STAG: Smart Tools and Applications in Graphics (2024)
A. Caputo, V. Garro, U. Castellani, and A. Giachetti (Editors)

S4A: Scalable Spectral Statistical Shape Analysis

F. Maccarone[1,2], G. Longari[1], G. Viganò[1], D. Peruzzo[2], F. Maggioli[1] and S. Melzi[1]

[1]University of Milano-Bicocca, Milan, Italy
[2]Neuroimaging Unit, IRCCS Scientific Institute E. Medea, Bosisio Parini, Lecco LC, Italy.

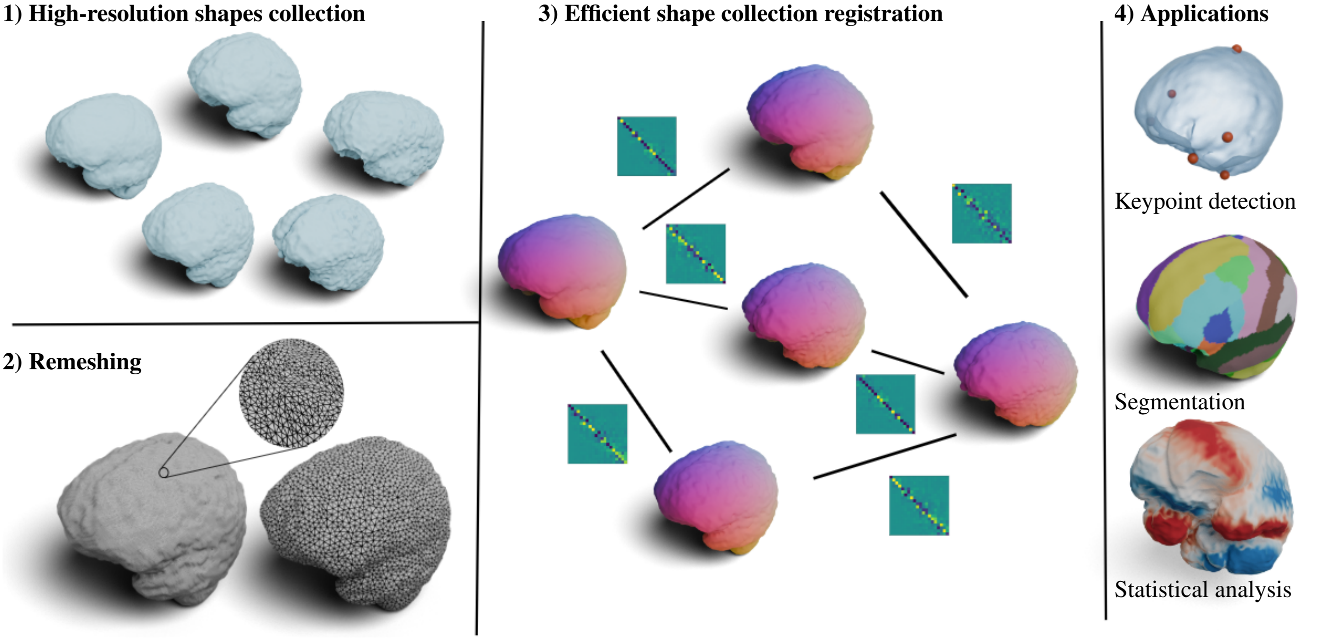
1) High-resolution shapes collection
2) Remeshing
3) Efficient shape collection registration
4) Applications
Keypoint detection
Segmentation
Statistical analysis

Figure 1: *An overview of the S4A pipeline: 1) The input of our pipeline is a collection of high-resolution shapes. 2) We perform remeshing to reduce the complexity of the data. 3) Thanks to the remeshed version of our data, we can efficiently compute correspondences via functional maps between the shapes of the collection and make them comparable. 4) We can use the registered collection to perform various applications*

.

Abstract

Statistical shape analysis is a crucial technique for studying deformations within collections of shapes, particularly in the field of Medical Imaging. However, the high density of meshes typically used to represent medical data poses a challenge for standard geometry processing tools due to their limited efficiency. While spectral approaches offer a promising solution by effectively handling high-frequency variations inherent in such data, their scalability is questioned by their need to solve eigendecompositions of large sparse matrices. In this paper, we introduce S4A, a novel and efficient method based on spectral geometry processing, that addresses these issues with a low computational cost. It operates in four stages: (i) establishing correspondences between each pair of shapes in the collection, (ii) defining a common latent space to encode deformations across the entire collection, (iii) computing statistical quantities to identify, highlight, and measure the most representative variations within the collection, and iv) performing information transfer from labeled data to large collections of shapes. Unlike previous methods, S4A provides a highly efficient solution across all stages of the process. We demonstrate the advantages of our approach by comparing its accuracy and computational efficiency to existing pipelines, and by showcasing the comprehensive statistical insights that can be derived from applying our method to a collection of medical data.

CCS Concepts

• ***Computing methodologies*** → ***Shape analysis;*** • ***Theory of computation*** → *Computational geometry;*

DOI: 10.2312/stag.20241343

1. Introduction

The analysis of shape deformations is a fundamental task in numerous scientific and engineering fields, particularly in medical imaging, where understanding the variability of anatomical structures is crucial for both accurate diagnosis and treatment planning. Statistical shape analysis aims to quantify and analyze these deformations across a collection of shapes, providing insights into the underlying patterns and variations. However, medical data often come in the form of very dense meshes, which complicates and limits the application of traditional geometry processing techniques in the context of high resolution meshes. These methods could greatly assist in various task in the medical field, including not only statistical shape analysis but also applications like organ shape segmentation, reconstruction, and alignment, by leveraging the rich information contained in 3D representations. However, these standard approaches often struggle with efficiency and scalability when dealing with large and complex datasets, which is typical of medical applications, making them not feasible and not usable.

To address these challenges, spectral geometry processing has emerged as a powerful tool for analyzing shape collections. Spectral methods are particularly effective at handling high-frequency variations, which are common in medical datasets due to noise, acquisition artifacts, or natural anatomical geometry of structures. By operating in the spectral domain, these methods can provide a compact and informative representation of shapes, facilitating more efficient and robust analysis. Despite the potential of spectral approaches, there remains a need for a comprehensive pipeline that can efficiently perform all steps of statistical shape analysis on large volumes of high resolution data, starting from the establishment of correspondences within the collection of shapes to the computation of meaningful statistical quantities. The establishment of such a workflow could impact the way statistical shape analysis is performed in the context of large shape collections with high resolution meshes, including the clinical field.

In this paper, we propose S4A, an innovative method that leverages spectral geometry processing to perform statistical shape analysis on dense shape collections with high efficiency. The proposed method operates in three main stages: (i) it finds correspondences between each pair of shapes in the collection; (ii) it defines a common latent space that encodes the deformations among the entire collection; (iii) it computes statistical quantities that highlight and measure the most representative variations within the input collection. This comprehensive approach ensures that S4A not only captures the detailed variations among shapes but does so with high accuracy and a level of computational efficiency that outperforms existing methods. Moreover, S4A represents an efficient building block for various tasks in shape analysis, from keypoint detection to segmentation transfer.

To summarize, our contributions are threefold. On one side, we introduce a novel pipeline that integrates spectral geometry processing techniques for the efficient analysis of dense and complex shape collections. At the same time, we provide an extensive evaluation of our method against existing approaches, demonstrating superior performance in terms of both accuracy and computational efficiency. Finally, we explore the practical applications of our method on real medical data, showing how S4A can provide valuable statistical insights into anatomical variability and facilitate the transfer of meaningful information, such as segmentation labels and anatomical landmarks.

2. Related work

2.1. Statistical shape analysis

Statistical Shape Analysis (SSA) applies mathematical and statistical methods to analyze and quantify the geometrical properties of shapes, for generating shape comparisons, averages, probability models, and hypothesis tests, that are particularly relevant in biomedical and anatomical contexts [HM09, MBT*23]. Central to SSA is shape modeling, which captures shapes in a quantitative framework, enabling the measurement of shape differences, estimating mean shapes (intuitively a purely algebraic template where functions from all shapes get averaged out [HAGO19]), and testing for variability and clustering. Tools for statistical shape analysis automate the process of establishing point correspondences across a set of shapes by solving an optimization problem that defines these correspondences. Once the shapes are aligned within a common coordinate system, techniques like principal component analysis (PCA) help uncover the key variations within shapes [SG02] and deformations between shapes are often studied using diffeomorphometry, which preserves smoothness in the transformation.

2.2. Spectral geometry processing

Given a 3D mesh, which consists of a connectivity structure and geometry associated with its vertices, the connectivity captures some of the correlation between the vertex positions in the space. One approach to extract this correlation is to express the coordinate functions x, y, and z as a linear combination of a small number of basis vectors. This framework, well described by Taubin [Tau23], forms the foundation of spectral geometry processing. Central to this approach is the Laplace-Beltrami operator, which allows the decomposition of the geometry of a shape into its spectral components via its eigenfunctions and eigenvalues, producing an orthogonal set of smooth basis functions ordered by frequency [Lev06]. Each eigenvalue acts as the frequency of the corresponding eigenfunction, with most of the geometric signal energy concentrated in low-frequency components [LZ10].

Shape correspondence The problem of shape correspondence has been a widely investigated research topic in recent years. For an in-depth treatment of this topic, we refer to [Sah20]. Our research is closely aligned with the functional map framework, initially introduced by [OBCS*12] . This framework leverages spectral geometry processing to address the correspondence problem efficiently. In this approach, each correspondence is represented as a compact linear map between real-valued functions defined on the surfaces of the shapes, known as a functional map. Harnessing this representation, the map can be efficiently solved using linear optimization techniques. A comprehensive overview of this area is provided in [OCB*17]. This seminal work inspired a full area of research, applying the concept in various contexts [EBC17, RCB*17, MMRC18, MMO*21], including machine learning applications [DSO20, MRMO20] and shape analysis tools

[HAGO19, HCO18]. Notably, [MRR*19] presents an efficient refinement method to enhance the accuracy of given functional map correspondences. The efficiency of the functional map framework offers significant advantages in scenarios requiring rapid data processing, and so it represents a fundamental block of our method.

2.3. Medical Data

State-of-the-art methods in computer graphics and computer vision generally do not target 3D medical data, which tend to be complex, dense, and yet not widespread. However, some works have attempted to address clinical challenges using spectral geometry and statistical shape analysis, as in [MMS*16] and [MBT*23]. The translation of these tools to the clinical context offers many advantages, including more accurate diagnosis through the detection of morphological changes, the ability to use computationally efficient algorithms on sparse manifolds, and the utilization of data that are less prone to vulnerability. Moreover, to encourage the translation of computer graphics and computer vision methods into the clinical context, a medical imaging dataset, namely MedShapeNet, is publicly available. This dataset provides an extensive collection of 3D shapes of various anatomical structures extracted from real imaging data of both pathological and healthy subjects [Li23].

2.4. Scalable methods

An important aspect of algorithms that impact both the graphic and the medical community is the need for scalable methods to process a large quantity of high-resolution data. Strongly related to our approach are the methods of [MO23] and [MBRM24]. Both methods propose pipelines to compute correspondences based on the functional map approach for high-resolution meshed. Moreover, [MBRM24] performs meshing of the surfaces, providing not only good correspondences but also triangulations able to preserve the geometric properties of the shapes. Our paper works in this direction, providing a pipeline that enables data analysis on shapes at very high-resolutions.

3. Background

This section introduces some background notions and the notation we adopt in the following.

3.1. Discrete surfaces

In this paper, we refer to shapes as 2-dimensional smooth manifolds embedded in $\mathbb{R}^3$, which we indicate as $\mathcal{M} \subset \mathbb{R}^3$. We refer to [dC92] for any details about the continuous representation of these objects. In the discrete setting, we represent $\mathcal{M}$ as a triangular mesh $\mathcal{M} = (V_\mathcal{M}, T_\mathcal{M})$, where $V_\mathcal{M}$ is the set of n vertices, $T_\mathcal{M}$ is the set of faces (*i.e.*, $t_{ijk} \in T_\mathcal{M}$ if and only if there exists a triangle connecting the vertices $v_i, v_j, v_k \in V_\mathcal{M}$). We store the 3D coordinates of the vertices in $V_\mathcal{M}$ in a matrix $X_\mathcal{M} \in \mathbb{R}^{n\times 3}$. Each row of $X_\mathcal{M}$ corresponds to the position in the 3D space of a vertex of $\mathcal{M}$.

3.2. Shape matching

The input of a shape-matching task is usually composed of two discrete meshes $\mathcal{M}$ and $\mathcal{N}$. Assuming an existing unknown correspondence $T : V_\mathcal{N} \rightarrow V_\mathcal{M}$ between $\mathcal{M}$ and $\mathcal{N}$ exists, the goal of shape matching is, given $\mathcal{M}$ and $\mathcal{N}$, to estimate this correspondence. The estimated map $\bar{T}$ has to be as close as possible to T, which means that $\bar{T}$ should assign to each vertex $y \in V_\mathcal{N}$ a vertex on $\mathcal{M}$ that is geodesically as close as possible to the one associated by T. We represent a correspondence either as a vector of vertex indices of size $n_\mathcal{N}$ or as an $n_\mathcal{N} \times n_\mathcal{M}$ matrix Π such that $\Pi_{ij} = 1$ if $T(v_i) = v_j$ and 0 otherwise.

Some shape-matching pipelines require a set of input landmarks, meaning couples of points $(y \in V_\mathcal{N}, x \in V_\mathcal{M})$ in correspondences, such that $T(y) = x$.

3.3. Functional maps

Given a pair of shapes $\mathcal{M}$ and $\mathcal{N}$ and a ground truth correspondences $T : V_\mathcal{N} \rightarrow V_\mathcal{M}$, the functional map framework [OBCS*12], instead of estimating T directly, searches for a correspondence among functions defined on $\mathcal{M}$ and $\mathcal{N}$. In the discrete setting, a real-valued function f defined on the surface of the shape $\mathcal{M}$ is given by a vector that associates to each vertex $x \in V_\mathcal{M}$ a value $f(x) \in \mathbb{R}$. The space of such a function is called $\mathcal{F}(\mathcal{M}, \mathbb{R})$.

Functional maps build upon the observation that a point-wise map T induces a *linear* operator $T_F : \mathcal{F}(\mathcal{M}, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{N}, \mathbb{R})$ that maps functions defined on $\mathcal{M}$ to functions defined on $\mathcal{N}$ via the composition:

$$T_F(f) = f \circ T\,, \qquad \forall f \in \mathcal{F}(\mathcal{M}, \mathbb{R})\,. \tag{1}$$

Given a set of orthonormal functions, namely a basis for the functional spaces $\Phi = \{\phi\}_i$ and $\Psi = \{\psi\}_j$ for $\mathcal{F}(\mathcal{M}, \mathbb{R})$ and $\mathcal{F}(\mathcal{N}, \mathbb{R})$, respectively, we can write

$$\begin{aligned} g = T_F(f) = T_F\left(\sum_i a_i \phi_i\right) = \sum_i a_i T_F(\phi_i) = \\ = \sum_i a_i \sum_j c_{ji} \psi_j = \sum_{ji} a_i c_{ji} \psi_j = \sum_j b_j \psi_j\,, \end{aligned}$$

where $\vec{a}$ and $\vec{b}$ are the coefficients of f and g for the basis Φ and Ψ respectively. The c_{ji} is computed as the projection of $T_F(\phi_i)$ on ψ_j and depends only on T_F and the two bases. Therefore T_F can be compactly represented by the matrix C such that $\vec{b} = C \cdot \vec{a}$.

In practice, only the first k atoms of the bases are used, truncating the series after k coefficients. Notably, k is independent of the number of vertices n in the meshes and usually $k \ll n$. Thus, matching two shapes in the functional map framework involves estimating a matrix C of size $k \times k$. The optimization consists of representing features, such as landmarks and descriptors as functions on $\mathcal{M}$ and $\mathcal{N}$, finding the functional map C that best preserves these functional alignment constraints in the least squares sense, we refer to [OCB*17] for a detailed description on the optimization of the functional map. Then, once the *functional correspondence* has been estimated, we can extract the corresponding point-wise map.

When truncating the basis for $\mathcal{F}(\mathcal{M}, \mathbb{R})$ to size k, we store it in a matrix $\Phi_\mathcal{M}$, where each column is a basis atom represented as a vector of real values, making $\Phi_{\mathcal{M}}$ of size $n \times k$. Each row of $\Phi_{\mathcal{M}}$ corresponds to the coefficients in the basis $\Phi_{\mathcal{M}}$ of a Delta function centered at x. Therefore, $\Phi_{\mathcal{M}}^{T}$ contains the coefficients of all Delta functions of $\mathcal{M}$ as column vectors.

Functional maps estimate a mapping between functional spaces built over $\mathcal{M}$ and $\mathcal{N}$, and this needs to be converted into a point-wise map $\bar{T} : V_{\mathcal{N}} \to V_{\mathcal{M}}$. A simple solution, proposed in [OBCS*12], consists in finding, for each column of $\Phi_{\mathcal{N}}^{T}$, the nearest neighbor in the columns of $C\Phi_{\mathcal{M}}^{T}$, such that

$$T_{\mathcal{N}\mathcal{M}} = KnnSearch(\Phi_{\mathcal{N}}, \Phi_{\mathcal{M}} C^{\top}) \,. \tag{2}$$

The full functional map pipeline for shape matching, as described in [OBCS*12], involves three steps: (1) selecting a truncated basis of size k for each mesh, (2) determining the matrix $C \in \mathbb{R}^{k \times k}$ that optimally preserves certain functional constraints, and (3) converting the functional map C into a dense point-wise map $\bar{T}$.

The functional map representation of the correspondence helps minimize the complexity of the problem, but it creates a trade-off between the optimization's complexity and the solution's accuracy. diverse methods have the goal of increasing the solution accuracy, starting from a low-dimensional initial map. The ZoomOut refinement [MRR*19] for instance, performs spectral upsampling, iteratively converting the functional map to its point-to-point counterpart, while increasing the basis dimension.

3.4. Collection of Shapes

The application of the functional map approach can be extended from pairs to collections of shapes, which we will indicate as $\{\mathcal{M}_i\}_{i=0}^{N}$. A notable work in this direction is the Consistent ZoomOut algorithm [HRWO20]. In this work, the authors perform spectral refinement on a collection of shapes exploiting the method from [MRR*19], building a functional map network (FMN), in which the nodes are the shapes and the edges are represented by the functional maps. In this way, it is possible to enforce cycle consistency on the maps and refine the correspondences taking into account all the shapes of a collection.

We stress that, to perform the refinement stage, the algorithm computes a *consistent latent basis* for each shape $\{Y_i\}_{i=0}^{N}$ and that we can retrieve a pointwise correspondence by performing a nearest neighbor search between consistent latent bases, as

$$T_{\mathcal{M}_i \mathcal{M}_j} = KnnSearch(Y_i, Y_j) \,. \tag{3}$$

3.5. Shape difference operator

A useful notion in this context is the shape difference operators proposed in [ROA*13, HRA*19]. This operator has the goal to represent the intrinsic distortion between shapes. Given a couple of shapes $\mathcal{M}, \mathcal{N}$ and a given map $C_{\mathcal{M},\mathcal{N}}$ [HAGO19], defines the area-based shape difference operator as $D = C_{\mathcal{M}\mathcal{N}}^{T} C_{\mathcal{M}\mathcal{N}}$. Considering a collection of shapes and the introduction of the consistent latent basis Y_i, we can rewrite the difference operator as $D_i = Y_i^{T} Y_i$. This operator has some useful properties since it allows to represent an arbitrary shape $\mathcal{M}_i$ in the collection via a fully-intrinsic compact matrix.

3.6. Statistical Shape analysis

As well explained by [SG02, HM09], there are numerous potential applications for modeling the shape variation. Considering a collection of shapes with a certain degree of correlation between the faces and vertices of each mesh, we can assume that statistical methods can capture and explain the variation within the collection. Consequently, a shape representation can be formulated to account for the correlations between points. For instance, if some points exhibit minimal variation, encoded as deformation in 3D space close to zero, then we can consider them fully correlated. This observation enables us to reduce the dimensionality of the data to explain deformations.

Principal Component Analysis In this paper, we utilize the PCA as a tool for performing statistical shape analysis, derived using linear algebra, to achieve dimensionality reduction. As done in [ROA*13] we first vectorize the area based and conformal shape difference matrices, then we apply PCA. This dimensionality reduction technique linearly transforms the data in a new coordinate system, where the new directions, also called principal components, capturing the largest variation in the data. Specifically, in our task, each principal component captures the largest deformation among the brain shapes. It is important to remark that these principal components are orthogonal to each other, representing a portion of the total variation. By retaining only the first L principal components, it is possible to represent the original data in a lower dimensional space.

4. Proposed method

The goal of our method is to perform statistical shape analysis on a collection of high-resolution geometric data $\{\mathcal{M}_i\}_{i=0}^{N}$, proving that it's possible in a reasonable amount of time utilizing a low-resolution strategy.

Our pipeline takes inspiration from the method of [MBT*23], in which the authors conduct a statistical shape analysis on medical data collection in a functional map-based pipeline. This work shows the efficacy of spectral methods in conducting statistical analysis of geometric deformations within a collection of shapes. However, this method does not scale well to high-resolution data: this problem led us to build S4A.

4.1. Pairwise Scalable Registration

To facilitate statistical shape analysis, the first step of our method requires computing non-rigid correspondences between each pair of shapes within the collection. Despite being a well-studied problem, accurately and efficiently matching two high-resolution shapes is not trivial.

As mentioned in Section 3, an efficient and flexible way to perform correspondences between shapes is the Functional Map approach. Nevertheless, when applying this method to real-world high-resolution shapes, the computation of the needed geometric operators can be unstable and unreliable, since these data often result in irregular or imperfect meshes. Moreover, they often do not satisfy the manifoldness assumption of the shapes on which Functional maps rely. Recently Rematching [MBRM24] and Scalable Fmaps [MO23] propose two methods to tackle this problem.

Our pipeline relies on the work of [MBRM24], in which each original shape $\mathcal{M}$ is remeshed to a low-resolution version $\tilde{\mathcal{M}}$, which preserves the key geometrical features of the original. Along with these remeshed shapes, the method computes an operator $U \in \mathbb{R}^{n_{\mathcal{M}} \times n_{\tilde{\mathcal{M}}}}$, This operator, which we will refer to as the *Projection operator*, has the role of projecting the discretization of each scalar function $\tilde{f} \in \mathcal{F}(\tilde{\mathcal{M}}_i, \mathbb{R})$ back to its high-resolution counterpart $f \in \mathcal{F}(\mathcal{M}_i, \mathbb{R})$. This allows us to reduce the mesh size by over 90% without losing any essential geometric information, as shown in the result section.

Leveraging the remeshed shape and this operator, we can compute a functional map C between the remeshed version of two shapes, which is then converted into a high-resolution point-to-point correspondence T. This latter task is achieved performing nearest neighbor between the columns of the projected basis, as

$$T_{\mathcal{M}\mathcal{N}} = KnnSearch(U_{\mathcal{N}}\phi_{\mathcal{N}}, (U_{\mathcal{M}}(\phi_{\mathcal{M}}C^{\top}))) \,. \tag{4}$$

This approach allows to accelerate the computation of correspondences between real-world dense shapes.

All the steps of our pipeline can also be easily adapted to the method of Scalable Fmaps [MO23]. Instead of a remeshed version of the shape, Scalable Fmaps provides a pipeline to compute a low-resolution Functional map and a projection operator similar to the one from Rematching. We provide an extensive comparison of the two methods in Section 5.

On these premises, our registration pipeline has the goal of computing accurate and efficient maps, merging both rigid and non-rigid registration techniques. It can be summarized as follows:

1. Remesh the high-resolution input shapes into lower size meshes $\tilde{M},\tilde{\mathcal{N}}$
2. Rigidly align the two shapes via the iterative closest point (ICP) algorithm.
3. Sample 10 points from the source shape and extract 10 corresponding points on the target shape, performing nearest neighbors among the 3D coordinates. These pairs of points will serve as landmarks to initialize the functional map.
4. Optimize for a Functional map $C_{\tilde{\mathcal{M}}\tilde{\mathcal{N}}}$ of dimension $k \times k$

Performing this pipeline, we can efficiently initialize a functional map between a pair of shapes. We stress that this functional map, despite being computed on the low-resolution pair of shapes, is a good functional map also for the high-resolution shapes. In Figure 2 we show an example of an initial functional map computed with this pipeline and their typical behavior with quasi-spherical and symmetrical shapes.

Afterwards, the map can be transformed into a point-to-point correspondence as follows:

1. we apply the ZoomOut algorithm [MRR*19] to upsample the map until $K > k$ efficiently;
2. we convert the obtained map to a point-to-point correspondence between the two original shapes using the projection operator, as explained before.

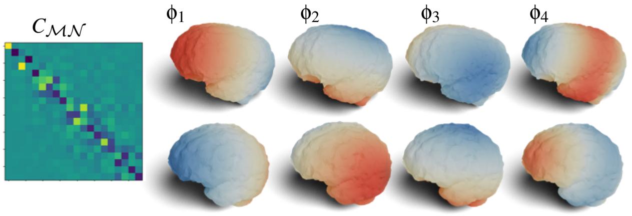

Figure 2: *Left: an example of a Functional map between two remeshed shapes from our dataset. Right: the first 4 non-constant bases are visualized as functions over the shape surface. We note that the second and the third basis are switched between the two shapes. This behavior is typical in cases in which the shape is symmetrical, and it is represented in the first 4x4 submatrix of the functional map.*

With these steps, we can obtain reliable high-resolution point-to-point correspondences. Moreover, the first part of the pipeline can serve to initialize functional maps to perform spectral statistical shape analysis, as we will describe in the next section.

4.2. Scalable statistical shape Analysis

Mean shape computation. From the previous section, we know how to extract pairwise low-frequency functional maps between the pairs of a collection of shapes. Given a collection of high-dimensional shapes, we can instantiate a Functional Map network by computing the map on the edges as described before. In this way, we can follow [MBT*23] to perform statistical shape analysis. Indeed we can perform Consistent ZoomOut [HRWO20] to upsample the functional maps, enforcing cycle consistency within the maps of the collection and obtaining a set of maps of dimension $K \times K$.

We have found that a naive application of Consistent ZoomOut leads to unsatisfactory refinement. This behavior is caused by the initial maps, which are too noisy to return good refinement. Better results can be obtained by performing a step of the ZoomOut algorithm in advance. Projecting the initial maps to the space of Maps which represents a permutation helps the refinement stage. Once we get high-frequency, cycle-consistent maps, we can perform our analysis. The first step is to find the *median shape* of the collection. Thanks to the refinement step of [HRWO20], we can compute a *limit shape* in the spectral domain. However, since this does not represent a real shape in the collection, we can select the shape that is the most intrinsically similar to this latent mean shape. In order to do so [MBT*23] introduces a metric to measure the intrinsic dissimilarity of each shape of the collection between this latent mean shape, which is:

$$E_{intrinsic}(\mathcal{M}_i) = \|D_i - I\|_{H_0^2} \,, \tag{5}$$

where D_i is the shape difference operator [HAGO19], I is the identity matrix of dimension K and $\|\cdot\|_{H_0^2}$ the norm in the Sobolev space [ROA*13]. Thus, we select as the median shape the one that minimizes this energy, which we indicate as $\bar{\mathcal{M}}$.

We stress that all these steps are performed on the functional space, and so they do not depend on the discretization of the original shape.

Shape Deformation Analysis. After identifying the median shape, we use it as the most representative shape of our collection. We consider this shape as a template shape for our collection. In order to perform statistical analysis, we register all the shapes of our collection to this *template*, and compare the deformation with respect to it.

To achieve this, we must extract point-to-point correspondences between the shapes. In [MBT*23] this correspondence is obtained by

$$T_{\bar{\mathcal{M}}\mathcal{M}_i} = KnnSearch(Y_{\bar{\mathcal{M}}}, Y_{\mathcal{M}_i}), \tag{6}$$

where $Y_{\mathcal{M}_i}$ are the consistent Latent basis that we extract from the network, as described in Equation (3)

We remark that, since we computed spectral basis from the remeshed shape, this point-to-point map is a map between the remeshed version of the shapes of the collections. We can use two strategies to retrieve a correspondence between the original high-dimensional shape.

- The first strategy is to compute low-resolution correspondences using the [MBT*23] original framework, using equation (3), then compute the respective pairwise functional map, and then convert it to the high-resolution correspondence using the projection operator on the basis, as in equation (4). We will refer to this approach as the *standard* one.
- The second possible approach is to use the projection operator directly on the consistent latent basis Y_i, to compute

  $$T_{\bar{\mathcal{M}}\mathcal{M}_i} = KnnSearch(U_{\bar{\mathcal{M}}}Y_{\bar{\mathcal{M}}}, U_{\mathcal{M}_i}Y_{\mathcal{M}_i}) \tag{7}$$

  Indeed, the consistent latent basis can be seen as scalar functions defined on the shape's surface. This second approach is much more efficient than the first one since it requires less nearest neighbor search in a high-dimensional space. We will refer to this approach as *ours*.

Having the pairwise correspondences between the median shape and the other shape of the collection, we can define statistics that represent the deformation of each shape of the collection. For each $\mathcal{M}_i$ we compute

$$d_i(x) = x - T^{H}_{\bar{\mathcal{M}}\mathcal{M}_i}(x), \quad \forall x \in \bar{\mathcal{M}}, \tag{8}$$

which is the displacement field between the point of the mean shape and the corresponding points on the target shape.

Projecting this displacement on the spectrum of the mean shape via $U_{\mathcal{M}}\phi_{\mathcal{M}}(U_{\mathcal{M}}\phi_{\mathcal{M}})^{\dagger}d_i$ we get a smooth version of the statistic and we can perform shape analysis on this feature. In the result section we will how to use this statistic to perform PCA and other analysis on the shape variation within the dataset.

4.3. Other applications

Other than being a good tool for analyzing deformations in a collection of shapes, our pipeline can also be used to efficiently register the entire collection to a fixed template. Indeed, if we have access to a template shape $\mathcal{M}_T$, matching this to the median shape of the collection can lead to a template to mesh registration of the full dataset. Having a set of shapes registered to a fixed template can be beneficial to transfer information, such as segmentation, between shapes. Moreover, our pipeline can be exploited to produce new training data reducing the data scarcity for training, which is common problem regarding 3D medical data. The segmentation and anatomical landmarks are just example of supervised data that can be extended to a given collection through the registration step. In the future, additional information, such as curves and displacements, could be transferred using the same process.

5. Results

5.1. Experimental Settings

Dataset. The present study relies on a collection of N = 51 triangular meshes of tumored brains from the MedShapeNet dataset. These data represent the outer membrane, namely the dura mater, of brains, where some shapes exhibit alterations due to real tumors and others due to lesions synthetically generated using generative adversarial networks. This shape collection presents several inherent challenges: the meshes are dense, with the average number of vertices across shapes exceeding 100k, some shapes have disconnected components, and the majority exhibit artifacts, primarily due to the acquisition and reconstruction process. At the same time, the dataset provides shapes having similar a 3D embedding. In the shape matching setting, this framework represents a facilitated condition that allows a proficient registration even with simple, extrinsic methods, such as ICP. More challenging conditions taking into account non-similarity among shapes embedding will be considered in future work.

Implementation Details. We base our implementation on the PyFM library and it is based on CPU computations. The resolution of the low-dimensional meshes has been set around 6000 vertices. The dimension of the functional Map is set to $k = 20$ for the initial maps, and to $K = 100$ for refined maps. The functional map network can have, to make it a clique, a maximum number of edges of 1275. To optimize computational cost, we reduced the number of edges to 500, while still ensuring that the graph is connected. This choice of parameters is the same for all the experiments. S4A is executed on a machine equipped with an AMD Ryzen 7 5800H processor (3.20 GHz), 16 GB RAM, and running Windows 11 (64-bit).

Metrics. To compare the proposed method we consider some metrics to evaluate the accuracy of the estimated correspondences. The standard benchmarks for shape matching provide ground truth correspondence, to compare the estimated ones. However, in our dataset, as common with medical data, we do not have access to ground truth correspondences, and so we are going to perform an unsupervised analysis considering some reconstruction metrics. Given two shapes $\mathcal{M}, \mathcal{N}$, we consider the following metrics:

- Chamfer distance:

$$Chamfer(\mathcal{M},\mathcal{N}) = \frac{1}{n_{\mathcal{M}}}\sum_{p\in\mathcal{M}}\min_{q\in\mathcal{N}} d(p,q) + \frac{1}{n_{\mathcal{N}}}\sum_{q\in\mathcal{N}}\min_{p\in\mathcal{M}} d(q,p) \quad (9)$$

- Hausdorff distance:

$$Hausdorff(\mathcal{M},\mathcal{N}) = \max_{p\in\mathcal{M}}\min_{q\in\mathcal{N}} d(p,q) + \frac{1}{n_{\mathcal{N}}}\max_{q\in\mathcal{N}}\min_{p\in\mathcal{M}} d(q,p) \quad (10)$$

Where $d(q,p)$ is the Euclidean distance between the two.

These metrics compute the distance between the points of two shapes, and we apply them to $\mathcal{M}$ and the image $T_{\mathcal{N}\mathcal{M}}(\mathcal{N})$ of the estimated map.

Furthermore, we consider the Dirichlet energy that evaluates the geometrical smoothness of the estimated map. Given a function $f \in \mathcal{F}(\mathcal{M},\mathbb{R})$, the Laplacian operator $L_{\mathcal{M}}$ and the mass matrix $A_{\mathcal{M}}$, the Dirichlet energy of f is defined as

$$Dirichlet(f) = \mathbf{f}^T \mathbf{A}_{\mathcal{M}} \mathbf{L}_{\mathcal{M}} \mathbf{f}. \quad (11)$$

In our experiment, we compute the sum of the energy of each coordinate of the correspondence image $T_{\mathcal{M}\mathcal{N}}(\mathcal{M})$.

5.2. Pair of High-Resolution Shapes

We selected a shape from our dataset as a reference, paired it with five other shapes being representative of the dataset and perform shape registration using five methods. We measured and recorded the computational time required, some reconstruction metrics, and qualitative results to evaluate their performance comprehensively.

The methods under comparison include k-nearest neighbors (kNN), ZO, FMAPS combined with ZoomOut (FMAP+ZoomOut), Scalable Functional Map (SCALABLE), and Rematching (RE-MATCH). The computational time is recorded for each registration method to understand the computational efficiency, while reconstruction metrics such as alignment accuracy and structural consistency are used to quantify the quality of the registration. Additionally, qualitative assessments are performed to provide a visual and interpretative evaluation of the registration results. The results are summarized in Table 1.

It is important to stress that the application of all pipelines require a preprocessing step for cleaning meshes, avoiding, for example, disconnected components and non-manifoldness. This is explicitly done for all the methods, while in our pipeline this is part of the remeshing step, in which the shape is remeshed to a fixed lower number of vertices, 6000 in our case. For fair comparison of the whole pipeline, we report the overall computational time, considering the shape loading and alignment part, the optimization for the map computation and, if available, the refinement of the map, and rescaling step in case of scalable methods. This steps comparison is outlined in Figure 3, with the exclusion of the negligible shape loading and cleaning part. Moreover, the reported runtimes consider a CPU implementation.

From the analysis of the results, we can derive some insights and comparisons of the above approaches. K-Nearest Neighbors (kNN) does not perform very well for shape registration in this context. This method operates solely in the 3D ambient space, which leads to a significant loss of intrinsic expressivity. Consequently, the registered shapes often lack the necessary detail and accuracy, making kNN an unsuitable choice for high-fidelity shape registration tasks, even though it requires very little computational costs. On the other hand, ZoomOut and FMAPS+ZoomOut operate in the intrinsic domain, however they both exhibit high computational costs, primarily due to the necessity of computing the Laplace-Beltrami (LB) basis on high-resolution meshes. While these methods can potentially yield accurate results, their computational expense makes them impractical for large-scale or time-sensitive applications, in particular in case of wide shapes collections. The Scalable method strikes a balance between efficiency and effectiveness, providing reliable results, comparable to the two aforementioned methods, but without incurring excessive computational costs. Rematching appears to be the method outperforming in both registration accuracy and runtime, relying on a fast preprocessing in which shapes are remeshed and basis computed on low resolution meshes. Its ability to deliver accurate registration in less than half of a minute makes it suitable for practical applications where both time and accuracy are critical considerations.

METHOD	TIME	Chamfer	Hausdorff	Dirichlet
kNN	9.51	0.22	0.41	0.16
ZO	301.97	0.0035	0.0341	1.3
FMAP+ZO	292	0.0031	0.026	0.93
SCALABLE FM	159.9	0.0035	0.028	0.85
RE-MATCH	20	0.0023	0.01	4.5

Table 1: *Processing time, expressed in seconds, and reconstructive metrics of the five methods compared. Dirichlet energy is reported in scale* 10^{12}.

To sum up, the analysis reveals that while kNN is inadequate for detailed shape registration, methods like ZoomOut and FMAPS+ZoomOut, though accurate, are prohibitively expensive in terms of computation. In contrast, Scalable and Rematching methods, working with a low resolution embedding and then scaling back to the original resolution, offer a promising balance, achieving efficient and accurate shape registration suitable for various practical scenarios, with Rematching being the most efficient solution.

5.3. High Resolution Shape Collections Registration

In this section, we compare the performance of the proposed method's pipeline using both the Rematching [MBRM24] and Scalable [MO23] variants. The goal of this analysis is to determine the most effective approach for high-resolution shape registration and to evaluate how each variant integrates high-resolution registration with statistical spectral shape analysis, as outlined in the Method section 4.

Breaking down the computational time,we subdivide the method in three blocks. The first block of our pipeline is the computation of the basis an the functional map network initialization,This step of the pipeline taken approximately 18 minutes for the Rematching approach and 20 minutes for the Scalable method. Next, we perform Consistent ZoomOut refinement, and the process takes 22 minutes with rematching and 27 minutes with Scalable. At this

	Chamfer		Time(min)	
	Standard	Ours	Standard	Ours
SCALABLE	0.0230	**0.0212**	60.45	13.59
RE-MATCH	0.0234	**0.0212**	57.38	**12.03**

Table 2: *Comparison of the Scalable and Rematching methods in terms of Chamfer Distance and computational time. The two methods are evaluated at high resolution, thus in the table is provided also a comparison between the two pipelines to extend the pipeline at the original shape.*

point, after the identification of the mean shape, we can compute the point-to-point correspondences between the mean shapeand the other shapes of the collection. It is important to highlight that both methods converge to the same median shape.

Once a satisfactory low resolution point to point correspondence is obtained, the subsequent task addressed to our pipeline is to correctly extend the functional mapping (matching) to the same brains but at high resolution. At this point, we calculate the average Chamfer distance between each mesh and the template. The results of our comparative analysis are summarized in Table 2. The table provides a detailed overview of the performance metrics for both the Rematching and Scalable variants, including computational time and registration accuracy.

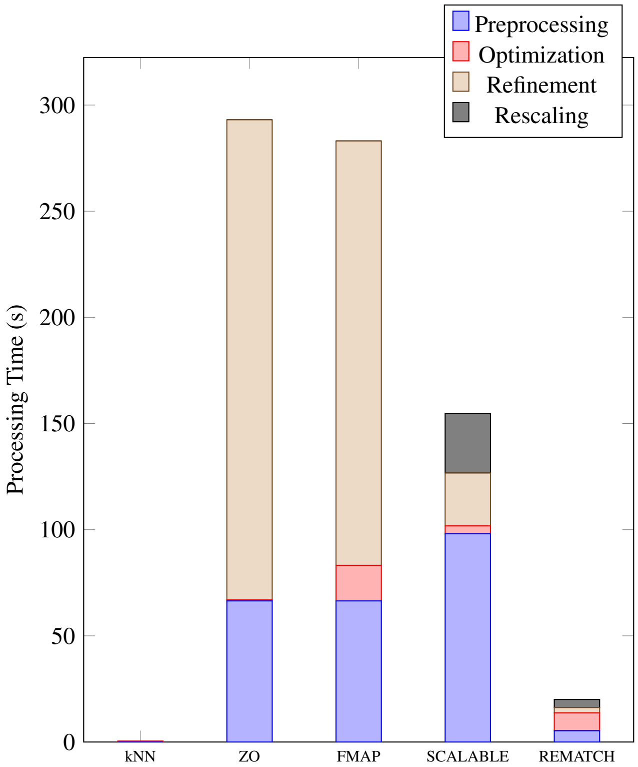

Figure 3: *Visualization of runtime expressed in seconds for each principal step composing the pipelines of the five compared methods.*

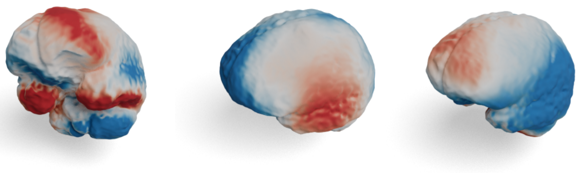

Figure 4: *Visualization of the deformation represented by the first three principal components. Blue areas indicate expansion, while red areas signify compression.*

From the results presented in Table 2, we can notice that both methods achieve comparable levels of accuracy in the final registration. However our method requires fewer computational resources and consequently, resulting in a significantly reduced runtime to achieve the same outcome. This is coherent with our expectations, indeed it makes just one KNN computation, with a smaller number of parameters. In particular, this efficiency is consistent in both the approaches, Scalable and in Rematching. The total runtime of the pipeline for the Rematching technique is about 52 minutes with our approach, compared to 98 minutes using the standard method. The Scalable technique are required 60 minutes for our approach and 107 minutes for the standard method. Overall, applying our methods improves runtime efficency by about 45%.

Moreover, it is important to note that the computational time of our method scales linearly with the number of nodes and edges of the graph. Therefore, for future work, it would be beneficial to fine-tune these parameters in order to further speed up the whole pipeline.

6. Applications

6.1. Analysis of the variation

Using the median shape $\bar{\mathcal{M}}$ as template, we generated for each shape $\mathcal{M}_i$ a vertex-wise deformation field $d^{(i)}$ deforming $\bar{\mathcal{M}}$ into $\mathcal{M}_i$. Exploiting the dissimilarity field, calculated as explained in section 4, we projected it into the truncated Laplacian basis of dimention K. Using this deformation field as descriptor for a shape, we perform PCA to extract the principal components $D_i^{p_{pca}}$ that describe the deformations. These can be interpreted as the deformations of the median shape, in the three dimensional space, and in Figure 4, we can visualize the first three components over $\bar{\mathcal{M}}$.

As we can see from the Figure 4, the first principal component highlights the lower region of the brain as the area with the greatest variability, meaning more deformation, among the data. This result is reflective of the dataset's nature utilized and the data acquisition process and shape reconstruction method. Indeed, during medical imaging process, it's not uncommon for parts of the organ to be incompletely captured, and "lacking" portions are mainly due to machine settings (e.g. field of view). The second principal component is more significant, as it captures the brain's expansion along a circular band around the transverse axis. Simultaneously, it also accounts for a symmetric compression on both temporal lobes.

Similarly, the third principal component describes a positive brain deformation along the band around the sagittal axis and a negative one on the frontal lobe and the occipital lobe.

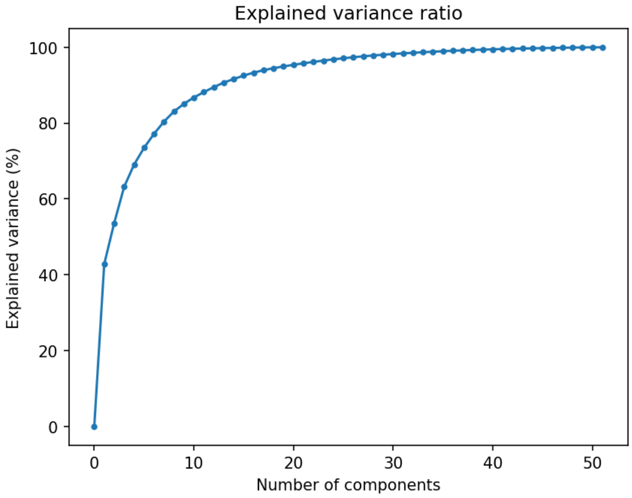

Figure 5: *Cumulative variance explained by each principal component at high resolution.*

Moreover, as shown in Figure 5, by considering the first ten principal components, we are able to explain almost the 86% of the deformations of the data at high resolution an about the 80% at low. This type of analysis may be used to perform relevant medical statistical analysis over a large dataset for clinical diagnosis along with a clear representation of the classification outcomes, such as the recognition and quantification of deformation over the brain shape due to a pathology.

6.2. Information Transfer

As stated in section 4, our pipeline can be used to register a collection of shapes to a template surface. This solution can be leveraged to transfer information between meshes, which is one of the key applications of maps between shapes. We proceeded in two different ways: 1) registering the template to the shapes pairwise; 2) registering the template to the mean shape found during shape collection registration. While using the template as the mean shapes can be more efficient, depending on the application, it can be more useful to identify a new more informative template within the collection.

Figure 6 and 7 refer to the transfer of information and segmentation, and the transfer of landmarks, respectively. The source shape (on the left) is a template representing a population-averaged brain model. To apply our method to a real case of possible medical interest, the considered template is a high-resolution shape, with more than 30k vertices, and comes with a fine-grained pre-segmentation providing a comprehensive map of the brain's anatomical structure, with 40 labels for each hemisphere. The template key-points have been located manually on the shape following expert's guidelines and clinical needs, as transferring this type of information has useful applications in obtaining meaningful metrics.

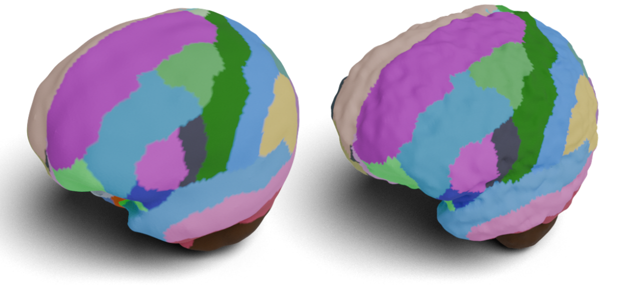

Figure 6: *Segmentation transfer from the template (left) to the target shape (right).*

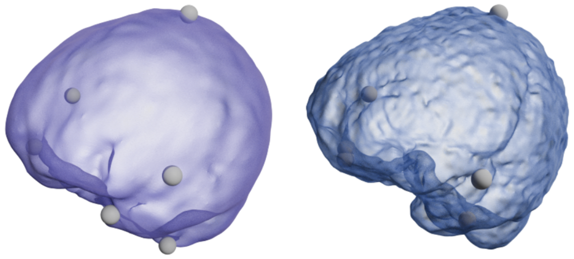

Figure 7: *Key-points transfer from the template (left) to the target shape (right).*

7. Conclusions and Future Works

In this paper, we introduced S4A, an efficient method for statistical shape analysis based on spectral geometry processing, addressing the challenges of dense mesh representations in medical data. S4A establishes correspondences between shape pairs, defines a common latent space, and computes statistical quantities, capturing representative variations while maintaining computational efficiency. Our results demonstrate that S4A outperforms existing methods in accuracy and processing time, proving its value in analyzing complex shape collections. But, despite its potentiality, it is important to note that our approach is limited to applications on meshes due to the underlying requirements of the rematching algorithm, which necessitates a mesh structure. This restricts our analysis to mesh-based structures and does not allow for other types of 3D representations, such as point clouds and implicit surfaces

Up to now, S4A has been tested on a specific and quite small shape collection from the medical world, which limits the evaluation of the robustness of our method. Looking ahead, we plan to enhance S4A's scalability and robustness for larger datasets, and test its accuracy on benchmarks like the SHREC19 dataset [MMR*19] and to an extension of the considered brain dataset. While this method has primarily been applied to genus-0 shapes, we anticipate it can be extended to shapes with varying topologies, as with the functional maps pipeline. We believe this approach should be adaptable to models of different genera, though additional experiments would be required to confirm its effectiveness. Additionally, S4A can be a foundation for studies and analyses on high-resolution shape collection. Thanks to its ability to transfer information, this pipeline can be exploited to transfer segmentation labels and landmarks onto unlabeled data, addressing both the lack of labeled data and the challenges involved in obtaining them. In accordance with that, a key future development is that of exploiting S4A for building a large dataset to train machine learning models on medical data for automatic segmentation or keypoint detection using feature extractors such as PointNet [QSMG16] or DiffusionNet [SACO22]. Overall, S4A has significant potential to advance research in medical and other scientific fields requiring detailed shape analysis.

By combining advanced spectral techniques with a carefully designed analysis pipeline, S4A represents a significant advancement in the field of statistical shape analysis, particularly for applications involving dense and complex medical data. Our work opens new possibilities for efficient and detailed analysis of shape deformations, with potential applications in medical research, diagnosis, and treatment planning.

Acknowledgments

This work was funded by the European Union – Next Generation EU within the project NRPP M4C2, Investment 1.,3 DD. 341 - 15 March 2022 – FAIR – Future Artificial Intelligence Research – Spoke 4 - PE00000013 - D53C22002380006. Part of this work wa supported by the MUR for REGAINS, the Department of Excellence DISCo at the University of Milano-Bicocca, the PRIN project GEOPRIDE Prot. 2022-NAZ-0115, CUP H53D23003400001, and by the NVIDIA Corporation with the RTX A5000 GPUs granted through the Academic Hardware Grant Program to the University of Milano-Bicocca for the project "Learned representations for implicit binary operations on real-world 2D-3D data".

References

[dC92] do Carmo M. P.: Riemannian geometry. *Bikhausen, Boston* (1992). 3

[DSO20] Donati N., Sharma A., Ovsjanikov M.: Deep geometric functional maps: Robust feature learning for shape correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8592–8601.

[EBC17] Ezuz D., Ben-Chen M.: Deblurring and denoising of maps between shapes. *Computer Graphics Forum 36*, 5 (2017), 165–174. 2

[HAGO19] Huang R., Achlioptas P., Guibas L., Ovsjanikov M.: Limit shapes – a tool for understanding shape differences and variability in 3d model collections. *Computer Graphics Forum 38*, 5 (2019), 187–202. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13799, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13799, doi:https://doi.org/10.1111/cgf.13799. 2, 3, 4, 5

[HCO18] Huang R., Chazal F., Ovsjanikov M.: On the stability of functional maps and shape difference operators. *Computer Graphics Forum 37*, 1 (2018), 145–158. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13238, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13238, doi:https://doi.org/10.1111/cgf.13238. 3

[HM09] Heimann T., Meinzer H.-P.: Statistical shape models for 3d medical image segmentation: A review. *Medical Image Analysis 13*, 4 (2009), 543–563. URL: https://www.sciencedirect.com/science/article/pii/S1361841509000425, doi:https://doi.org/10.1016/j.media.2009.05.004. 2, 4

[HRA*19] Huang R., Rakotosaona M.-J., Achlioptas P., Guibas L., Ovsjanikov M.: Operatornet: Recovering 3d shapes from difference operators, 2019. URL: https://arxiv.org/abs/1904.10754, arXiv:1904.10754. 4

[HRWO20] Huang R., Ren J., Wonka P., Ovsjanikov M.: Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum 39*, 5 (2020), 265–278. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.14084, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.14084, doi:https://doi.org/10.1111/cgf.14084. 4, 5

[Lev06] Levy B.: Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry. In *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)* (2006), pp. 13–13. 2

[Li23] Li J.: Medshapenet – a large-scale dataset of 3d medical shapes for computer vision, 2023. URL: https://arxiv.org/abs/2308.16139, arXiv:2308.16139. 3

[LZ10] Lévy B., Zhang H.: Spectral mesh processing. In *ACM SIGGRAPH 2010 Courses*. 2010, pp. 1–312. 2

[MBRM24] Maggioli F., Baieri D., Rodolà E., Melzi S.: Rematching: Low-resolution representations for scalable shape correspondence. In *Proc. ECCV* (2024). 3, 5, 7

[MBT*23] Magnet R., Bloch K., Taverne M., Melzi S., Geoffroy M., Khonsari R. H., Ovsjanikov M.: Assessing craniofacial growth and form without landmarks: A new automatic approach based on spectral methods. *Journal of Morphology 284*, 8 (2023), e21609. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jmor.21609, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jmor.21609, doi:https://doi.org/10.1002/jmor.21609. 2, 3, 4, 5, 6

[MMO*21] Maggioli F., Melzi S., Ovsjanikov M., Bronstein M. M., Rodolà E.: Orthogonalized fourier polynomials for signal approximation and transfer. In *Computer Graphics Forum* (2021), vol. 40, Wiley Online Library, pp. 435–447. 2

[MMR*19] Melzi S., Marin R., Rodolà E., Castellani U., Ren J., Poulenard A., Wonka P., Ovsjanikov M.: Shrec 2019: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval* (2019), vol. 7, The Eurographics Association, p. 3. 9

[MMRC18] Marin R., Melzi S., Rodolà E., Castellani U.: Farm: Functional automatic registration method for 3d human bodies, 2018. 2

[MMS*16] Melzi S., Mella A., Squarcina L., Bellani M., Perlini C., Ruggeri M., Altamura C. A., Brambilla P., Castellani U.: Functional maps for brain classification on spectral domain. *Spectral and Shape Analysis in Medical Imaging 10126* (2016), 25–36. 3

[MO23] Magnet R., Ovsjanikov M.: Scalable and efficient functional map computations on dense meshes. In *Computer Graphics Forum* (2023), vol. 42, Wiley Online Library, pp. 89–101. 3, 5, 7

[MRMO20] Marin R., Rakotosaona M.-J., Melzi S., Ovsjanikov M.: Correspondence learning via linearly-invariant embedding, 2020. arXiv:2010.13136. 2

[MRR*19] Melzi S., Ren J., Rodolà E., Sharma A., Wonka P., Ovsjanikov M.: Zoomout: spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG) 38*, 6 (2019), 1–14. 4, 5

[OBCS*12] Ovsjanikov M., Ben-Chen M., Solomon J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (ToG) 31*, 4 (2012), 1–11. 2, 3, 4

[OCB*17] OVSJANIKOV M., CORMAN E., BRONSTEIN M., RODOLÀ E., BEN-CHEN M., GUIBAS L., CHAZAL F., BRONSTEIN A.: Computing and processing correspondences with functional maps. In *ACM SIGGRAPH 2017 Courses* (2017), pp. 5:1–5:62. 2, 3

[QSMG16] QI C. R., SU H., MO K., GUIBAS L. J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593* (2016). 10

[RCB*17] RODOLÀ E., COSMO L., BRONSTEIN M. M., TORSELLO A., CREMERS D.: Partial functional correspondence. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 222–236. 2

[ROA*13] RUSTAMOV R. M., OVSJANIKOV M., AZENCOT O., BEN-CHEN M., CHAZAL F., GUIBAS L.: Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 1–12. 4, 5

[SACO22] SHARP N., ATTAIKI S., CRANE K., OVSJANIKOV M.: Diffusionnet: Discretization agnostic learning on surfaces, 2022. URL: https://arxiv.org/abs/2012.00888, arXiv:2012.00888. 10

[Sah20] SAHILLIOĞLU Y.: Recent advances in shape correspondence. *The Visual Computer 36*, 8 (2020), 1705–1721. 2

[SG02] STEGMANN M. B., GOMEZ D. D.: A brief introduction to statistical shape analysis. *Informatics and mathematical modelling, Technical University of Denmark, DTU 15*, 11 (2002). 2, 4

[Tau23] TAUBIN G.: *A Signal Processing Approach To Fair Surface Design*, 1 ed. Association for Computing Machinery, New York, NY, USA, 2023. URL: https://doi.org/10.1145/3596711.3596724. 2