline, and propose an algorithm for optimizing it efficiently, which leads to high-quality results in challenging settings. Specifically, we first formulate the Dirichlet energy of the pulled-back shape coordinates, as a way to evaluate smoothness of a pointwise map across discrete surfaces. We then extend the recently proposed discrete solver and show how a strategy based on auxiliary variable reformulation allows us to optimize pointwise map smoothness alongside desirable functional map properties such as bijectivity. This leads to an efficient map refinement strategy that simultaneously improves functional and point-to-point correspondences, obtaining smooth maps even on non-isometric shape pairs. Moreover, we demonstrate that several previously proposed methods for computing smooth maps can be reformulated as variants of our approach, which allows us to compare different formulations in a consistent framework. Finally, we compare these methods both on existing benchmarks and on a new rich dataset that we introduce, which contains non-rigid, non-isometric shape pairs with inter-category and cross-category correspondences. Our work leads to a general framework for optimizing and analyzing map smoothness, both conceptually and in challenging practical settings.

5.1 Introduction

Shape correspondence is a fundamental task in Geometry Processing, acting as a building block for many downstream applications [224, 188, 59]. One of the key challenges in designing a successful general-purpose shape matching method is the choice of the objective function that should promote high quality correspondences and, at the same time, be easy enough to optimize in order to be applicable on complex, densely sampled geometric objects.

A widely acknowledged desirable objective in non-rigid shape matching is *smoothness*, which intuitively promotes local consistency or continuity of computed correspondences, while being less restrictive than, e.g., isometries or conformal maps. Several works have incorporated smoothness into the map computation pipelines either via auxiliary energy terms [77], or by structuring the search space privileging continuous, often low frequency, correspondences or deformation fields, e.g., [69, 70].

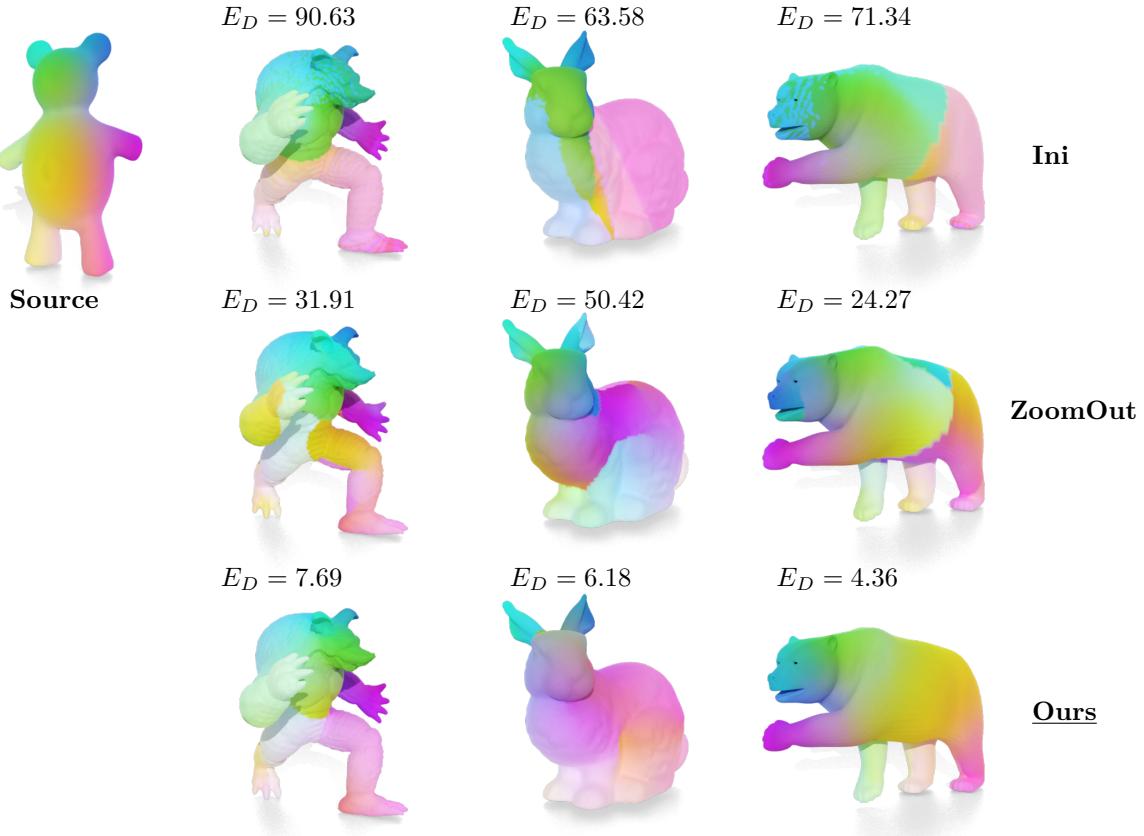


Figure 5.1: Our method can deal with noisy inputs and produce high-quality and smooth pointwise maps for non-isometric shape pairs. As a comparison, ZoomOut [145], the current state-of-the-art refinement method, cannot explicitly control the map smoothness and can have large discontinuous patches in the obtained maps. We report the smoothness metric E_D for each map.

Despite the utility of smoothness as a supervising signal in map computation, existing approaches can either be difficult to scale to dense meshes or are incorporated in an ad-hoc manner. Moreover, there is no coherent framework for comparing different existing strategies for promoting map smoothness using a single consistent computational and conceptual formalism.

In this chapter we focus on the functional map framework, which was originally proposed as a tool for near-isometric shape matching [159] and has since then been significantly extended to different tasks [186, 98] and correspondence models [181, 117, 60], among many others. The key advantages of this framework are its efficiency and flexibility. The efficiency of functional maps-based approaches stems from representing maps as small matrices using a reduced basis, which leads to small-scale optimization problems. At the same time, this framework is flexible and can incorporate a wide range of desirable constraints using simple linear algebraic formulations, e.g., [74, 153, 162].

Although originally functional map-based methods focused on constraints in the functional (spectral) domain, recent works have started to highlight and exploit links that exist between pointwise and functional map representations, while lever-



Figure 5.2: **DeformThings4D-Matching Dataset.** We construct a new dataset for non-isometric shape matching based on the DEFORMTHINGS4D [124]. We show some example humanoid shapes and visualize the cross/inter-category correspondences via color transfer. Note that the shapes in the same category are remeshed independently (zoom in to see the mesh wireframes).

aging the strengths of both [174, 77, 177]. Specifically, a recent discrete optimization scheme was proposed in [177], demonstrating that many desirable map properties can be optimized directly in the pointwise map representation. Unfortunately, while the class of energies considered in [177] covers many existing functional map objectives, such as bijectivity or commutativity with the Laplacian, it does not address desirable pointwise map properties such as map smoothness. This can lead to local inconsistencies, such as discontinuous mapped patches, thus severely limiting the utility of the computed maps in practice.

In this chapter, we introduce a novel method that allows to explicitly promote pointwise map smoothness within the functional map framework. Our method is based on, first, formulating smoothness as the optimization of the Dirichlet energy of the pointwise map, and second, an iterative method for solving this energy optimization by extending the method introduced in [177]. This allows our approach to be used alongside other desirable objectives, while explicitly promoting smooth and locally consistent maps. We therefore both extend the scope of discrete map optimization to new energies not covered in [177] and use this insight to develop an efficient non-rigid shape matching approach that directly promotes pointwise map smoothness.

In addition to introducing a novel method for promoting smooth maps within the functional maps framework, we also investigate multiple previous approaches for computing smooth maps in different settings [77, 70, 209, 9] and show how they can be interpreted as variants of each other and thus compared within a unified formalism. This allows us to design a *family* of different approaches, parametrized by the choice of the smoothness energy and its associated optimization strategy. We propose a coherent formalism within which various energies can be compared and demonstrate their relative utility in different settings. Finally, we observe that most public datasets focus on near-isometric pairs, making it non-trivial to evaluate accuracy and smoothness in more realistic scenarios, which can involve diverse and non-isometric shapes. To fill this gap, we introduce a new challenging dataset based on DEFORMTHINGS4D [124], but with additional cross-category ground truth maps (Fig. 5.2). We use this dataset alongside existing benchmarks in a comprehensive comparison of various approaches to computing smooth correspondences. To summarize, our key **contributions** include:

1. We show how pointwise map smoothness can be formulated and optimized

within the functional map framework, by extending the discrete solver proposed in [177].

2. Based on this construction, we introduce a simple and effective map refinement method that is both computationally efficient and leads to high-quality results in non-isometric settings (Fig. 5.1).
3. We show how several previously proposed methods are intimately related both to our approach and within themselves, and propose a coherent framework, allowing us to directly compare ways to promote smoothness within a consistent formalism and computational strategy.
4. We construct a new dataset for non-rigid shape matching tasks with inter-category correspondences for animal shapes, and inter-/cross-category correspondences for humanoid shapes that are independently remeshed.

5.2 Related Work

In this section, we briefly review the previous works of shape matching, commonly used map evaluation metrics, and various map solvers, that are most related to this work. We refer to recent surveys [59, 188, 24] for more thorough discussions of shape matching.

Shape Matching Our work focuses on the problem of shape matching, that looks for dense correspondences between two non-rigid 3D shapes. One solution to shape matching is to solve for correspondences directly by minimizing an explicit and carefully designed energy [34, 94, 158], which can lead to complex combinatorial problems with high computational complexity. An alternative solution is to find correspondences between parametric representations, where the input shapes are mapped into a canonical domain [125, 8, 7]. Our work is based on the functional map representation [159, 162], which computes correspondences between functions defined on the shapes. Different regularizers have been proposed to promote the accuracy of functional maps [153, 154, 174, 84, 230, 231]. Computing a functional map is usually reduced to solving a least-square system, which has a relatively low computation cost, but recovering a point-wise correspondence from the computed functional map is error-prone [180, 75, 177]. To further improve the accuracy of the recovered point-wise correspondences, different refinement methods have been proposed as a post-processing step [208, 136, 227, 226]. A common technique for map refinement in the functional maps framework is to iteratively update functional maps and the underlying pointwise maps according to different energies, such as Dirichlet energy and bijectivity [159, 77, 174, 145, 175, 177]. In this work, we present a new refinement method that can robustly deal with noisy input and efficiently produce smooth maps in the functional maps framework.

Metrics for Map Quality Evaluation Different criterion have been taken into consideration to evaluate map quality, which are incorporated into map computation. The most commonly-used metric is the map accuracy, which is measured by comparing the geodesic distance between the mapped position and the pre-specified ground-truth position. Some previous work [9, 136, 84] adopt a landmark term to enforce map accuracy. To achieve a fully automatic solution, other metrics such as smoothness, bijectivity, conformality, and coverage are considered for map optimization other than accuracy which needs manually specified landmarks. For example, Reversible Harmonic Maps [77] proposes to optimize the Dirichlet energy together with the bijectivity of the pointwise maps. Smooth Shells [70] adopts the ARAP energy [209] to compute a smooth deformation field, which potentially leads to a smooth pointwise map. [109] blends across multiple maps to get a smooth one. [174] proposes heuristics to improve the bijectivity, smoothness, and coverage of the pointwise map in both spatial and spectral domain. In this work, we observe how several previous proposed approaches are closely related in formulating map smoothness. We show different variants can be compared in a coherent way within a consistent formalism.

Map Solver Previous methods adopt different search space for maps and hence need different solvers. For example, some work [80, 208, 66, 164] solve for maps that are represented by doubly stochastic matrices. Functional maps framework [159, 153, 154, 174] usually solves a least-square system for functional maps. Quadratic-splitting technique [75, 77] is also used to solve vertex-to-point (also called precise) maps. [177] introduces a discrete solver to optimize commonly used functional map energies constrained on the proper functional maps, which is a subset of functional maps that are associated with pointwise maps. In this work, we introduce map smoothness into functional map pipeline and present an efficient algorithm to minimize the smoothness which extends the scope of discrete solver.

5.3 Notation & Background

Notation Given a triangle mesh $\mathcal{S} = (X, F)$ with the vertex positions X and face set F , we denote the cotangent weight matrix by W and the diagonal lumped mass matrix by A [146]. By solving the generalized eigenvalue problem $W\phi_j = \lambda_j A\phi_j$, we can obtain the Laplace-Beltrami basis Φ by collecting the first k eigenfunctions as columns, i.e., $\Phi = [\phi_1 \dots \phi_k]$ and the corresponding eigenvalues in a diagonal matrix, denoted as $\Delta = \text{diag}(\lambda_1 \dots \lambda_k)$. We then have $\Phi^\top A \Phi = I$. A pointwise map is denoted as $\Pi_{ij} : \mathcal{S}_i \rightarrow \mathcal{S}_j$, where the subscript indicates the map direction. Specifically, $\Pi_{ij} \in \{0, 1\}^{n_i \times n_j}$ (n_i is the number of vertices in \mathcal{S}_i) is a binary matrix indicating the correspondences between the two shapes. For example, if the p -th vertex on \mathcal{S}_i is mapped to the q -th vertex on shape \mathcal{S}_j , we then have $\Pi_{ij}(p, q) = 1$ and $\Pi_{ij}(p, t) = 0$ for $\forall t \neq q$.

Functional Maps Framework The goal of shape matching is to find a semantically meaningful and continuous pointwise map for a given shape pair. In this work, we follow the functional map framework [159] and encode a point-wise map as a linear transformation (called functional map) in the Laplace-Beltrami basis. Specifically, for a pointwise map $\Pi_{ij} : \mathcal{S}_i \rightarrow \mathcal{S}_j$, the associated functional map is given as $C_{ji} = \Phi_i^\dagger \Pi_{ij} \Phi_j$. Note that C_{ji} is a pull-back linear operator that maps functions on shape \mathcal{S}_j to functions on shape \mathcal{S}_i . In the original pipeline [159], a functional map is computed by solving a least-squared system in the continuous linear operator space, i.e., $C_{21} = \arg \min_{C \in \mathbb{R}^{k_1 \times k_2}} E(C)$, where $E(\cdot)$ is a functional map energy that preserves input descriptors or landmarks, surface area or angles, multiplicative operators, or shape orientation [159, 154, 174, 95]. Solving for a function map in the unconstrained search space simplifies the optimization problem, but can lead to errors when converting the computed functional map to a pointwise one [180, 75, 177]. Thus, additional post-processing techniques are used to improve the quality of the pointwise maps [174, 77, 145, 164].

Discrete Optimization A recent work [177] has proposed a *discrete solver* for functional map pipeline which constrains the optimization problem to the space of *proper functional maps*. Specifically, the functional map, $C_{21} = \arg \min_{C \in \mathcal{P}_{21}} E(C)$, is solved in a discrete search space $\mathcal{P}_{21} = \{C_{21} \mid \exists \Pi_{12} \text{ s.t. } C_{21} = \Phi_1^\dagger \Pi_{12} \Phi_2\}$, i.e., the set of functional maps arising from *some* pointwise correspondence. The general strategy to solve this constrained problem, advocated in [177] mimics the Augmented Lagrangian methods with variable splitting [81] and consists of the following two main steps: **(i)** reformulate the energy $E(\cdot)$ by making C_{21} and Π_{12} independent variables, and adding a *coupling* term:

$$E_{\text{couple}}(C_{21}, \Pi_{12}) = \|C_{21} - \Phi_1^\dagger \Pi_{12} \Phi_2\|_F^2, \quad (5.1)$$

(ii) iteratively solve for C_{21} and Π_{12} with the other variables fixed. This approach is shown to be efficient and leads to high-quality and well-regularized functional maps. Key to the success of this strategy is the ability to reformulate the given functional map energy so that the resulting optimization problems for C_{21} and Π_{12} in step (ii) can be solved in closed form. In [177], a range of energies is considered including bijectivity, landmarks preservation, orthogonality, and Laplacian commutativity.

Dirichlet Energy Given two Riemannian manifolds \mathcal{S}_1 and \mathcal{S}_2 , the Dirichlet energy of a map $f : \mathcal{S}_1 \rightarrow \mathcal{S}_2$ is defined as $E_D(f) = \frac{1}{2} \int_{\mathcal{S}_1} \|df\|^2 d\mu_{\mathcal{S}_1}$, with df the map differential, which intuitively acts as a measure of the stretch induced by the map (see, e.g., [77] for a discussion). A *smooth* map f is therefore characterized as being a minimizer of the Dirichlet energy. In the discrete setting, a map $f : \mathcal{S}_1 \rightarrow \mathcal{S}_2$ can be seen as a function between the two surface embeddings (i.e., $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$) and is assumed to be affine on each face. We can then define the discrete Dirichlet energy [167]:

$$E_D(f) = \sum_{(x_i, x_j) \in \mathcal{E}(\mathcal{S}_1)} w_{ij} \|f(x_i) - f(x_j)\|^2, \quad (5.2)$$



Figure 5.3: Previous methods focus on improving map accuracy and do not have explicit control over the map smoothness. Here we show an example of a non-isometric pair. We report the Dirichlet energy (E_D) of maps after refinement by different methods.

where $\mathcal{E}(S_1)$ is the set of edges on S_1 and w_{ij} the cotangent weight of edge (i, j) . We can rewrite Equation (5.2) in a more compact way: $E_D(f) = \text{Trace}(f^\top W_1 f) := \|f\|_{W_1}^2$, where W_1 is the cotangent weight matrix of shape \mathcal{S}_1 .

Note that in practice one only needs to store the value of f at each vertex of S_1 and therefore if f is a pointwise map from S_1 to S_2 , we can represent it in matrix form $f = \Pi_{12}X_2$, where the value at row i gives the coordinates $f(x_i)$. We therefore define the Dirichlet energy of the map Π_{12} as the Dirichlet energy of f , which is the W -norm of the pull-back vertex coordinates:

$$E_D(\Pi_{12}) = \|\Pi_{12}X_2\|_{W_1}^2. \quad (5.3)$$

Note that [77] adopts a similar formulation to measure the smoothness of a given map, but pulls-back a high-dimensional embedding, in which the L^2 distance approximates the geodesic distance, and that is computed via multidimensional scaling [52].

While the Dirichlet energy defines a measure of distortion induced by a map, we note that mapping all vertices in S_1 to a single vertex in S_2 leads to zero energy, as seen by setting $f(x_i) = y$ for some fixed y in Equation (5.2). The Dirichlet energy thus only contains partial information about the quality of the map, and one needs to use additional constraints to obtain a *non-trivial* smooth map.

5.4 Discrete Solver for Dirichlet Energy

While functional maps intrinsically represent correspondences using low frequency eigenfunctions, thus inducing some smoothness, they do not provide any explicit control over the pointwise map smoothness (see Fig. 5.3). The discrete solver proposed in [177] has shown that many desirable map properties can be promoted directly on the functional maps, including bijectivity, landmarks preservation or conformality, the latter being unable to effectively promote smoothness as shown in the supplementary material. In this work, we therefore seek to extend this framework by introducing a pointwise map smoothness constraint that can be efficiently used alongside other objectives.

5.4.1 Problem Formulation

As discussed in Section 5.3, the Dirichlet energy, seen as a measure of smoothness, is globally minimized by constant maps. To avoid such trivial solutions, we propose to couple a smoothness energy with bijectivity constraints, which can be enforced in the spectral domain using the discrete optimization framework [177].

Specifically, given two shapes \mathcal{S}_1 and \mathcal{S}_2 we consider functional maps C_{ij} and pointwise maps Π_{ij} from *both* directions, where $(i, j) \in \{1, 2\}^2$ indicates the map direction. The discrete solver framework [177] introduces a bijectivity energy which reads:

$$E_{\text{bij}}(\Pi, C) = \sum_{ij} \left\| \Pi_{ji} \Phi_i C_{ji} - \Phi_j \right\|_{A_j}^2 + \alpha \left\| \Phi_j C_{ij} - \Pi_{ji} \Phi_i \right\|_{A_j}^2 \quad (5.4)$$

where the first term is derived from a spectral bijectivity energy and the second is a coupling term between functional maps C_{ij} and pointwise maps Π_{ji} (note the change in map directions). Note that variables Π and C contain maps in *both* directions in order to simplify notations. We refer the reader to [177] for a detailed derivation.

In this work, we augment this energy using smoothness constraints, acting on the primal domain instead of the functional (dual) one, which reads:

$$\min_{C, \Pi} E_{\text{bij}}(\Pi, C) + \gamma E_{\text{smooth}}(\Pi) \quad (5.5)$$

where E_{smooth} penalizes non-smooth pointwise maps, its most basic version being the sum of the Dirichlet energies of the pointwise maps $E_{\text{smooth}}(\Pi) = \sum_{ij} E_D(\Pi_{ij})$ with E_D being defined in Equation (5.3). In Section 5.5, we highlight how other common energies for smoothness can be expressed as variations of this Dirichlet energy, thus enabling their straightforward introduction within our formulation.

5.4.2 Smoothness-promoting Discrete Solver

We aim at solving Equation (5.5) using a similar algorithm to the standard discrete solver discussed in Section 5.3. However, as long as the energy E_{smooth} includes quadratic terms in Π_{ij} , for instance the Dirichlet energy, this solver cannot be applied as it assumes row-separable variables (see Lemma 4.1 in [177]). Since quadratic terms in the Dirichlet energy appear as W -norms of terms $\Pi_{ij} X_j$, we introduce auxiliary variables Y_{ij} as surrogate for products $\Pi_{ij} X_j$, and add a corresponding coupling term between the two, resulting in a new *coupled* smoothness energy:

$$E_{\text{sm}}^c(\Pi, Y) = E_{\text{smooth}}(\Pi, Y) + \beta \sum_{ij} \left\| Y_{ij} - \Pi_{ij} X_j \right\|_{A_i}^2 \quad (5.6)$$

where the second term is a spatial coupling term and, using some abuse of notations, $E_{\text{smooth}}(\Pi, Y)$ is obtained by replacing products $\Pi_{ij} X_j$ in $E_{\text{smooth}}(\Pi)$ by Y_{ij} . In the particular case where $E_{\text{smooth}} = E_D$, the coupled smoothness energy is now *row-separable* for Π :

$$E_{\text{sm}}^c(\Pi, Y) = \sum_{ij} \left\| Y_{ij} \right\|_{W_i}^2 + \beta \left\| Y_{ij} - \Pi_{ij} X_j \right\|_{A_i}^2 \quad (5.7)$$

Note that this particular half-quadratic splitting was used in [77] to handle similar constraints. Furthermore, we will show in Section 5.5 that multiple common energy for smoothness can benefit from this similar technique, resulting in a row-separable problem for Π in all cases.

Algorithm 4 Meta-algorithm

```

1: procedure METAALGORITHM
2:   Initialization:  $\Pi_{ij}^{(0)} = \Pi_{ij}^{\text{in}}$ ,  $Y_{ij}^{(0)} = \Pi_{ij}^{(0)} X_j$  for  $i, j \in \{1, 2\}$ 
3:   while Not converged do
4:      $C^{(k+1)} = \arg \min_C E_{\text{bij}}(\Pi^{(k)}, C)$ 
5:      $Y^{(k+1)} = \arg \min_Y E_{\text{sm}}^c(\Pi^{(k)}, Y)$ 
6:      $\Pi^{(k+1)} = \arg \min_{\Pi} E_{\text{ours}}(\Pi, C^{(k+1)}, Y^{(k+1)})$ 
7:   end while
8: end procedure

```

Total energy Eventually, the initial optimization problem, Equation (5.5), has been relaxed into a problem of the form $\min_{\Pi, C, Y} E_{\text{ours}}(\Pi, C, Y)$ with

$$E_{\text{ours}}(\Pi, C, Y) = E_{\text{bij}}(\Pi, C) + \gamma E_{\text{sm}}^c(\Pi, Y) \quad (5.8)$$

Crucially, this reformulation makes the total energy row-separable w.r.t. the pointwise maps Π . We can therefore propose a general iterative method (summarized in Algorithm 4) to minimize the total energy, in the spirit of the discrete solver, which iteratively updates each variable Π, C, Y with the other two sets fixed.

Solver The solver described in Algorithm 4 is divided in three optimization problems, for which we present the solution procedure. (1) Computing $C^{(k+1)}$ from $\Pi^{(k)}$ reduces to a simple $K \times K$ linear system, which has actually been introduced as *bijective ZoomOut* in [175]. (2) Computing $Y^{(k+1)}$ from $\Pi^{(k)}$ also reduces to a sparse linear system whose form depends on the choice of smoothness energy E_{smooth} , some of which are given in Section 5.5. In the case of $E_{\text{smooth}} = E_D$, computing Y_{ij} requires solving $(W_i + \beta A_i)Y_{ij} = \beta A_i \Pi_{ji} X_j$ where the system can be *prefactored* to further improve efficiency. (3) Since introducing auxiliary variables leads to a row-separable problem for Π , computing $\Pi^{(k+1)}$ from $C^{(k+1)}$ and $Y^{(k+1)}$ reduces to a simple nearest neighbor search. Note that this step is done in a high-dimensional space obtained by concatenating several terms, and can be heavily accelerated by only using coupling terms from equations Equations (5.4) and (5.6), which significantly reduces the embedding dimension on which to perform nearest neighbor. Finally, following [177], we also increase the size K of the functional map as iterations grow, which has shown to be a great regularization procedure in many spectral algorithms.

5.5 Smoothness Analysis in Unified Framework

In this section, we formulate several existing formulations for promoting map smoothness, including non-rigid ICP (nICP) [9], as-rigid-as-possible (ARAP) [209], reversible harmonic maps (RHM) [77], and Smooth Shells [70]. Our first objective is to provide a coherent formulation of various smoothness terms in the form of the Dirichlet energy on either a map or a deformation. Secondly, we aim to show how different energy terms and solvers can ultimately be introduced in our smoothness-promoting Discrete Solver. This will form the basis for our quantitative evaluation in the next section, in which we compare different terms within our solver. We remain succinct regarding the following derivations and their incorporation in our algorithm, and refer the interested reader to the supplementary material for a more complete overview.

nICP was originally proposed to wrap a source shape \mathcal{S}_1 onto a target shape \mathcal{S}_2 via a per-vertex affine deformation field \mathbf{D} . nICP implicitly maintains a pointwise map Π_{12} such that the deformed coordinates $\mathbf{D} \circ X_1$ approximate the pointwise map $\Pi_{12}X_2$. The total energy reads

$$E_{\text{nicip}}(\Pi_{12}, \mathbf{D}) = \left\| \mathbf{D} \right\|_{W_1}^2 + \beta \left\| \mathbf{D} \circ X_1 - \Pi_{12}X_2 \right\|_{A_1}^2 \quad (5.9)$$

with $\left\| \mathbf{D} \right\|_{W_1}^2 = \sum_{i \sim j} w_{ij} \left\| D_i - D_j \right\|_F^2$ extends the Dirichlet energy to per-vertex matrices. In our algorithm, this energy may be used as a surrogate for E_{sm}^c , given in Equation (5.7).

ARAP is a commonly-used energy that aims at promoting *local rigidity* of the shape deformation by enforcing the deformation to remain locally close to a rotation. ARAP optimizes both for expected vertex coordinates Y_{12} and per-vertex rotations \mathbf{R} . The total reformulated energy reads:

$$E_{\text{arap}}(Y_{12}, \mathbf{R}) = \left\| Y_{12} \right\|_{W_1}^2 + \lambda E_{\text{arap}}^{\text{rigid}}(Y_{12}, \mathbf{R}), \quad (5.10)$$

where $E_{\text{arap}}^{\text{rigid}}$ is a bilinear term promoting local rigid deformations. One can augment the energy using the coupling term from Equation (5.6) to use the ARAP energy in our algorithm.

Smooth Shells models the deformation \mathbf{D} as a simple per-vertex translation seen as a function $\mathcal{S}_1 \rightarrow \mathbb{R}^3$, restricted to lie in the *spectral* basis of size K , i.e., $\mathbf{D} \in \mathbb{R}^{K \times 3}$. In addition, smooth shells uses the ARAP energy to enforce the smoothness of the deformation. Specifically, if $Y_{12} = X_1 + \Phi_1 \mathbf{D}$ denotes the updated vertex positions, the shells energy is defined as

$$E_{\text{shells}}(\mathbf{D}, \mathbf{R}, \Pi_{12}) = E_{\text{arap}}(Y_{12}, \mathbf{R}) \quad (5.11)$$

which is augmented with a coupling term $\|X_1 + \Phi_1 \mathbf{D} - \Pi_{12}X_2\|_{A_1}^2$ to remain close to given correspondences.

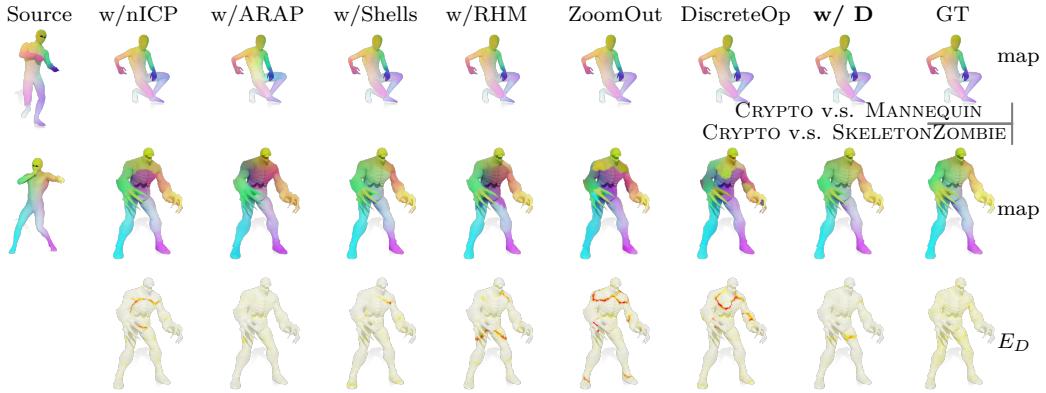


Figure 5.4: Qualitative evaluation on two pairs from DEFORMTHINGS4D-MATCHING. For a near-isometric shape pair shown on the *top*, all methods achieve smooth maps. For a shape pair that is far from isometry shown on *bottom*, nICP, ARAP, RHM, and Shells achieve relatively smooth maps but contain large patch of back-to-front ambiguity. The maps obtained by ZoomOut and Discrete Solver are locally smooth due to their spectral representation, but fail to maintain global smoothness. As a comparison, our methods can be robustly generalized to non-isometric shape maps and achieve globally smooth maps.

RHM directly minimizes the Dirichlet energy of a map without manipulating deformation fields. To avoid making the map collapse, the authors look for bijective maps with the lowest possible Dirichlet energy. Specifically, using notations of Sec. 5.4, smoothness is enforced by minimizing the same energy as in Equation (5.6) extended with a pointwise bijectivity term $\sum_{ij} \|\Pi_{ij} Y_{ji} - X_i\|_{A_i}$, resulting in a slower solver.

All these smoothness terms can be incorporated quickly within our solver, only affecting steps 2. and 3. of Algorithm 4. Furthermore, note that, for fairness of comparison, we ignored additional building blocks used in these works like normal preservation, high-dimensional embeddings, etc. More details on these two points can be found in the supplementary material.

5.6 Experiments

5.6.1 DeformThings4D-Matching Dataset

We propose DEFORMTHINGS4D-MATCHING (Fig. 5.2), a new dataset based on the DEFORMTHINGS4D dataset [124], a rich synthetic dataset with significant variations in both identities and types of motions, containing 1,972 animation sequences spanning 31 categories of humanoids and animals. However, using DEFORMTHINGS4D to evaluate shape matching is difficult since: (1) most shape models are disconnected; (2) the meshes belong to the same category are in the same triangulation, which provides perfect ground-truth but can lead to over-fitting

issues for matching algorithms [174], while cross-category ground truth is missing; (3) some meshes of the synthesized poses have unrealistic distortions such as large self-intersections and unnatural twists. We therefore select 56 animal categories and 8 humanoid categories from DEFORMTHINGS4D, each containing 15–50 poses selected from different motion clips, while ensuring large enough pose variations. We then apply LRVd [243] to *independently* remesh all the meshes in the same category. For the humanoid shapes, we further use the commercial software ¹R3DS to non-rigidly fit one shape into another to get *cross-category* correspondences. See Fig. 5.2 for some examples, where the corresponding vertices are assigned the same color. See supplementary materials for more details of how we construct the dataset and obtain the ground-truth correspondences between the remeshed shapes with different triangulations. The dataset is available at https://github.com/l1orZ/3DV22_DeformingThings4DMatching_dataset.

5.6.2 Comparison on Smoothness Formulation

We evaluate our method on the standard benchmark for non-isometric shape matching TOSCA non-Isometric Dataset [35], and the cross-category humanoid shape pairs from our DEFORMTHINGS4D-MATCHING Dataset. Note that on standard benchmarks like the FAUST dataset [28], existing methods already perform well as shapes remain near-isometric. We provide some results in Table 5.2 to show our method performs similarly in these simple cases, and refer to supplementary material for additional discussions.

Evaluation Metrics We follow [174] to measure the *accuracy*, *bijectivity*, *coverage* and *runtime* to compare different methods. Additionally, We apply Equation (5.3) to compute the Dirichlet energy on the obtained pointwise maps to evaluate the *smoothness*. See supplementary materials for detailed definitions and discussions.

Initialization & Baselines Since the tested shape pairs are highly non-isometric and challenging, standard shape descriptors failed to produce reasonable initialization, as shown in supplementary. We therefore compute each initial map from a 5×5 functional map obtained by using 5 landmarks. Our baselines can be categorized into three groups: (1) We compare to ZoomOut (ZO) [145] and Discrete Solver (DO) [177], the current-state-of-the-art refinement methods in functional maps pipeline. (2) We compare the standard Dirichlet Energy with the different variants presented in Section 5.5, namely nICP [9], ARAP [209], Shells [70] and RHM [77], all using the same algorithm. We highlight the Dirichlet energy (ours w/ D) and the RHM energy (ours w/ RHM) as respectively the simplest and globally best performing energies within our algorithm, which we both advocate. (3) We also include the results using original implementations of RHM and Shells for reference only, since additional regularizers besides smoothness are included.

¹<https://www.russian3dscanner.com/>

Table 5.1: DEFORMTHINGS4D-MATCHING Dataset: Summary over 433 shape pairs. We highlight the best two in blue, except those of Shells and RHM (see text for details).

| methods | <i>accuracy</i> | <i>bijection</i> | <i>smoothness</i> | <i>coverage</i> | <i>runtime (s)</i> |
|----------------|-----------------|------------------|-------------------|-----------------|--------------------|
| Init | 12.71 | 11.70 | 3.60 | 24.57% | - |
| RHM | 11.8 | 1.6 | 0.50 | 56.6% | |
| Shells | 11.4 | 5.1 | 1.50 | 50.8% | |
| Ours w/ ARAP | 12.16 | 11.70 | 0.71 | 31.0% | 25.3 |
| Ours w/ nICP | 9.56 | 3.89 | 1.72 | 40.4% | 100.8 |
| Ours w/ Shells | 8.41 | 2.59 | 2.18 | 51.7% | 48.2 |
| ZO | 8.57 | 7.14 | 4.02 | 67.0% | 17.5 |
| DO | 9.01 | 1.78 | 3.21 | 62.4% | 40.9 |
| Ours w/ D | 8.19 | 2.63 | 1.56 | 50.4% | 21.4 |
| Ours w/ RHM | 8.10 | 2.18 | 1.47 | 56.0% | 42.1 |

Table 5.2: Results on a random subset of 200 pairs of the FAUST dataset. We highlight the best two in blue.

| methods | <i>accuracy</i> | <i>bijection</i> | <i>smoothness</i> | <i>coverage</i> |
|-------------|-----------------|------------------|-------------------|-----------------|
| Init | 6.45 | 5.51 | 2.67 | 38.47 % |
| ZO | 3.95 | 2.16 | 0.79 | 82.16 % |
| DO | 4.07 | 1.08 | 0.86 | 77.96 % |
| Ours w/ D | 4.43 | 1.83 | 0.64 | 67.47 % |
| Ours w/ RHM | 3.94 | 1.11 | 0.71 | 79.26 % |

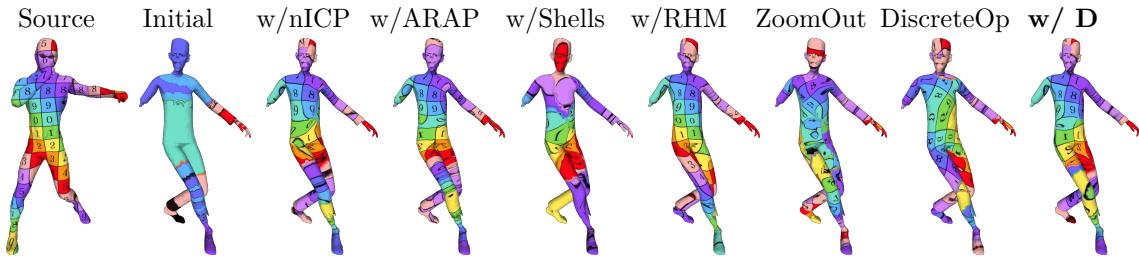


Figure 5.5: Starting from a poor initial map, our method can produce a more smooth and accurate map compared to the baseline methods.

Table 5.3: TOSCA Non-Isometric Dataset: Summary over 95 shape pairs. We highlight the best two in blue, except those of Shells and RHM (see text for details).

| methods | <i>accuracy</i> | <i>bijectivity</i> | <i>smoothness</i> | <i>coverage</i> | <i>runtime (s)</i> |
|----------------|-----------------|--------------------|-------------------|-----------------|--------------------|
| Init | 7.51 | 7.23 | 1.94 | 26.9% | - |
| RHM | 9.20 | 1.37 | 1.55 | 54.3 % | 818 |
| Shells | 10.20 | 6.72 | 5.58 | 45.6 % | 29.0 |
| ours w/ ARAP | 7.55 | 8.35 | 0.83 | 48.6% | 42.8 |
| Ours w/ nICP | 7.78 | 3.63 | 1.16 | 40.2% | 178 |
| Ours w/ Shells | 11.85 | 7.40 | 1.18 | 37.8% | 72.5 |
| ZO | 12.47 | 8.17 | 6.53 | 56.8% | 33.7 |
| DO | 13.30 | 1.90 | 5.51 | 53.4% | 79.2 |
| Ours w/ D | 7.25 | 3.02 | 1.22 | 42.2% | 33.3 |
| Ours w/ RHM | 6.26 | 1.87 | 1.39 | 53.1% | 40.1 |

DeformThings4D-Matching Dataset We report the average metrics over 433 cross-category shape pairs from the humanoid shapes from our DEFORMTHINGS4D-MATCHING dataset in Table 5.1. Among all the baseline methods, our method achieves the best accuracy. Compared to ZoomOut (ZO) and Discrete Solver (DO), our two selected energies achieve 3× better smoothness on average with comparable bijectivity and coverage. It suggests that, our method, as an extended algorithm of discrete solver by adding a smoothness term, is indeed effective to promote map smoothness. In supplementary, we also report *per-category* map evaluation. We show two qualitative examples in Figure 5.4, where the obtained maps are visualized by color transfer. For the pair between CRYPTO and SKELETONZOMBIE, we also visualize the per-vertex smoothness error for each map. We additionally display texture transfer for a difficult pair in Figure 5.5, using [75] to obtain a vertex-to-point map for each method to improve visualization. While this figure shows that our maps clearly outperform standard spectral method starting from poor initialization, there is room for improvement for all energies.

TOSCA Non-Isometric Dataset contains cross-category correspondences among one gorilla shape (with 5 different poses), one male shape (with 7 different poses),

and one female shape (with 12 different poses). We use all 95 non-isometric shape pairs between the gorilla shapes and the human (male and female) shapes. The summary evaluation is shown in Table 5.3. See supplementary for qualitative examples. Enforcing the smoothness of the pointwise maps via Dirichlet energy (Ours + D) help us achieve much more accurate and $5\times$ smoother maps. We additionally highlight that adding extra pointwise bijectivity (ours w/ RHM) has a positive effect on the metrics, but results in a slower solver. Finally, while ARAP and nICP energies perform quite well, the Shells energy seems to suffer from the high level of non-isometry in the dataset as it mainly relies on spectral quantities.

5.6.3 Implementation & Parameters

We implemented all the baselines (based on their released code) and our methods in Python to guarantee a fair comparison. We follow the discrete solver [177] to adopt the progressive upsampling technique into our algorithm, which is introduced in [145], and gradually increase the spatial coupling term weight γ to avoid over-smoothing in the earlier iterations. Detailed parameters can be found in supplementary, or in the released version of the code at <https://github.com/RobinMagnet/smoothFM>.

5.7 Conclusion, Limitations & Future Work

In this work, we extended the discrete solver formulation from [177] to optimize the Dirichlet energy to promote map smoothness. We then proposed an efficient algorithm that can produce high-quality and smooth maps from noisy initial maps for between non-isometric surfaces. Furthermore, we demonstrated that multiple previously proposed methods for computing smooth maps, including nICP [9], ARAP [209], RHM [77], and Smooth Shells [70], can be reformulated in a coherent framework. This allowed us to compare and analyze different formulations for smoothness using a single algorithm. Finally, to address the scarcity of evaluation data, we proposed a new dataset based on DEFORMTHINGS4D, with variable mesh structure, and dense ground truth cross-category correspondences for eight challenging categories. We believe both our framework and this dataset can be helpful for the shape matching community.

Our method still has some limitations. First, optimizing the Dirichlet energy can indeed greatly improve the smoothness compared to spectral methods. This, however, can come at the expense of loss of coverage, and we observe that our maps can still collapse locally, as seen from the texture transfer of Figure 5.5. It would be interesting to investigate techniques that to further prevent *local collapse* and obtain a smooth map with high coverage. Second, our results show that the proposed method improves significantly results from ZoomOut and discrete solver on complete shapes, even for non-isometric cases. However, for the partial matching setting, though our maps still outperform ZoomOut and discrete solver, there is still a lot of room for further improvement. Finally, our energy is a weighted sum of

a bijectivity and a smoothness term, which can become hard to balance across all initialization quality.

In the future, we would like to study different energies for partial matching and to ways to prevent local map collapse. It will also be interesting to apply our approach for computing dense correspondences in other domains, such as point clouds, graphs, or even 2D images.

Acknowledgments The authors thank the anonymous reviewers for their valuable comments and suggestions. Parts of this work were supported by the ERC Starting Grant No. 758800 (EXPROTEA), the ERC Consolidator Grant No. 101003104 (MYCLOTH), and the ANR AI Chair AIGRETTE.

Scalable and Efficient Functional Map Computations on Dense Meshes

We propose a new scalable version of the functional map pipeline that allows to efficiently compute correspondences between potentially very dense meshes. Unlike existing approaches that process dense meshes by relying on ad-hoc mesh simplification, we establish an integrated end-to-end pipeline with theoretical approximation analysis. In particular, our method overcomes the computational burden of both computing the basis, as well the functional and pointwise correspondence computation by approximating the functional spaces and the functional map itself. Errors in the approximations are controlled by theoretical upper bounds assessing the range of applicability of our pipeline. With this construction in hand, we propose a scalable practical algorithm and demonstrate results on dense meshes, which approximate those obtained by standard functional map algorithms at the fraction of the computation time. Moreover, our approach outperforms the standard acceleration procedures by a large margin, leading to accurate results even in challenging cases.

6.1 Introduction

Processing and analyzing complex 3D objects is a major area of study with applications in computer graphics, medical imaging and other domains. The underlying structure of such data can be highly detailed and require dense point sets and meshes to capture important features. At the same time, shape analysis methods are often designed to only handle objects that consist of tens of thousands of points, thus requiring decimation algorithms to process meshes containing millions of points that can arise in real-world applications. While mesh simplification can lead to good results, it suffers from several drawbacks. First, the simplification process might lead to artifacts and significant loss of detail. Second, for many applications, it remains highly non-trivial to accurately transfer the results of analysis from the simplified to original shapes. Finally, the transfer process can introduce errors and aliasing artifacts.

In this work, we focus on computing correspondences between non-rigid shapes. This is a long-standing problem in Geometry Processing and related fields, with a

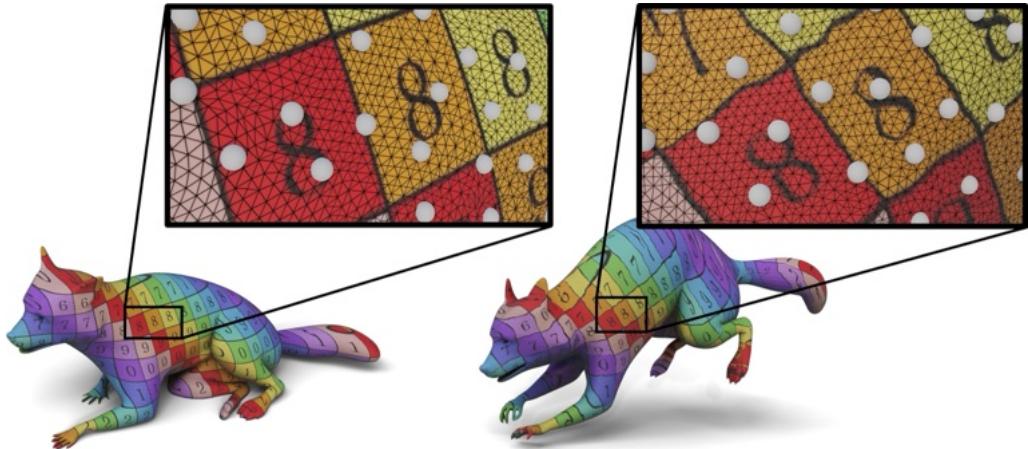


Figure 6.1: Our method produces point-to-point correspondences between dense meshes *efficiently*, using values only located at sparse samples, displayed in white. The source and target shapes from the DEFORMINGTHINGS4D dataset [124] are composed of roughly 100 000 vertices, and correspondences are displayed using texture transfer. The map computation (including all preprocessing) took 60 seconds on a standard machine.

wide range of techniques developed in the past few years [59, 188]. A notable line of work in this domain uses the so-called functional map framework, which is based on manipulating correspondences as matrices in a reduced basis [159]. Methods based on this framework have recently achieved high accuracy on a range of difficult non-rigid shape matching tasks [144, 145, 65]. Unfortunately, these approaches require costly and time-consuming precomputation of the Laplacian basis and, potentially, other auxiliary data-structures [174]. As a result, these techniques do not scale well to densely sampled meshes and, thus, are most often applied to simplified shapes. Moreover, while accelerated versions of some methods [145] have recently been proposed, these lack theoretical approximation guarantees, and can be error-prone.

At the same time, several approaches have recently been proposed for efficient approximation of the Laplace-Beltrami basis [149, 148]. These approaches can successfully scale to very large meshes, and are especially effective for computing low frequency eigenfunctions. While these methods have been shown to be efficient when, e.g., using approximated spectra as shape descriptors [178] or for individual shape processing, they can come short when applied in *shape correspondence scenarios*. Conceptually, this is because the objectives and guarantees in [149, 148] only apply at a *global* scale of individual shapes, instead of the local function approximation or function transfer required for functional and point-to-point map computation.

In this work, we make a step towards creating scalable and efficient non-rigid shape correspondence methods, which can handle very large meshes, and are backed by theoretical approximation bounds. We focus on the functional map framework [162] and especially its recent variants based on spectral upsampling, such as the ZoomOut method [145] and its follow-up works [100, 240, 177]. These methods

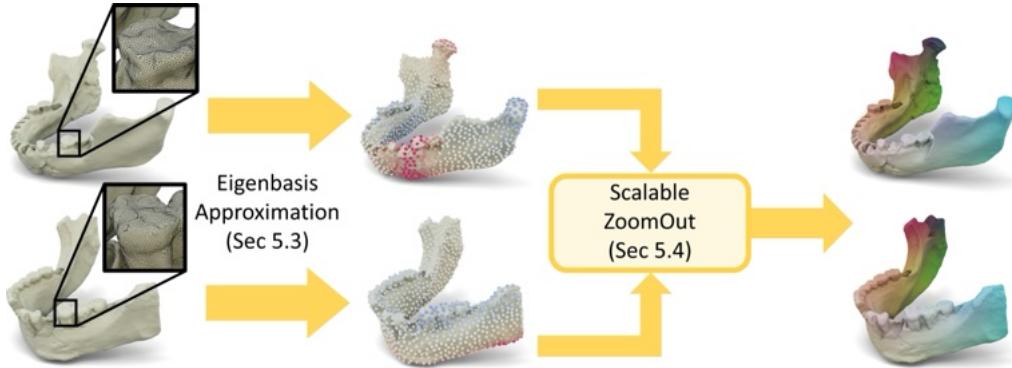


Figure 6.2: Overall pipeline of our method, using real data from [176]. Given two dense input shapes, we first generate an approximate eigenbasis computation by using a modified version of the approach introduced in [148] (Sec. 6.5.3). We then propose a new, scalable version of ZoomOut (Sec. 6.5.4), which exploits our functional map approximation (Sec. 6.5.1) and comes with theoretical approximation bounds. Ultimately, this leads to dense pointwise correspondences between the two input shapes visualized here via color transfer.

are based on iteratively updating functional and point-to-point maps and have been shown to lead to high-quality results in a wide range of cases. Unfortunately, the two major steps: basis pre-computation and iterative updating of the pointwise maps can be costly for dense shapes.

To address this challenge, we propose an integrated pipeline that helps to make both of these steps scalable and moreover comes with approximation guarantees. For this, we first establish a new functional space inspired by [149], and demonstrate how it can be used to define an approximation of functional maps without requiring either a dense pointwise correspondence or even a basis on the dense meshes. We then provide theoretical approximation bounds for this construction that, unlike the original definition in [159] is fully agnostic to the number of points in the original mesh. Following this analysis, we extend the approach introduced in [149] to improve our functional map approximation, and present an efficient and scalable algorithm for map refinement, based on our constructions, which eventually produces accurate results in the fraction of the time required for standard processing, as displayed on Figure 6.2.

6.2 Related Works

Our main focus is on designing a scalable and principled approach for non-rigid shape correspondence, within the functional map framework. We therefore review works that are most closely related to ours, especially those using spectral techniques for shape matching, and refer the interested readers to recent surveys [224, 214, 24, 188, 59] for a more comprehensive overview of other approaches.

Spectral methods in shape matching The idea of using the spectral properties of the Laplace-Beltrami, and especially its eigenfunctions for shape correspondence, has been investigated in many existing works. Early approaches focused on directly aligning the eigenfunctions, seen as descriptors, [140, 103] or using diffusion processes to derive descriptors or embedding spaces, *e.g.* [195, 158], among others.

A more principled framework was introduced in [159], based on the idea of functional maps. The overall strategy is to express the pull-back of functions as an operator in a reduced basis, and to formulate objective functions based on desirable properties of such an operator. The main advantage of this approach is that it leads to small-scale optimization problems, with the number of unknowns independent of the size of the underlying meshes.

Despite the simplicity of the original approach, its performance is strongly dependent on accurate descriptors and hyperparameter tuning. As a result, this basic strategy has been extended significantly in many follow-up works, based both on geometric insights [117, 4, 161, 37, 74], improved optimization strategies [118, 153, 175, 177], and richer correspondence models going beyond isometries across complete shapes, [181, 186, 128], among others.

Functional and pointwise maps While many approaches in the functional map literature focus on the optimization in the spectral domain, it has also been observed that the *interaction* between pointwise and functional correspondences can lead to significant improvement in practice. This was used in the form of the Iterative Closest Point (ICP) refinement in the original article and follow-up works [159, 139, 161] and has since then been extended to map deblurring and denoising [75], as well as powerful refinement, and even map optimization strategies [145, 174, 100, 70, 177]. All of these works are based on the insight that manipulating maps in *both* the spectral and spatial (primal) domains can lead to overall improvement in the quality of the results.

Unfortunately, such approaches can often come at a cost of scalability, since the complexity of pointwise maps is directly dependent on the mesh resolution, making it difficult to scale them to highly dense meshes.

Multi-resolution spectral approaches Our work is also related to multi resolution techniques for approximating spectral quantities, as, *e.g.*, in [225], and especially to recent developments for accurate and scalable eigen-solvers geared towards Laplacian eigenfunctions on complex meshes [149, 148]. The latter set of methods have been shown to lead to excellent performance and scalability on tasks involving individual shapes, such as computing their Shape-DNA [178] descriptors, or performing mesh filtering. Similarly, there exist several spectral coarsening and simplification approaches [129, 119, 45] that explicitly aim to coarsen operators, such as the Laplacian, while preserving their low frequency eigenpairs. Unfortunately, these methods typically rely on the eigenfunctions on the dense shapes, while the utility of the former approaches in the context of *functional maps* has not yet been fully analyzed and exploited, in part, since, as we show below, this requires *local* approximation bounds.

Finally, we mention that our work is also related to hierarchical techniques, including functional maps between subdivision surfaces proposed in [202], and even more closely, to refinement via spectral upsampling [145]. However, the former approach relies on a subdivision hierarchy, while the acceleration strategy of the latter, as we discuss below, is based on a scheme that unfortunately can fail to converge in the presence of full information.

Limitations of existing techniques and our contributions To summarize, the scalability of existing functional maps-based methods is typically limited by two factors: first, the pre-processing costs associated with the computation of the eigenfunctions of the Laplace-Beltrami operator, and second, the complexity of simultaneously manipulating pointwise and functional correspondences.

In this context, our key contributions include:

1. We define an approximation of the functional map, which requires only a sparse correspondence, and provide a theoretical basis for this construction.
2. We analyze the basis approximation approach in [149] for functional map computation, obtaining explicit theoretical upper bounds. We then modify this approach to improve the approximation guarantees, leading to more accurate maps.
3. We present a principled and scalable algorithm for functional map refinement, based on our constructions, which produces accurate results at a fraction of the time of comparable methods.

6.3 Method Overview

As mentioned above, our overall goal is to design a scalable pipeline for non-rigid shape matching that can handle potentially very dense meshes. We base our approach on the ZoomOut variant of the functional map framework [145]. However, our constructions can be easily extended to other recent functional maps methods, *e.g.*, [175, 177], which share the same general algorithmic structure. Specifically, ZoomOut and related methods are based on two main building blocks: computing the eigenfunctions of the Laplace-Beltrami operator first, and then iterating between updating the point-to-point and functional maps.

Our general pipeline is displayed on Figure 6.2 and consists of the following major steps. First, we generate for each shape a sparse set of samples and a factorized functional space using a modification of the approach introduced in [149], described in Section 6.5.3. Secondly, we use the approximation of the functional map that we introduce (Sec. 6.5.1) to define a scalable version of the ZoomOut algorithm, producing a sparse pointwise map. Finally, we extend this sparse map to a dense pointwise map with sub-sample accuracy, by using the properties of the functional subspaces we consider.

The rest of the chapter is organized as follows: in Section 6.4 we introduce the notations and background necessary for our approach.

In Section 6.5.1, we introduce our functional map approximation based on the basis construction approach in [149]. Section 6.5.2 provides explicit approximation errors and Section 6.5.3 describes our modification of the method of [149], which helps to improve the theoretical upper bounds we obtained for functional map computation.

Given these constructions, we show in Section 6.5.4 how ZoomOut-like algorithms can be defined, first by iteratively updating functional and pointwise maps in the reduced functional spaces, and then how the computed functional map can be extended onto the dense shapes efficiently.

Section 6.5.5 provides implementation details, while Section 6.6 is dedicated to extensive experimental evaluation of our approach.

6.4 Notations & Background

6.4.1 Notations

For a triangle mesh, we denote by \mathbf{W} and \mathbf{A} its stiffness and mass matrices that together define the (positive semi-definite) Laplace-Beltrami Operator as $L = \mathbf{A}^{-1}\mathbf{W}$. Given two shapes \mathcal{N} and \mathcal{M} with, respectively, n and m vertices, any vertex-to-vertex map $T : \mathcal{N} \rightarrow \mathcal{M}$ can be represented as a binary matrix $\boldsymbol{\Pi} \in \{0, 1\}^{n \times m}$ with $\boldsymbol{\Pi}_{ij} = 1$ if and only if $T(x_i) = y_j$, where x_i denotes the i -th vertex on \mathcal{N} and y_j the j -th vertex in \mathcal{M} .

The eigenfunctions of the Laplace-Beltrami operator can be obtained by solving a generalized eigenproblem:

$$\mathbf{W}\psi_i = \lambda_i \mathbf{A}\psi_i, \quad (6.1)$$

where, in practice, we typically consider the eigenfunctions corresponding to the K smallest eigenvalues.

6.4.2 Functional Maps and the ZoomOut algorithm

Functional maps were introduced in [159] as a means to perform dense non-rigid shape matching. The key insight is that any pointwise map $T : \mathcal{N} \rightarrow \mathcal{M}$ can be transformed into a functional map via composition $F_T : f \in \mathcal{F}(\mathcal{M}) \mapsto f \circ T \in \mathcal{F}(\mathcal{N})$, where $\mathcal{F}(\mathcal{S})$ is the space of real-valued functions on a surface \mathcal{S} . Since F_T is linear, it can be represented as a matrix in the given basis for each space $(\psi_i^{\mathcal{M}})_i$ and $(\psi_i^{\mathcal{N}})_i$.

If the basis on shape \mathcal{N} is orthonormal with respect to $\mathbf{A}^{\mathcal{N}}$, the functional map \mathbf{C} can be expressed in the truncated basis of size K on each shape as a $K \times K$ matrix:

$$\mathbf{C} = (\Psi^{\mathcal{N}})^{\top} \mathbf{A}^{\mathcal{N}} \boldsymbol{\Pi} \Psi^{\mathcal{M}}, \quad (6.2)$$

where each basis function on \mathcal{M} (resp. \mathcal{N}) is stacked as a column of $\Psi^{\mathcal{M}}$ (resp. $\Psi^{\mathcal{N}}$), $\boldsymbol{\Pi}$ is the matrix representing the underlying pointwise map, and we use $^{\top}$ to denote the matrix transpose.

ZoomOut Given the Laplace-Beltrami eigenbasis, the ZoomOut algorithm [145] allows recovering high-quality correspondences starting from an approximate initialization, by iterating between two steps: (1) Converting a $k \times k$ functional map to a pointwise map, (2) converting the pointwise map to a functional map of size $k+1 \times k+1$. This method has also been extended to other settings, to both promote cycle consistency [100] and optimize various energies [177] among others. Unfortunately, although simple and efficient, the scalability of this approach is limited, first, by the precomputation of the Laplacian basis, and second by the pointwise map recovery which relies on possibly expensive nearest-neighbor search queries across dense meshes.

Several ad-hoc acceleration strategies have been proposed in [145]. However, as we discuss below, these do not come with approximation guarantees and indeed can fail to converge in the limit of complete information.

6.4.3 Eigenbasis approximation

To improve the scalability of spectral methods, recent works [149, 242] have tried to develop approximations of the Laplace-Beltrami eigenbasis, via the reduction of the search space. Specifically, in [149], the authors first sample a set of p points $\mathcal{S} = \{v_1, \dots, v_p\}$ on shape \mathcal{M} and create a set of p local functions (u_1, \dots, u_p) , each centered on a particular sample point. Each function u_j is built from an unnormalized function \tilde{u}_j supported on a geodesic ball of radius ρ around the sample v_j , which decreases with the geodesic distance from the center:

$$\tilde{u}_j : x \in \mathcal{M} \mapsto \chi_\rho(d^\mathcal{M}(x, v_j)) \in \mathbb{R} \quad (6.3)$$

where $d^\mathcal{M}$ is the geodesic distance on shape \mathcal{M} and $\chi_\rho : \mathbb{R}_+ \rightarrow \mathbb{R}$ is a differentiable non-increasing function with $\chi_\rho(0) = 1$ and $\chi_\rho(x) = 0$ for $x \geq \rho$. Choices for χ are discussed in Appendix D.1. Finally, local functions u_j are defined to satisfy the partition of the unity by using:

$$u_j(x) = \frac{\tilde{u}_j(x)}{\sum_k \tilde{u}_k(x)} \quad \forall x \in \mathcal{M} \quad (6.4)$$

Now only considering functions that lie in the Span $\{u_1, \dots, u_p\}$, the original eigendecomposition system in Equation (6.1) reduces to a generalized eigenproblem of size $p \times p$:

$$\bar{\mathbf{W}} \bar{\phi}_i = \bar{\lambda}_i \bar{\mathbf{A}} \bar{\phi}_i \quad (6.5)$$

with $\bar{\mathbf{W}} = \mathbf{U}^\top \mathbf{W} \mathbf{U}$ and $\bar{\mathbf{A}} = \mathbf{U}^\top \mathbf{A} \mathbf{U}$ where \mathbf{W} and \mathbf{A} are the stiffness and area matrices of \mathcal{M} , and \mathbf{U} a *sparse* matrix whose columns are values of functions $(u_j)_j$.

Eigenvectors $\bar{\phi}_i$ are p -dimensional vectors describing the coefficients with respect to $(u_j)_j$, which define the approximated eigenvectors as $\bar{\psi}_i = \mathbf{U} \bar{\phi}_i$. Note that since $(\bar{\phi}_i)_i$ are orthonormal with respect to $\bar{\mathbf{A}}$, this implies that $(\bar{\psi}_i)_i$ are orthonormal with respect to \mathbf{A} .

While the original work [149] focused on global per-shape applications such as filtering and Shape-DNA [178] computation, we build on and modify this pipeline in order to obtain reliable functions to perform dense *shape correspondence*.

6.5 Our approach

In this section, we first present a functional map definition using the basis approximation strategy from [149], and provide theoretical bounds on the approximation error (Secs. 6.5.1 and 6.5.2 respectively). Based by these results, we then introduce our modification of [149] in Section 6.5.3 which we use in our approach in order to minimize the computed bound. Finally, we present our Extended ZoomOut algorithm and provide implementation details in Sections 6.5.4 and 6.5.5.

6.5.1 Approximate Functional Map

As mentioned in, Section 6.4.3 the eigenfunctions computed using the approach in [149] are, by construction, orthonormal with respect to the area matrix \mathbf{A} . Thus, they can be used to compute a functional map following Equation (6.2). This leads to the following definition:

Definition 6.1. *Given two shapes \mathcal{M} and \mathcal{N} , with approximated eigenfunctions $(\Psi_i^{\mathcal{M}})_i$, stacked as columns of matrix $\bar{\Psi}^{\mathcal{M}}$ (resp. with \mathcal{N}), the reduced functional map associated to a pointwise map $\Pi : \mathcal{N} \rightarrow \mathcal{M}$ is defined as:*

$$\bar{\mathbf{C}} = (\bar{\Psi}^{\mathcal{N}})^T \bar{\mathbf{A}}^{\mathcal{N}} \Pi \bar{\Psi}^{\mathcal{M}} \quad (6.6)$$

Note that this functional map definition uses the approximated bases. However, it still relies on the knowledge of a full point-to-point map between complete (possibly very dense) shapes.

To alleviate this constraint, we introduce another functional map $\bar{\mathbf{C}}$ that only relies on maps between samples, independently of the original number of points:

Definition 6.2. *Using the same setting as in Definition 6.1, with eigenfunctions arising from Equation (6.5), $(\bar{\phi}_i^{\mathcal{M}})_i$ (resp. with \mathcal{N}) being stacked as columns of a matrix $\bar{\Phi}^{\mathcal{M}}$ (resp. with \mathcal{N}), given a point-wise map $\bar{\Pi} : \mathcal{S}^{\mathcal{N}} \rightarrow \mathcal{S}^{\mathcal{M}}$, our restricted functional map $\bar{\mathcal{F}}_K(\mathcal{N})$ is defined as:*

$$\hat{\mathbf{C}} = (\bar{\Phi}^{\mathcal{N}})^T \bar{\mathbf{A}}^{\mathcal{N}} \bar{\Pi} \bar{\Phi}^{\mathcal{M}} \quad (6.7)$$

Recall that, as mentioned in Section 6.4.3, \mathcal{S} denotes the sparse set of samples on each shape. Therefore, in order to define $\hat{\mathbf{C}}$, we only need to have access to a pointwise map between the *sample points* on the two shapes. This restricted functional map $\hat{\mathbf{C}}$ is a pull-back operator associated to the reduced spaces $\text{Span}\{\bar{\phi}_k^{\mathcal{M}}\}_k$ and $\text{Span}\{\bar{\phi}_k^{\mathcal{N}}\}_k$, since both families are orthonormal with respect to $\bar{\mathbf{A}}$. Furthermore, using the factorization $\bar{\Psi} = \mathbf{U} \bar{\Phi}$ on each shape in Eq. (6.6) as well as the definition of $\bar{\mathbf{A}}$, we remark that going from Eq. (6.6) to Eq. (6.7) only requires the approximation $\Pi \mathbf{U}^{\mathcal{M}} \simeq \mathbf{U}^{\mathcal{N}} \bar{\Pi}$, for which we will later derive an upper bound in Proposition 6.2. Note that one might want to replace $\bar{\Phi}^{\mathcal{M}}$ by $\bar{\Psi}^{\mathcal{M}}$ in Equation (6.7) so that the map $\bar{\Pi}$ actually transports pointwise values rather than coefficients. In

practice, as evaluated in Appendix D.2, we did not observe any improvement using this modification.

The first benefit of the approximated functional map in Equation (6.7) compared to the exact one in Equation (6.6) is the exclusive use of small-sized matrices. Observe that functions $(\bar{\phi}_i)_i$, are associated with the area and stiffness matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{W}}$, which define the L_2 and W_1 inner products, thus allowing to use *all* functional map related algorithms in a straightforward way *without* using any extra approximation or acceleration heuristics. Eventually, a dense pointwise map between complete shapes can be obtained by identifying the two pull-back operators $\hat{\mathbf{C}}$ and $\bar{\mathbf{C}}$, as described later in Section 6.5.4. As we will see, the resulting correspondences outperform those obtained using remeshed versions of shape and nearest neighbor extrapolation, as our functional map produces sub-sample accuracy.

Secondly, as shown in the following section, our approach is backed by strong theoretical convergence guarantees, providing bounds on approximation errors. In contrast, previous approaches, such as the accelerated version of ZoomOut [145] (Sec. 4.2.3) *might not* converge to the true functional maps even when using all available information. Namely, Fast ZoomOut indeed samples q points on shapes \mathcal{M} and \mathcal{N} , and approximates $\bar{\mathbf{C}}$ using

$$\mathbf{C}_{\text{F-ZO}} = \arg \min_{\mathbf{X}} \|\mathbf{Q}^{\mathcal{N}} \Psi^{\mathcal{N}} \mathbf{X} - \boldsymbol{\Pi} \mathbf{Q}^{\mathcal{M}} \Psi^{\mathcal{M}}\|_F^2 \quad (6.8)$$

where $\mathbf{Q}^{\mathcal{N}} \in \{0, 1\}^{q \times n^{\mathcal{N}}}$ with $\mathbf{Q}_{ij}^{\mathcal{N}} = 1$ if and only if x_j is the i^{th} sample on shape \mathcal{N} (similarly for \mathcal{M}). Using all points means \mathbf{Q} matrices are identity. This approximation gives equal importance to all sampled points regardless of their area, and thus fails to converge towards the underlying \mathbf{C} as the number of samples increases. This means a near uniform sampling strategy is required in practice, which is difficult to achieve on very dense meshes.

In the following section, we provide approximation error bounds for our functional map definition, which we later use to modify the approach from [149] to reduce these errors and obtain a more accurate and principled correspondence approach.

6.5.2 Approximation Errors

Most expressions above involve a given pointwise map $\boldsymbol{\Pi}$ between surfaces \mathcal{N} and \mathcal{M} . The following lemma provides simple assumptions to obtain a Lipschitz constant for its associated functional map, which will be very useful to derive bounds on the approximation errors of our estimators:

Lemma 6.1. *Let \mathcal{M} and \mathcal{N} be compact surfaces and $T : \mathcal{N} \rightarrow \mathcal{M}$ a diffeomorphism. Then there exists $B_T \in \mathbb{R}$ so that:*

$$\|f \circ T\|_{\mathcal{N}} \leq B_T \|f\|_{\mathcal{M}} \quad \forall f \in L^2(\mathcal{M}) \quad (6.9)$$

the proof of which can be found in [97] (Proposition 3.3).

Our overall goal is to use the newly designed functional map $\bar{\mathbf{C}}$ within a ZoomOut-like functional map estimation algorithm. We therefore expect the approximated

functional map to mimic the underlying map \mathbf{C} when the computed eigenvectors $\bar{\Psi}_k$ approximate well the true ones Ψ_k . The following proposition bounds the error between the two functional maps:

Proposition 6.1. *Let $\bar{\Psi}^N$ (resp. $\bar{\Psi}^M$) and Ψ^N (resp. Ψ^M) the approximated and true first K eigenvectors of the Laplacian on N (resp. M). Let \mathbf{C} and $\bar{\mathbf{C}}$ be the original and reduced (see Eq. (6.6)) functional maps of size K , associated to the map T . Suppose that T is a diffeomorphism, and let B_T be the bound given by Lemma 6.1. If there exists $\varepsilon \in \mathbb{R}_+^*$ so that for any $j \in \{1, \dots, K\}$:*

$$\|\Psi_j^N - \bar{\Psi}_j^N\|_\infty \leq \varepsilon \text{ and } \|\Psi_j^M - \bar{\Psi}_j^M\|_\infty \leq \varepsilon$$

Then:

$$\frac{1}{K} \|\mathbf{C} - \bar{\mathbf{C}}\|_2^2 \leq \varepsilon^2 (1 + B_T^2) \quad (6.10)$$

The proof can be found in Appendix D.3. This proposition ensures that a good estimation of the spectrum implies an accurate functional map approximation, and thus its good behavior within matching algorithms.

A more fundamental error to control is the estimation error between the functional maps $\bar{\mathbf{C}}$ from Def. 6.1 and $\hat{\mathbf{C}}$ from Def. 6.2. As mentioned above, the estimation relies on the identification $\Pi \bar{\Psi}^M \simeq \mathbf{U}^N \bar{\Pi} \bar{\Phi}^M$, where $\bar{\Pi}$ is a map between the two sets of samples \mathcal{S}^N and \mathcal{S}^M , which we expect to be similar to Π on these spaces. This approximation treats equivalently the two following procedures: 1) interpolating between values on \mathcal{S}^M then transferring using the map Π , 2) transferring values on \mathcal{S}^M to values on \mathcal{S}^N using $\bar{\Pi}$ and then interpolating on N . The following proposition bounds the error of this approximation:

Proposition 6.2. *Let $T : N \rightarrow M$ be a pointwise map between the shapes represented by Π , and let B_T be the bound given by Lemma 6.1. Suppose that $T|_{\mathcal{S}^N} : \mathcal{S}^N \rightarrow \mathcal{S}^M$ is represented by $\bar{\Pi}$.*

Let $\alpha = \min_j u_j^M(v_j) \in [0, 1]$. Suppose further that there exists $\varepsilon > 0$ so that for any $k \in \{1, \dots, K\}$ and $x, y \in \mathcal{S}^M$:

$$d^M(x, y) \leq \rho^M \Rightarrow |\bar{\Psi}_k^M(x) - \bar{\Psi}_k^M(y)| \leq \varepsilon \quad (6.11)$$

and

$$d^M(x, y) \leq \rho^M \Rightarrow |\bar{\Phi}_k^M(x) - \bar{\Phi}_k^M(y)| \leq \varepsilon. \quad (6.12)$$

Then

$$\frac{1}{K} \|\Pi \bar{\Psi}^M - \mathbf{U}^N \bar{\Pi} \bar{\Phi}^M\|_N^2 \leq \varepsilon^2(1 - \alpha) + \varepsilon^2 B_T^2 \quad (6.13)$$

The proof is given in Appendix D.4. This proposition shows that the estimation error depends on two parameters: 1) the variation ε of the eigenfunctions w.r.t to the sample distance ρ , 2) the *self-weights* $u_j(v_j)$ from the local functions defined in the basis approximation. Note that since the basis functions u_j verify $0 \leq u_j \leq 1$ and satisfy the partition of unity, they can be interpreted as interpolation weights from values at sampled points to values on the entire shape. This makes the dependence in α more intuitive, as our approximation relies on the local identification of

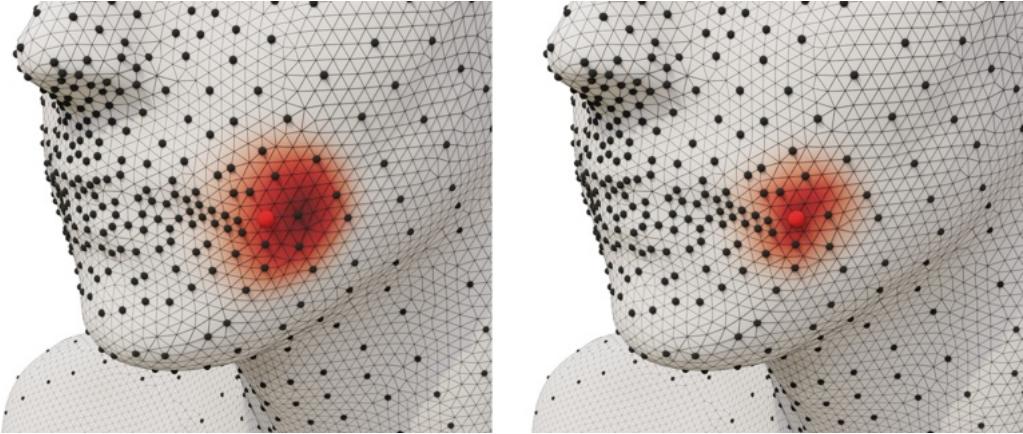


Figure 6.3: Example of a local function u_j (red color) centered on v_j (red vertex), visualized without (Left) and with (Right) our adaptive radius strategy. Other samples v_k are displayed in black.

basis coefficients with function values. A discussion on the numerical values of the quantities used in Proposition 6.2 are provided in Appendix D.5.

In the following, we will therefore seek to modify the basis approximation [149] in order to maximize α while retaining both the quality of the approximation of the true Laplacian spectrum, necessary to apply functional maps-related algorithms.

6.5.3 Improved Eigenbasis Approximation

In this section, we propose a modification of the approach from [149], based on the theoretical bounds introduced above. For the rest of this section, we focus on a single shape, as the basis computations are done on each shape independently.

As seen from Proposition 6.2, high self weights allow stabilizing our functional map approximation. Interestingly, with the construction in [149], the value $u_j(v_j)$ only depends on the geodesic distance between v_j and other sampled points v_i for $i \neq j$:

$$u_j(v_j) = \frac{1}{1 + \sum_{i \neq j} \tilde{u}_i(v_j)}. \quad (6.14)$$

where $(\tilde{u}_i)_i$ are the unnormalized local functions. We modify the pipeline from [149] in order to increase these values as follows: we first define a per-sample radius ρ_j for $j \in \{1, \dots, p\}$, instead of a single global value ρ . Given a sample point v_j with a small self-weight $u_j(v_j)$, radius ρ_j is kept untouched as it has no influence on the self-weight, but we instead reduce the radius ρ_i of its most influential neighbor - that is the radius of the point v_i with the highest value $\tilde{u}_i(v_j)$. Following Equation (6.14) this eventually increases the value of $u_j(v_j)$. Note that this modification doesn't change the value $u_i(v_i)$ and increase the self weights of all its neighbors. This way, all self weights are non-decreasing during the algorithm, with at least one of them increasing. This extra adaptation additionally comes at a negligible computational cost, as it only requires re-evaluating u_j at a set of fixed vertices. In particular,

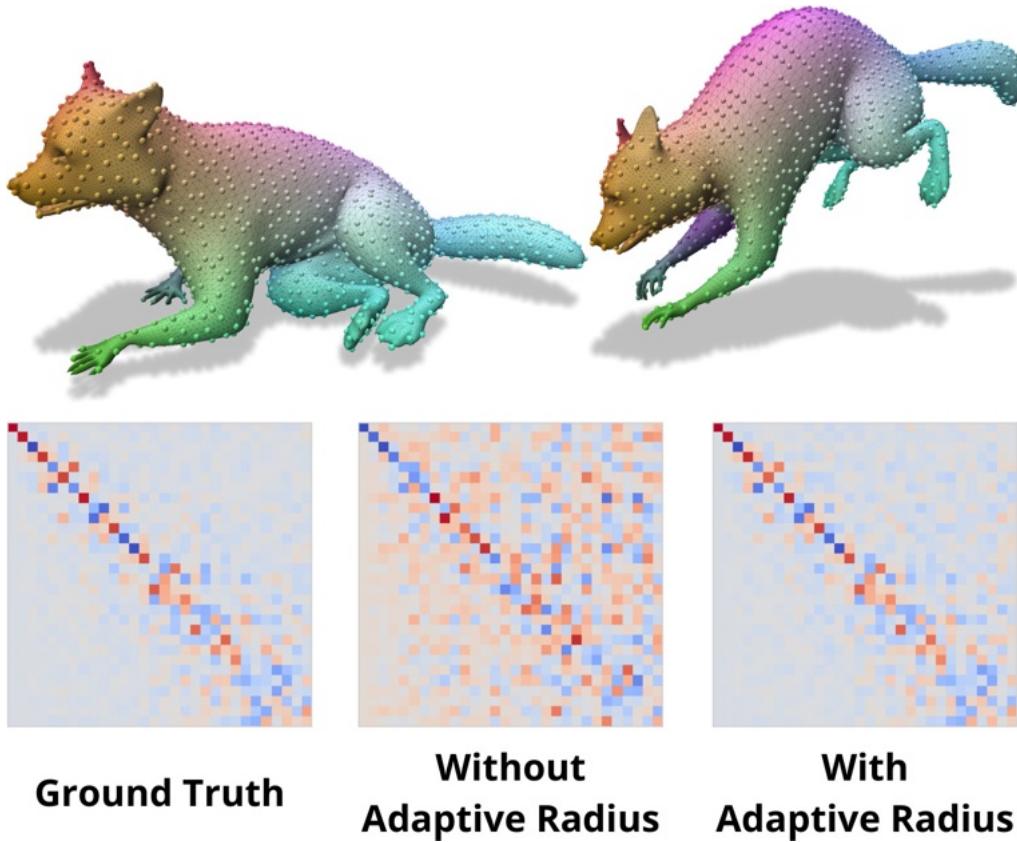


Figure 6.4: Effect of the adaptive radius on functional map approximation. Top row displays a pointwise map T from the right mesh to the left mesh using color transfer. Bottom row displays $\bar{\mathbf{C}}$ (Left), $\hat{\mathbf{C}}$ when using the pipeline from [149] (Middle) and our functional map $\hat{\mathbf{C}}$ (Right).

this does not require additional local geodesic distance computations. More details are provided in Section 6.5.5, and the algorithm to compute these new functions is displayed in Algorithm 5. We observe that the adaptive radius strategy generates better local functions than those introduced in [149], especially for non-uniform sampling, as can be seen on a surface from the DFaust dataset [29] in Figure 6.3. Note that since we focus on *local* analysis, a desirable property of the local interpolation function is the consistency across different shapes when only values at the samples are provided. With a single global radius, we see on Figure 6.3 that these functions can be heavily distorted by the normalization procedure, which is corrected by our approach. However, increasing the self-weights too close to 1 also deteriorates the results, as any vertex x within the radius of a single sample will be given the value of the sample point. There thus exists a limit at which this procedure ceases to be helpful, and the only solution then lies in increasing the number of samples on the shape.

The positive effect of our adaptive radius algorithm for functional map estimation is further visualized in Figure 6.4, where given a single pointwise map T , we display the exact functional map on the approximated spaces $\bar{\mathbf{C}}$, and two approximated

Algorithm 5 Computation of local functions with adaptive radius

Require: Mesh \mathcal{M} , samples $(v_k)_k$, initial ρ_0 , threshold ε

- 1: $\rho_j \leftarrow \rho_0 \quad \forall j$
 - 2: Compute local functions U with radius ρ : Equations (6.3) and (6.4)
 - 3: Add sample points if necessary
 - 4: **while** some k with $u_k(v_k) < \varepsilon$ **do**
 - 5: $j \leftarrow \arg \max_{i \neq k} u_i(v_k)$
 - 6: $\rho_j \leftarrow \rho_j / 2$
 - 7: update all u using Equations (6.3) and (6.4)
 - 8: **end while**
 - 9: Add unseen vertices in the sample
-

functional maps $\hat{\mathbf{C}}$, one being computed with a shared radius [149] and the other with our adaptive radius scheme. We highlight that the ground truth functional map actually differ for each approximation $\hat{\mathbf{C}}$ as the reduced functional spaces are modified, which makes values not directly comparable. However, we observe that the two ground truth maps have nearly identical sparsity structure (see Appendix D.6), which is why we only display one in Figure 6.4. Note that using the adaptive radius strategy then generates a sparsity pattern on matrix $\hat{\mathbf{C}}$ very close to the ground truth one.

6.5.4 Scalable ZoomOut

In light of the previous discussions and theoretical analysis, we now describe how to use the approximated functional map $\hat{\mathbf{C}}$ within a standard ZoomOut pipeline [145]. Our complete pipeline is summarized in Algorithm 6, where the notation $\bar{\Phi}_{1:k}$ indicates that we only use the first k column of matrix $\bar{\Phi}_{1:k}$.

Algorithm 6 Scalable ZoomOut

Require: Meshes \mathcal{M} and \mathcal{N} , threshold ε , initial map

- 1: Sample $\mathcal{S}^{\mathcal{M}}$ and $\mathcal{S}^{\mathcal{N}}$ using Poisson Disk Sampling
 - 2: Compute $\mathbf{U}^{\mathcal{M}}$ and $\mathbf{U}^{\mathcal{N}}$ using Algo. 5
 - 3: Approximate eigenvectors $\bar{\Phi}^{\mathcal{M}}$ and $\bar{\Phi}^{\mathcal{N}}$ using Eq. (6.5)
 - 4: Set $\bar{\Psi}^{\mathcal{M}} = \mathbf{U}^{\mathcal{M}} \bar{\Phi}^{\mathcal{M}}$ and $\bar{\Psi}^{\mathcal{N}} = \mathbf{U}^{\mathcal{N}} \bar{\Phi}^{\mathcal{N}}$
 - 5: Obtain $\bar{\Pi}$ between *samples* using the initial map
 - 6: **for** $k = k_{\text{init}}:k_{\text{final}}$ **do**
 - 7: $\hat{\mathbf{C}} = (\bar{\Phi}_{1:k}^{\mathcal{N}})^{\top} \bar{\mathbf{A}}^{\mathcal{N}} \bar{\Pi} \bar{\Phi}_{1:k}^{\mathcal{M}}$
 - 8: $\bar{\Pi} = \text{NNsearch}(\bar{\Phi}_{1:k}^{\mathcal{M}}, \bar{\Phi}_{1:k}^{\mathcal{N}} \hat{\mathbf{C}})$ potentially using Eq. (6.16)
 - 9: **end for**
 - 10: $\Pi = \text{NNsearch}(\bar{\Psi}_{1:k}^{\mathcal{M}}, \bar{\Psi}_{1:k}^{\mathcal{N}} \hat{\mathbf{C}})$
 - 11: **Return** Π
-

As mentioned earlier, using $\hat{\mathbf{C}}$ and matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{W}}$ allows applying the ZoomOut algorithm directly, as if it was applied on remeshed versions of the shapes with only p vertices. This results in a refined functional map $\hat{\mathbf{C}}^*$ and a refined pointwise map *between samples* $\bar{\boldsymbol{\Pi}}^*$. The last remaining non-trivial task consists in converting the refined functional map into a global pointwise map $\boldsymbol{\Pi}^*$ between the original dense meshes.

Standard approaches using remeshed versions of the shapes extend maps via nearest neighbors, resulting in locally constant maps. Instead, we identify $\hat{\mathbf{C}}$ and $\bar{\mathbf{C}}$, which then allows us to compute the pointwise map $\boldsymbol{\Pi}^*$ by solving the standard least square problem:

$$\boldsymbol{\Pi}^* = \arg \min_{\boldsymbol{\Pi}} \|\bar{\boldsymbol{\Psi}}^{\mathcal{N}} \hat{\mathbf{C}}^* - \boldsymbol{\Pi} \bar{\boldsymbol{\Psi}}^{\mathcal{M}}\|_{\mathbf{A}^{\mathcal{N}}}^2. \quad (6.15)$$

Since \mathbf{A} is diagonal this problem reduces to a nearest neighbor search for each vertex $x \in \mathcal{N}$. This way, the obtained pointwise map is no longer locally constant which results in a significant gain of quality with respect to typical approaches.

On meshes containing millions of vertices, this nearest neighbor search can, however, still be very slow. In these cases, we propose to use the computed pointwise map $\bar{\boldsymbol{\Pi}}$ as a guide to reduce the search space as follows: for $x \in \mathcal{N}$, we first select the indices of its nearest sample points $N(x) = \{j \mid u_j^{\mathcal{N}}(x) > 0\}$, and create the set of possible *images* as the points in \mathcal{M} close to the image of this set under the map $\bar{\boldsymbol{\Pi}}$, that is

$$\mathcal{I}(x) = \{y \mid \exists j \in N(x), u_{\bar{T}(j)}(y) > 0\} \quad (6.16)$$

where \bar{T} is the function representation of $\bar{\boldsymbol{\Pi}}$. Since local functions u_j are compactly supported, in practice, they are stored as sparse vectors and extracting the set of possible images of a given vertex therefore can be done efficiently through simple indexing queries.

6.5.5 Implementation

We implement the complete algorithm in Python and provide the code at https://github.com/RobinMagnet/Scalable_FM, built on top of the PYFM library, found at <https://github.com/RobinMagnet/pyFM>

Following [149], we generate sparse samples \mathcal{S} using Poisson Disk sampling, and run a fixed-radius Dijkstra algorithm starting at all sampled points v_j to build local functions u_j . Values can be stored in a sparse $n \times p$ matrix, where p is the number of samples. Note that the adaptive radius algorithm presented in Section 6.5.3 does not require additional geodesic distance computations. Furthermore, finding the set of potential images for a point as described in Section 6.5.4 simply reduces to checking non-zero indices in a sparse matrix. More details are provided in Appendix D.7.

6.6 Results

In this section, we evaluate our method, while focusing on two aspects. Firstly, we verify that our method outperforms existing approaches in terms of speed at

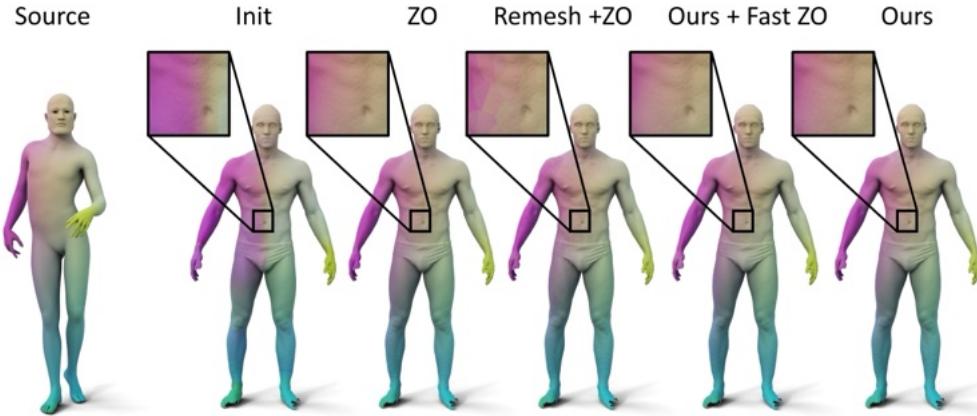


Figure 6.5: Qualitative results on the SHREC19 dataset. Although processing time differ heavily, there is no significant difference between our method and results from ZoomOut. However, remeshing the surface before ZoomOut results in locally constant correspondences.

Table 6.1: Timing in seconds for different methods when processing a pair with 50K and 200K vertices and applying ZoomOut from spectral size 20 to 100

| methods | <i>Preprocess</i> | <i>LBO</i> | <i>ZoomOut</i> | <i>Conversion</i> | <i>Total (s)</i> |
|---------|-------------------|------------|----------------|-------------------|------------------|
| ZO | 1 | 132 | 410 | 83 | 626 |
| Fast ZO | 10 | 132 | 1 | 44 | 187 |
| R + ZO | 14 | 2 | 3 | 1 | 21 |
| Ours | 10 | 7 | 5 | 44 | 65 |

all steps of the pipeline - that is, pre-processing as well as the ZoomOut algorithm. Secondly, we show this gain in speed comes at a minimal cost in terms of quantitative metrics. In particular, we verify that although our pipeline relies on sparse samples, we eventually obtain clear sub-sample accuracy in the correspondences.

6.6.1 Timings

The method introduced in [149] aimed at approximating the spectrum of the Laplace-Beltrami Operator efficiently. As mentioned above, the additional building blocks we introduced in Section 6.5.3 come at a nearly negligible computational cost as the main bottleneck lies in local geodesic distances computations, which are not recomputed. The main benefit of our method appears when considering the processing time of the ZoomOut algorithm. Indeed, since our algorithm does not involve any n -dimensional matrices, its running time becomes entirely agnostic to the original number of vertices. Only the final conversion step, which converts the refined functional map into a dense point-wise map, scales with the number of vertices.

Table 6.2: Evaluation of different methods on the complete SHREC19 dataset. Blue highlights the best two methods.

| methods | <i>Accuracy</i> | <i>Coverage</i> | <i>Smoothness</i> |
|-----------------|-----------------|-----------------|-------------------|
| Init | 60.18 | 26.5 % | 9.5 |
| GT | — | 33.0 % | 10.43 |
| ZO | 26.84 | 61.5 % | 6.2 |
| R + ZO | 28.57 | 18.0 % | 15.0 |
| Ours w/o radius | 71.35 | 29 % | 52.2 |
| Ours + Fast ZO | 29.5 | 59.7 % | 6.4 |
| Ours | 27.78 | 56.7 % | 5.6 |

Table 6.1 displays an example of timings when applying the ZoomOut algorithm between two meshes with respectively 50 and 200 thousands vertices. We compare the standard ZoomOut algorithm (ZO), the Fast ZoomOut algorithm (Fast ZO), the standard ZoomOut applied to remeshed versions of the shapes with nearest neighbor extrapolation (R+ZO) and our complete pipeline with $p = 3000$ samples on each shape. Notice that farthest point sampling used in Fast ZoomOut can become quite slow on dense meshes compared to Poisson sampling, which explains the similar preprocessing timings between our method and Fast ZoomOut.

6.6.2 Evaluation

Dataset As most shape matching methods scale poorly with the number of vertices, there are few benchmarks with dense meshes and ground truth correspondences for evaluation. The SHREC19 dataset [144] consists of 430 pairs of human shapes with different connectivity, all of which come with initial correspondences. Meshes in this dataset have on average 38 000 vertices, with the smallest and largest number of vertices having respectively 4700 and 200 000 vertices. Due to the limitations of existing shape matching methods, a remeshed version of this dataset is commonly used. In contrast, we display results on the *complete dense dataset*, and show that our method obtains similar results as ZoomOut [145] in only a fraction of the required time.

Metrics We evaluate different methods using standard metrics [174] for dense shape correspondence, that is accuracy, coverage, and smoothness. The accuracy of a computed dense map $T : \mathcal{N} \rightarrow \mathcal{M}$ gives the average geodesic distance between $T(x)$ and $T^*(x)$ for all $x \in \mathcal{N}$ where T^* denotes the ground truth map. Note that since maps on SHREC19 are only evaluated on a small subset of 6890 points this metric only captures partial information, and locally constant maps can still achieve high accuracy. Coverage and smoothness metrics provide additional information on the quality of correspondences and are sensitive to locally constant correspondences.

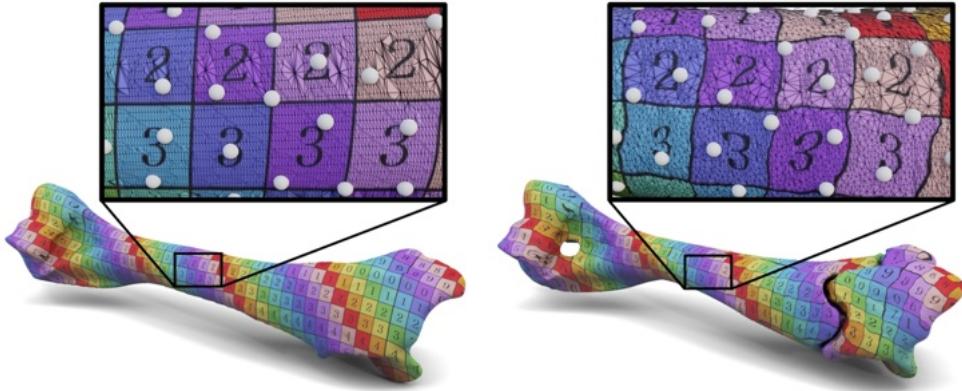


Figure 6.6: Texture transfer using our scalable version of ZoomOut. Samples used in the pipeline are shown as white dots.

Coverage is defined as the ratio of area covered by the pointwise map, and smoothness is the Dirichlet energy defined as the squared L^2 norm of the gradient of the transferred coordinates.

ZoomOut We compare our method (Ours) using 3000 samples first to the same algorithm without adaptive radius (Ours w/o radius), to the standard ZoomOut [145] algorithm applied on the dense meshes (ZO) and on remeshed versions with 3000 vertices (R+ZO). We don’t compare to other standard shape matching baseline [70, 77] first since we only wish to approximate results from ZoomOut, but also because these baselines don’t scale to high number of vertices. Additionally, despite the lack of theoretical guarantees, we evaluate a new version of Fast ZoomOut which uses functional map approximation (6.8) on the approximated functional space $\bar{\mathcal{F}}$ introduced in Section 6.5.1 (Ours + Fast ZO). Table 6.2 shows the values of the evaluation metrics on the SHREC19 dataset where the accuracy curves can be found on Figure 6.7, and Figure 6.5 shows an example of a map computed on two dense meshes. We see that all methods but R+ZO produce similar metrics, although processing times vary significantly. In contrast, the fastest method R+ZO produces locally constant maps as seen on Figure 6.5, which results in poor coverage and smoothness metrics. While our results are similar to ZoomOut and Fast ZoomOut, we stress that our results were obtained at a fraction of the processing time of ZoomOut, and come with theoretical upper bounds and control parameters on approximations which Fast ZoomOut does not have.

Sub-sample accuracy One Figure 6.6, we provide a result using texture transfer after applying our scalable ZoomOut on a pair of real scans of humerus bones obtained using a CT scanner [186]. This figure shows how our algorithm obtain sub-sample accuracy, as the transferred texture remains smooth even though samples are quite sparse on each shape. We display similar results using texture transfer on the SHREC19 dataset on Figure 6.8 and in Appendix D.8, which provides further details on the shapes.

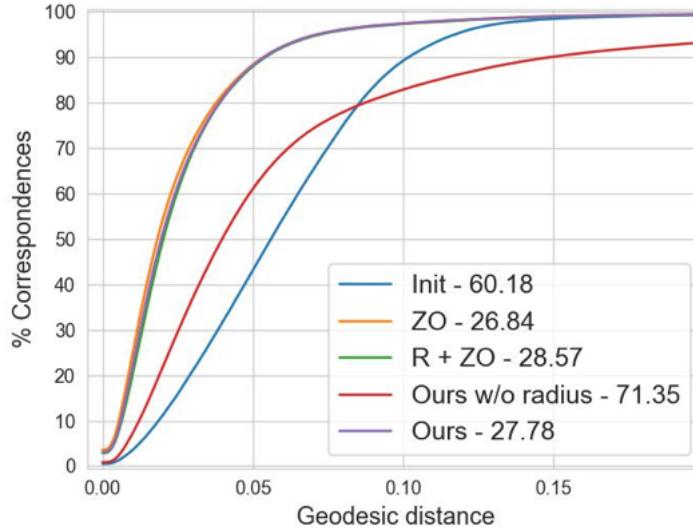


Figure 6.7: Accuracy curves for different methods presented in Table 6.2. Numbers in the legend provide the average geodesic error ($\times 10^3$).

Table 6.3: Norm of the estimation error Δ with and without adaptive radius on the SHREC19 dataset

| | <i>w/o radius</i> | <i>Ours</i> |
|----------------------|-------------------|-------------|
| $\Delta (\times 10)$ | 1.486 | 0.018 |

Adaptive Radius While results on Table 6.2 highlight the efficiency of the adaptive radius scheme, we additionally evaluate how this heuristic allows improving the estimation $\Delta = \|\bar{\mathbf{C}} - \hat{\mathbf{C}}\|$ presented in Section 6.5. For this, we simply compute $\bar{\mathbf{C}}$ and $\hat{\mathbf{C}}$ with $K = 20$ for all initial maps of the SHREC19 dataset, and evaluate the norms of the estimation errors Δ which we provide in Table 6.3. In this experiment, we notice our method improves the baseline by two orders of magnitude.

6.7 Conclusion, Limitations, and Future Work

In this chapter, we introduced a new scalable approach for computing correspondences between non-rigid shapes, represented as possibly very dense meshes. Our method is based on the efficient approach for estimating the Laplace-Beltrami eigenbasis [149] using optimization of coefficients of local extension functions built from a sparse set of samples. Key to our approach is careful analysis of the relation between functional spaces on the samples and those on the original dense shapes. For this, we extend this approach proposed in [149] and demonstrate how better behaved local functions can be obtained with very little additional effort. We use this construction

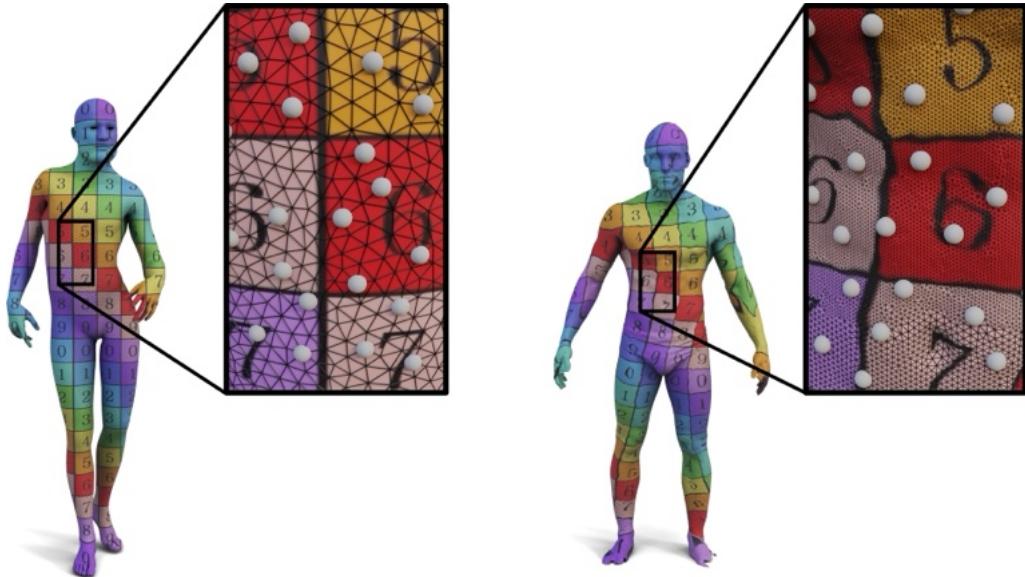


Figure 6.8: Texture transfer using our scalable version of ZoomOut on a pair of the SHREC19 dataset. Samples used in the pipeline are shown as white dots.

to define a functional map approximation that only relies on information stored at the samples, and provide theoretical guarantees for this construction. Finally, we use these insights to propose a scalable variant of the ZoomOut algorithm [145], which allows to compute high-quality functional and point-to-point maps between very dense meshes at the fraction of the cost of the standard approach.

Although our method achieves high-quality results, it still has several limitations. First, it relies heavily on the mesh structure, and is not directly applicable to other representations, such as point clouds. Second, our method depends on a critical hyperparameter, which is the number of samples. We have observed that 3000 samples perform well on a very wide range of settings, but it would be interesting to investigate the optimal number, depending on the size of the spectral basis. Furthermore, we use Poisson sampling as advocated in [149], which gives good results in practice. However, the optimal choice of the sampling procedure, depending on the geometric properties of shapes under consideration, would be an equally interesting venue for investigation. Lastly, an out-of-core implementation, capable of handling meshes with 10s of millions to billions of vertices, while possible in principle, would be an excellent practical future extension of our approach.

Acknowledgments The authors thank the anonymous reviewers for their valuable comments and suggestions. Parts of this work were supported by the ERC Starting Grant No. 758800 (EXPROTEA) and the ANR AI Chair AIGRETTE.

Memory-Scalable and Simplified Functional Map Learning

Deep functional maps have emerged in recent years as a prominent learning-based framework for non-rigid shape matching problems. While early methods in this domain only focused on learning in the functional domain, the latest techniques have demonstrated that by promoting consistency between functional and pointwise maps leads to significant improvements in accuracy. Unfortunately, existing approaches rely heavily on the computation of large dense matrices arising from soft pointwise maps, which compromises their efficiency and scalability. To address this limitation, we introduce a novel memory-scalable and efficient functional map learning pipeline. By leveraging the specific structure of functional maps, we offer the possibility to achieve identical results without ever storing the pointwise map in memory. Furthermore, based on the same approach, we present a differentiable map refinement layer adapted from an existing axiomatic refinement algorithm. Unlike many functional map learning methods, which use this algorithm at a post-processing step, ours can be easily used at train time, enabling to enforce consistency between the refined and initial versions of the map. Our resulting approach is both simpler, more efficient and more numerically stable, by avoiding differentiation through a linear system, while achieving close to state-of-the-art results in challenging scenarios.

7.1 Introduction

Automatically computing dense correspondences between non-rigid shapes is a classical problem in computer vision, forming the foundation of various downstream applications like shape registration [29], deformation [209, 59], and analysis [188]. A popular approach to tackle this problem involves the functional map pipeline [159], which represents correspondences as linear operators between functional spaces derived from the intrinsic Laplacian [146] on each shape. Numerous early methods [154, 174, 153] have leveraged this framework using handcrafted descriptors to generate functional maps, which can lack fine detail. Many algorithms [145, 177, 75, 133] have therefore successfully been developed in order to refine such imprecise maps into high quality dense correspondences.

Building upon pioneering efforts by [60], recent advancements [198, 71, 212] have successfully explored the possibility of *learning* descriptors directly from data for

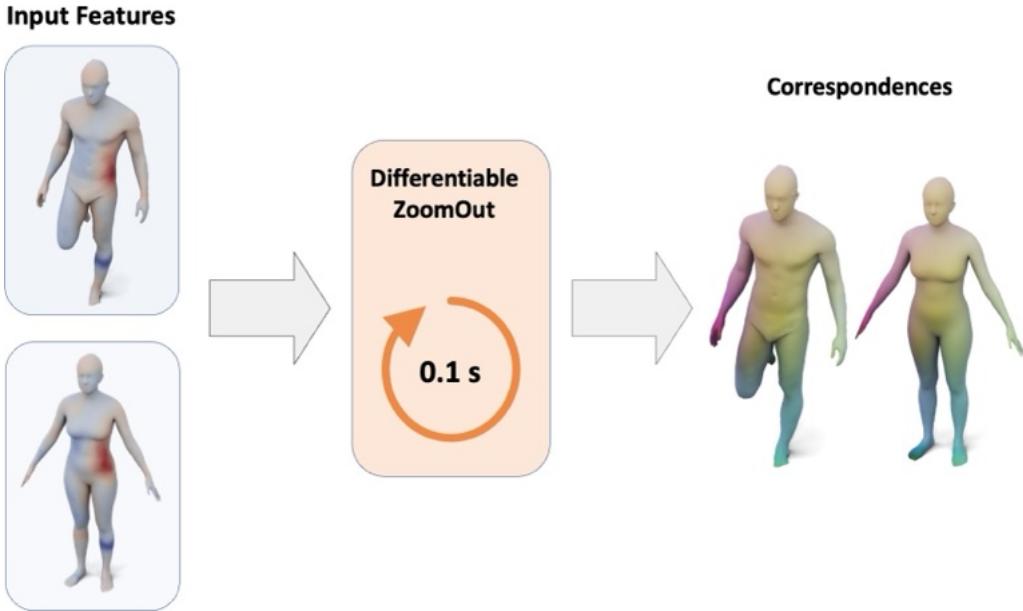


Figure 7.1: Our method takes a set of point features as input, which can be learned and uses a differentiable version of the ZoomOut algorithm to produce correspondences. Due to its light memory cost and attractive processing time this can be used while training a network, or when running the network on very dense meshes.

subsequent functional map computations, adapting the original pipeline introduced by [159, 184]. Notably, the most recent developments in this area have observed that promoting functional maps to be “proper” (i.e., functional maps arising from pointwise ones) can lead to significant improvement in accuracy. The concept of “proper” functional maps was introduced in the optimization setting [177] and then quickly adopted within the learning context. Specifically, recent deep functional map methods have constructed dual-branch networks [12, 123, 40, 212] that enforce the connection between pointwise and functional maps and that have demonstrated impressive performance across multiple datasets. Interestingly, these studies highlighted the necessity of retaining the original functional map branch [212] to achieve optimal performance, despite its inherent instability when differentiating through the linear system solver [68].

In all these works, however, the “properness” of functional maps is enforced by first computing a soft point-to-point map which is then converted to a functional map using matrix multiplication. This heavily limits the scalability of these approaches, as the dense pointwise map has to be stored in memory, which scales quadratically with the number of vertices. While common shape matching benchmarks only use meshes with low number of vertices, using these methods on real meshes is a serious challenge.

To address this limitation, we propose an approach that can compute the functional map associated with the soft p2p map, without *ever* storing the dense matrix in memory. Key to our approach is the fact that the proper functional map is defined as a matrix product between the soft pointwise map and the Laplacian ba-

sis [159, 177]. By exploiting this structure and GPU acceleration [44], we show that such matrix product can be computed directly without the necessity of storing the pointwise map, thus significantly improving both the speed and scalability of related approaches.

Our work additionally demonstrates the feasibility of discarding the original functional branch while preserving result quality. Our approach involves the transformation of a widely adopted map refinement algorithm [145], originally implemented on CPU, into a differentiable and memory-efficient GPU version using a similar pointwise map computation. Utilizing this refined map allows us to impose constraints on the structure of the learned functional map through a form of self-supervision. This, in turn, replaces the need for a consistency loss with the traditional functional map branch as in [40, 212], providing a novel simple and efficient solution for maintaining result quality in the absence of the original functional branch. Overall, our contributions can be summarized as follows:

- We propose efficient GPU implementation of differentiable pointwise map or functional map learning with minimal space complexity and numerical stability.
- We use a novel GPU adapted refinement algorithm at train time to provide self-supervision to the network.
- We introduce the first single-branch network for functional map learning without differentiating through a linear system solver.

7.2 Related Works

Shape matching and in particular functional map correspondence computation is a very wide and established a field of research. We here only review the works the closest to our work, and refer the interested reader to [162, 188] for an in-depth description of related works.

Functional Maps Our work is built upon the functional map framework, originally developed in [159] and later extended in various ways [154, 174, 145, 177, 61], an overview being provided in [162]. This approach encodes correspondences between shapes as small sized matrices independently of the original number of vertices, offering an efficient way to compute maps. This then allows to efficiently enforce constraints on the correspondences such as bijectivity or area preservation using simple linear algebra. The most effective functional map algorithms are map refinement algorithms [145, 177, 133, 70], which take initial correspondences as input and iteratively refine them. While highly robust, obtaining initialization without landmarks often relies on the use of handcrafted descriptors such as HKS [36], WKS [15] or SHOT [219].

Deep Functional Maps A more recent line of research focus on learning descriptor functions directly from the surface itself. Originating with FMNet [127, 184] and further developed in [60, 198], these approaches typically take handcrafted descriptors as inputs and yield refined descriptor functions. These functions are then used in a standard functional map pipeline [159], and are usually post-processed at test time using off-the-shelf map refinement algorithms [145, 227, 75, 174]. Using modern feature extractors for surfaces and point cloud [198, 216], these works obtained impressive results despite the unstable differentiation through a linear system solver [68]. While these initial approaches primarily focused on supervised learning, contemporary research in functional map learning emphasizes unsupervised learning of correspondences [212, 39, 40, 123]. This is achieved using functional map priors, that is, explicitly promoting structural properties on the learned functional map such as orthogonality - which corresponds to area preservation in the spatial domain. Recent advancements [177] have highlighted the importance of using extra structural constraint in the form of “proper” functional maps, that are functional maps obtained from pointwise correspondences, a guarantee not provided in the original pipeline [159] or learning-based approaches [60]. This led to the development of methods computing a second functional map at train-time using soft correspondences, resulting in dual-branches networks [12, 212, 40, 39]. These approaches were however recognized [212] as unable to scale to large meshes, due to large dense matrix computations, and had to use mesh resampling to avoid memory and speed issues.

Differentiable Refinement In a context also aligned with our work, it was noted in [123] that proper functional maps were guaranteed by many map refinement algorithms [145, 177, 133]. Subsequently, this refinement was partially integrated into a network as a differentiable post-processing step for the initially learned functional map. However, the design from [123] still relies on the original linear system solver, and their adaptation of [145] was only partial. This partial adaptation was necessitated by the potential memory overflow resulting from numerous dense map computations. Additionally, the output functional map was only a weighted sum of proper functional maps, thus lacking a guarantee of being proper itself.

7.3 Background & Motivation

Our method builds upon the functional map framework [159], and in particular of its recent development, using learning-based descriptors inspired by GeoFMaps [60]. Before describing our approach in Section 7.4, we provide an overview of the foundation of this pipeline. Interested readers are encouraged to explore numerous related works [12, 212, 123, 39, 40, 162] for additional insights into various adaptations and nuances of this framework.

Notations We will suppose to be given two shapes S_1 and S_2 with respectively n_1 and n_2 vertices. For each shape S_i , we compute its intrinsic Laplacian [146], and

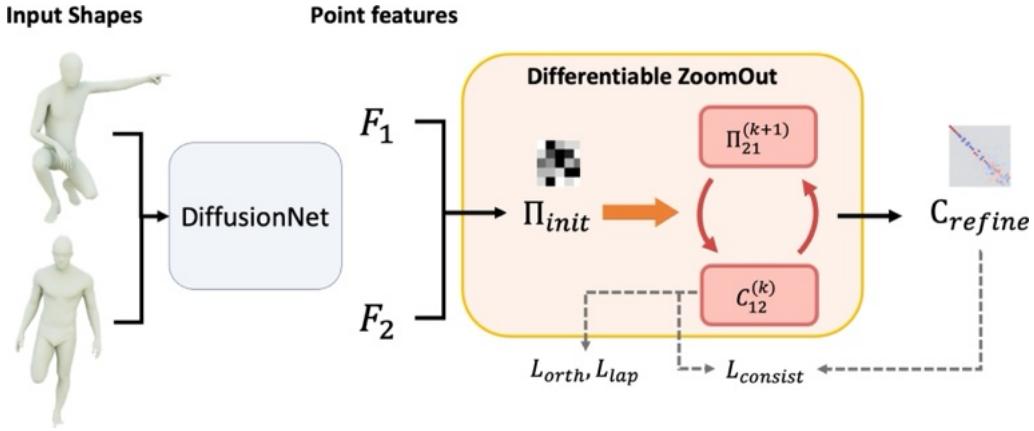


Figure 7.2: Our pipeline takes as input two shapes and use a feature extractor network to obtain pointwise features. These features are used to compute an initial pointwise map and then fed to our Differentiable ZoomOut block. All the pointwise maps Π are our scalable dense maps, which are fast and memory efficient.

store its eigenfunctions as columns of a matrix $\Phi_i \in \mathbb{R}^{n_i \times K}$. We denote $\Phi_i^\dagger = \Phi_i^\top A_i$ its pseudo-inverse, with A_i being the diagonal vertex-area matrix. Given any matrix B , we denote $[B]_i$ the vector consisting of the i -th line of B .

Deep Functional Maps The standard deep functional map pipeline [60] takes 2 shapes S_1 and S_2 as input, and use a feature extractor network \mathcal{F}_θ to generate p descriptor functions on each shape, stored as columns of matrices $F_i = \mathcal{F}_\theta(S_i) \in \mathbb{R}^{n_i \times p}$. Following the standard functional map pipeline [159] these descriptors are first projected into the Laplacian basis $\mathbf{A}_i = \Phi_i^\dagger F_i \in \mathbb{R}^{K \times p}$ and a functional map is obtained by solving the linear system:

$$\arg \min_{\mathbf{C}} \|\mathbf{CA}_1 - \mathbf{A}_2\|_2^2. \quad (7.1)$$

This linear system is further usually regularized using an extra Laplacian term [60, 173]. During training, losses are then imposed on the computed functional map $\mathbf{C}(\mathcal{F}_\theta(S_1), \mathcal{F}_\theta(S_2))$. At test time, a pointwise map can be recovered from the map \mathbf{C} and the eigenfunctions Φ_i using nearest neighbor search [159, 164].

Two branches networks Recent works in functional map literature [177] have highlighted the positive effects of using *proper* functional maps. A functional map is proper if it arises from *some* underlying pointwise map. Specifically, a proper functional map is *defined* as the pull-back of a pointwise map $T : S_2 \rightarrow S_1$:

$$\mathbf{C} = \Phi_2^\dagger \Pi \Phi_1 \quad (7.2)$$

where $\Pi \in \{0, 1\}^{n_2 \times n_1}$ is the matrix representation of the map T . Several works [12, 212, 40] adopt a differentiable approach to compute Π before deriving \mathbf{C}_{proper} using

Equation (7.2). Typically, the map Π is computed from the features F_1 and F_2 using a Gaussian kernel:

$$\Pi_{ij} = \frac{\exp(\delta_{ij})}{\sum_k \exp(\delta_{ik})} \quad (7.3)$$

with $\delta_{ij} = -\frac{1}{2\sigma^2} \| [F_2]_i - [F_1]_j \|^2$ the distance between rows of the feature matrices, where σ a temperature - or blur parameter. For training purposes, a consistency loss between \mathbf{C} , obtained with Eq. (7.1), and $\mathbf{C}_{\text{proper}}$, derived using Eqs. (7.2) and (7.3), is employed. This approach is taken *in addition* to the standard orthogonality or bijectivity losses presented in [60].

ZoomOut A popular map refinement algorithm named ZoomOut [145] has often been used to obtain high-quality correspondences from low quality initial functional maps such as those obtained from learning pipelines. ZoomOut iteratively computes functional maps using Eq. (7.2) and pointwise map using nearest neighbor search between the rows of $\Phi_1 \mathbf{C}^T$ and Φ_2 . Note that due to its iterative nature, ZoomOut is *guaranteed* to produce proper functional maps. A recent *approximation* [131] made the algorithm scalable to dense meshes on CPU, but however relies on sampling, a longer pre-processing and a final slow conversion from the samples back to the full shapes.

Drawbacks and motivation Despite achieving high quality results on shape matching benchmarks, the modern two-branches approaches presented above suffer from three notable drawbacks. Firstly, computing Π using Eq. (7.3) involves storing and differentiating through a dense $n_2 \times n_1$ matrix, making the method scale poorly in terms of memory. In particular, because of the linear system used in the other branch, features are required to be of high dimension (usually 128 or 256) to ensure invertibility of the feature matrix, thus heavily slowing down computations. Secondly, a naive implementation of Eq. (7.3) can result in underflows in the forward or backward pass for low values of σ . Thirdly, as remarked in some previous works [68] despite its necessity for achieving satisfactory results, the original functional map branch from [60] poses a risk of instability due to differentiation through the linear system solver.

In this work, we seek to address these challenges by establishing soft point-wise maps as a stable and memory-scalable option to learn functional maps, without approximations such as those presented in [131]. A second goal lies in trying to completely remove the spectral branch from the learning procedure. The necessity of the spectral branch suggested in [212] hints that properness might not be a sufficient constraint alone for efficient learning of correspondences. To overcome this challenge, we further refine the structural constraints by introducing the expectation that the functional map aligns with its refined version, produced by [145]. This leads to the first deep functional map method that completely avoids solving a linear system inside the network, enables unsupervised training, is scalable, efficient and leads to high quality results.

7.4 Method

In this section, we introduce our scalable approach to proper functional maps, which we then apply to design a novel GPU based differentiable version of the ZoomOut [145] algorithm. Finally, using these two elements, we introduce our new single branch network for functional map learning without a linear system solver.

7.4.1 Scalable Dense Maps

In this section, we introduce our scalable approach to proper functional maps, which we then apply to design a novel GPU based differentiable version of the ZoomOut [145] algorithm. Finally, using these two elements, we introduce our new single branch network for functional map learning without a linear system solver.

7.4.2 Scalable Dense Maps

The key observation to this work is that all dense pointwise maps computed in deep functional map pipelines [123, 212, 13, 40] are used *exclusively* to compute functional maps using Equation (7.2). In particular, they are invariably found in a matrix product of the form $\Pi\Phi_1$. Previous work did not seek to exploit this fact, and instead computed the complete dense matrix Π separately before performing the matrix product. In contrast, we argue it is possible to compute the result of this inner product without ever computing any dense $n_2 \times n_1$ matrix.

Observe first that we can explicitly write the i -th line of $\Pi\Phi_1$, using Equation (7.3) as:

$$[\Pi\Phi_1]_i = \sum_{j=1}^{n_1} \frac{\exp(\delta_{ij})}{\sum_k \exp(\delta_{ik})} [\Phi_1]_j \quad (7.4)$$

$$= L_i^{-1} \sum_{j=1}^{n_1} K([F_2]_i, [F_1]_j) [\Phi_1]_j. \quad (7.5)$$

where K is an RBF Kernel, and L_i the row normalization. By rewriting the proper functional map definition in this kernel form, we can now leverage existing methods for heavily scalable and fast GPU computation with kernels [185, 142, 44]. These methods rely on, in particular, the fact that the entry (i, j) of the Kernel matrix $K = (\exp(\delta_{ij}))_{ij}$ *only* depends on the vectors $[F_2]_i$ and $[F_1]_j$. This allows to compute the sum in Equation (7.5) in a block-wise manner, where the values of K are computed during summation. This is highlighted in Figure 7.3, where we represent the dense matrix on which summation is applied in Equation (7.5). The per-row sum can then be computed first for each contiguous memory block before summing all the outputs to obtain the value of $\Pi\Phi_1$.

In practice, we rely on the Keops library [44], which applies such operations on very large dense matrices whose entries can be described by mathematical formulas applied to the inputs. Keops uses symbolic matrices, and computes reduction on-the-fly using per-block operations for fast computation without ever fitting the dense matrix in memory.

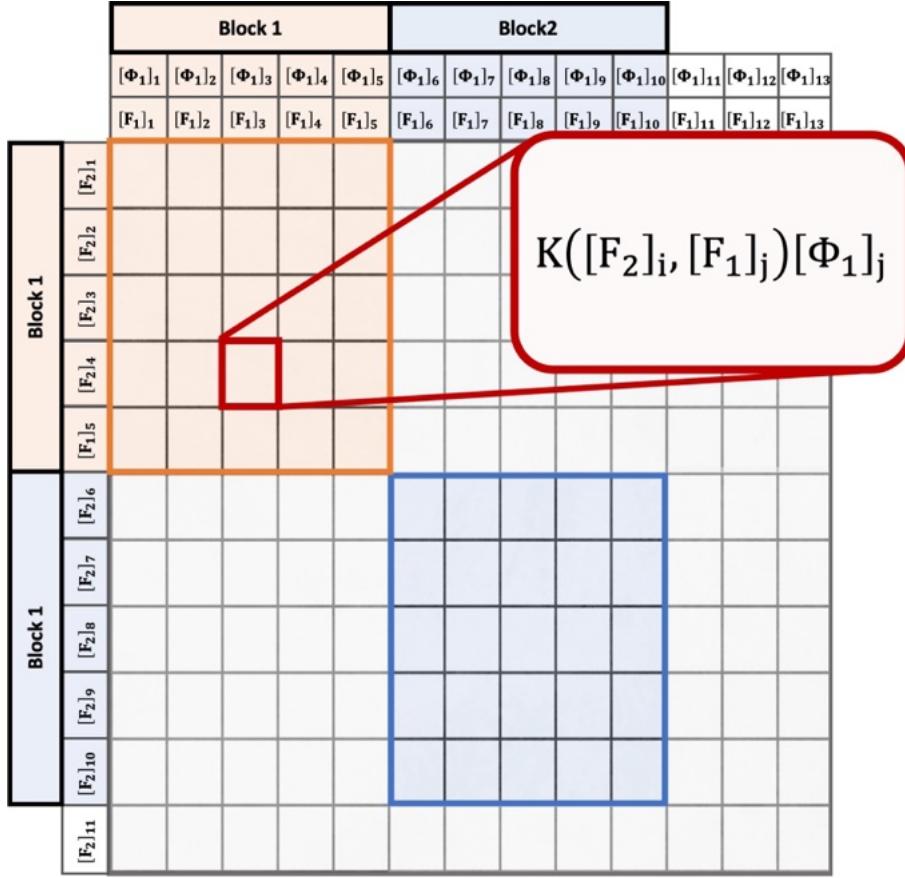


Figure 7.3: Our scalable dense maps relies on the underlying structure of Eq. (7.5), where the sum is computed for each contiguous memory block highlighted in the image. The entries are evaluated on the fly while performing summation, and results from each block are then accumulated to obtain the final per-rows values. The implementation is provided by the Keops package [44].

Note that the normalization L_i can additionally be handled using efficient stabilized logsumexp reductions and incorporated into the Kernel K to avoid underflow or overflow in the exponential. Furthermore, the gradient of $\Pi\Phi_1$ with respect to F_1 and F_2 can be computed using a similar trick [44].

At test time, a vertex-to-vertex map can be extracted from Π by looking for the indices of the per-row maximal value, which is equivalent to running nearest neighbor search between the rows of F_1 and the rows of F_2 . This can again be run efficiently on GPU without computing the dense distance matrix, using GPU-based nearest neighbor implementations [44, 105].

Ultimately, our dense pointwise map only stores values for F_1 and F_2 as well as the type of Kernel we use, and has therefore a linear memory cost.

7.4.3 Differentiable ZoomOut

As mentioned in Section 7.3, the ZoomOut algorithm [145] iteratively performs pointwise map computations using nearest neighbor queries between rows of $\Phi_1 \mathbf{C}_{12}^\top$ and of Φ_2 , and functional map computations using Eq. (7.2), while increasing the size K of the spectral basis. By replacing the nearest neighbor queries by differentiable soft maps that we store using our scalable versions, we introduce Differentiable ZoomOut, a fast and fully differentiable block, with negligible memory cost. The algorithm is presented in detail in the supplementary material.

Since ZoomOut acts as a powerful map refinement algorithm, we would like to enforce consistency between the output and input functional maps of the ZoomOut algorithm in order to help training. We expect such a loss to provide meaningful guidance to the features.

However, we note that the output functional map $\mathbf{C}_{\text{refine}}$ has a larger size than the initial map \mathbf{C}_{init} . This is due to ZoomOut using an increasing size of spectral basis. However, given a proper functional map of size $K_2 \times K_1$ associated to a pointwise map Π , the principal submatrix composed of its first K'_2 rows and K'_1 column from the proper functional map of size $K'_2 \times K'_1$ associated to the same map Π . This stems from the definition of proper functional maps [159], and we refer to the supplementary for details on this aspect. Therefore, our new consistency loss only uses a principal submatrix of the refined functional map:

$$L_{\text{consist}}(\mathbf{C}_{\text{init}}, \mathbf{C}_{\text{refine}}) = \|\mathbf{C}_{\text{init}} - [\mathbf{C}_{\text{refine}}]_{1:K_{\text{init}}, 1:K_{\text{init}}} \|_2^2 \quad (7.6)$$

where K_{init} is the size of the input functional map.

7.4.4 Overall Pipeline and Implementation

We would first like to highlight that our scalable dense maps can be used in any existing functional map base model using dense pointwise maps, with no impact on the results. Furthermore, we present a novel single-branch network for functional map prediction which exploits the structural properties of proper functional maps and does not require solving or differentiating through a linear system. We therefore present separate implementations first for our scalable dense maps and differentiable ZoomOut at <https://github.com/RobinMagnet/ScalableDenseMaps>, and of our entire pipeline at <https://github.com/RobinMagnet/SimplifiedFmapsLearning>.

As shown in Figure 7.2, our algorithm first extracts features from surfaces S_1 and S_2 using DiffusionNet [198]. This produces matrices of features $F_1 \in \mathbb{R}^{n_1 \times p}$ and $F_2 \in \mathbb{R}^{n_2 \times p}$. Importantly, we select $p = 32$ instead of the 128 or 256 Features produced by standard pipelines [123, 39, 212, 40, 12], as our approach does not require invertibility of a linear system obtained from the learned features.

An initial soft pointwise map Π_{init} is produced from the features using Equation (7.3), and then fed into our Differentiable ZoomOut algorithm presented in Section 7.4.3 where we perform 10 iteration with a spectral step size of 10 starting with $K_{\text{init}} = 30$. This results in a refined map $\mathbf{C}_{\text{final}}$ of size $K_{\text{final}} = 130$. This whole process uses a blur parameter $\sigma = 10^{-2}$, which is much lower than previous implementations [200, 123, 212].

| Train | F | | | S | | | F+S | | |
|----------------------------|------------|------------|------------|------|------------|------------|------------|------------|------------|
| Test | F | S | S19 | F | S | S19 | F | S | S19 |
| BCICP [174] | 6.1 | - | - | - | 11. | - | - | - | - |
| ZoomOut [145] | 6.1 | - | - | - | 7.5 | - | - | - | - |
| SmoothShells [70] | 2.5 | - | - | - | 4.7 | - | - | - | - |
| DiscreteOp [177] | 5.6 | - | - | - | 13.1 | - | - | - | - |
| GeomFmaps [60] | 3.5 | 4.8 | 8.5 | 4.0 | 4.3 | 11.2 | 3.5 | 4.4 | 7.1 |
| Deep Shells [71] | 1.7 | 5.4 | 27.4 | 2.7 | 2.5 | 23.4 | 1.6 | 2.4 | 21.1 |
| NeuroMorph [72] | 8.5 | 28.5 | 26.3 | 18.2 | 29.9 | 27.6 | 9.1 | 27.3 | 25.3 |
| DUO-FMNet [62] | 2.5 | 4.2 | 6.4 | 2.7 | 2.6 | 8.4 | 2.5 | 4.3 | 6.4 |
| UDMSM [39] | 1.5 | 7.3 | 21.5 | 8.6 | 2.0 | 30.7 | 1.7 | 3.2 | 17.8 |
| ULRSSM [40] | 1.6 | 6.4 | 14.5 | 4.5 | 1.8 | 18.5 | 1.5 | 2.0 | 7.9 |
| ULRSSM (w/ fine-tune) [40] | 1.6 | 2.2 | 5.7 | 1.6 | 1.9 | 6.7 | 1.6 | 2.1 | 4.6 |
| AttentiveFMaps [123] | 1.9 | 2.6 | 5.8 | 1.9 | 2.1 | 8.1 | 1.9 | 2.3 | 6.3 |
| ConsistentFMaps [212] | 2.3 | 2.6 | 3.8 | 2.5 | 2.4 | 4.5 | 2.2 | 2.3 | 4.3 |
| Ours | 1.9 | 2.4 | 4.2 | 1.9 | 2.4 | 6.9 | 1.9 | 2.3 | 3.6 |

Table 7.1: Mean geodesic errors ($\times 100$) when training and testing on the Faust, Scape and Shrec19 datasets. Best result is shown in bold.

Our unsupervised training loss consists in 3 terms. First, an orthogonality constraint $L_{\text{orth}}(\mathbf{C}_{\text{init}}) = \|\mathbf{C}_{\text{init}}^\top \mathbf{C}_{\text{init}} - I\|_2^2$ is applied to the initial functional map, with a weight of 1. The ZoomOut consistency loss from Equation (7.6) is applied with an initial weight of 10^{-4} , gradually increased to 10^{-1} . This term is therefore ignored during the first epochs until decent initialization has been found. We refer to the supplementary for more details on this aspect. We finally regularize the result using a Laplacian commutativity term as presented in [173, 39], which is a residual from the spectral branch we discarded. This final term receives a weight of 10^2 . Eventually, we train our network using ADAM optimizer [110] with an initial learning rate of 10^{-3} . We refer the reader to the supplementary for some more precise details on the implementation.

7.4.5 Properties of learned features

An interesting aspect of the two-branches networks [12, 212, 40] is that each branch offers a different interpretation of the learned features. On the one hand, the standard functional map branch [159, 60] uses features F_1 and F_2 as *functions* on the shapes, expected to correspond, and forces the functional map to effectively transfer them when solving the linear system in Equation (7.1). On the other hand, the pointwise-map based branch solely relies on distances between rows of the feature

matrices (Eq. (7.3)), viewing features as pointwise embeddings only.

Using a consistency loss between the two branches enables to merge the two effects, and, as highlighted in [212], removing the spectral branch has a serious impact on the results. In our experiments in Section 7.5, we observe that features learned without the spectral branch usually exhibited undesirable high-frequency variations. As emphasized by [12], smoothness of features is a key aspect for the generalization for functional map based methods. While all networks using the spectral branch provide relatively smooth features, we show in Section 7.5.4 that replacing this branch using a refinement consistency loss also promotes smoothness in features in our pipeline.

7.5 Results

In this section, we conduct a series of experiments to assess various aspects of our proposed method. In order to validate the capability of our entire pipeline, we first compare our method to several other works on multiple shape matching benchmarks. We additionally wish to highlight our scalable dense maps appear as a valuable tool for many functional map based networks using dense pointwise maps, independently of our complete pipeline. We therefore emphasize the memory scalability of our GPU-based ZoomOut algorithm compared to existing implementations of the algorithm.

Finally, inspired by [12] we analyze how our novel ZoomOut consistency loss we introduced at train-time influences the features learned by our feature extractor.

7.5.1 Datasets

We evaluate the shape matching performance of our algorithm across four widely-used human datasets, commonly employed as benchmarks. The evaluation includes the remeshed [174] version FAUST dataset [28] which contains 100 shapes, split in 80 and 20 shapes for training and testing as introduced in [60]. We also use the remeshed [174] version of the SCAPE dataset [10] with 71 humans divided in 51 shapes for training and 20 for testing. For testing purposes only, the remeshed version of the SHREC19 dataset [144], composed of 44 shapes, is also included.

While these datasets mostly contain near-isometric shapes, we also evaluate our method on the remeshed [133] Deforming Things 4D dataset [124], a challenging non-isometric dataset of humanoid shapes. In particular, we focus on the adapted version DT4D-H defined in [123], which defines 198 shapes for training and 95 for testing. Results on the SMAL [247] dataset, with PCK curves, can be found in the supplementary material.

7.5.2 Shape Matching Results

In this work, we exclusively evaluate unsupervised learning performances, and therefore discard baselines focusing on pure supervised learning [222, 127, 87, 237]. As

| Train | DT4D-H | |
|-----------------------|-------------|-------------|
| Test | intra-class | inter-class |
| Deep Shells [71] | 3.4 | 31.1 |
| DUO-FMNet [62] | 2.6 | 15.8 |
| AttentiveFMaps [123] | 1.2 | 14.6 |
| ULRSSM [40] | 0.9 | 4.4 |
| ConsistentFMaps [212] | 1.2 | 6.1 |
| Ours | 1.8 | 4.1 |

Table 7.2: Mean geodesic errors ($\times 100$) on the DeformingThing4D dataset subset from [123] (DT4D-H). Best results are highlighted in bold.

a reference, we provide results using axiomatic functional map algorithms such as ZoomOut [145], Discrete Optimization [177], BCICP [174] and SmoothShells [70].

Our method can directly be compared to the following baselines: GeomFmaps [60], DUO-FMNet [62], DeepShells [71], NeuroMorph [72], AttentiveFMaps [123], ConsistentFMaps [212], UDMSM [39], and ULRSSM [40]. Note that we all results are presented without test time refinement for fairness. In particular, ULRSSM [40] relies on fine-tuning the network for each shape in the test dataset independently, which we turn off to obtain the result. We provide results with fine-tuning using the “w/ fine-tune” tag. Note that we provide comparison with a more complete set of methods in the supplementary materials, as well as results of our pipeline without using the consistency loss.

Table 7.1 provides the mean geodesic error for all the aforementioned baselines, as well as for our pipeline described in Section 7.4.4. We evaluate our methods on combinations of the Faust (**F**), Scape (**S**) and Shrec19 (**S19**), when training either on Faust and Scape independently, or jointly (**F+S**). This table shows our simple pipeline provides similar performance to state of the arts methods, all the while being greatly scalable to large meshes and removing the need for differentiation through a linear system solver.

In addition, we evaluate our network on the DEFORMINGTHINGS4D dataset, and in particular on the subset provided in [123] for evaluation, as displayed on Table 7.2. Our method achieves better performance than existing baselines on the inter-class category, which shows its capabilities even in non-isometric scenarios.

7.5.3 Scalability to Dense Meshes

In this section, we discuss the memory efficiency of our scalable maps, and highlight its speed performance in the case of very dense meshes where standard methods would go out of GPU-memory.

A first observation, provided in Figure 7.4 shows the GPU memory usage, using varying number of vertices, of current state-of-the-art methods for unsupervised

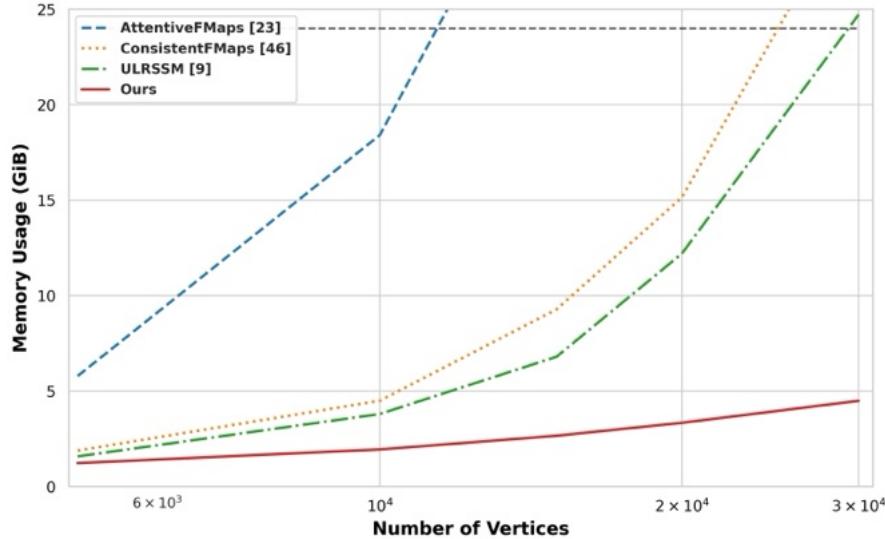


Figure 7.4: GPU memory usage when processing a single pair of shapes, depending on their vertex count. Note, e.g., that AttentiveFMaps [123] runs out of 24GB memory after 11k vertices.

shape matching. In particular, we notice that AttentiveFMaps [123], due to its multiple dense pointwise map computations, quickly runs out of 24 GiB GPU memory. On the other hand, while our method uses 11 different pointwise maps, its memory footprint remains significantly lower than competing methods [212, 40], in particular for large number of vertices.

Secondly, we analyze our results on the standard axiomatic ZoomOut algorithm [145], often used independently of learning pipelines, *e.g.* as a means to obtain maps from simple landmarks. In that case, we observe that usual implementations never leverage GPU acceleration and were only run on CPU, and we easily ported the code to GPU using PyTorch.

In Table 7.3, we first compare the CPU, GPU, and our version of ZoomOut, and show that the processing time in the presence of dense meshes remain reasonable. Our version of ZoomOut (“Our ZoomOut”) uses the same tools used to implement our Differentiable ZoomOut in Section 7.4.3, with a scalable version of brute force nearest neighbor in Keops [44], which again does not require fitting the distance matrix in memory. We additionally compare a naïve PyTorch implementation of our Differentiable ZoomOut (Sec. 7.4.3) with one using our scalable dense maps. Finally, we add results by porting the approximation from [131] to GPU and using scalable dense maps (“Our + [131]”). More details on this mix and its usage are provided in the supplementary material.

Table 7.3 presents the results of applying these algorithms to shapes of varying sizes. In the initial experiment with meshes of around 5000 vertices, all methods exhibit similar performance, significantly outperforming the CPU-based algorithm due to GPU utilization. However, with denser meshes containing 10^5 vertices, conventional methods encounter GPU memory limitations, while our scalable dense maps offer notable improvements over existing approaches. Moreover, our modifica-

| | Sparse (5K) | Dense (100k) |
|-------------------|-------------|--------------|
| CPU ZoomOut | 3.6 s | 700 s |
| GPU ZoomOut | 0.1 s | OOM |
| GPU Diff. ZoomOut | 0.1 s | OOM |
| Our ZoomOut | 0.1 s | 2.4 s |
| Our Diff. ZoomOut | 0.1 s | 5 s |
| Our + [131] | 0.1 s | 0.4 s |

Table 7.3: Average processing time in seconds, between CPU, naïve and scalable GPU implementations of ZoomOut and Differentiable ZoomOut

tion of [131], which approximates the algorithm, presents the fastest results without memory overloading. This solves the main speed bottleneck presented in [131], with more details provided in the supplementary.

Our method therefore allows using several dense pointwise maps simultaneously, or training and testing functional maps network directly on dense shapes. We refer the interested reader to the supplementary material for such experiments on dense meshes, including texture transfer visualization.

7.5.4 Learned Features

As highlighted in [12], analyzing the features learned in deep functional map networks valuable insights into their performance. In particular, it was shown that achieving smooth features positively impacts the network’s generalization capabilities.

The authors of [12] thus advocated explicitly enforcing features smoothness using spectral projection. This was used in [212] as well as in AttentiveFMaps [123]. In contrast, we do not enforce such constraints and no loss in our pipeline directly promotes smoothness. In particular, the dense pointwise map Π built from the features do not use any neighboring information.

However, we show that the consistency loss introduced in Section 7.4.3 actually pushes the feature extractor to learn smooth features. To observe this, we retrain our network on the Scape dataset while removing the consistency loss from Equation (7.6), and visualize the learned features on test datasets. Figure 7.5 shows example of feature functions produced by the networks when trained with and without the consistency loss on a random surface from the SHREC19 dataset. On the left side of this image, we observe that without refinement consistency, the features seem to highlight multiple small patches on the surface. In contrast, the feature functions learned by our method, displayed on the rightmost part of the image, present nicer patterns where large geodesic patches of the surfaces are highlighted.

We argue that obtaining an orthogonal functional map from a soft pointwise map built with features does not require such features to exhibit smoothness. However, the introduction of the consistency loss serves a dual purpose. While its primary

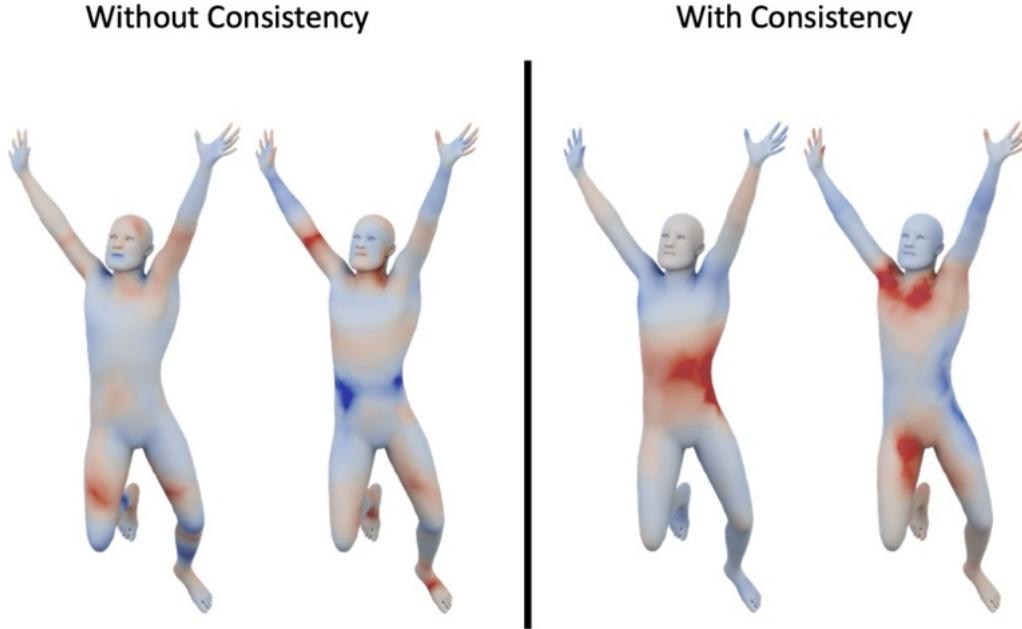


Figure 7.5: Example of feature functions learned by our model, with or without consistency loss. As noted by [12], smoother features are generally preferred for generalization purposes.

role is to align the functional map with the output of a refinement algorithm, it inadvertently acts as a compelling constraint that encourages the learning of smoother features. As this property has been noted as key to performance [12, 212], these results lead us to believe that incorporating such a loss into existing pipelines holds significant promise for enhancing overall performance in functional map learning.

7.6 Conclusion, Limitations & Future Work

In this work, we presented a novel approach to compute functional maps using soft pointwise maps, without ever storing the dense matrix in memory. This novel implementation enables use to derive a fast, differentiable and memory efficient version of the ZoomOut algorithm [145]. In turn, we use this algorithm while training and derive a new consistency loss between the initial and refined version of the predicted functional map. We notice this loss appears particularly effective and allows us to use a new single-branch architecture for functional map learning, which does not require differentiating through a linear system.

One major limitation of our method is its dependence on the computation of the spectrum Laplacian of the input shapes, which can become prohibitively slow with larger shapes. Furthermore, the ZoomOut algorithm, while particularly fit to handle near-isometric shapes, is prone to fail in the presence of highly non-isometric deformations or partiality [133]. The guidance provided by the consistency loss would then be unfit for the problem.

Future research could therefore seek to handle meshes with higher differences

such as partiality, noise or simply high distortion. This would potentially require incorporating other refinement algorithms into the training pipeline. Investigating the impact of our new consistency loss in various pipelines would also contribute to a comprehensive understanding of its applicability and effectiveness.

Acknowledgements. The authors thank the anonymous reviewers for their valuable comments and suggestions. Parts of this work were supported by the ERC Starting Grant 758800 (EXPROTEA), ERC Consolidator Grant 101087347 (VEGA), ANR AI Chair AIGRETTE, as well as gifts from Ansys and Adobe Research.

Conclusion

Throughout this thesis, we presented our contributions to the field of shape matching, particularly focusing on spectral shape matching. Our work began with Chapter 3, where we proposed an approach for extracting local information about differences between shapes using correspondences. Building upon this foundation, Chapter 4 leveraged similar tools for disease detection using real scans, highlighting the disparities between the practical needs of practitioners and the predominant interests in shape correspondence within the research field.

Subsequent chapters expanded upon our contributions, with Chapter 5 explicitly addressing the promotion of smoothness — an often overlooked aspect in traditional methodologies. Chapters 6 and 7 were dedicated to the advancement of modern functional map methods, aiming at enhancing their scalability to real-world dense meshes.

8.1 Evolution of the Field and Impact of Our Work

The field of geometry processing has seen rapid advancements throughout the duration of this thesis, with our research contributing to recent progress.

Our exploration of shape differences, detailed in Chapters 3 and 4, represents one of the first efforts in applying functional map methods to real-world data analysis [143, 176, 217]. This underscores the potential of recent advancements in shape matching for novel applications in practical domains, such as disease detection [73]. Following this, our introduction of smoothness-promoting functional maps in Chapter 5 has established a framework for developing hybrid methods that leverage both spatial and spectral information, offering significant potential when applied to real data. The dataset we released, a remeshed version of the DEFORMINGTHINGS4D dataset [124] with ground truth correspondences, continues to serve as a standard challenging benchmark for evaluating modern shape matching pipelines [39, 123, 40, 212]. Lastly, our research towards defining scalable functional maps also represents some of the earliest attempts to explicitly tackle this constraint [149], which is essential for providing usable tools across various types of data.

Throughout the duration of this thesis, the field of 3D shape analysis has also evolved. Functional map methods have seen enhancements, including extensions of computations to vector fields on shapes rather than solely real-valued functions [61, 62], as well as the incorporation of novel extrinsic information using an elastic energy basis [92, 19], among several others. Moreover, the domain of learning on 3D data has witnessed significant success, with the development of specialized architectures [198, 205] and learning strategies [241, 245] that have notably improved the performance of shape matching pipelines, positioning functional maps as the state-of-the-art framework to address this problem [123, 40, 212]. Alternative data representations such as implicit surfaces and signed distance fields have also gained popularity [63, 46], particularly when used with deep neural networks. Lastly, with the development of powerful networks for the analysis and generation of images [156, 183, 111] and text [172, 122], a novel area of research seeks to leverage such tools to perform shape analysis [1, 2].

8.2 Limitations and Future Work

Despite its quick evolution, it is important to acknowledge the existing limitations and challenges faced by the field of shape matching and its applications. As we conclude this thesis, we therefore highlight several avenues for future exploration and development in this field.

Shape Matching

While significant advancements have been made in automating the process of establishing correspondences between objects, the field of shape matching continues to face various challenges.

One prominent issue across all shape correspondence approaches, whether based on deformations or intrinsic constraints like functional maps, is partial shape matching. Existing methods primarily consider pairs of shapes in full correspondence, often struggling with missing parts in one or both shapes. While some techniques address partial matching, they often rely on ad hoc implementation strategies [9, 64] or necessitate prior knowledge about the pair with, for example, the area of the missing part [145, 181, 128]. Introducing novel energies into the pipeline, as demonstrated in Chapter 5 by employing smoothness constraints, holds promise for enhancing results on partial data. Additionally, detecting shared parts in two shapes while computing the correspondences could offer a more general solution than partial shape matching, an aspect rarely explored in the field [126, 13].

More generally, intrinsic methods, which aim to preserve geodesic distances between points before and after matching, encounter difficulties when dealing with highly non-isometric shapes, as presented in Chapter 5. Hybrid approaches that combine intrinsic and extrinsic information have recently shown some promise [70, 71, 92], hinting at a potential avenue for generating novel matching algorithms.

Moreover, while many surface deformation models have been developed [9, 209], exploration of volume deformation methods for surface matching remains limited [64, 69].

Furthermore, modern shape matching methods often leverage deep neural networks to extract features from surfaces [232, 198]. Although these pipelines are rapidly evolving [60, 40, 212], they still struggle with generalizing to new shapes. Addressing the robustness of these methods to novel data presents a crucial area for further research. In this direction, transfer learning in 3D poses unique challenges compared to images, given the inconsistent effectiveness of pretraining networks on 3D data [241, 18, 246]. In contrast, leveraging powerful deep neural networks on images [156, 111] and text [172, 122] to obtain features on shapes could also be a promising direction.

Finally, an emerging direction in shape matching involves exploring alternative surface representations, such as signed distance functions [157], which have gained popularity, particularly in learning applications [166, 63]. This volumetric representation of surfaces enables using methods from volumetric data analysis [58], such as MRI data processing. Integrating surface information and shape analysis tools into these pipelines could potentially significantly improve the results.

Downstream Applications

Despite the efficiency achieved in shape matching, leveraging these outcomes for downstream applications still poses a considerable challenge.

A notable objective of many multi-shape analysis pipelines is to characterize the disparity between two shapes using both a distance metric and visual feedback on the localized areas of difference. In Chapter 3, we derived a local descriptor of deformations based on a global embedding of intrinsic deformations, called shape difference operators [186]. In Chapter 4, we used the same shape difference operators to define a meaningful metric between shapes. However, we had to resort to a simple deformation model to effectively highlight zones of significant difference and conduct statistical analyses.

Developing automated tools to analyze such disparities is imperative. These tools should perform robustly even in scenarios with imprecise correspondences or surfaces exhibiting defects, as demonstrated in Chapter 4.

Moreover, introducing more rigorous benchmarks with highly non-isometric shapes and diverse downstream applications would significantly advance the field.

Scalability

The scalability of shape matching methods has usually not been a primary concern, as advancements in hardware performance have typically sufficed. However, with diminishing improvements in CPU capabilities, there is now a pressing need for faster correspondence processing, given that many of these methods are CPU-dependent [109, 77]. Notably, since shape matching often serves as the initial step in an analysis pipeline, *e.g.* applied to all pairs within a small collection, faster algorithms enable using more complex data processing pipelines.

In Chapter 6, we introduced a method to scale the functional map pipeline to very dense meshes containing hundreds of thousands of vertices. The approximations we presented rely on relatively loose bounds, and we believe that additional or enhanced approximations tailored specifically to shape matching could be developed. Furthermore, obtaining the final dense correspondence remains a significant speed bottleneck, and further methods could be developed in order to improve this last step.

While CPUs are experiencing slower improvements, GPU performance is rapidly increasing. A significant future direction involves porting shape analysis algorithms to GPUs. However, shape matching methods for triangulated surfaces heavily depend on sparse matrices and solving sparse linear systems [159, 77], which do not benefit as much from GPUs as other matrix manipulations. Treating these sparse matrices as dense ones on GPUs is not a viable solution due to their quadratic scaling with the number of vertices. Thus, CPU-based algorithms remain partially prevalent in this domain. In Chapter 7, we successfully ported a popular functional map refinement algorithm to the GPU and utilized the specific structure of functional map computations to avoid storing certain dense matrices in memory. We believe that developing algorithms explicitly designed for GPUs represents a compelling direction of research with substantial impact for practitioners.

However, even when using GPUs, processing 3D data remains slower than processing 2D data. To improve processing time and performance, volumetric algorithms employ multiscale accelerations using voxels, typically using image downscaling and upscaling algorithms [16, 17]. However, deriving such algorithms for shapes, particularly integrating them into learning-based methods, poses significant challenges. While some graph pooling layers have been developed, there remains a lack of efficient surface pooling methods, which could greatly enhance the scalability of surface-based deep neural networks.

Ethical Impact

In light of the increasing utilization of artificial intelligence, it is important to address the ethical implications of our work. We recognize the potential risks associated with the use of 3D data, especially in contexts such as military applications of deepfakes.

We do not endorse such use of our algorithms.

Furthermore, we believe that the primary beneficiaries of our research are practitioners, in particular in the medical and 3D animation fields. This perspective underscores our commitment to ethical considerations and responsible utilization of AI technology.

APPENDIX A

DWKS: A Local Descriptor of Deformations Between Meshes and Point Clouds: Supplementary Material

A.1 Spectral Properties of SDO

A.1.1 Theoretical properties

Notations. Given two discrete shapes \mathcal{M} and \mathcal{N} , we note their respective Laplacian $L^{\mathcal{M}} = (A^{\mathcal{M}})^{-1} W^{\mathcal{M}}$ (resp. $L^{\mathcal{N}} = (A^{\mathcal{N}})^{-1} W^{\mathcal{N}}$) with A being a diagonal matrix filled with per-vertex areas, and W the stiffness matrix (*e.g.* the standard cotangent weight matrix [167]).

The discrete $L^2(\mathcal{S})$ inner product on shape \mathcal{S} is defined as

$$\langle f, g \rangle_{L^2(\mathcal{S})} = f^\top A^{\mathcal{S}} g \quad (\text{A.1})$$

and the discrete $H_0^1(\mathcal{S})$ inner product as

$$\langle f, g \rangle_{H_0^1(\mathcal{S})} = \langle f, L^{\mathcal{S}} g \rangle_{L^2(\mathcal{S})} = f^\top W^{\mathcal{S}} g \quad (\text{A.2})$$

We recall Theorem 3.1:

Theorem 3.1. *Given a non-degenerate functional map F , both the area-based and conformal shape difference operators are positive (semi)-definite, provided that the area and stiffness matrices of the Laplacian are positive (semi)-definite.*

Theorem 3.1 assumes the area-matrices $A^{\mathcal{M}}$ and $A^{\mathcal{N}}$ to be positive definite, which means no vertex has a 0 area. The assumption for a semi definite stiffness matrices however means the kernel of W only consists in the space of constant functions on the shape, which is the 0-set of the H_0^1 norm.

Proposition A.1. *Under the assumptions of Theorem 3.1, the area-based shape difference operator is positive definite.*

Proof. The area-based shape difference operator V is defined implicitly via

$$\langle f, Vg \rangle_{L^2(\mathcal{M})} = \langle Ff, Fg \rangle_{L^2(\mathcal{N})} \quad \forall f, g \in L^2(\mathcal{M}) \quad (\text{A.3})$$

Our discretized version of the L^2 inner product defines a positive-definite form, since we suppose the area matrix A to be non-degenerate. V is therefore self-adjoint operator with respect to the L^2 inner product.

Therefore, given $f \in L^2(\mathcal{M})$,

$$\langle f, Vf \rangle_{L^2(\mathcal{M})} = \|Ff\|_{L^2(\mathcal{N})}^2 \geq 0 \quad (\text{A.4})$$

where the last inequality is an equality if and only if $f = 0$ in $L^2(\mathcal{M})$ since the functional map F is supposed to be non-degenerate.

V is therefore self-adjoint positive-definite, and the spectral theorem states its eigenvalues are real and positive. \square

Proposition A.2. *Under the assumptions of Theorem 1, the conformal-based shape difference operator is positive semi-definite.*

Proof. The conformal-based shape difference operator R is defined implicitly via

$$\langle f, Rg \rangle_{H_0^1(\mathcal{M})} = \langle Ff, Fg \rangle_{H_0^1(\mathcal{N})} \quad \forall f, g \in H_0^1(\mathcal{M}) \quad (\text{A.5})$$

Given non-degenerate matrices A and W , the H_0^1 inner product is positive definite (on H_0^1), and

$$\langle f, Rf \rangle_{H_0^1(\mathcal{M})} = \|Ff\|_{H_0^1(\mathcal{N})}^2 \geq 0 \quad (\text{A.6})$$

for $f \in H_0^1(\mathcal{M})$ with equality if and only if $f = 0$ in $H_0^1(\mathcal{M})$, since F is supposed to be non-degenerate.

R is therefore self-adjoint positive definite on H_0^1 , and the spectral theorem states its eigenvalues are real and positive.

In practice, since the operator R is extended to the entire L^2 space by setting it to 0 for constants, the operator is only positive semidefinite. \square

Proof of Theorem 3.1

Proof. Under the assumption of Theorem 3.1, Propositions A.1 and A.2 apply from which the result follows. \square

A.1.2 Practical Computation

In practice, given two discrete shapes \mathcal{M} and \mathcal{N} with a low dimensional functional map $\mathbf{C} \in \mathbb{R}^{k_N \times k_M}$ between them, the shape difference operators are computed in the spectral basis using the following formulas [186] :

$$\mathbf{V} = \mathbf{C}^\top \mathbf{C} \quad (\text{A.7})$$

$$\mathbf{R} = (\Delta^{\mathcal{M}})^\dagger \mathbf{C}^\top \Delta^{\mathcal{N}} \mathbf{C} \quad (\text{A.8})$$

where $\Delta^{\mathcal{M}}$ and $\Delta^{\mathcal{N}}$ are diagonal matrices of the first $k_{\mathcal{M}}$ (resp. $k_{\mathcal{N}}$) eigenvalues of the Laplace-Beltrami operator on \mathcal{M} (resp. \mathcal{N}).

The area-based shape difference operator \mathbf{V} being symmetric as shown by Equation (A.7), its eigendecomposition is easily computed, while the eigendecomposition of the conformal-based shape difference operator \mathbf{R} is obtained by solving a *generalized* eigenvalue problem

$$(\mathbf{C}^\top \Delta^{\mathcal{N}} \mathbf{C}) \Psi = \Delta^{\mathcal{M}} \Psi \Lambda \quad (\text{A.9})$$

where $\mathbf{C}^\top \Delta^{\mathcal{N}} \mathbf{C}$ and $\Delta^{\mathcal{M}}$ are both symmetric and positive semi-definite.

A.1.3 Algebraic structure

In this section, we provide some insight on the discussion in Section 3.4.1, namely about the claim that the spectrum of shape difference operators is better expressed using the log-scale.

By definition the shape difference operators between isometric shapes are identity operators, and all functions are preserved by these operators. Therefore, the absence of deformations leads to 1 eigenvalues or 0 log-eigenvalues. More generally, if a function f on \mathcal{M} only takes non-zeros values on zones undergoing no deformation (regarding the one between \mathcal{M} and \mathcal{N}), then it will be preserved by the shape difference operator $D_{\mathcal{M},\mathcal{N}}$, which means it becomes an eigenvector with eigenvalue 1.

It is straightforward to see the non-zero eigenvalues of $D_{\mathcal{N},\mathcal{M}}$ are the inverse of those from $D_{\mathcal{M},\mathcal{N}}$, which makes the log-eigenvalues opposite of each other.

The somewhat more complex case lies in the composition of shape difference operators. While for three shapes $\mathcal{M}, \mathcal{P}, \mathcal{N}$ the equality $D_{\mathcal{M},\mathcal{N}} = D_{\mathcal{M},\mathcal{P}} D_{\mathcal{P},\mathcal{N}}$ always holds, there is in general no simple relationship between the eigenvalues of each term. One scenario where the eigenvalues of the composition of the two operators are the product of the individual ones happens when one can find common eigenvectors for the two composed operators. This is known as codiagonalization of matrices and is only possible when the two operators commute.

A.2 Parameters

A.2.1 Optimization Objective

The optimization objective described in Section 3.4.4 consists of 5 different terms. Given a functional maps $\mathbf{C} \in \mathbb{R}^{k_{\mathcal{N}} \times k_{\mathcal{M}}}$, we discuss here the influence of each term in order to bring some intuition on how the hyperparameters can be tuned.

The first term $E_d(\mathbf{C}) = \|\mathbf{CA} - \mathbf{B}\|^2$ simply enforces descriptor preservation with respect to the L^2 norm on the target shape \mathcal{N} , where descriptors are the columns of matrices \mathbf{A} and \mathbf{B} .

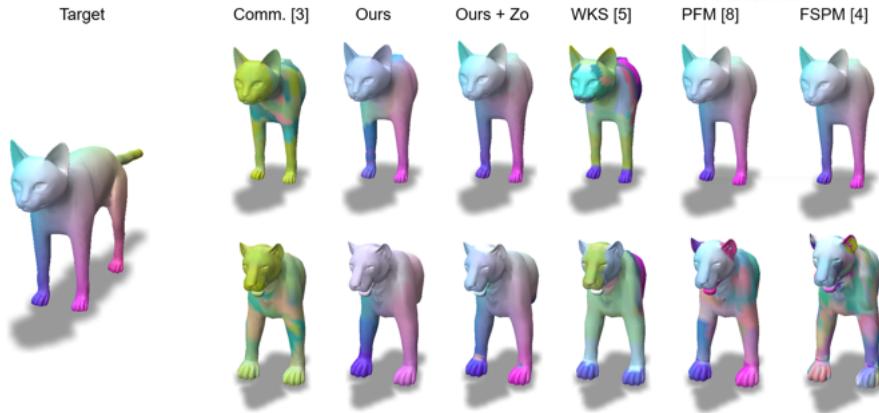


Figure A.1: Results on the cats and lions dataset. Left mesh shows the base shape of the collection of complete cats. Right meshes display the computed pointwise maps in the case of a collection of partial cats (top row) and partial lions (bottom row).

The second term $E_{dc}(\mathbf{C})$ was introduced in [154] in order to improve the descriptor preservation via the introduction of new operators built from individual descriptor functions.

The third term $E_l(\mathbf{C})$ corresponds to commutativity of the functional map with the two Laplace-Beltrami operators, which enforces the functional map to represent a near-isometric map.

The fourth and fifth terms $E_c(\mathbf{C})$ and $E_a(\mathbf{C})$, introduced in [193], simply seek to preserve the action of the respectively conformal and area shape difference operators under the functional map.

A.2.2 Hyperparameters

DWKS. When computing DWKS descriptor, one has to choose a set of energy values (e_1, \dots, e_p) as well as a scale parameter σ . We advocate using moderate energy values to ignore extreme shape difference eigenvalues, often created by noise in the functional correspondences. Using $e_1 = -\log 3$ and $e_p = \log 3$ has led to satisfying results on our side. The σ parameter describes how far an eigenvector will be spread on the energy-scale. To our knowledge, there is no provably efficient procedure to fit this parameter for standard WKS descriptors [15], which is why we settle for a constant parameter across our experiments.

Optimization. Using the previous section, the optimization parameters are tuned so that the descriptors are well-preserved, as well as the action of the shape difference operators. Note that in our experiments using partial shapes, the near-isometric assumption doesn't hold, which justifies the absence of the Laplacian commutativity term in our experiments. Furthermore, because the DFaust collections contains

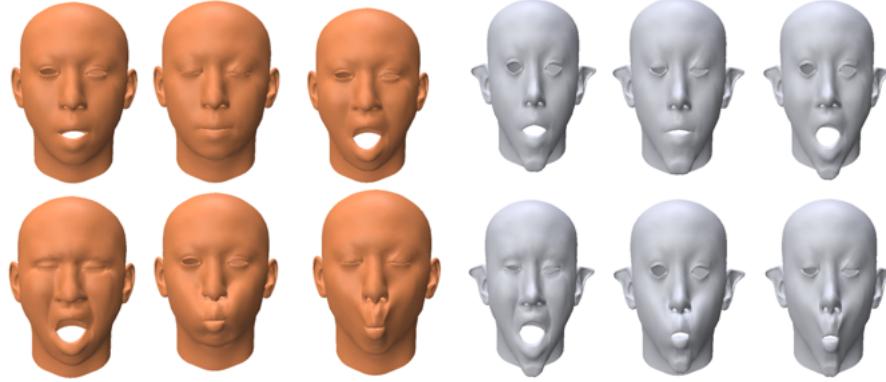


Figure A.2: Visualization of the Synthetic Faces dataset.

many holes and cuts, we found it beneficial to set $\mu_a = 0$ and therefore ignore the area shape difference operators.

Projection to low dimension When projecting the computed pointwise map into a low-dimension functional map, we ignore the fraction α of vertices which are the hardest to match (as defined by the descriptor distance). This is due to the fact shape difference operators and therefore DWKS descriptors can detect some intrinsic distortion near cuts and holes (usually due to noisy intra-collection maps). This explains why this parameter is set to a high value 20% when using partial shapes instead of the 5% for complete shapes.

Refinement. In the case of partial shape, we adapt the ZoomOut algorithm with step size $(1, \lfloor \frac{1}{\lambda} \rfloor)$ with λ the approximate ratio of area between the complete shape and the partial one, which eventually leads to a rectangular functional map.

A.3 Synthetic faces dataset

In Chapter 3, we refer to a synthetic faces dataset [186] shown on Figure A.2, on which we compare our pipeline with the standard baseline [48] as well as the usual functional map pipeline [154] using WKS descriptors.

On this dataset, we show that our pipeline can ignore the area-based shape difference operators and still obtain great results, unlike the pipeline from [48] as seen on Figure A.3.

A.4 Comparison with partial spectral matching

The most direct competitor to our method is [48] extending the pipeline from [193], which leverages on intra-collection maps to compute cross-collection correspondences. We presented extensive comparisons with this approach in Chapter 3. How-

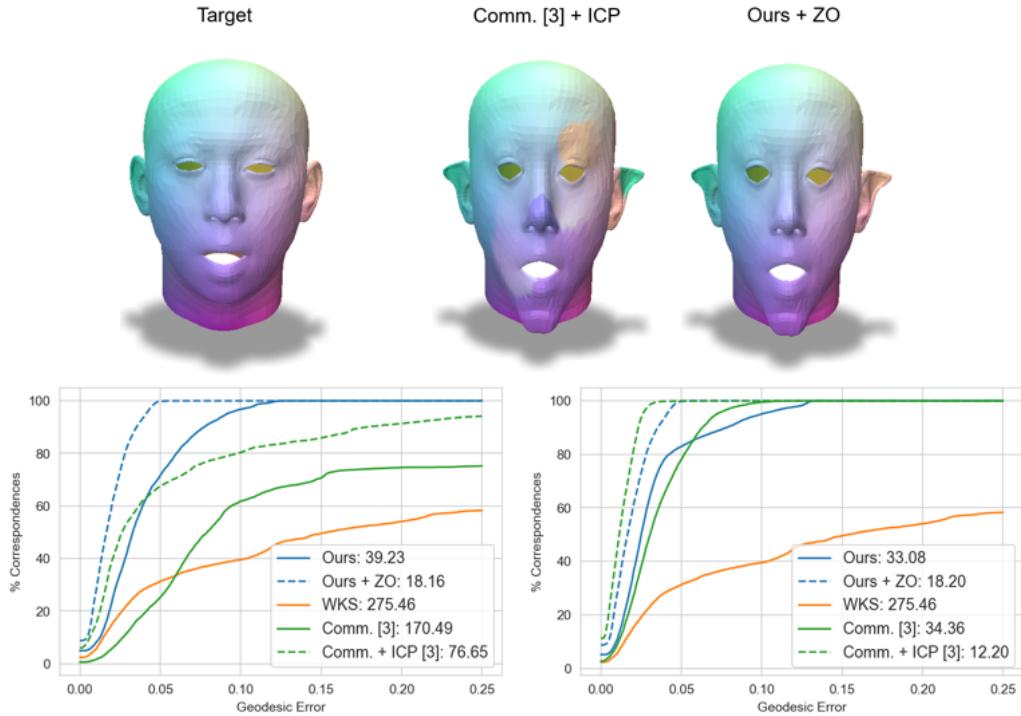


Figure A.3: Results on the Synthetic Faces dataset. Top row shows pointwise maps after refinement when ignoring the area shape difference operators. Bottom row show the accuracy curves for this setting (left) and when using both type shape difference operators (right). Notice the stability of our method.

ever, since these correspondences are only computed between the base shapes of each collection, any matching pipeline could be applied to this problem without using information given by the collections. Namely, the standard functional correspondence computation problem [154] is known to be very efficient to compute correspondences between near-isometric shapes, and specific derivations have been obtained in [181, 128] in the challenging scenario of partial matching.

In the following, we present more quantitative and qualitative results on the two datasets presented in Chapter 3, comparing to additional baselines [154, 181, 128].

Cats and Lions. We first focus on the Sumner dataset, consisting in similar meshes of cats and lions as seen on Figure A.4. Additionally, to the experiment presented in Chapter 3, we apply the same pipeline trying to match a collection of half-cat to the collection of complete cats. Qualitative results are displayed on Figure A.1. Quantitative results associated to these two experiments are displayed on Figure A.5, where the top graphs show results in the case of a partial cat matched to a complete cat, and the bottom one those in the case of a partial lion matched to a complete cat. Note that in the first case, where the partial shape *is exactly isometric* to a subset of the complete shape, partial matching methods [181, 128] obtain excellent results, whereas performance drops very significantly in the second

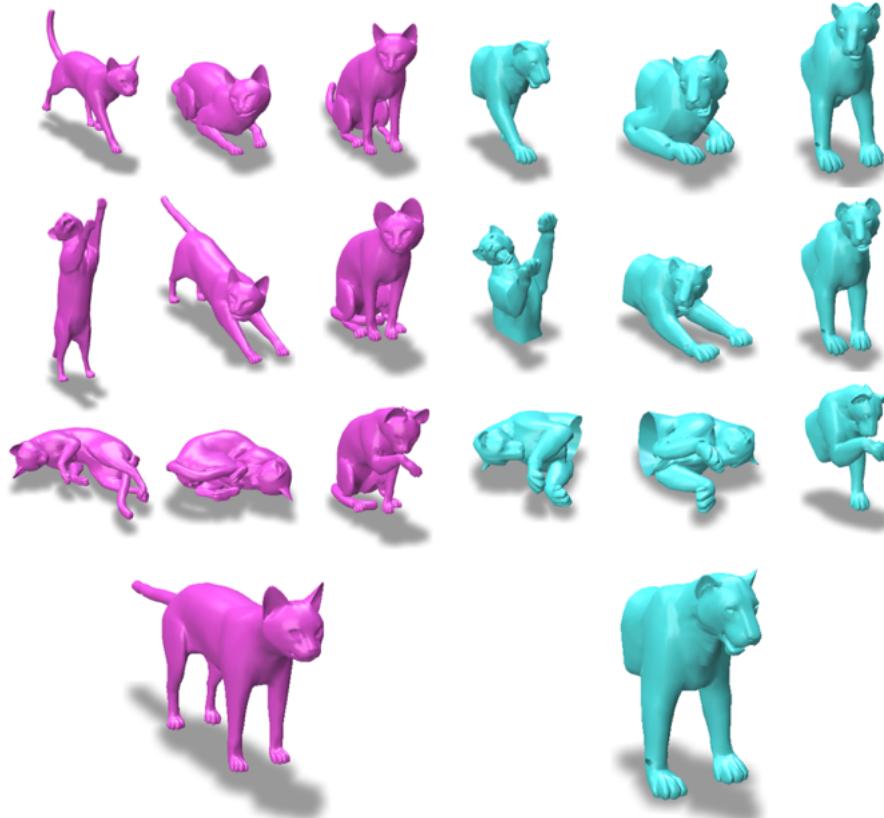


Figure A.4: Visualization of the Cats and Lions dataset. The two bottom shapes are matched together using the deformations shown above them.

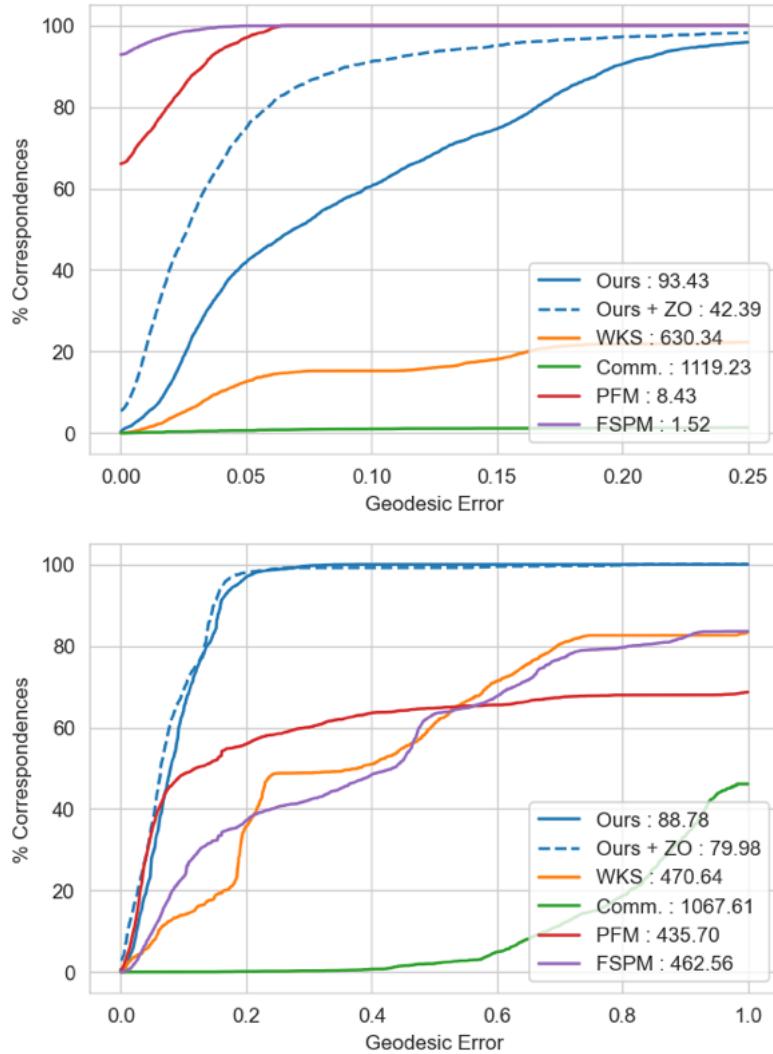
experiment in the case of near-isometry. Furthermore, the standard functional shape matching pipeline [154] fails to disambiguate left and right symmetry in both cases, and results from [48] are unable to obtain meaningful results. Our method however performs quite well in both cases, displaying an efficient use of common deformations to match vertices and disambiguate symmetries. We used the same parameters for both experiments, using the recommended values of [181, 128].

DFaust. We also compared our method to the same baselines on the DFaust [29] dataset. Note that since the dataset consists of point clouds without normals, we replace the SHOT [219] descriptors in [128] by standard WKS descriptors [15], and remove comparison to [181] which requires information about faces.

Our dataset consists in scans of 3 individuals in two different motions, namely jumping on one leg and running on a spot, which results in 6 collections of point clouds. Given a motion, we match the related collection of the first individual, which consists in full scans, with the similar collections of the second and third individual, which consist in scans of the lower half of their bodies as shown on Figure A.6.

Qualitative result using the second individual and the jumping motion are shown on Figure A.7. Accuracy evaluations in all 4 cases are given on Figure A.8, where

the first line displays results when matching respectively the second (left) and third (right) individual to the first one using the jumping motion, and the second line similar results but using the running motion. We remark again on Figure A.7 that the standard functional map method [154] is unable to disambiguate left and right, and that in this case both partial matching [128] and the collection based method [48] can't to produce meaningful correspondences. On the other hand, our method clearly disambiguates the left-right symmetry, and strongly benefits from the ZoomOut refinement steps. This claim is reinforced by the accuracy results given on Figure A.8, which show that our method significantly outperforms existing baselines.



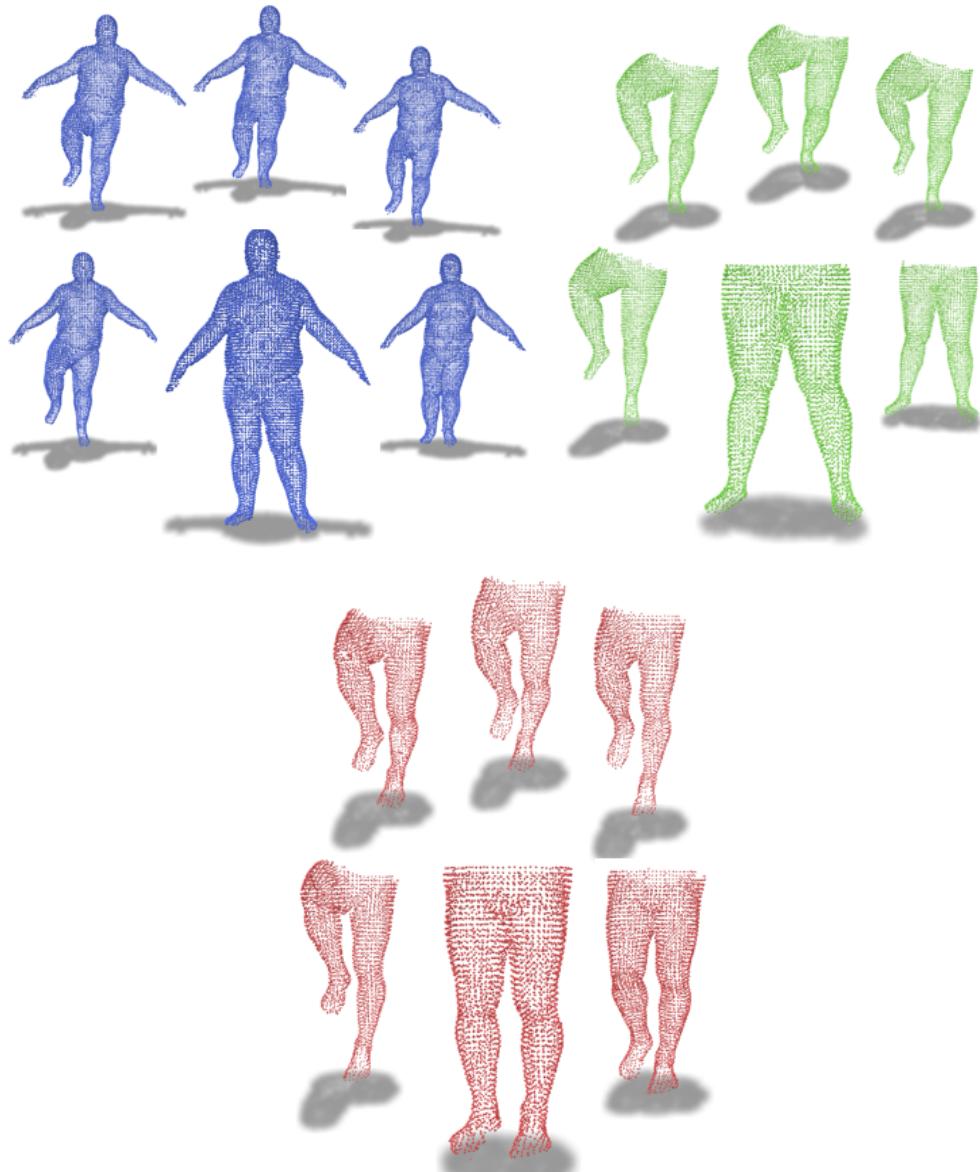


Figure A.6: Visualization of the three collections corresponding to the jumping motion. The biggest shape represents the base shape of the collection and the smaller ones the used deformations.

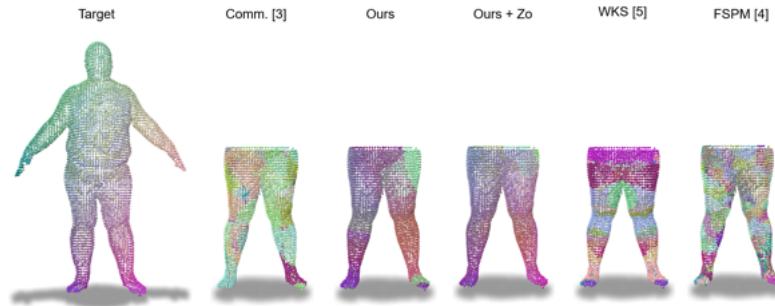


Figure A.7: Examples of results on the DFAust dataset for the jumping motion. Left mesh shows the base shape of the first individual. Right meshes display the computed pointwise maps in the case of a collection of partial cats (top row) and partial lions (bottom row).

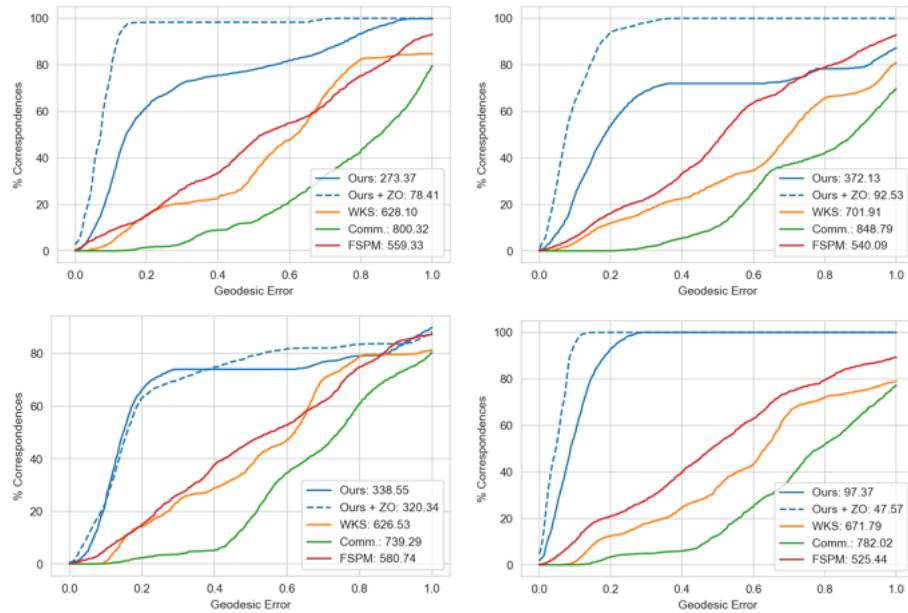


Figure A.8: Results on the DFAust dataset. Each line show results for a given motion, and each column results for a given individual.

APPENDIX B

Assessing Craniofacial Growth and Form Without Landmarks: A New Automatic Approach Based on Spectral Methods

B.1 Age Distribution

As mentioned in Section 4.2.2, the dataset consisted in $N = 155$ CT-scans of skulls including $n_t = 85$ patients with trigonocephaly (mean age 219.3 ± 81.4 days), $n_m = 27$ patients with a metopic ridge (mean age 379.25 ± 224.7 days), and $n_c = 43$ control patients (mean age 218.7 ± 107.8 days). The three distributions (Figure B.1) covered the same age range, excepting some older patients with metopic crest.

B.2 Using age as a feature

As the age distribution were slightly distinguishable, a potential bias of the algorithm lied in using age as a distinguishing factor. The algorithm may have been biased because it used age as a distinguishing factor. To ensure this was not the case, logistic regression was performed on the dataset using only age as a feature. Using 5-fold cross-validation and reweighting data-points to account for the class size difference, The resulting True Positive Rates were 43.8% 3-class problem (**C-M-T**), and a 58% for the binary classification problems ((**C+M**)-T) and (**C-T**).

We concluded from these results that discriminating from age resulted in poor classification accuracy, especially for our binary classification problems ((**C+M**)-T) and (**C-T**) where the TPR are close to random decision.

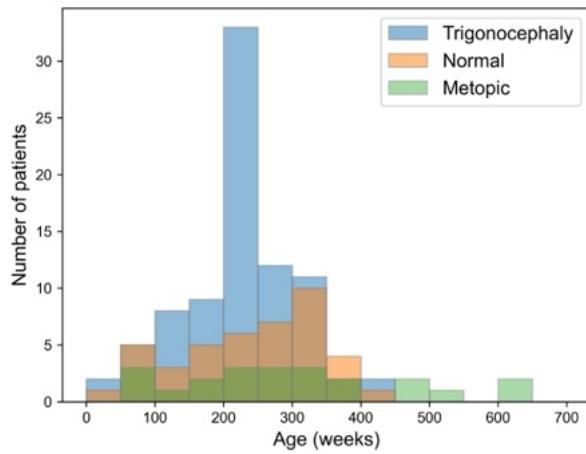


Figure B.1: Age Distribution of the three groups in the dataset.

Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization: Additional Results

C.1 Smoothness Reformulation

In this section, we give details on the reformulation of smoothness methods provided in Chapter 5.

C.1.1 Non-Rigid ICP

Non-Rigid ICP (nICP) [9] deforms a source shape \mathcal{S}_1 into a target shape \mathcal{S}_2 using a per-vertex affine deformation \mathbf{D} . Global energy reads, trying to fit a pointwise map Π_{12}

$$E_{\text{nicp}}(\Pi_{12}, \mathbf{D}) = \left\| \mathbf{D} \right\|_{W_1}^2 + \beta \left\| \mathbf{D} \circ X_1 - \Pi_{12} X_2 \right\|_{A_1}^2 \quad (\text{C.1})$$

with $\left\| \mathbf{D} \right\|_{W_1}^2 = \sum_{i \sim j} w_{ij} \left\| D_i - D_j \right\|_F^2$ and $\mathbf{D} \circ X_1$ the deformed vertex coordinates. Given a point-wise map Π_{12} , one can directly incorporate this energy in our algorithm, where solving for Y_{12} is replaced by solving for \mathbf{D} , and then setting $Y_{12} = \mathbf{D} \circ X_1$. Solving for \mathbf{D} reduces to a simple linear system, as explained in [9]. Note that in the original work, nICP algorithm uses graph Laplacian instead of cotangent Laplacian, but we find that using cotangent weights is more stable in the case of triangle meshes. We furthermore ignored landmarks preservation terms, borders skipping heuristic, normals preservation and self-intersection verification procedures for simplicity.

C.1.2 As-Rigid-As-Possible

As-Rigid-As-Possible (ARAP) [209] promotes *local rigidity* of the deformation of a shape \mathcal{S}_1 using per-vertex rotations \mathbf{R} , which results in minimizing the following energy:

$$E_{\text{arap}}(\mathbf{R}, Y) = \sum_{i \sim j} w_{ij} \left\| (y_i - y_j) - R_i(x_i - x_j) \right\|_F^2 \quad (\text{C.2})$$

where y_i are the expected vertex coordinates and x_i the undeformed coordinates.

We observe that the ARAP energy can be decomposed into two main components, including a smoothness term and a rigidity term:

$$E_{\text{arap}}(\mathbf{R}, Y) = E_{\text{arap}}^{\text{smooth}}(Y) - 2E_{\text{arap}}^{\text{rigid}}(\mathbf{R}, Y) + \text{const.} \quad (\text{C.3})$$

$$E_{\text{arap}}^{\text{smooth}}(Y) = \sum_{(x_i, x_j) \in \mathcal{E}(\mathcal{S}_1)} w_{ij} \|y_i - y_j\|_F^2 \quad (\text{C.4})$$

$$E_{\text{arap}}^{\text{rigid}}(\mathbf{R}, Y) = \sum_{(x_i, x_j) \in \mathcal{E}(\mathcal{S}_1)} w_{ij} (y_i - y_j)^T R_i (x_i - x_j) \quad (\text{C.5})$$

with $E_{\text{arap}}^{\text{smooth}} = \|Y\|_{W_1}^2 = E_D(Y)$. Note, however, that the default ARAP energy does not have a coupling term to ensure that Y remains on the surface of \mathcal{S}_2 . Therefore, to avoid a trivial solution, such an energy must rely on pre-existing landmarks to make sure that the deformation maps onto the target shape. In our algorithm, given a pointwise map Π_{12} , we instead decide to add a coupling term between the expected coordinates Y_{12} and transferred coordinates $\Pi_{12}X_1$, which slightly modifies the linear system to solve when minimizing over Y_{12} , but doesn't involve the rotations \mathbf{R} . Therefore, given a pointwise map Π_{12} , one first needs to compute local rotations \mathbf{R} and can then obtain the expected coordinates Y_{12} by solving a linear system.

C.1.3 Smooth Shells

Smooth Shells [70] models the deformation \mathbf{D} as a simple per-vertex translation seen as a function $\mathcal{S}_1 \rightarrow \mathbb{R}^3$, which is restricted to lie in the *spectral* basis of size K , i.e., $\mathbf{D} \in \mathbb{R}^{K \times 3}$. In addition, smooth shells uses the ARAP energy to enforce the smoothness of the deformation, which therefore adds additional local rotation \mathbf{R} . Specifically, $X_1 + \Phi_1 \mathbf{D}$ would give the updated vertex positions and the smoothness is then defined as:

$$E_{\text{shells}}^{\text{smooth}}(\mathbf{D}, \mathbf{R}) = E_{\text{arap}}(\mathbf{R}, X_1 + \Phi_1 \mathbf{D}) \quad (\text{C.6})$$

The smoothness energy is again associated with a coupling term which ensures the deformed shape remains close to the current correspondences $\|X_1 + \Phi_1 \mathbf{D} - \Pi_{12}X_2\|_{A_1}^2$. Note that in the original work, vertices X_1 and X_2 are also projected to a spectral basis, and extra feature and normal preservation terms are added. In practice, solving for \mathbf{D} reduces to solving a $K \times K$ linear system, compared to the $n \times n$ linear system obtained with standard ARAP.

C.1.4 Reversible Harmonic Maps

Reversible Harmonic Maps (RHM) [77] directly minimizes the Dirichlet energy of a map without manipulating deformation fields. To avoid making the map collapse, the authors look for bijective maps with the lowest possible Dirichlet energy. Vertices of the pull-back shape $\Pi_{ij}X_j$ for $(i, j) \in \{(1, 2), (2, 1)\}$ are again estimated via an auxiliary variable Y_{ij} and the energy reads as the sum in both directions of $E_{\text{rhm}}^{\text{half}}$ with :

$$\begin{aligned} E_{\text{rhm}}^{\text{half}}(\Pi_{ij}, \Pi_{ji}, Y_{ij}, Y_{ji}) &= E_D(Y_{ij}) + E_{\text{rhm}}^{\text{bij}}(\Pi_{ji}, Y_{ij}) \\ &\quad + E_{\text{rhm}}^{\text{couple}}(\Pi_{ij}, Y_{ij}). \end{aligned} \quad (\text{C.7})$$

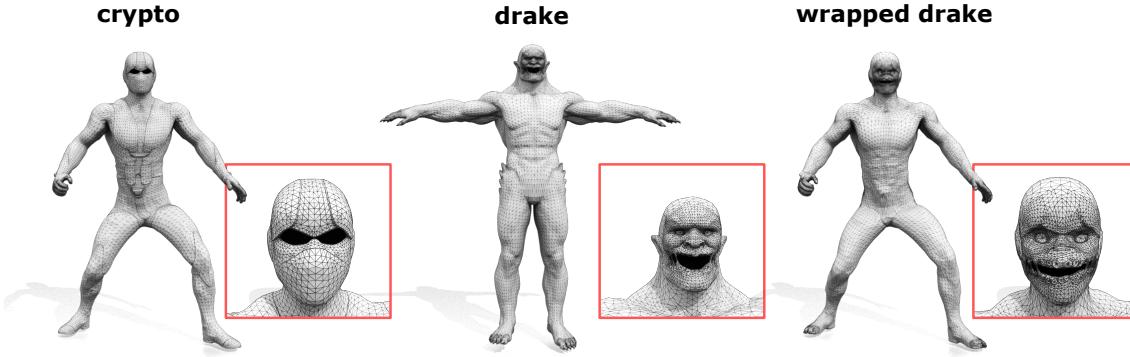


Figure C.1: Example of wrapping a DRAKE shape to a CRYPTO shape to establish cross-category correspondences.

Here, again, we recognize the Dirichlet energy of the estimated map $E_D(Y_{ij})$, and two terms $E_{\text{rhm}}^{\text{bij}}$ and $E_{\text{rhm}}^{\text{couple}}$ which respectively enforce bijectivity and coupling:

$$E_{\text{rhm}}^{\text{bij}}(\Pi_{ji}, Y_{ij}) = \left\| \Pi_{ji} Y_{ij} - X_j \right\|_{A_j}^2 \quad (\text{C.8})$$

$$E_{\text{rhm}}^{\text{couple}}(\Pi_{ij}, Y_{ij}) = \left\| Y_{ij} - \Pi_{ij} X_j \right\|_{A_i} \quad (\text{C.9})$$

This formulation leads to a computationally expensive iterative solver, which can obtain great results given an already good initialization. Additionally, the authors use a high-dimensional embedding obtained via MDS [52] which mimics the geodesic distance, instead of directly using the embedding coordinates.

C.2 DeformThings4D-Matching Dataset

Here we discuss in details how we construct our dataset from the DEFORMTHINGS4D [124] for shape matching task:

- Select Models.** We first pick models in DEFORMTHINGS4D that are close to watertight. Specifically, we only keep the models where the number of vertices in the largest connected components is more than 75% of the total number of vertices. Then the largest connected component is taken if the model is disconnected. As a result, we get 56 animal models and 8 humanoid models.
- Select Poses.** For each watertight model, we collect all motion clips in DEFORMTHINGS4D and select poses from all the frames that are sufficiently different from each other. Specifically, we first pick a base pose that is close to an A-pose: we find the pose that has wide range in z -axis and has relatively small range in xy -axis. We then recursively find new pose from the collection that have the largest difference in vertex positions to the chosen ones, until we get 50 poses or all the poses are included. We then manually check each chose pose and remove unrealistic poses with large distortion or self-intersection. As a result, the number of poses for each model has a range from 30 to 50.

3. **Remeshing.** The chosen poses in each model are in the same triangulation, which can lead to overfitting issues for some shape matching methods [174]. We therefore apply a geometry-aware remeshing algorithm, LRVD [243], to independently remesh all the poses to the resolution of around 8K vertices. The correspondences between the remeshed shapes are propagated by nearest-neighbor searching between the remeshed shapes and the original shapes. To fix the potential topological errors in the nearest neighbor map, we apply spectral ICP [159] at dimension 500 of the Laplace-Beltrami Basis.
4. **Wrapping.** We also provide cross-category correspondences for the 8 humanoid models. Specifically, we use the commercial software R3DS to wrap the rest 7 models (ZLORP, MANNEQUIN, DRAKE, NINJA, PRISONER, PUMPKINHULK, SKELETONZOMBIE) to the chosen model (CRYPTO, the left-most shapes in Fig. 5.2 in Chapter 5). For each pair, we manually select 50-80 landmarks on shapes for wrapping. Note here we wrap the original models and propagate the correspondences to the remeshed shapes afterward. Specifically, the cross-category correspondences among the original poses can be established by nearest-neighbor searching between the wrapped shape and the target shape (see Fig. C.1 for an example of a wrapped shape), which are then propagated to the remeshed poses similar to step 3. Note that, since some shapes are far from isometry or even incomplete, the wrapped results are not perfect, and hence the established correspondences via map compositions can be inaccurate. In general, as illustrated in Figure 5.2, the established correspondences are in reasonable accuracy.

C.3 FAUST dataset

The FAUST dataset [28] consists of 100 meshes of 10 individuals in 10 different poses.

This dataset is used as a standard benchmark for most shape-matching algorithms. However, as all shapes are near-isometric, many methods achieve smooth and accurate results for this dataset. This therefore gives very little room for improvement regarding the smoothness.

We provide results on a random subset of 200 pairs in Section 5.6, where pairs we selected so that only cross-individual ones are considered.

C.4 Additional Results

We evaluate different methods using accuracy, bijectivity, coverage, and smoothness of the maps as metrics. We also report runtime to compare the efficiency. Specifically, we compute the geodesic distances between the obtained maps T_{ij} and the ground-truth maps (if available) to measure the *accuracy* (see Table C.1). Similarly, we compute the geodesic distances between the composite maps $T_{ij} \circ T_{ji}$ and the identity map I_{n_i} to measure the *bijectivity* of the pointwise maps (see Table C.2).

Table C.1: Accuracy on DEFORMTHINGS4D-MATCHING

| methods | <i>near-isometric</i> | | | <i>partial</i> | <i>non-iso</i> | |
|----------------|-----------------------|-------|-----------|----------------|----------------|--------|
| | ZLORP | DRAKE | MANNEQUIN | NINJA | PRISONER | ZOMBIE |
| Init | 11.49 | 9.59 | 8.62 | 10.43 | 20.78 | 15.33 |
| Ours w/ ARAP | 11.22 | 9.04 | 8.10 | 9.88 | 19.91 | 14.83 |
| Ours w/ nICP | 7.29 | 7.07 | 4.61 | 5.25 | 21.18 | 11.95 |
| Ours w/ Shells | 3.25 | 7.78 | 4.11 | 4.73 | 20.27 | 10.32 |
| ZO | 3.43 | 5.74 | 3.33 | 4.61 | 20.59 | 13.71 |
| DO | 3.26 | 5.95 | 3.64 | 5.10 | 19.59 | 16.53 |
| Ours w/ D | 3.72 | 6.93 | 4.18 | 4.80 | 19.81 | 9.71 |
| Ours w/ RHM | 3.70 | 5.63 | 3.94 | 5.46 | 18.85 | 11.00 |

Table C.2: Bijectivity on DEFORMTHINGS4D-MATCHING

| methods | <i>near-isometric</i> | | | <i>partial</i> | <i>non-iso</i> | |
|----------------|-----------------------|-------|-----------|----------------|----------------|--------|
| | ZLORP | DRAKE | MANNEQUIN | NINJA | PRISONER | ZOMBIE |
| Init | 11.69 | 7.17 | 6.58 | 10.69 | 22.53 | 11.52 |
| Ours w/ ARAP | 11.93 | 7.25 | 7.69 | 10.42 | 21.71 | 11.18 |
| Ours w/ nICP | 3.63 | 2.73 | 2.58 | 2.49 | 7.17 | 4.71 |
| Ours w/ Shells | 1.67 | 2.16 | 2.22 | 2.23 | 3.56 | 3.71 |
| ZO | 2.14 | 4.05 | 1.37 | 3.99 | 21.19 | 10.11 |
| DO | 1.27 | 1.55 | 1.63 | 1.46 | 2.26 | 2.52 |
| Ours w/ D | 1.77 | 2.12 | 2.30 | 2.25 | 3.60 | 3.74 |
| Ours w/ RHM | 1.42 | 1.84 | 1.82 | 1.94 | 2.81 | 3.24 |

We compute the Dirichlet energy on the obtained pointwise maps to evaluate the smoothness (defined in Eq. (5.3)) as shown in Table 5.1. Here we additionally evaluate the conformal distortion [77, 175], another popular smoothness metric, as shown in Table C.4. We finally compute coverage of a pointwise map T , i.e., the area ratio of the target shape that is covered by the map T , which evaluates the map surjectivity (see Table C.3). This metric must be considered in pair with smoothness to detect the degenerate case of trivial maps with perfect smoothness. For example, a trivial map where all vertices on the source are mapped to the same vertex on the target, is perfectly smooth w.r.t. the Dirichlet energy, but its coverage is close to zero. Therefore, in the ideal case, the best map is the one with zero Dirichlet energy and 100% coverage. All metrics are reported as an average over all the tested shape pairs.

In Figure C.2, we show some qualitative results on the TOSCA non-isometric dataset.

Table C.3: Coverage on DEFORMTHINGS4D-MATCHING

| methods | <i>near-isometric</i> | | | <i>partial</i> | <i>non-iso</i> | |
|----------------|-----------------------|-------|-----------|----------------|----------------|--------|
| | ZLORP | DRAKE | MANNEQUIN | NINJA | PRISONER | ZOMBIE |
| Init | 22% | 34% | 35% | 28% | 8% | 20% |
| Ours w/ ARAP | 28% | 37% | 36% | 34% | 22% | 29% |
| Ours w/ nICP | 38% | 50% | 53% | 53% | 20% | 30% |
| Ours w/ Shells | 61% | 55% | 56% | 57% | 39% | 43% |
| ZO | 72% | 70% | 71% | 71% | 59% | 60% |
| DO | 68% | 66% | 65% | 66% | 55% | 55% |
| Ours w/ D | 59% | 55% | 54% | 56% | 37% | 41% |
| Ours w/ RHM | 64% | 60% | 60% | 60% | 45% | 47% |

Table C.4: Smoothness on DEFORMTHINGS4D-MATCHING, via Conformal Distortion

| methods | <i>near-isometric</i> | | | <i>partial</i> | <i>non-iso</i> | |
|----------------|-----------------------|-------|-----------|----------------|----------------|--------|
| | ZLORP | DRAKE | MANNEQUIN | NINJA | PRISONER | ZOMBIE |
| Ours w/ ARAP | 2.33 | 2.99 | 2.21 | 2.24 | 3.02 | 2.10 |
| Ours w/ nICP | 4.14 | 5.15 | 2.59 | 2.90 | 10.49 | 4.58 |
| Ours w/ Shells | 3.22 | 4.68 | 3.77 | 4.56 | 14.04 | 7.13 |
| ZO | 3.05 | 5.03 | 2.51 | 4.22 | 24.80 | 15.76 |
| DO | 3.23 | 5.30 | 3.77 | 4.69 | 21.10 | 16.09 |
| Ours w/ D | 2.85 | 3.70 | 2.81 | 3.05 | 9.89 | 4.54 |
| Ours w/ RHM | 2.88 | 3.92 | 2.78 | 3.07 | 10.05 | 4.72 |

C.5 Parameters

In all experiments, we use the same set of parameters, where those of each smoothness energy were tuned independently. Parameters can also be found in the released implementation at <https://github.com/RobinMagnet/smoothFM>.

Spectral Energy. For all experiments, we weighted the spectral bijectivity term by 1 and the coupling term by 10^{-1} , as advocated in the Discrete Optimization implementation [177].

Smoothness Energy. Each smoothness energy required its own set of parameters. The Dirichlet energy was weighted by 1 for all of them for consistency. In particular, for RHM energy, we used a coupling weight of 1 and a bijectivity weight of 10^4 . We used a coupling weight of 10^{-1} for ARAP, 10^{-2} for nICP and 10^{-3} for Shells.

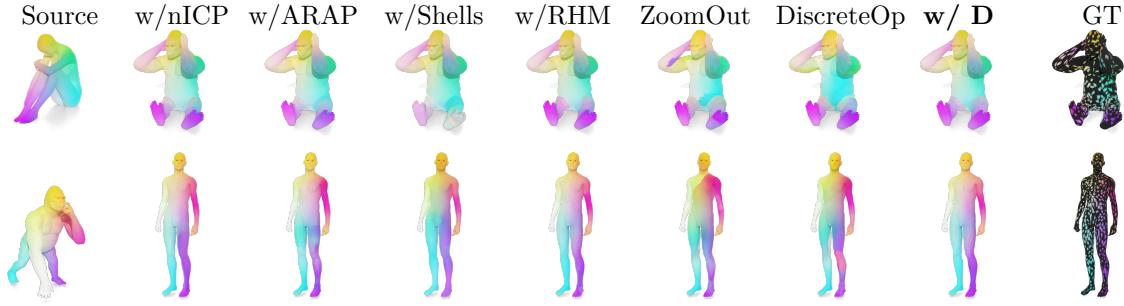


Figure C.2: We show two non-isometric shape pairs from TOSCA dataset can compare pointwise maps obtained from different methods via color transfer. Note that TOSCA non-isometric dataset only provide *sparse* ground-truth correspondences. We therefore color the vertices that do not have GT correspondences in *black*.

Table C.5: Results on TOSCA nonIsometric using WKS initialization

| methods | <i>accuracy</i> | <i>bijection</i> | <i>smoothness</i> | <i>coverage</i> |
|-------------|-----------------|------------------|-------------------|-----------------|
| Init | 56.56 | 39.50 | 93.24 | 15.48 % |
| Zo | 54.61 | 43.23 | 19.27 | 52.48 % |
| DO | 53.65 | 2.33 | 16.47 | 50.04 % |
| Ours w/ D | 51.38 | 22.30 | 2.46 | 16.72 % |
| Ours w/ RHM | 54.07 | 4.18 | 3.92 | 35.29 % |

Coupling. We globally reweighted the smoothness energy by a parameter γ , gradually increasing from 10^{-1} to 1 across iterations.

C.6 Initialization

For all datasets, we obtain initial dense correspondences by computing a 5×5 functional map using 5 landmarks.

We chose this kind of initialization, as standard shape descriptors like WKS [15] could not provide meaningful correspondences in the presence of high levels of non-isometry.

Indeed, Table C.5 provides results using WKS descriptor as initialization for all methods. Note that the accuracy is unable to significantly go down from initialization. It thus becomes difficult to read into these results in a meaningful manner.

C.7 Discrete Optimization

The discrete optimization framework [177] proposes a large set of spectral energies, along which the conformal energy promoting functional maps associated to conformal pointwise correspondences. While this energy does help smoothness, we did not

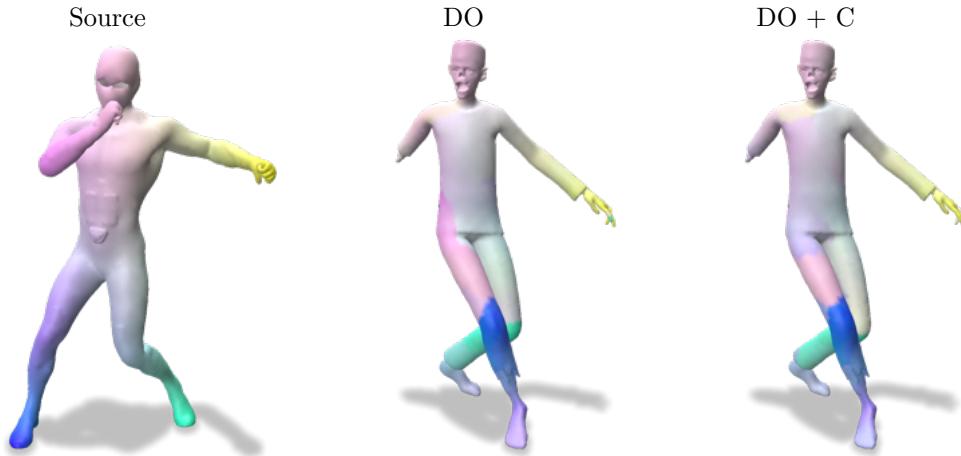


Figure C.3: Example of correspondences without (enter) and with (right) the conformal term of Discrete Optimization. While some parts are smoother, the overall effect is marginal.

notice significant improvements regarding discontinuities in the correspondences.

On Figure C.3, we display an example of correspondences obtained by the standard Discrete Optimization (center) and by adding the conformal term (right). While some parts have been made smoother, the effect remains quite marginal.

In practice, this term can provide meaningful regularization in some cases but appears quite hard to tune to obtain a consistent effect.

APPENDIX D

Scalable and Efficient Functional Map Computations on Dense Meshes: Additional Results

D.1 Function χ

The function $\chi : \mathbb{R}_+ \rightarrow [0, 1]$, differentiable with $\chi(0) = 1$ and $\chi(x) = 0$ for $x \leq 1$ can be defined two ways. Following [149], we use the polynomial interpolation function $\chi : x \mapsto 1 - 3x^2 + 2x^3$. Another possibility is to use the standard C^∞ compactly supported function $\chi : x \mapsto \exp\left(1 - \frac{1}{1-x^2}\right)$, which we found not as good as the polynomial interpolation regarding results. Both functions are displayed on Figure D.1.

D.2 Coefficient weighting

Table D.1 compares our algorithm (Ours) with a similar one (Ours + reweight), where we replace $\bar{\Phi}^M$ by $\bar{\Psi}^M$ in Equation (6.7) so that the map $\bar{\Pi}$ actually transports pointwise values rather than coefficient, as mentioned in Section 6.5.1. This amounts to reweighting local coefficients to actually become function values. We notice this method does not improve our pipeline on the SHREC19 dataset.

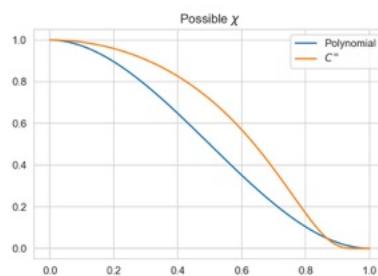


Figure D.1: Possible choices for function χ

Table D.1: Evaluation of the reweighting scheme on the SHREC19 dataset.

| methods | <i>Accuracy</i> | <i>Coverage</i> | <i>Smoothness</i> |
|--------------------|-----------------|-----------------|-------------------|
| Init | 60.18 | 26.5 % | 9.5 |
| Ours + Reweighting | 28.1 | 54.6 % | 6.3 |
| Ours | 27.78 | 56.7 % | 5.6 |

D.3 Proof of Proposition 6.1

Proposition 6.1. Let $\bar{\Psi}^{\mathcal{N}}$ (resp. $\bar{\Psi}^{\mathcal{M}}$) and $\Psi^{\mathcal{N}}$ (resp. $\Psi^{\mathcal{M}}$) the approximated and true first K eigenvectors of the Laplacian on \mathcal{N} (resp. \mathcal{M}). Let \mathbf{C} and $\bar{\mathbf{C}}$ be the original and reduced (see Eq. (6.6)) functional maps of size K , associated to the map T . Suppose that T is a diffeomorphism, and let B_T be the bound given by Lemma 6.1. If there exists $\varepsilon \in \mathbb{R}_+^*$ so that for any $j \in \{1, \dots, K\}$:

$$\|\Psi_j^{\mathcal{N}} - \bar{\Psi}_j^{\mathcal{N}}\|_{\infty} \leq \varepsilon \text{ and } \|\Psi_j^{\mathcal{M}} - \bar{\Psi}_j^{\mathcal{M}}\|_{\infty} \leq \varepsilon$$

Then:

$$\frac{1}{K} \|\mathbf{C} - \bar{\mathbf{C}}\|_2^2 \leq \varepsilon^2 (1 + B_T^2) \quad (6.10)$$

Proof. We first note that the entries of the functional maps \mathbf{C} and $\bar{\mathbf{C}}$ can be written

$$\mathbf{C}_{i,j} = \langle \psi_j^{\mathcal{N}}, \psi_i^{\mathcal{M}} \circ T \rangle_{\mathcal{N}} \quad (D.1)$$

$$\bar{\mathbf{C}}_{i,j} = \langle \bar{\psi}_j^{\mathcal{N}}, \bar{\psi}_i^{\mathcal{M}} \circ T \rangle_{\mathcal{N}} \quad (D.2)$$

Furthermore, given f_1, g_1, f_2, g_2 functions on \mathcal{N} .

Then for any $x \in \mathcal{N}$

$$\begin{aligned} f_1(x)g_1(x) - f_2(x)g_2(x) &= f_1(x)(g_1(x) - g_2(x)) \\ &\quad + g_2(x)(f_1(x) - f_2(x)) \end{aligned} \quad (D.3)$$

With $f_1 = \psi_i^{\mathcal{N}}$, $g_1 = \psi_j^{\mathcal{M}} \circ T$, $f_2 = \bar{\psi}_j^{\mathcal{N}}$ and $g_2 = \bar{\psi}_i^{\mathcal{M}} \circ T$, we have by hypothesis

$$\begin{aligned} \|f_2 - f_1\|_{\infty} &\leq \varepsilon \\ \|g_2 - g_1\|_{\infty} &\leq \varepsilon \\ \|g_2\|_{\mathcal{N}} &\leq B_T \end{aligned} \quad (D.4)$$

Therefore

$$\begin{aligned}
|\langle f_1, g_1 \rangle_{\mathcal{N}} - \langle f_2, g_2 \rangle_{\mathcal{N}}|^2 &= \left| \int_{\mathcal{N}} (f_1(x)g_1(x) - f_2(x)g_2(x)) d\mu^{\mathcal{N}}(x) \right|^2 \\
&\leq \int_{\mathcal{N}} f_1(x)^2 (g_1(x) - g_2(x))^2 d\mu^{\mathcal{N}}(x) \\
&\quad + \int_{\mathcal{N}} g_2(x)^2 (f_1(x) - f_2(x))^2 d\mu^{\mathcal{N}}(x) \\
&\leq \varepsilon \int_{\mathcal{N}} f_1(x)^2 d\mu^{\mathcal{N}}(x) + \varepsilon \int_{\mathcal{N}} g_2(x)^2 d\mu^{\mathcal{N}}(x) \\
&\leq \varepsilon (\|f_1\|_{\mathcal{N}}^2 + \|g_2\|_{\mathcal{N}}^2) \\
&\leq \varepsilon^2 (1 + B_T^2)
\end{aligned}$$

Summing for all elements of the matrix \mathbf{C} gives the result. \square

D.4 Proof of Proposition 6.2

Proposition 6.2. *Let $T : \mathcal{N} \rightarrow \mathcal{M}$ be a pointwise map between the shapes represented by $\boldsymbol{\Pi}$, and let B_T be the bound given by Lemma 6.1. Suppose that $T|_{\mathcal{S}^{\mathcal{N}}} : \mathcal{S}^{\mathcal{N}} \rightarrow \mathcal{S}^{\mathcal{M}}$ is represented by $\overline{\boldsymbol{\Pi}}$.*

Let $\alpha = \min_j u_j^{\mathcal{M}}(v_j) \in [0, 1]$. Suppose further that there exists $\varepsilon > 0$ so that for any $k \in \{1, \dots, K\}$ and $x, y \in \mathcal{S}^{\mathcal{M}}$:

$$d^{\mathcal{M}}(x, y) \leq \rho^{\mathcal{M}} \Rightarrow |\overline{\Psi}_k^{\mathcal{M}}(x) - \overline{\Psi}_k^{\mathcal{M}}(y)| \leq \varepsilon \quad (6.11)$$

and

$$d^{\mathcal{M}}(x, y) \leq \rho^{\mathcal{M}} \Rightarrow |\overline{\Phi}_k^{\mathcal{M}}(x) - \overline{\Phi}_k^{\mathcal{M}}(y)| \leq \varepsilon. \quad (6.12)$$

Then

$$\frac{1}{K} \left\| \boldsymbol{\Pi} \overline{\boldsymbol{\Psi}}^{\mathcal{M}} - \mathbf{U}^{\mathcal{N}} \overline{\boldsymbol{\Pi}} \overline{\boldsymbol{\Phi}}^{\mathcal{M}} \right\|_{\mathcal{N}}^2 \leq \varepsilon^2(1 - \alpha) + \varepsilon^2 B_T^2 \quad (6.13)$$

The proof relies on the following proposition,

Proposition D.1. *Given, \mathcal{M} a surface, $(v_j)_j$ and $(u_j)_j$ built as described in Section 6.4.3.*

Given $f : \mathcal{M} \rightarrow \mathbb{R}$, suppose there exists $\varepsilon > 0$ so that for any $x, y \in \mathcal{M}$, $d(x, y) \leq \rho \implies |f(x) - f(y)| \leq \varepsilon$.

Then the interpolation error between f and $\tilde{f} = \sum_j f(v_j)u_j$ is bounded by ε :

$$|\tilde{f}(x) - f(x)| \leq \varepsilon \quad \forall x \in \mathcal{M} \quad (D.5)$$

And for any $j \in \{1, \dots, p\}$

$$|\tilde{f}(v_j) - f(v_j)| \leq \varepsilon (1 - u_j(v_j)) \quad (D.6)$$

Proof. of Proposition D.1

Let $f : \mathcal{M} \rightarrow \mathbb{R}$, $\mathcal{S} = \{v_1, \dots, v_p\}$ a sample of \mathcal{M} associated to a radius ρ . Since (u_j) verify $\sum_j u_j = \mathbb{1}$, for any $x \in \mathcal{M}$ we have

$$f(x) = \sum_{j=1}^p u_j(x) f(x) \quad (\text{D.7})$$

Therefore if $\tilde{f} = \sum_j f(v_j) u_j$

$$\begin{aligned} f(x) - \tilde{f}(x) &= \sum_{j=1}^p u_j(x)(f(x) - f(v_j)) \\ &= \sum_{j, d(v_j, x) < \rho} u_j(x)(f(x) - f(v_j)) \end{aligned} \quad (\text{D.8})$$

This gives, using triangular inequality and $u_j(x)^2 \leq u_j(x)$ (since $0 \leq u_j(x) \leq 1$):

$$\begin{aligned} |f(x) - \tilde{f}(x)|^2 &\leq \sum_{j, d(v_j, x) < \rho} u_j(x)^2 |f(x) - f(v_j)|^2 \\ &\leq \sum_{j, d(v_j, x) < \rho} u_j(x) |f(x) - f(v_j)|^2 \end{aligned} \quad (\text{D.9})$$

Which gives $|f(x) - \tilde{f}(x)|^2 \leq \epsilon$ using the hypothesis of the proposition and the fact $\sum_j u_j = \mathbb{1}$.

Furthermore, if there exit k so that $x = v_k$, we can remove the term of index k and we have

$$\begin{aligned} |f(x) - \tilde{f}(x)|^2 &\leq \sum_{j \neq k, d(v_j, x) < \rho} u_j(v_k)^2 |f(v_k) - f(v_j)|^2 \\ &\leq \epsilon^2 \sum_{j \neq k, d(v_j, x) < \rho} u_j(v_k) \\ &\leq \epsilon^2 (1 - u_k(v_k)) \end{aligned} \quad (\text{D.10})$$

□

Proof. of Proposition 6.2

We again suppose all shapes to be area-normalized. Using the \tilde{f} notation from Proposition D.1, we can use the triangular inequality on $\|\Pi \bar{\Psi}^{\mathcal{M}} - \mathbf{U}^{\mathcal{N}} \bar{\Pi} \bar{\Phi}^{\mathcal{M}}\|_{\mathcal{N}}$:

$$\begin{aligned} \|\Pi \bar{\Psi}^{\mathcal{M}} - \mathbf{U}^{\mathcal{N}} \bar{\Pi} \bar{\Phi}^{\mathcal{M}}\|_{\mathcal{N}}^2 &\leq \left\| \Pi \bar{\Psi}^{\mathcal{M}} - \widetilde{\Pi \bar{\Psi}^{\mathcal{M}}} \right\|_{\mathcal{N}}^2 \\ &\quad + \left\| \widetilde{\Pi \bar{\Psi}^{\mathcal{M}}} - \mathbf{U}^{\mathcal{N}} \bar{\Pi} \bar{\Phi}^{\mathcal{M}} \right\|_{\mathcal{N}}^2 \end{aligned} \quad (\text{D.11})$$

The first term can be decomposed as a sum of the norms of its K columns, where each term is in the form $\|\bar{\Psi}_j^{\mathcal{M}} \circ T - \widetilde{\bar{\Psi}_j^{\mathcal{M}}} \circ T\|_{\mathcal{N}}^2$, and can be controlled by applying

the bound on interpolation error from Proposition D.1 associated with the bounded distortion lemma, that is

$$\left\| \boldsymbol{\Pi} \bar{\boldsymbol{\Psi}}^{\mathcal{M}} - \widetilde{\boldsymbol{\Pi} \bar{\boldsymbol{\Psi}}^{\mathcal{M}}} \right\|_{\mathcal{N}}^2 \leq KB_T^2\epsilon^2 \quad (\text{D.12})$$

Focusing on the second term, the following lemma will be very useful in order to bound it :

Lemma D.1. *Given $\beta \in \mathbb{R}^{p^{\mathcal{N}}}$, $\|\mathbf{U}^{\mathcal{N}}\beta\|_{\mathcal{N}}^2 \leq \|\beta\|_F^2$*

We indeed notice the second term can be written in the form $\|\mathbf{U}^{\mathcal{N}}\mathbf{A} - \mathbf{U}^{\mathcal{N}}\mathbf{B}\|_{\mathcal{N}}^2$. Using Lemma D.1, we can now focus on bounding $\|A - B\|_F^2$ and especially on the squared norm of each of the columns of $A - B$. In practice, each column can be written as $(\bar{\psi}_j^{\mathcal{M}} \circ T(v_k) - \bar{\phi}_j^{\mathcal{M}} \circ T|_{\mathcal{S}^{\mathcal{N}}}(v_k))_k$, since we supposed that $T|_{\mathcal{S}^{\mathcal{N}}}$ was well-defined between the subsamples.

Given $k \in \{1, \dots, p^{\mathcal{N}}\}$, there exists $i_0 \in \{1, \dots, p^{\mathcal{M}}\}$ so that $T(v_i^{\mathcal{N}}) = v_{i_0}^{\mathcal{M}}$. Furthermore, by definition of the approximated eigenvectors $\bar{\psi}_j^{\mathcal{M}}$, for all $x \in \mathcal{M}$ we have $\bar{\psi}_j^{\mathcal{M}}(x) = \sum_{k=1}^{p^{\mathcal{M}}} \bar{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}}) u_k^{\mathcal{M}}(x)$. Therefore, denoting $\Delta_j(i) = \bar{\psi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) - \bar{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})$

$$\Delta_j(i) = \sum_{k=1}^{p^{\mathcal{M}}} \bar{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}}) u_k^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) - \bar{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) \quad (\text{D.13})$$

$$= \sum_{k=1}^{p^{\mathcal{M}}} u_k^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) (\bar{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}}) - \bar{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})) \quad (\text{D.14})$$

The exact same procedure as in the proof of Proposition D.1 can now be applied, which allows bounding the term

$$\left\| \widetilde{\boldsymbol{\Pi} \bar{\boldsymbol{\Psi}}^{\mathcal{M}}} - \mathbf{U}^{\mathcal{N}} \boldsymbol{\Pi} \bar{\boldsymbol{\Phi}}^{\mathcal{M}} \right\|_{\mathcal{N}}^2 \leq K\varepsilon(1 - \alpha) \quad (\text{D.15})$$

Summing terms from Equations (D.12) and (D.15) produce the upper bound of Proposition 6.2. □

D.5 Values of theoretical quantities

We here provide values for the named values from Proposition 6.2. We again highlight the proposed bounds are not tight and only serves as guidance to select parameters.

First, note that B_T is a Lipschitz-constant, which is 1 whenever T is an isometry, and is else related to the area-distortion induced by T .

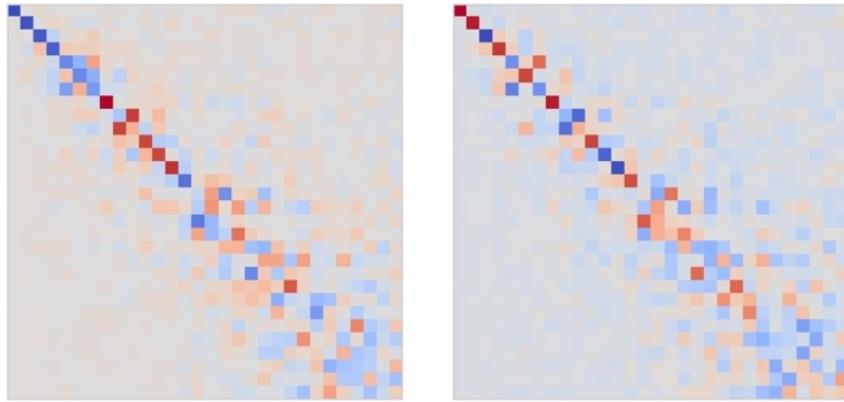


Figure D.2: Ground truth functional map $\bar{\mathbf{C}}$ using the functional space $\bar{\mathcal{F}}$ without (Left) and with (Right) adaptive radius. Notice that up to a change of sign, both functional maps look similar

α can then vary between 0 and 1, but our adaptive radius scheme ensures the minimal value is 0.3. In practice, the average value is higher, around 0.43 on average on the SHREC19 dataset.

Finally, ε controls the variation of the approximated eigenvectors in a local neighborhood, and can be set arbitrarily small by decreasing the value for ρ (and potentially increasing the number of samples to ensure partition of unity). Note that the higher the frequency of the eigenvector, the higher the maximal value of ε is, where the maximum is taken across all local neighborhoods. In practice, we observe maximum values between 0 and 8 for the first 150 eigenvectors, when using around 1500 sampled points. In comparison, we obtain values between 0 and 4 by comparing values of the *exact* eigenvectors simply across edges.

D.6 Functional Map approximation

We here display on Figure D.2 images of the ground truth functional maps for Figure 6.4

D.7 Implementation details

We here provide additional details on parameters and algorithm for implementation.

Per vertex radii are initially set to the same initial value ρ_0 , defined as $\rho_0 = 3\tilde{\rho}_0$ with $\tilde{\rho}_0 = \sqrt{\frac{\text{Area}(\mathcal{M})}{p\pi}}$. The value of $\tilde{\rho}_0$ is obtained by expecting each sample point v_j to occupy a geodesic disk of radius $\tilde{\rho}_0$, which would eventually cover the complete shape - that is $p\pi\rho^2 = \text{Area}(\mathcal{M})$. If the choice of the sample is free, we recommend using Poisson Disk Sampling to obtain roughly evenly spaced samples in a fast manner.

Local Dijkstra starting from samples can be accelerated by both parallelization and reduction of the search space to a Euclidean ball of radius ρ_0 around each

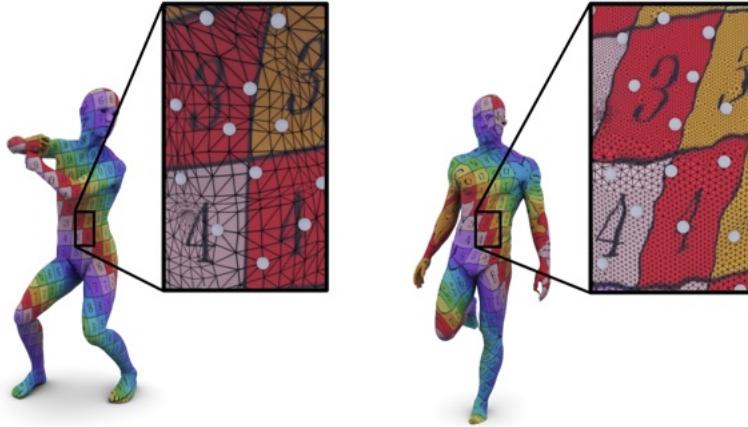


Figure D.3: Texture transfer using our scalable version of ZoomOut on a pair of the SHREC19 dataset. Samples used in the pipeline are shown as white dots.

sample, as we have $d^{\mathcal{M}}(x_i, x_j) \leq \|x_i - x_j\|_2$.

If some points $x_i \in \mathcal{M}$ have not been reached during this process, one should either increase the initial radius ρ_0 or simply add x_i to the sample set \mathcal{S} and run an extra local Dijkstra starting from x_i .

Values of $(\tilde{u}_j)_j$ can now be computed and stored in a sparse $n \times p$ matrix $\widetilde{\mathbf{U}}$, where each column stores a local function. Eventually in order to detect too small self-weights $u_j(v_j)$, we notice from Equation (6.14) that $u_j(v_j) \leq \alpha$ is equivalent to $\sum_i \tilde{u}_i(v_j) \geq \frac{1}{\alpha}$, where the first term is the sum of a row of a $p \times p$ submatrix extracted from $\widetilde{\mathbf{U}}$. Reducing the radius ρ_j of a sample only consists in recomputing the j -th column of $\widetilde{\mathbf{U}}$ from the *same* distance values as computed by the first Dijkstra run. This way, no additional Dijkstra is run, which leads to a somewhat costless improvement of the local functions.

D.8 Texture transfer

We further show the efficiency of our method in terms of accuracy by displaying another example of texture transfer on a pair of dense meshes part of the SHREC19 dataset, as seen on Figure D.3. Here the leftmost shape contains 50 000 vertices and rightmost 200 000, but we only use nearly 1500 samples to obtain such correspondences. Note that in this case, where the number of vertices on the target shape is larger than the number of vertices on the source shape, texture transfer is especially challenging as multiple vertices of the target shape are projected into the same triangles. This makes texture transfer very sensitive to the quality of the estimated map. We stress this pipeline obtains sub-sample accuracy in the correspondences, all in a fraction of the required time to run the exact ZoomOut pipeline. We further highlight that texture at the elbows and shoulder is not smooth on the source shape, which explains the distortion on the target shape. This results simply serves as visualization.

Memory-Scalable and Efficient Functional Map Computations on Dense Meshes: Additional Results

E.1 Implementation Details

In this section, we provide more detailed information on the implementation of our model described in Figure 7.2 and Sec. 7.4.4.

Our model takes as input shapes using 128 WKS descriptors computed from 128 eigenfunctions of the Laplace-Beltrami operator. Similarly to [123, 40], each descriptor function is normalized on the shape with respect to the standard L^2 inner product on a mesh. These descriptors are then fed to a DiffusionBlock with 4 Diffusion blocks of width of 256, in a standard manner [198, 12, 40, 212]. The main difference with these implementations is that we only output 32 feature functions instead of the 128 or 256 usually used.

Features produced by DiffusionNet are used in our Differentiable ZoomOut block which first normalizes the pointwise features, and then computes a scalable dense map equivalent to standard two-branch networks, as shown in Section 7.3 and Eq. (7.3). Using this map, an initial functional map \mathbf{C}_{init} of size $K_{\text{init}} = 30$ is computed, and is fed into a ZoomOut algorithm [145] for 10 iterations with a spectral upsampling step of 10, where the pointwise maps are replaced by our scalable dense maps. This eventually produces our refined map $\mathbf{C}_{\text{refined}}$ of size $K_{\text{refined}} = 130$.

Our loss consists in 3 terms, a orthogonality loss $L_{\text{orth}}(\mathbf{C}_{\text{init}}) = \|\mathbf{C}_{\text{init}}^\top \mathbf{C}_{\text{init}} - I\|_2^2$, a consistency loss $L_{\text{consist}}(\mathbf{C}_{\text{init}}, \mathbf{C}_{\text{refined}}) = \|\mathbf{C}_{\text{init}} - \mathbf{C}_{\text{refined}}\|_2^2$, and a Laplacian bijectivity loss $L_{\text{lap}}(\mathbf{C}_{\text{init}}) = \|\Delta \odot \mathbf{C}_{\text{init}}\|_2^2$, where \odot denotes element-wise product and Δ is obtained from [173, 39]. More precisely, if $\lambda^{(1)}, \lambda^{(2)} \in \mathbb{R}^{K_{\text{init}}}$ denote the vector of eigenvalues of S_1 and S_2 , then Δ is defined element-wise as

$$\begin{aligned} \Delta_{ij}^2 &= \left(\frac{\sqrt{\lambda_i^{(2)}}}{1 + \lambda_i^{(2)}} - \frac{\sqrt{\lambda_j^{(1)}}}{1 + \lambda_j^{(1)}} \right)^2 \\ &\quad + \left(\frac{1}{1 + \lambda_i^{(2)}} - \frac{1}{1 + \lambda_j^{(1)}} \right)^2 \end{aligned} \tag{E.1}$$

This is an extension of the standard Laplacian commutativity loss, which has

| Train | F | | | S | | | F+S | | |
|--------------------------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | F | S | S19 | F | S | S19 | F | S | S19 |
| Test | | | | | | | | | |
| BCICP [174] | 6.1 | - | - | - | 11. | - | - | - | - |
| ZoomOut [145] | 6.1 | - | - | - | 7.5 | - | - | - | - |
| SmoothShells [70] | 2.5 | - | - | - | 4.7 | - | - | - | - |
| DiscreteOp [177] | 5.6 | - | - | - | 13.1 | - | - | - | - |
| GeomFmaps [60] | 3.5 | 4.8 | 8.5 | 4.0 | 4.3 | 11.2 | 3.5 | 4.4 | 7.1 |
| Deep Shells [71] | 1.7 | 5.4 | 27.4 | 2.7 | 2.5 | 23.4 | 1.6 | 2.4 | 21.1 |
| NeuroMorph [72] | 8.5 | 28.5 | 26.3 | 18.2 | 29.9 | 27.6 | 9.1 | 27.3 | 25.3 |
| DUO-FMNet [62] | 2.5 | 4.2 | 6.4 | 2.7 | 2.6 | 8.4 | 2.5 | 4.3 | 6.4 |
| UDMSM [39] | 1.5 | 7.3 | 21.5 | 8.6 | 2.0 | 30.7 | 1.7 | 3.2 | 17.8 |
| ULRSSM [40] | 1.6 | 6.4 | 14.5 | 4.5 | 1.8 | 18.5 | 1.5 | 2.0 | 7.9 |
| ULRSSM (w/ Opt) [40] | 1.6 | 2.2 | 5.7 | 1.6 | 1.9 | 6.7 | 1.6 | 2.1 | 4.6 |
| AttentiveFMaps Fast [123] | 1.9 | 2.6 | 5.8 | 1.9 | 2.1 | 8.1 | 1.9 | 2.3 | 6.3 |
| AttentiveFMaps [123] | 1.9 | 2.6 | 6.4 | 2.2 | 2.2 | 9.9 | 1.9 | 2.3 | 5.8 |
| ConsistentFMaps [212] | 2.3 | 2.6 | 3.8 | 2.4 | 2.5 | 4.5 | 2.2 | 2.3 | 4.3 |
| ConsistentFMaps (dim 80) [212] | 1.7 | 2.6 | 5.5 | 2.2 | 2.0 | 5.8 | 1.7 | 2.2 | 5.6 |
| Ours | 1.9 | 2.4 | 4.2 | 1.9 | 2.4 | 6.9 | 1.9 | 2.3 | 3.6 |

Table E.1: Mean geodesic errors ($\times 100$) when training and testing on the Faust, Scape and Shrec19 datasets. Due to the fine-tuning strategy on ULRSSM (w/ fine-tune), we do not highlight its results. See text for details.

| Method | SMAL |
|----------------------------|------------|
| ULRSSM [40] | 6.9 |
| ULRSSM (w/ fine-tune) [40] | 3.5 |
| AttentiveFMaps [123] | 5.4 |
| ConsistentFMaps [212] | 5.4 |
| Ours (w/o Consistency) | 6.7 |
| Ours | 5.9 |

Table E.2: Mean geodesic errors ($\times 100$) when training and testing on the SMAL dataset.

been used in most existing implementations since GeoFMaps [60].

We do not enforce orthogonality of $\mathbf{C}_{\text{refined}}$, since ZoomOut is proven to promote orthogonal functional maps [145]. We notice that given a sound initialization, ZoomOut produces great results, which inspires us mostly to penalize \mathbf{C}_{init} . Furthermore, during the first iterations, initial functional maps produced by the network have no guarantee to be sound, and we therefore tune down the consistency loss L_{consist} initially until the network converges towards good initialization. The consistency loss then provides meaningful guidance to the network. In practice, we increase the weight of this loss from 10^{-4} to 10^{-1} in 5 epochs using a multiplicative schedule.

The complete implementation is available at <https://github.com/RobinMagnet/SimplifiedFmapsLearning>.

E.2 More Baselines & Ablation

We here present additional results on the standard baselines presented in the manuscript. In particular, some works [123, 212] provided multiple versions of their algorithm. Furthermore, we display results from [40] using further test-time optimization. Note that this test-time optimization fine-tunes the network for each shape on the test set and should be applied to all other methods for fairness. All these additional baselines can be found on Table E.1.

We additionally provide results on the SMAL dataset [247], where we additionally show the result of our pipeline without using the consistency loss (“w/o Consistency”), which serves as an ablation study similar to the one presented in [212]. However, in this ablation, we still use the ZoomOut algorithm at test-time, only the consistency loss was removed. PCK curves for ULRSSM [40] and AttentiveFMaps [123] are also provided on Figure E.1.

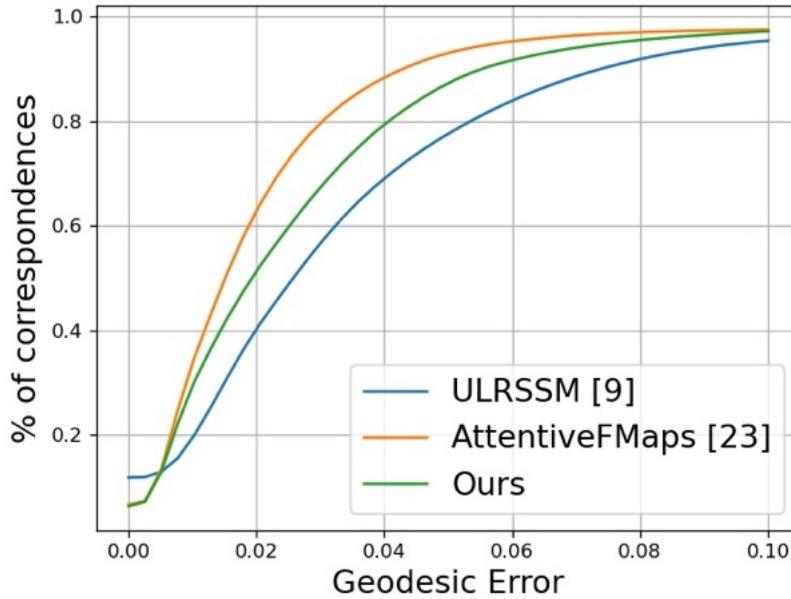


Figure E.1: PCK curves on the SMAL dataset

Algorithm 7 The ZoomOut algorithm

Require: Initial pointwise map $\Pi_{21} \in \{0, 1\}^{n_2 \times n_1}$ from S_2 to S_1 , eigenvectors $\Phi_1 \in \mathbb{R}^{n \times k_1}$ and $\Phi_2^{n \times k_2}$ on each shape.

```

1: for  $k = k_{\text{init}}$  to  $k_{\text{final}}$  do
2:   Compute  $C_{12} = [\Phi_2]_{[:,k]}^\dagger \Pi_{21} [\Phi_1]_{[:,k]}$ 
3:   Compute  $\Pi_{21} = \text{NN}\left([\Phi_1]_{[:,k]} C_{12}^\top, [\Phi_2]_{[:,k]}\right)$ 
4: end for
5: Return  $C_{12}, \Pi_{21}$ 

```

E.3 ZoomOut Algorithm

The ZoomOut algorithm [145] is a simple functional map refinement algorithm, which uses iterative conversions between functional and pointwise maps.

The algorithm is presented on Algorithm 7, where NN denotes the nearest neighbor query between the rows of the two arguments.

E.4 Adapting Scalable ZoomOut

In [131], the authors present an approximation of the functional map for dense shapes using only sparse samples.

This approximation allows running the ZoomOut algorithm on a sparse subset of the vertices of both shapes, only using a complete high dimensional nearest neighbor query at the last step of the algorithm. This last step appears as the heaviest speed bottleneck of the algorithm as presented in [131].

When porting [131] to GPU, this query makes the GPU run out of memory on very dense meshes, which we solve by using our scalable dense maps.

However, this algorithm adds a layer of approximation, which can potentially hinder the results. Furthermore, since it only uses values at sparse samples, the gradient can only propagate through these samples and not to the entire vertex-wise embeddings. In particular, it is not possible to use different samples each time the shape is used in training, as the preprocessing time is not negligible. This refrains us from using this adapted version within our learning framework.

E.5 Dense Meshes

In this section, we provide more information on dense mesh processing using our pipeline, using meshes from the original version of the SHREC19 dataset [144].

While DiffusionNet needs to store the eigenvectors of each shape of size $N \times K$ in memory, it is still able to compute features quickly for each shape. Due to its discretization-agnostic architecture, the features obtained on the dense and remeshed version are similar, as noted on Figure E.2, where each mesh contains $N = 2 \cdot 10^5$ vertices. However, fitting a dense pointwise map would for this mesh require 10^7 MiB of GPU memory, without even storing the gradient, which is infeasible in most cases.

In contrast, our scalable dense map can easily compute these maps. In particular, at test time when no gradient information is stored, our DifferentiableZoomOut has a negligible memory cost since intermediate maps don't need to be stored.

We show an example of texture transfer on another pair of this dataset in Figure E.3. Here, we used our network, trained on the standard remeshed [174] versions of the Faust [28] and Scape [10] datasets, and evaluate at test time on shapes with around 10^5 vertices. We transform the output functional map into a precise map [75]. This demonstrates our pipeline can be trained on simple remeshed versions of datasets, but then used at test time on denser shapes without issues.

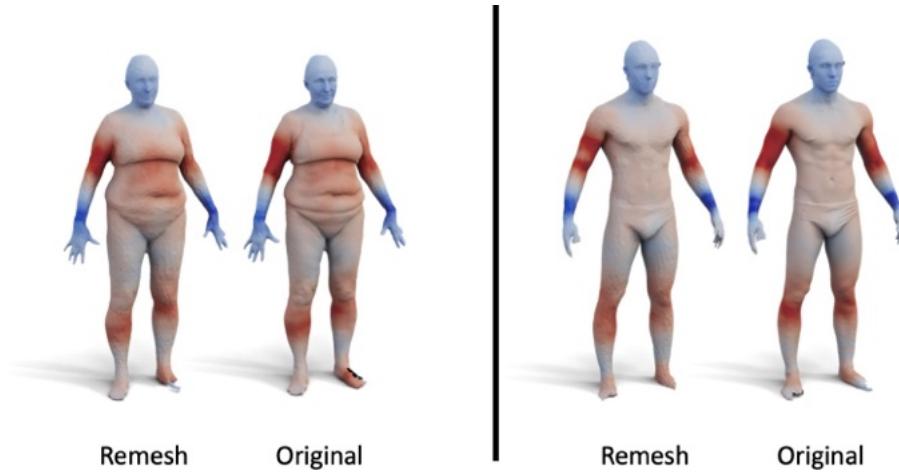


Figure E.2: We leverage on the capacity of DiffusionNet [198] to perform on various discretization of the same shape. Left and right are two shapes from the SHREC19 dataset [144]. We show on each shape features obtained on the remeshed and original version of the dataset.

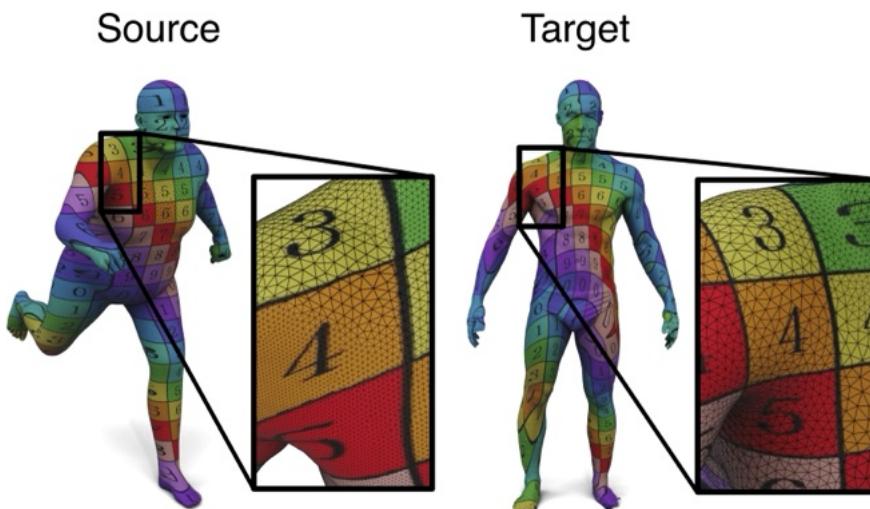


Figure E.3: Example of texture transfer of our method on the SHREC19 [144] dataset using our pipeline.

Bibliography

- [1] A. Abdelreheem, A. Eldesokey, M. Ovsjanikov, and P. Wonka. Zero-Shot 3D Shape Correspondence. In *SIGGRAPH Asia 2023 Conference Papers*, SA '23, 2023. Association for Computing Machinery. (Cited on page 108).
- [2] A. Abdelreheem, I. Skorokhodov, M. Ovsjanikov, and P. Wonka. SATR: Zero-Shot Semantic Segmentation of 3D Shapes. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. IEEE. (Cited on page 108).
- [3] D. C. Adams, F. J. Rohlf, and D. E. Slice. Geometric morphometrics: Ten years of progress following the ‘revolution’. *Italian Journal of Zoology*, 71(1), 2004. (Cited on page 38).
- [4] Y. Aflalo and R. Kimmel. Spectral multidimensional scaling. *Proceedings of the National Academy of Sciences*, 110(45), 2013. (Cited on page 74).
- [5] Y. Aflalo, A. Bronstein, and R. Kimmel. On convex relaxation of graph isomorphism. *Proceedings of the National Academy of Sciences*, 112(10), 2015. (Cited on pages 5 and 15).
- [6] M. Agrawala, A. C. Beers, and M. Levoy. 3D painting on scanned surfaces. In *Proceedings of the 1995 Symposium on Interactive 3D Graphics*, I3D '95, 1995. Association for Computing Machinery. (Cited on pages 4 and 14).
- [7] N. Aigerman and Y. Lipman. Hyperbolic orbifold tutte embeddings. *ACM Transactions on Graphics*, 35(6), 2016. (Cited on pages 22 and 58).
- [8] N. Aigerman, R. Poranne, and Y. Lipman. Seamless surface mappings. *ACM Transactions on Graphics*, 34(4), 2015. (Cited on page 58).
- [9] B. Amberg, S. Romdhani, and T. Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007. IEEE. (Cited on pages 4, 5, 6, 14, 15, 16, 57, 59, 64, 66, 69, 108, 109, and 127).
- [10] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis. SCAPE: Shape completion and animation of people. *ACM Transactions on Graphics*, 24(3), 2005. (Cited on pages 7, 17, 101, and 147).

- [11] G. Arnqvist and T. Mårtensson. Measurement error in geometric morphometrics: Empirical strategies to assess and reduce its impact on measures of shape. *Acta Zoologica Academiae Scientiarum Hungaricae*, 44, 1998. (Cited on page 38).
- [12] S. Attaiki and M. Ovsjanikov. Understanding and Improving Features Learned in Deep Functional Maps. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. IEEE. (Cited on pages xii, 92, 94, 95, 99, 100, 101, 104, 105, and 143).
- [13] S. Attaiki, G. Pai, and M. Ovsjanikov. DPFM: Deep Partial Functional Maps. In *2021 International Conference on 3D Vision (3DV)*, 2021. IEEE. (Cited on pages 6, 17, 38, 97, and 108).
- [14] M. Attene, M. Campen, and L. Kobbelt. Polygon mesh repairing: An application perspective. *ACM Computing Surveys*, 45(2), 2013. (Cited on pages 4 and 14).
- [15] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011. IEEE. (Cited on pages 6, 16, 23, 24, 25, 27, 29, 93, 116, 119, and 133).
- [16] B. B. Avants, C. L. Epstein, M. Grossman, and J. C. Gee. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, 12 (1), 2008. (Cited on pages 6, 16, and 110).
- [17] G. Balakrishnan, A. Zhao, M. R. Sabuncu, J. Guttag, and A. V. Dalca. Voxelmorph: A Learning Framework for Deformable Medical Image Registration. *IEEE Transactions on Medical Imaging*, 38(8), 2019. (Cited on pages 6, 16, and 110).
- [18] A. Bardes, J. Ponce, and Y. LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. In *International Conference on Learning Representations*, 2021. (Cited on page 109).
- [19] L. Bastian, Y. Xie, N. Navab, and Z. Lähner. Hybrid Functional Maps for Crease-Aware Non-Isometric Shape Matching, 2023. (Cited on page 108).
- [20] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. (Cited on pages 1 and 11).
- [21] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1), 2000. (Cited on pages 4 and 14).

- [22] P. J. Besl and N. D. McKay. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1992. (Cited on pages 4, 6, 15, 16, and 43).
- [23] R. Bhalodia, L. A. Dvoracek, A. M. Ayyash, L. Kavan, R. Whitaker, and J. A. Goldstein. Quantifying the Severity of Metopic Craniosynostosis: A Pilot Study Application of Machine Learning in Craniofacial Surgery. *Journal of Craniofacial Surgery*, 31(3), 2020. (Cited on page 51).
- [24] S. Biasotti, A. Cerri, A. Bronstein, and M. Bronstein. Recent Trends, Applications, and Perspectives in 3D Shape Similarity Assessment. *Computer Graphics Forum*, 35(6), 2016. (Cited on pages 38, 58, and 73).
- [25] C. B. Birgfeld, B. S. Saltzman, A. V. Hing, C. L. Heike, P. C. Khanna, J. S. Gruss, and R. A. Hopper. Making the diagnosis: Metopic ridge versus metopic craniosynostosis. *The Journal of Craniofacial Surgery*, 24(1), 2013. (Cited on page 38).
- [26] C. B. Birgfeld, C. L. Heike, F. Al-Mufarrej, A. Oppenheimer, S. E. Kamps, W. Adidharma, and B. Siebold. Practical Computed Tomography Scan Findings for Distinguishing Metopic Craniosynostosis from Metopic Ridging. *Plastic and Reconstructive Surgery. Global Open*, 7(3), 2019. (Cited on page 38).
- [27] K. Bloch, M. Geoffroy, M. Taverne, L. van de Lande, E. O'Sullivan, C. Liang, G. Paternoster, M. Moazen, S. Laporte, and R. Khonsari. New diagnostic criteria for metopic ridges and trigonocephaly: a 3d geometric approach. Unpublished, 2023. (Cited on page 51).
- [28] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and Evaluation for 3D Mesh Registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014. (Cited on pages 4, 7, 15, 17, 66, 101, 130, and 147).
- [29] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black. Dynamic FAUST: Registering Human Bodies in Motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. IEEE. (Cited on pages 1, 4, 7, 11, 15, 17, 33, 82, 91, and 119).
- [30] F. L. Bookstein. Foundations of Morphometrics. *Annual Review of Ecology and Systematics*, 13, 1982. (Cited on page 38).
- [31] F. L. Bookstein. *Morphometric Tools for Landmark Data: Geometry and Biology*. Cambridge University Press, Cambridge, 1992. (Cited on page 38).
- [32] F. L. Bookstein. Pathologies of Between-Groups Principal Components Analysis in Geometric Morphometrics. *Evolutionary Biology*, 46(4), 2019. (Cited on page 38).
- [33] M. Botsch, L. Kobelt, M. Pauly, P. Alliez, and B. Lévy. *Polygon Mesh Processing*. AK Peters / CRC Press, 2010. (Cited on pages 1, 3, 4, 11, 13, and 14).

- [34] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. Generalized multidimensional scaling: A framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences*, 103(5), 2006. (Cited on page 58).
- [35] A. M. Bronstein, M. M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Monographs in Computer Science. Springer, New York, NY, 2009. (Cited on page 66).
- [36] M. M. Bronstein and I. Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. IEEE. (Cited on pages 27 and 93).
- [37] O. Burghard, A. Dieckmann, and R. Klein. Embedding shapes with Green's functions for global shape matching. *Computers & Graphics*, 68, 2017. (Cited on page 74).
- [38] R. Calandrelli, F. Pilato, L. Massimi, A. Marrazzo, M. Panfili, C. Di Rocco, and C. Colosimo. Orbito-facial dysmorphology in patients with different degrees of trigonocephaly severity: Quantitative morpho-volumetric analysis in infants with non-syndromic metopic craniosynostosis. *Child's Nervous System: ChNS: Official Journal of the International Society for Pediatric Neurosurgery*, 36(6), 2020. (Cited on page 38).
- [39] D. Cao and F. Bernard. Unsupervised Deep Multi-shape Matching. In S. Avi-dan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision – ECCV 2022*, volume 13663. Springer Nature Switzerland, Cham, 2022. (Cited on pages 94, 99, 100, 102, 107, 143, and 144).
- [40] D. Cao, P. Roetzer, and F. Bernard. Unsupervised Learning of Robust Spectral Shape Matching. *ACM Transactions on Graphics*, 42(4), 2023. (Cited on pages 6, 7, 17, 18, 92, 93, 94, 95, 97, 99, 100, 102, 103, 107, 108, 109, 143, 144, and 145).
- [41] T. Chakraborty, M. K. Driscoll, E. Jeffery, M. M. Murphy, P. Roudot, B.-J. Chang, S. Vora, W. M. Wong, C. D. Nielson, H. Zhang, V. Zhemkov, C. Hiremath, E. D. De La Cruz, Y. Yi, I. Bezprozvanny, H. Zhao, R. Tomer, R. Heintzmann, J. P. Meeks, D. K. Marciano, S. J. Morrison, G. Danuser, K. M. Dean, and R. Fiolka. Light-sheet microscopy of cleared tissues with isotropic, subcellular resolution. *Nature Methods*, 16(11), 2019. (Cited on pages 1, 3, 11, and 13).
- [42] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. ShapeNet: An Information-Rich 3D Model Repository, 2015. (Cited on pages 1 and 11).

- [43] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. IEEE. (Cited on pages 2 and 12).
- [44] B. Charlier, J. Feydy, J. A. Glaunès, F.-D. Collin, and G. Durif. Kernel operations on the gpu, with autodiff, without memory overflows. *Journal of Machine Learning Research*, 22(74), 2021. (Cited on pages xii, 9, 19, 93, 97, 98, and 103).
- [45] H. Chen, H.-T. D. Liu, A. Jacobson, and D. I. W. Levin. Chordal decomposition for spectral coarsening. *ACM Transactions on Graphics*, 39(6), 2020. (Cited on page 74).
- [46] Z. Chen, A. Tagliasacchi, T. Funkhouser, and H. Zhang. Neural dual contouring. *ACM Transactions on Graphics*, 41(4), 2022. (Cited on page 108).
- [47] M.-J. Cho, A. A. Kane, J. R. Seaward, and R. R. Hallac. Metopic "ridge" vs. "craniosynostosis": Quantifying severity with 3D curvature analysis. *Journal of Cranio-Maxillo-Facial Surgery: Official Publication of the European Association for Cranio-Maxillo-Facial Surgery*, 44(9), 2016. (Cited on page 38).
- [48] A. Cohen and M. Ben-Chen. Robust Shape Collection Matching and Correspondence from Shape Differences. *Computer Graphics Forum*, 39(2), 2020. (Cited on pages ix, 22, 23, 24, 26, 27, 28, 31, 32, 33, 34, 117, 119, and 120).
- [49] E. Corman. Functional Characterization of Intrinsic and Extrinsic Geometry. *ACM Transactions on Graphics*, 2016. (Cited on page 24).
- [50] E. Corman. *Functional representation of deformable surfaces for geometry processing*. Theses, Université Paris Saclay (COmUE), 2016. (Cited on pages 28 and 29).
- [51] L. Cosmo, G. Minello, M. Bronstein, L. Rossi, and A. Torsello. The average mixing kernel signature. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, 2020. Springer International Publishing. (Cited on page 23).
- [52] M. A. A. Cox and T. F. Cox. Multidimensional Scaling. In C.-h. Chen, W. Härdle, and A. Unwin, editors, *Handbook of Data Visualization*, Springer Handbooks Comp.Statistics. Springer, Berlin, Heidelberg, 2008. (Cited on pages 61 and 129).
- [53] K. Crane, C. Weischedel, and M. Wardetzky. The heat method for distance computation. *Communications of the ACM*, 60(11), 2017. (Cited on pages 4 and 14).
- [54] M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. *Advances in Neural Information Processing Systems*, 26, 2013. (Cited on pages 6 and 16).

- [55] A. Daboul, T. Ivanovska, R. Bülow, R. Biffr, and A. Cardini. Procrustes-based geometric morphometrics on MRI images: An example of inter-operator bias in 3D landmarks and its impact on big datasets. *PLOS ONE*, 13(5), 2018. (Cited on page 38).
- [56] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. (Cited on pages viii, 1, 2, 11, and 12).
- [57] A. V. Dalca, G. Balakrishnan, J. Guttag, and M. R. Sabuncu. Unsupervised Learning for Fast Probabilistic Diffeomorphic Registration. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, 2018. Springer International Publishing. (Cited on pages 6 and 16).
- [58] A. V. Dalca, M. Rakic, J. Guttag, and M. R. Sabuncu. Learning conditional deformable templates with convolutional networks. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 2019. (Cited on pages 6, 16, and 109).
- [59] B. Deng, Y. Yao, R. M. Dyke, and J. Zhang. A Survey of Non-Rigid 3D Registration. *Computer Graphics Forum*, 41(2), 2022. (Cited on pages 4, 15, 55, 58, 72, 73, and 91).
- [60] N. Donati, A. Sharma, and M. Ovsjanikov. Deep Geometric Functional Maps: Robust Feature Learning for Shape Correspondence. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. IEEE. (Cited on pages 6, 7, 17, 18, 21, 23, 56, 91, 94, 95, 96, 100, 101, 102, 109, 144, and 145).
- [61] N. Donati, E. Corman, S. Melzi, and M. Ovsjanikov. Complex Functional Maps: A Conformal Link Between Tangent Bundles. *Computer Graphics Forum*, 41(1), 2022. (Cited on pages 93 and 108).
- [62] N. Donati, E. Corman, and M. Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in Shape Matching. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. IEEE. (Cited on pages 100, 102, 108, and 144).
- [63] Y. Duan, H. Zhu, H. Wang, L. Yi, R. Nevatia, and L. J. Guibas. Curriculum DeepSDF. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, 2020. Springer International Publishing. (Cited on pages 108 and 109).
- [64] S. Durrleman, M. Prastawa, N. Charon, J. R. Korenberg, S. Joshi, G. Gerig, and A. Trouvé. Morphometry of anatomical shape complexes with dense deformations and sparse parameters. *NeuroImage*, 101, 2014. (Cited on pages 6, 7, 16, 17, 108, and 109).

- [65] 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, 2020. (Cited on pages 38 and 72).
- [66] N. Dym, H. Maron, and Y. Lipman. DS++: A flexible, scalable and provably tight relaxation for matching problems. *ACM Transactions on Graphics*, 36(6), 2017. (Cited on page 59).
- [67] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, 1995. Association for Computing Machinery. (Cited on pages 3 and 14).
- [68] O. Efroni, D. Ginzburg, and D. Raviv. Spectral Teacher for a Spatial Student: Spectrum-Aware Real-Time Dense Shape Correspondence. In *2022 International Conference on 3D Vision (3DV)*, 2022. (Cited on pages 92, 94, and 96).
- [69] M. Eisenberger, Z. Lähner, and D. Cremers. Divergence-Free Shape Correspondence by Deformation. *Computer Graphics Forum*, 38(5), 2019. (Cited on pages 5, 6, 15, 16, 55, and 109).
- [70] 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*, 2020. (Cited on pages 6, 7, 17, 18, 45, 55, 57, 59, 64, 66, 69, 74, 87, 93, 100, 102, 108, 128, and 144).
- [71] 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, 2020. (Cited on pages 6, 7, 17, 18, 45, 91, 100, 102, 108, and 144).
- [72] 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 *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. IEEE Computer Society. (Cited on pages 6, 17, 100, 102, and 144).
- [73] I. Esteban, P. Schmidt, A. Desgrange, M. Raiola, S. Temiño, S. M. Meilhac, L. Kobbelt, and M. Torres. Pseudodynamic analysis of heart tube formation in the mouse reveals strong regional variability and early left-right asymmetry. *Nature Cardiovascular Research*, 1(5), 2022. (Cited on page 107).
- [74] D. Eynard, E. Rodolà, K. Glashoff, and M. M. Bronstein. Coupled Functional Maps. In *2016 Fourth International Conference on 3D Vision (3DV)*, 2016. (Cited on pages 56 and 74).

- [75] D. Ezuz and M. Ben-Chen. Deblurring and Denoising of Maps between Shapes. *Computer Graphics Forum*, 36(5), 2017. (Cited on pages 24, 58, 59, 60, 68, 74, 91, 94, and 147).
- [76] D. Ezuz, B. Heeren, O. Azencot, M. Rumpf, and M. Ben-Chen. Elastic Correspondence between Triangle Meshes. *Computer Graphics Forum*, 38(2), 2019. (Cited on pages 5, 6, 15, and 16).
- [77] D. Ezuz, J. Solomon, and M. Ben-Chen. Reversible Harmonic Maps between Discrete Surfaces. *ACM Transactions on Graphics*, 38(2), 2019. (Cited on pages 5, 6, 16, 22, 24, 55, 57, 58, 59, 60, 61, 63, 64, 66, 69, 87, 110, 128, and 131).
- [78] A.-C. Fabre, C. Bardua, M. Bon, J. Clavel, R. N. Felice, J. W. Streicher, J. Bonnel, E. L. Stanley, D. C. Blackburn, and A. Goswami. Metamorphosis shapes cranial diversity and rate of evolution in salamanders. *Nature Ecology & Evolution*, 4(8), 2020. (Cited on page 38).
- [79] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. V. Miller, S. Pieper, and R. Kikinis. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, 30(9), 2012. (Cited on page 41).
- [80] F. Fogel, R. Jenatton, F. Bach, and A. D' Aspremont. Convex Relaxations for Permutation Problems. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. (Cited on page 59).
- [81] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1), 1976. (Cited on page 60).
- [82] S. Gallot, D. Hulin, and J. Lafontaine. *Riemannian Geometry*. Universitext. Springer, Berlin, Heidelberg, 2004. (Cited on pages 1 and 11).
- [83] M. Gao, Z. Lahner, J. Thunberg, D. Cremers, and F. Bernard. Isometric Multi-Shape Matching. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. IEEE. (Cited on page 24).
- [84] A. Gehre, M. Bronstein, L. Kobbelt, and J. Solomon. Interactive Curve Constrained Functional Maps. *Computer Graphics Forum*, 37(5), 2018. (Cited on pages 25, 58, and 59).
- [85] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012. (Cited on pages 1 and 11).

- [86] A. Gillet, B. Frédéric, and E. Parmentier. Divergent evolutionary morphology of the axial skeleton as a potential key innovation in modern cetaceans. *Proceedings of the Royal Society B: Biological Sciences*, 286(1916), 2019. (Cited on page 38).
- [87] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry. 3d-coded: 3d correspondences by deep deformation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. (Cited on pages 22 and 101).
- [88] P. Gunz and P. Mitteroecker. Semilandmarks: A method for quantifying curves and surfaces. *Hystrix, the Italian Journal of Mammalogy*, 24(1), 2013. (Cited on page 38).
- [89] B. Hallgrímsson, C. Percival, R. Green, N. Young, W. Mio, and R. Marcucio. Morphometrics, 3D Imaging, and Craniofacial Development. *Current Topics in Developmental Biology*, 115, 2015. (Cited on pages 4 and 14).
- [90] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll. Learning a variational network for reconstruction of accelerated MRI data. *Magnetic Resonance in Medicine*, 79(6), 2018. (Cited on pages 3 and 13).
- [91] P. Hanrahan and P. Haeberli. Direct WYSIWYG painting and texturing on 3D shapes. In *Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '90, 1990. Association for Computing Machinery. (Cited on pages 4 and 14).
- [92] F. Hartwig, J. Sassen, O. Azencot, M. Rumpf, and M. Ben-Chen. An Elastic Basis for Spectral Shape Correspondence. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings*, 2023. ACM. (Cited on pages 9, 20, and 108).
- [93] Q. Huang, F. Wang, and L. Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics*, 33(4), 2014. (Cited on page 24).
- [94] Q.-X. Huang, B. Adams, M. Wicke, and L. J. Guibas. Non-Rigid Registration Under Isometric Deformations. *Computer Graphics Forum*, 2008. (Cited on page 58).
- [95] R. Huang and M. Ovsjanikov. Adjoint Map Representation for Shape Analysis and Matching. *Computer Graphics Forum*, 36(5), 2017. (Cited on pages 23 and 60).
- [96] R. Huang, F. Chazal, and M. Ovsjanikov. On the Stability of Functional Maps and Shape Difference Operators. *Computer Graphics Forum*, 2017. (Cited on page 28).

- [97] R. Huang, F. Chazal, and M. Ovsjanikov. On the Stability of Functional Maps and Shape Difference Operators. *Computer Graphics Forum*, 37(1), 2018. (Cited on page 79).
- [98] R. Huang, P. Achlioptas, L. Guibas, and M. Ovsjanikov. Limit Shapes – A Tool for Understanding Shape Differences and Variability in 3D Model Collections. *Computer Graphics Forum*, 38(5), 2019. (Cited on pages 5, 15, 23, 35, 39, 42, 44, and 56).
- [99] R. Huang, M.-J. Rakotosaona, P. Achlioptas, L. Guibas, and M. Ovsjanikov. OperatorNet: Recovering 3D Shapes From Difference Operators. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. IEEE. (Cited on pages 23, 24, and 35).
- [100] R. Huang, J. Ren, P. Wonka, and M. Ovsjanikov. Consistent ZoomOut: Efficient Spectral Map Synchronization. *Computer Graphics Forum*, 39(5), 2020. (Cited on pages 24, 38, 39, 42, 43, 72, 74, and 77).
- [101] X. Huang and D. Xu. Aspect-ratio based triangular mesh smoothing. In *ACM SIGGRAPH 2017 Posters*, SIGGRAPH '17, 2017. Association for Computing Machinery. (Cited on pages 4 and 14).
- [102] W. Hübner, G. P. McNerney, P. Chen, B. M. Dale, R. E. Gordon, F. Y. S. Chuang, X.-D. Li, D. M. Asmuth, T. Huser, and B. K. Chen. Quantitative 3D Video Microscopy of HIV Transfer Across T Cell Virological Synapses. *Science*, 323(5922), 2009. (Cited on pages 3 and 13).
- [103] V. Jain, H. Zhang, and O. Van Kaick. Non-rigid spectral correspondence of triangle meshes. *International Journal of Shape Modeling*, 13(01), 2007. (Cited on page 74).
- [104] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. Mistral 7B, 2023. (Cited on pages 2 and 12).
- [105] J. Johnson, M. Douze, and H. Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3), 2019. (Cited on page 98).
- [106] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), 2021. (Cited on pages 1 and 11).

- [107] R. Kellogg, A. C. Allori, G. F. Rogers, and J. R. Marcus. Interfrontal angle for characterization of trigonocephaly: Part 1: Development and validation of a tool for diagnosis of metopic synostosis. *The Journal of Craniofacial Surgery*, 23(3), 2012. (Cited on page 38).
- [108] B. Kerbl, G. Kopanas, T. Leimkuehler, and G. Drettakis. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics*, 42(4), 2023. (Cited on pages 1 and 11).
- [109] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. *ACM Transactions on Graphics*, 30(4), 2011. (Cited on pages 5, 6, 15, 16, 59, and 110).
- [110] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015. (Cited on page 100).
- [111] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. (Cited on pages 108 and 109).
- [112] L. Kitchell, D. Bullock, S. Hayashi, and F. Pestilli. Shape Analysis of White Matter Tracts via the Laplace-Beltrami Spectrum. In M. Reuter, C. Wachinger, H. Lombaert, B. Paniagua, M. Lüthi, and B. Egger, editors, *Shape in Medical Imaging*, volume 11167. Springer International Publishing, Cham, 2018. (Cited on pages 38 and 50).
- [113] A. Klein, S. S. Ghosh, F. S. Bao, J. Giard, Y. Häme, E. Stavsky, N. Lee, B. Rossa, M. Reuter, E. Chaibub Neto, and A. Keshavan. Mindboggling morphometry of human brains. *PLOS Computational Biology*, 13(2), 2017. (Cited on pages 38 and 50).
- [114] C. P. Klingenberg. Morphometrics and the role of the phenotype in studies of the evolution of developmental mechanisms. *Gene*, 287(1), 2002. (Cited on page 38).
- [115] C. P. Klingenberg. Size, shape, and form: Concepts of allometry in geometric morphometrics. *Development Genes and Evolution*, 226(3), 2016. (Cited on page 38).
- [116] J. C. Kolar and E. M. Salter. Preoperative anthropometric dysmorphology in metopic synostosis. *American Journal of Physical Anthropology*, 103(3), 1997. (Cited on page 37).
- [117] A. Kovnatsky, M. M. Bronstein, A. M. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. *Computer Graphics Forum*, 32(2pt4), 2013. (Cited on pages 25, 56, and 74).

- [118] A. Kovnatsky, K. Glashoff, and M. M. Bronstein. MADMM: A Generic Algorithm for Non-smooth Optimization on Manifolds. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, Lecture Notes in Computer Science, 2016. Springer International Publishing. (Cited on page 74).
- [119] T. Lescoat, H.-T. D. Liu, J.-M. Thiery, A. Jacobson, T. Boubekeur, and M. Ovsjanikov. Spectral Mesh Simplification. *Computer Graphics Forum*, 39(2), 2020. (Cited on page 74).
- [120] B. Levy. Laplace-Beltrami Eigenfunctions Towards an Algorithm That "Understands" Geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*, 2006. IEEE. (Cited on page 25).
- [121] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics*, 21 (3), 2002. (Cited on pages 3 and 14).
- [122] J. Li, D. Li, S. Savarese, and S. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *ICML'23*, 2023. JMLR.org. (Cited on pages 2, 12, 108, and 109).
- [123] L. Li, N. Donati, and M. Ovsjanikov. Learning Multi-resolution Functional Maps with Spectral Attention for Robust Shape Matching. *Advances in Neural Information Processing Systems*, 35, 2022. (Cited on pages xii, xiv, 7, 9, 18, 20, 92, 94, 97, 99, 100, 101, 102, 103, 104, 107, 108, 143, 144, and 145).
- [124] Y. Li, H. Takehara, T. Taketomi, B. Zheng, and M. Niesner. 4DComplete: Non-Rigid Motion Estimation Beyond the Observable Surface. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. IEEE. (Cited on pages viii, x, xi, 1, 2, 4, 7, 8, 11, 12, 15, 17, 19, 57, 65, 72, 101, 107, and 129).
- [125] Y. Lipman and T. Funkhouser. Möbius voting for surface correspondence. *ACM Transactions on Graphics*, 28(3), 2009. (Cited on page 58).
- [126] O. Litany, E. Rodolà, A. M. Bronstein, M. M. Bronstein, and D. Cremers. Non-Rigid Puzzles. *Computer Graphics Forum*, 35(5), 2016. (Cited on page 108).
- [127] O. Litany, T. Remez, E. Rodola, A. Bronstein, and M. Bronstein. Deep Functional Maps: Structured Prediction for Dense Shape Correspondence. In *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. IEEE. (Cited on pages 21, 23, 94, and 101).
- [128] O. Litany, E. Rodolà, A. M. Bronstein, and M. M. Bronstein. Fully Spectral Partial Shape Matching. *Computer Graphics Forum*, 36(2), 2017. (Cited on pages 6, 17, 22, 23, 74, 108, 118, 119, and 120).

-
- [129] H.-T. D. Liu, A. Jacobson, and M. Ovsjanikov. Spectral coarsening of geometric operators. *ACM Transactions on Graphics*, 38(4), 2019. (Cited on page 74).
 - [130] R. Magnet and M. Ovsjanikov. DWKS : A Local Descriptor of Deformations Between Meshes and Point Clouds. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. IEEE. (Cited on pages 9 and 20).
 - [131] R. Magnet and M. Ovsjanikov. Scalable and Efficient Functional Map Computations on Dense Meshes. *Computer Graphics Forum*, 42(2), 2023. (Cited on pages 9, 20, 96, 103, 104, and 147).
 - [132] R. Magnet and M. Ovsjanikov. Memory scalable and simplified functional map learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2024. (Cited on pages 10 and 20).
 - [133] R. Magnet, J. Ren, O. Sorkine-Hornung, and M. Ovsjanikov. Smooth Non-Rigid Shape Matching via Effective Dirichlet Energy Optimization. In *2022 International Conference on 3D Vision (3DV)*, 2022. IEEE. (Cited on pages 9, 20, 91, 93, 94, 101, and 105).
 - [134] R. Magnet, K. Bloch, M. Taverne, S. Melzi, M. Geoffroy, R. H. Khonsari, and M. Ovsjanikov. Assessing craniofacial growth and form without landmarks: A new automatic approach based on spectral methods. *Journal of Morphology*, 284(8), 2023. (Cited on pages 9 and 20).
 - [135] C. Mallet, R. Cornette, G. Billet, and A. Houssaye. Interspecific variation in the limb long bones among modern rhinoceroses—extent and drivers. *PeerJ*, 7, 2019. (Cited on page 38).
 - [136] M. Mandad, D. Cohen-Steiner, L. Kobbelt, P. Alliez, and M. Desbrun. Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics*, 36(4), 2017. (Cited on pages 58 and 59).
 - [137] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner. Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. *Journal of Cognitive Neuroscience*, 19(9), 2007. (Cited on pages viii, 1, 2, 11, and 12).
 - [138] R. Marin, S. Melzi, E. Rodolà, and U. Castellani. FARM: Functional Automatic Registration Method for 3D Human Bodies. *Computer Graphics Forum*, 39(1), 2020. (Cited on pages 4 and 15).
 - [139] H. Maron, N. Dym, I. Kezurer, S. Kovalsky, and Y. Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics*, 35(4), 2016. (Cited on page 74).

- [140] D. Mateus, R. Horaud, D. Knossow, F. Cuzzolin, and E. Boyer. Articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008. (Cited on page 74).
- [141] I. M. J. Mathijssen and Working Group Guideline Craniosynostosis. Updated Guideline on Treatment and Management of Craniosynostosis. *The Journal of Craniofacial Surgery*, 32(1), 2021. (Cited on page 37).
- [142] G. Meanti, L. Carratino, L. Rosasco, and A. Rudi. Kernel Methods Through the Roof: Handling Billions of Points Efficiently. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. (Cited on page 97).
- [143] S. Melzi, A. Mella, L. Squarcina, M. Bellani, C. Perlini, M. Ruggeri, C. A. Altamura, P. Brambilla, and U. Castellani. Functional Maps for Brain Classification on Spectral Domain. In M. Reuter, C. Wachinger, and H. Lombaert, editors, *Spectral and Shape Analysis in Medical Imaging*, Lecture Notes in Computer Science, 2016. Springer International Publishing. (Cited on pages 50 and 107).
- [144] S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Matching Humans with Different Connectivity. In S. Biasotti, G. Lavoué, and R. Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. (Cited on pages xiii, 7, 17, 72, 86, 101, 147, and 148).
- [145] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov. ZoomOut: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics*, 38(6), 2019. (Cited on pages ix, x, 6, 17, 23, 24, 30, 31, 32, 38, 39, 42, 56, 58, 60, 66, 69, 72, 74, 75, 77, 79, 83, 86, 87, 89, 91, 93, 94, 96, 97, 99, 100, 102, 103, 105, 108, 143, 144, 145, and 147).
- [146] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In G. Farin, H.-C. Hege, D. Hoffman, C. R. Johnson, K. Polthier, H.-C. Hege, and K. Polthier, editors, *Visualization and Mathematics III*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. (Cited on pages 4, 14, 59, 91, and 94).
- [147] A. Myronenko, X. Song, and M. Carreira-Perpiñán. Non-rigid point set registration: Coherent Point Drift. In *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. (Cited on pages 6 and 16).
- [148] A. Nasikun and K. Hildebrandt. The hierarchical subspace iteration method for laplace–beltrami eigenproblems. *ACM Transactions on Graphics (TOG)*, 41(2), 2022. (Cited on pages xi, 72, 73, and 74).

- [149] A. Nasikun, C. Brandt, and K. Hildebrandt. Fast Approximation of Laplace-Beltrami Eigenproblems. *Computer Graphics Forum*, 37(5), 2018. (Cited on pages xi, 72, 73, 74, 75, 76, 77, 78, 79, 81, 82, 83, 84, 85, 88, 89, 107, and 135).
- [150] A. Nguyen, M. Ben-Chen, K. Welnicka, Y. Ye, and L. Guibas. An Optimization Approach to Improving Collections of Shape Maps. *Computer Graphics Forum*, 30(5), 2011. (Cited on pages 24 and 42).
- [151] M. Niethammer, M. Reuter, F.-E. Wolter, S. Bouix, N. Peinecke, M.-S. Koo, and M. E. Shenton. Global Medical Shape Analysis Using the Laplace-Beltrami Spectrum. In N. Ayache, S. Ourselin, and A. Maeder, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2007*, volume 4791. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. (Cited on pages 38 and 50).
- [152] M. J. Nitzken, M. F. Casanova, M. D. G. Gimel'farb, T. Inanc, J. M. Zurada, and A. El-Baz. Shape Analysis of Human Brain: A Brief Survey. *IEEE TRANSACTIONS ON INFORMATION TECHNOLOGY IN BIOMEDICINE*, 2013. (Cited on pages 38 and 50).
- [153] D. Nogneng and M. Ovsjanikov. Informative Descriptor Preservation via Commutativity for Shape Matching. *Computer Graphics Forum*, 36(2), 2017. (Cited on pages 6, 17, 23, 25, 32, 33, 56, 58, 59, 74, and 91).
- [154] D. Nogneng, S. Melzi, E. Rodolà, U. Castellani, M. Bronstein, and M. Ovsjanikov. Improved Functional Mappings via Product Preservation. *Computer Graphics Forum*, 37(2), 2018. (Cited on pages 6, 17, 58, 59, 60, 91, 93, 116, 117, 118, 119, and 120).
- [155] P. O'Higgins and N. Jones. Facial growth in Cercopithecus torquatus: An application of three-dimensional geometric morphometric techniques to the study of morphological variation. *Journal of Anatomy*, 193 (Pt 2)(Pt 2), 1998. (Cited on page 38).
- [156] M. Oquab, T. Darzet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Bal-las, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*, 2023. (Cited on pages 108 and 109).
- [157] S. Osher and R. Fedkiw. Constructing Signed Distance Functions. In S. Osher and R. Fedkiw, editors, *Level Set Methods and Dynamic Implicit Surfaces*. Springer, New York, NY, 2003. (Cited on page 109).
- [158] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One Point Isometric Matching with the Heat Kernel. *Computer Graphics Forum*, 29(5), 2010. (Cited on pages 58 and 74).

- [159] 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*, 31(4), 2012. (Cited on pages 5, 6, 15, 17, 23, 24, 25, 32, 38, 42, 56, 58, 59, 60, 72, 73, 74, 76, 91, 92, 93, 94, 95, 99, 100, 110, and 130).
- [160] M. Ovsjanikov, M. Ben-Chen, F. Chazal, and L. Guibas. Analysis and Visualization of Maps Between Shapes. *Computer Graphics Forum*, 32(6), 2013. (Cited on page 8).
- [161] M. Ovsjanikov, Q. Mérigot, V. Pătrăucean, and L. Guibas. Shape Matching via Quotient Spaces. *Computer Graphics Forum*, 32(5), 2013. (Cited on page 74).
- [162] 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 *ACM SIGGRAPH 2017 Courses*, SIGGRAPH '17, 2017. Association for Computing Machinery. (Cited on pages 6, 17, 25, 38, 42, 56, 58, 72, 93, and 94).
- [163] O. Özyeşil, V. Voroninski, R. Basri, and A. Singer. A survey of structure from motion. *Acta Numerica*, 26, 2017. (Cited on pages 3 and 13).
- [164] 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 *CVPR*, 2021. (Cited on pages 59, 60, and 95).
- [165] A. Palci and M. S. Y. Lee. Geometric morphometrics, homology and cladistics: Review and recommendations. *Cladistics*, 35(2), 2019. (Cited on page 38).
- [166] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019. (Cited on page 109).
- [167] U. Pinkall and K. Polthier. Computing Discrete Minimal Surfaces and Their Conjugates. *Experimental Mathematics*, 2(1), 1993. (Cited on pages 4, 14, 60, and 113).
- [168] G. Podobnik, P. Strojan, P. Peterlin, B. Ibragimov, and T. Vrtovec. HaN-Seg: The head and neck organ-at-risk CT and MR segmentation dataset. *Medical Physics*, 50(3), 2023. (Cited on pages 1 and 11).
- [169] A. Poulenard, M.-J. Rakotosaona, Y. Ponty, and M. Ovsjanikov. Effective Rotation-Invariant Point CNN with Spherical Harmonics Kernels. In *2019 International Conference on 3D Vision (3DV)*, 2019. IEEE. (Cited on pages 4 and 14).
- [170] A. Punjani and D. J. Fleet. 3D variability analysis: Resolving continuous flexibility and discrete heterogeneity from single particle cryo-EM. *Journal of Structural Biology*, 213(2), 2021. (Cited on pages 3 and 13).

- [171] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 30, 2017. (Cited on pages 2 and 12).
- [172] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021. (Cited on pages 108 and 109).
- [173] J. Ren, M. Panine, P. Wonka, and M. Ovsjanikov. Structured Regularization of Functional Map Computations. *Computer Graphics Forum*, 38(5), 2019. (Cited on pages 95, 100, and 143).
- [174] J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics*, 37(6), 2019. (Cited on pages 7, 18, 21, 23, 24, 25, 38, 57, 58, 59, 60, 66, 72, 74, 86, 91, 93, 94, 100, 101, 102, 130, 144, and 147).
- [175] J. Ren, S. Melzi, M. Ovsjanikov, and P. Wonka. MapTree: Recovering multiple solutions in the space of maps. *ACM Transactions on Graphics*, 39(6), 2020. (Cited on pages 21, 23, 58, 63, 74, 75, and 131).
- [176] J. Ren, P. Wonka, G. Harihara, and M. Ovsjanikov. Geometric analysis of shape variability of lower jaws of prehistoric humans. *L'Anthropologie*, 124(5), 2020. (Cited on pages viii, xi, 3, 13, 73, and 107).
- [177] J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov. Discrete Optimization for Shape Matching. *Computer Graphics Forum*, 40(5), 2021. (Cited on pages 6, 8, 17, 19, 38, 42, 57, 58, 59, 60, 61, 62, 63, 66, 69, 72, 74, 75, 77, 91, 92, 93, 94, 95, 100, 102, 132, 133, and 144).
- [178] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace–Beltrami spectra as ‘Shape-DNA’ of surfaces and solids. *Computer-Aided Design*, 38(4), 2006. (Cited on pages 72, 74, and 77).
- [179] M. Reuter, F.-E. Wolter, M. Shenton, and M. Niethammer. Laplace–Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis. *Computer-Aided Design*, 41(10), 2009. (Cited on pages 25, 38, 46, and 48).
- [180] E. Rodolà, M. Moeller, and D. Cremers. *Point-Wise Map Recovery and Refinement from Functional Correspondence*. The Eurographics Association, 2015. (Cited on pages 58 and 60).
- [181] E. Rodolà, L. Cosmo, M. M. Bronstein, A. Torsello, and D. Cremers. Partial Functional Correspondence: Partial Functional Correspondence. *Computer Graphics Forum*, 36(1), 2017. (Cited on pages 6, 17, 22, 23, 24, 25, 31, 32, 38, 56, 74, 108, 118, and 119).

- [182] P. Roetzer, P. Swoboda, D. Cremers, and F. Bernard. A Scalable Combinatorial Solver for Elastic Geometrically Consistent 3D Shape Matching. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. (Cited on pages 6 and 16).
- [183] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2022. (Cited on pages 2, 12, and 108).
- [184] J.-M. Roufosse, A. Sharma, and M. Ovsjanikov. Unsupervised Deep Learning for Structured Shape Matching. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. IEEE. (Cited on pages 7, 18, 21, 23, 92, and 94).
- [185] A. Rudi, L. Carratino, and L. Rosasco. FALKON: An Optimal Large Scale Kernel Method. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (Cited on page 97).
- [186] R. M. Rustamov, M. Ovsjanikov, O. Azencot, M. Ben-Chen, F. Chazal, and L. Guibas. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics*, 32(4), 2013. (Cited on pages 4, 5, 8, 14, 15, 23, 24, 25, 26, 28, 33, 44, 56, 74, 87, 109, 114, and 117).
- [187] J. W. Rutland, C. P. Bellaire, A. Yao, A. Arrighi-Allisan, J. G. Napoli, B. N. Delman, and P. J. Taub. The Expanding Role of Geometric Morphometrics in Craniofacial Surgery. *Journal of Craniofacial Surgery*, 32(3), 2021. (Cited on page 38).
- [188] Y. Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36(8), 2020. (Cited on pages 21, 23, 38, 55, 58, 72, 73, 91, and 93).
- [189] M. Schaufelberger, R. Kühle, A. Wachter, F. Weichel, N. Hagen, F. Ringwald, U. Eisenmann, J. Hoffmann, M. Engel, C. Freudlsperger, and W. Nahm. A Radiation-Free Classification Pipeline for Craniosynostosis Using Statistical Shape Modeling. *Diagnostics*, 12(7), 2022. (Cited on page 51).
- [190] P. Schmidt, J. Born, M. Campen, and L. Kobbelt. Distortion-minimizing injective maps between surfaces. *ACM Transactions on Graphics*, 38(6), 2019. (Cited on pages 5, 6, 16, and 22).
- [191] P. Schmidt, D. Pieper, and L. Kobbelt. Surface Maps via Adaptive Triangulations. In *Computer Graphics Forum*, volume 42, 2023. (Cited on pages 5, 6, and 16).
- [192] J. A. Sethian et al. *Level Set Methods and Fast Marching Methods*, volume 98. Cambridge Cambridge UP, 1999. (Cited on pages 4 and 14).

- [193] N. Shapira and M. Ben-Chen. Cross-Collection Map Inference by Intrinsic Alignment of Shape Spaces. *Computer Graphics Forum*, 33(5), 2014. (Cited on pages 22, 23, 24, 26, 27, 28, 31, 116, and 117).
- [194] A. Sharf, T. Lewiner, G. Shklarski, S. Toledo, and D. Cohen-Or. Interactive topology-aware surface reconstruction. *ACM Transactions on Graphics*, 26(3), 2007. (Cited on pages 4 and 14).
- [195] A. Sharma and R. Horaud. Shape matching based on diffusion embedding and on mutual isometric consistency. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010. (Cited on page 74).
- [196] A. Sharma and M. Ovsjanikov. Weakly Supervised Deep Functional Maps for Shape Matching. *Advances in Neural Information Processing Systems*, 33, 2020. (Cited on page 23).
- [197] N. Sharp and K. Crane. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum*, 39(5), 2020. (Cited on pages 23 and 34).
- [198] N. Sharp, S. Attaiki, K. Crane, and M. Ovsjanikov. DiffusionNet: Discretization Agnostic Learning on Surfaces. *ACM Transactions on Graphics*, 41(3), 2022. (Cited on pages xiii, 2, 4, 5, 6, 7, 12, 14, 17, 18, 91, 94, 99, 108, 109, 143, and 148).
- [199] Y. Shen, Q. Huang, N. Srebro, and S. Sanghavi. Normalized spectral map synchronization. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. (Cited on page 24).
- [200] Y. Shi, M. Xu, S. Yuan, and Y. Fang. Unsupervised Deep Shape Descriptor With Point Distribution Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. IEEE. (Cited on page 99).
- [201] R. Shishegar, F. Pizzagalli, N. Georgiou-Karistianis, G. F. Egan, N. Jahan-shad, and L. A. Johnston. A gyrification analysis approach based on Laplace Beltrami eigenfunction level sets. *NeuroImage*, 229, 2021. (Cited on pages 38 and 50).
- [202] M. Shoham, A. Vaxman, and M. Ben-Chen. Hierarchical Functional Maps between Subdivision Surfaces. *Computer Graphics Forum*, 38(5), 2019. (Cited on page 75).
- [203] D. E. Slice, editor. *Modern Morphometrics in Physical Anthropology*. Springer US, Boston, MA, 2005. (Cited on pages 4 and 14).
- [204] D. E. Slice. Geometric Morphometrics. *Annual Review of Anthropology*, 36(1), 2007. (Cited on page 38).

- [205] D. Smirnov and J. Solomon. HodgeNet: Learning spectral geometry on triangle meshes. *ACM Transactions on Graphics*, 40(4), 2021. (Cited on pages 2, 12, and 108).
- [206] J. Smisek, M. Jancosek, and T. Pajdla. 3D with Kinect. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011. IEEE. (Cited on pages 1, 3, 11, and 13).
- [207] Y. Soliman, D. Slepčev, and K. Crane. Optimal cone singularities for conformal flattening. *ACM Transactions on Graphics*, 37(4), 2018. (Cited on pages 3 and 14).
- [208] J. Solomon, G. Peyré, V. G. Kim, and S. Sra. Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics*, 35(4), 2016. (Cited on pages 6, 16, 58, and 59).
- [209] O. Sorkine and M. Alexa. As-Rigid-As-Possible Surface Modeling. In A. Belyaev and M. Garland, editors, *Geometry Processing*. The Eurographics Association, 2007. (Cited on pages 4, 5, 6, 14, 15, 16, 57, 59, 64, 66, 69, 91, 109, and 127).
- [210] 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*, SGP '04, 2004. Association for Computing Machinery. (Cited on pages 4 and 14).
- [211] J. Sun, M. Ovsjanikov, and L. Guibas. A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. *Computer Graphics Forum*, 28(5), 2009. (Cited on pages 6, 16, 23, 25, and 27).
- [212] M. Sun, S. Mao, P. Jiang, M. Ovsjanikov, and R. Huang. Spatially and Spectrally Consistent Deep Functional Maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. (Cited on pages 6, 7, 17, 18, 91, 92, 93, 94, 95, 96, 97, 99, 100, 101, 102, 103, 104, 105, 107, 108, 109, 143, 144, and 145).
- [213] K. Takayama. Compatible intrinsic triangulations. *ACM Transactions on Graphics*, 41(4), 2022. (Cited on pages 6 and 16).
- [214] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3D Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7), 2013. (Cited on page 73).
- [215] G. Taubin. Curve and surface smoothing without shrinkage. In *Proceedings of IEEE International Conference on Computer Vision*, 1995. (Cited on pages 4 and 14).

- [216] 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/CVF International Conference on Computer Vision*, 2019. (Cited on page 94).
- [217] O. O. Thomas, H. Shen, R. L. Raauw, W. E. Harcourt-Smith, J. D. Polk, and M. Hasegawa-Johnson. Automated morphological phenotyping using learned shape descriptors and functional maps: A novel approach to geometric morphometrics. Preprint, Bioinformatics, 2021. (Cited on pages 4, 14, and 107).
- [218] R. Tibrewala, T. Dutt, A. Tong, L. Ginocchio, M. B. Keerthivasan, S. H. Baete, S. Chopra, Y. W. Lui, D. K. Sodickson, H. Chandarana, and P. M. Johnson. FastMRI Prostate: A Publicly Available, Biparametric MRI Dataset to Advance Machine Learning for Prostate Cancer Imaging, 2023. (Cited on pages 1, 3, 11, and 13).
- [219] F. Tombari, S. Salti, and L. Di Stefano. Unique Signatures of Histograms for Local Surface Description. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, Lecture Notes in Computer Science, 2010. Springer. (Cited on pages 6, 16, 93, and 119).
- [220] N. Toussaint, Y. Redhead, M. Vidal-García, L. Lo Vercio, W. Liu, E. M. C. Fisher, B. Hallgrímsson, V. L. J. Tybulewicz, J. A. Schnabel, and J. B. A. Green. A landmark-free morphometrics pipeline for high-resolution phenotyping: Application to a mouse model of Down syndrome. *Development (Cambridge, England)*, 148(18), 2021. (Cited on page 38).
- [221] H. Touvron, T. Lavigra, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. LLaMA: Open and Efficient Foundation Language Models, 2023. (Cited on pages 2 and 12).
- [222] G. Trappolini, L. Cosmo, L. Moschella, R. Marin, S. Melzi, and E. Rodolà. Shape Registration in the Time of Transformers. In *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021. (Cited on page 101).
- [223] J. van der Meulen, R. van der Hulst, L. van Adrichem, E. Arnaud, D. Chin-Shong, C. Duncan, E. Habets, J. Hinojosa, I. Mathijssen, P. May, D. Morriss, H. Nishikawa, P. Noons, D. Richardson, S. Wall, J. van der Vlugt, and D. Renier. The increase of metopic synostosis: A pan-European observation. *The Journal of Craniofacial Surgery*, 20(2), 2009. (Cited on page 37).
- [224] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6), 2011. (Cited on pages 4, 15, 21, 55, and 73).

- [225] A. Vaxman, M. Ben-Chen, and C. Gotsman. A multi-resolution approach to heat kernels on discrete surfaces. *ACM Transactions on Graphics*, 29(4), 2010. (Cited on page 74).
- [226] M. Vestner, Z. Lähner, A. Boyarski, O. Litany, R. Slossberg, T. Remez, E. Rodola, A. Bronstein, M. Bronstein, R. Kimmel, and D. Cremers. Efficient Deformable Shape Correspondence via Kernel Matching. In *2017 International Conference on 3D Vision (3DV)*, 2017. (Cited on page 58).
- [227] M. Vestner, R. Litman, E. Rodola, A. Bronstein, and D. Cremers. Product Manifold Filter: Non-rigid Shape Correspondence via Kernel Density Estimation in the Product Space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. IEEE. (Cited on pages 6, 16, 58, and 94).
- [228] C. Voisin. *Hodge Theory and Complex Algebraic Geometry I*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2002. (Cited on pages 1 and 11).
- [229] N. von Cramon-Taubadel, B. C. Frazier, and M. M. Lahr. The problem of assessing landmark error in geometric morphometrics: Theory, methods, and modifications. *American Journal of Physical Anthropology*, 134(1), 2007. (Cited on page 38).
- [230] L. Wang, A. Gehre, M. M. Bronstein, and J. Solomon. Kernel Functional Maps. *Computer Graphics Forum*, 37(5), 2018. (Cited on pages 23, 25, and 58).
- [231] Y. Wang, B. Liu, K. Zhou, and Y. Tong. Vector Field Map Representation for Near Conformal Surface Correspondence. *Computer Graphics Forum*, 37(6), 2018. (Cited on page 58).
- [232] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, 38(5), 2019. (Cited on pages 2, 7, 12, 18, and 109).
- [233] S. K. T. S. Wärmländer, H. Garvin, P. Guyomarc'h, A. Petaros, and S. B. Sholts. Landmark Typology in Applied Morphometrics Studies: What's the Point? *The Anatomical Record*, 302(7), 2019. (Cited on page 38).
- [234] A. Watanabe and D. E. Slice. The utility of cranial ontogeny for phylogenetic inference: A case study in crocodylians using geometric morphometrics. *Journal of Evolutionary Biology*, 27(6), 2014. (Cited on page 38).
- [235] M. Webster and H. D. Sheets. A Practical Introduction to Landmark-Based Geometric Morphometrics. *The Paleontological Society Papers*, 16, 2010. (Cited on page 38).
- [236] C. Weischedel. *A Discrete Geometric View on Shear-Deformable Shell Models*. PhD Thesis, Georg-August-Universität, Göttingen, 2012. (Cited on pages 6 and 16).

- [237] R. Wiersma, E. Eisemann, and K. Hildebrandt. CNNs on surfaces using rotation-equivariant features. *ACM Transactions on Graphics*, 39(4), 2020. (Cited on page 101).
- [238] T. Windheuser, U. Schlickwei, F. R. Schmidt, and D. Cremers. Geometrically consistent elastic matching of 3D shapes: A linear programming solution. In *2011 International Conference on Computer Vision*, 2011. (Cited on pages 6 and 16).
- [239] T. Windheuser, U. Schlickwei, F. R. Schimdt, and D. Cremers. Large-Scale Integer Linear Programming for Orientation Preserving 3D Shape Matching. *Computer Graphics Forum*, 30(5), 2011. (Cited on pages 6 and 16).
- [240] R. Xiang, R. Lai, and H. Zhao. A Dual Iterative Refinement Method for Non-Rigid Shape Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. (Cited on page 72).
- [241] S. Xie, J. Gu, D. Guo, C. R. Qi, L. Guibas, and O. Litany. PointContrast: Unsupervised Pre-training for 3D Point Cloud Understanding. In A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, editors, *Computer Vision – ECCV 2020*, 2020. Springer International Publishing. (Cited on pages 108 and 109).
- [242] C. Xu, H. Lin, H. Hu, and Y. He. Fast calculation of Laplace-Beltrami eigenproblems via subdivision linear subspace. *Computers & Graphics*, 97, 2021. (Cited on page 77).
- [243] D.-M. Yan, G. Bao, X. Zhang, and P. Wonka. Low-Resolution Remeshing Using the Localized Restricted Voronoi Diagram. *IEEE Transactions on Visualization and Computer Graphics*, 20(10), 2014. (Cited on pages 66 and 130).
- [244] D. Yang, S. Zhang, Z. Yan, C. Tan, K. Li, and D. Metaxas. Automated anatomical landmark detection on distal femur surface using convolutional neural network. In *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, 2015. (Cited on pages 4 and 15).
- [245] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun. Point Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. (Cited on page 108).
- [246] J. Zhou, J. Wang, B. Ma, Y.-S. Liu, T. Huang, and X. Wang. Uni3D: Exploring Unified 3D Representation at Scale. In *The Twelfth International Conference on Learning Representations*, 2023. (Cited on page 109).
- [247] S. Zuffi, A. Kanazawa, D. W. Jacobs, and M. J. Black. 3D Menagerie: Modeling the 3D Shape and Pose of Animals. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. (Cited on pages 1, 7, 11, 17, 101, and 145).

Titre : Méthodes spectrales robustes pour l'analyse des formes et l'évaluation des déformations.

Mots clés : géometrie, correspondance, maillage, cartes fonctionnelles, apprentissage profond

Résumé : Le traitement et l'analyse automatiques des formes 3D est un domaine actif de la recherche moderne, avec des retombées dans de nombreux domaines. L'un des principaux défis de ce domaine réside dans la comparaison efficace des formes, par exemple pour détecter des anomalies dans les scanners d'organes, ce qui nécessite souvent d'établir des correspondances entre les surfaces. En particulier, le cadre des cartes fonctionnelles, basé sur l'analyse spectrale des formes, offre une approche flexible pour représenter et calculer ces correspondances, servant de base pour les analyses ultérieures. Cette thèse cherche à améliorer ces méthodes spectrales afin de réaliser des comparaisons de formes robustes et efficaces applicables aux données du monde réel. Dans la première partie, nous nous concentrons sur la caractérisation des différences entre les formes. Nous introduisons un descripteur des différences entre les formes, offrant des informations sur la distorsion autour de chaque point. Nous appliquons ensuite des outils similaires à un ensemble de scans de crânes afin de détecter une maladie craniofaciale, en mettant en évidence certaines exigences spécifiques des

praticiens. Nous soulignons notamment l'importance de la continuité des correspondances et de l'application rapide des méthodes à des maillages denses. Dans la deuxième partie, nous répondons à ces besoins, tout d'abord en présentant un nouvel algorithme de correspondance de formes promouvant la continuité, ainsi qu'un nouveau jeu de données complexe pour l'évaluation. Nous nous concentrerons ensuite sur l'amélioration des méthodes de cartes fonctionnelles afin de traiter des maillages réels de haute résolution. Nous présentons alors une approximation de la carte fonctionnelle, permettant le calcul rapide de correspondances sur des maillages possédant des centaines de milliers de sommets. Finalement, nous présentons une nouvelle approche d'apprentissage profond pour le calcul de cartes fonctionnelles, en supprimant le stockage de grandes matrices denses dans la mémoire du GPU, améliorant ainsi l'extensibilité de la méthode. Dans l'ensemble, cette thèse propose des outils efficaces pour analyser les différences entre les formes et fournit des méthodes générales pour simplifier et accélérer les calculs de correspondance, facilitant ainsi les applications en aval.

Title : Robust spectral methods for shape analysis and deformation assessment

Keywords : geometry, shape matching, mesh, functional maps, deep learning

Abstract : Automatically processing and analyzing 3D shapes is an active area in modern research with implications in various fields. A key challenge in shape analysis lies in efficiently comparing shapes, for example to detect abnormalities in scans of organs, which often requires automatically deforming one shape into another, or establishing correspondences between surfaces. In this context, the functional map framework, based on spectral shape analysis, offers a flexible approach to representing and computing these correspondences, serving as a foundation for subsequent analysis. This thesis seeks to address the limitations of existing spectral methods, with the ultimate goal to achieve robust and efficient shape comparisons applicable to real-world data. In the first part, we concentrate on assessing deformations between shapes effectively, and introduce a descriptor of differences between shapes, capturing information about the distortion around each point. Next, we apply similar tools on a set of skull scans for craniofacial disease detection, highlighting the specific requirements of shape matching practitioners. Notably, we underscore the significance of correspon-

dence smoothness and scalability to dense meshes. In the second part, we address these needs by extending existing functional map methods. Firstly, we introduce a novel shape correspondence pipeline, which explicitly promotes smoothness of computed correspondences, alongside a new challenging shape matching dataset. Secondly, we focus on enhancing the scalability of functional map pipelines to handle real-world dense meshes. For this, we present an approximation of the functional map, enabling the computation of correspondences on meshes with hundreds of thousands of vertices in a fraction of the processing time required by standard algorithms. Finally, we introduce a new learning-based approach, by modifying existing techniques for functional map computations, eliminating the need for large dense matrix storage in GPU memory, thereby improving scalability and numerical stability. Overall, our work contributes efficient tools for analyzing differences between shapes and provides general methods to simplify and accelerate correspondence computations, facilitating downstream applications.