EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 2

Scalable and Efficient Functional Map Computations on Dense Meshes

Robin Magnet[1] and Maks Ovsjanikov[1]

[1]LIX, Ecole Polytechnique, IP Paris

Abstract

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.

CCS Concepts

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

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 wide range of techniques developed in the past few years [DYDZ22, Sah20]. 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 [OBS*12]. Methods based on this framework have recently achieved high accuracy on a range of difficult non-rigid shape matching tasks [MMR*19, MRR*19, DLR*20]. Unfortunately, these approaches require costly and time-consuming precomputation of the Laplacian basis and, potentially, other auxiliary data-structures [RPWO19]. As a result, these techniques do not scale well to densely sampled meshes and, thus, are most often applied on simplified shapes. Moreover, while accelerated versions of some methods [MRR*19] have recently been proposed, these lack theoretical approximation guarantees, and can be error-prone.

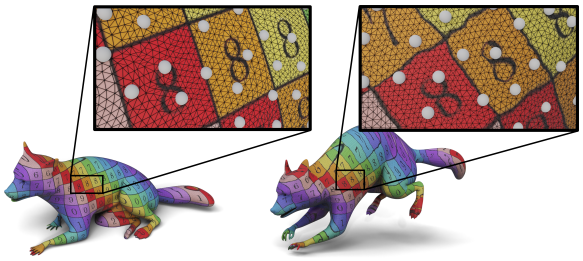

Figure 1: *Our method produces point-to-point correspondences between dense meshes efficiently, using values only located at sparse samples, displayed in white. The source and target shapes from the* DEFORMINGTHINGS4D *dataset [LTT*21] are composed of roughly* 100 000 *vertices, and correspondences are displayed using texture transfer. The map computation (including all preprocessing) took 60 seconds on a standard machine.*

At the same time, several approaches have recently been proposed for efficient approximation of the Laplace-Beltrami basis

[NBH18, NH22]. 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 [RWP06] or for individual shape processing, they can come short when applied in *shape correspondence scenarios*. Conceptually, this is because the objectives and guarantees in [NBH18, NH22] 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 [OCB*17] and especially its recent variants based on spectral upsampling, such as the ZoomOut method [MRR*19] and its follow-up works [HRWO20, XLZ21, RMWO21]. These methods 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 interative 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 [NBH18], and demonstrate how it can be used to define an approximation of functional maps without requiring either a dense pointwise correspondence or a even basis on the dense meshes. We then provide theoretical approximation bounds for this construction that, unlike the original definition in [OBS*12] is fully agnostic to the number of points in the original mesh. Following this analysis, we extend the approach introduced in [NBH18] 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 2.

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 [VKZHCO11, TCL*12, BCBB16, Sah20, DYDZ22] 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, [MHK*08, JZVK07] or using diffusion processes to derive descriptors or embedding spaces, e.g., [SH10, OMMG10], among others.

A more principled framework was introduced in [OBS*12], 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 hyper-parameter tuning. As a result, this basic strategy has been extended significantly in many follow-up works, based both on geometric insights [KBB*12, AK13, OMPG13, BDK17, ERGB16], improved optimization strategies [KGB16, NO17, RMOW20, RMWO21], and richer correspondence models going beyond isometries across complete shapes, [RCB*17, ROA*13, LRBB17], 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 [OBS*12, MDK*16, OMPG13] and has since then been extended to map deblurring and denoising [EB17], as well as powerful refinement, and even map optimization strategies [MRR*19, RPWO19, HRWO20, ELC20, RMWO21]. 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 [VBCG10], and especially to recent developments for accurate and scalable eigen-solvers geared towards Laplacian eigenfunctions on complex meshes [NBH18, NH22]. 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 [RWP06] descriptors, or performing mesh filtering. Similarly, there exist several spectral coarsening and simplification approaches [LJO19, LLT*20, CLJL20] that explicitly aim to coarsen operators, such as the Laplacian while preserving their low frequency eigenapairs. 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 [SVBC19], and even more closely, to refinement via spectral upsampling [MRR*19]. 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 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

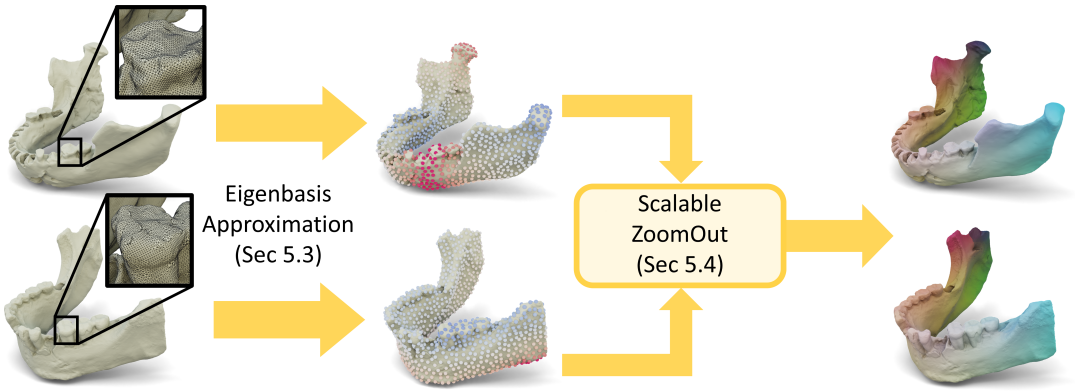

Figure 2: *Overall pipeline of our method, using real data from [RWHO20]. Given two dense input shapes, we first generate an approximate eigenbasis computation by using a modified version of the approach introduced in [NH22] (Sec. 5.3). We then propose a new scalable version of ZoomOut (Sec. 5.4), which exploits our functional map approximation (Sec 5.1) and comes with theoretical appromation bounds. Ultimately, this leads to dense pointwise correspondences between the two input shapes visualized here via color transfer.*

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 [NBH18] 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.

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 famework [MRR*19]. However, our constructions can be easily extended to other recent functional maps methods, e.g., [RMOW20, RMWO21], 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 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 [NBH18], described in Sec 5.3. Secondly, we use the approximation of the functional map that we introduce (Sec. 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 paper is organized as follows: in Section 4 we introduce the notations and background necessary for our approach.

In Section 5.1, we introduce our functional map approximation based on the basis construction approach in [NBH18]. Section 5.2 provides explicit approximation errors and Section 5.3 describes our modification of the method of [NBH18], which helps to improve the theoretical upper bounds we obtained for functional map computation. Given these constructions, we show in Sec. 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 5.5 provides implementation details, while Section 6 is dedicated to extensive experimental evaluation of our approach.

4. Notations & Background

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 $T : \mathcal{N} \to \mathcal{M}$ can be represented as a binary matrix $\mathbf{\Pi} \in \{0,1\}^{n\times m}$ with $\mathbf{\Pi}_{ij} = 1$ 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:

$$\mathbf{W}\psi_i = \lambda_i \mathbf{A}\psi_i, \tag{1}$$

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

4.2. Functional Maps and the ZoomOut algorithm

Functional maps were introduced in [OBS*12] as a means to perform dense non-rigid shape matching. The key insight is that any

pointwise map $T : \mathcal{N} \to \mathcal{M}$ 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 $\left(\psi_i^{\mathcal{M}}\right)_i$ and $\left(\psi_i^{\mathcal{N}}\right)_i$.

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:

$$\mathbf{C} = \left(\mathbf{\Psi}^{\mathcal{N}}\right)^{\top} \mathbf{A}^{\mathcal{N}} \mathbf{\Pi} \mathbf{\Psi}^{\mathcal{M}}, \tag{2}$$

where each basis function on $\mathcal{M}$ (resp. $\mathcal{N}$) is stacked as a column of $\mathbf{\Psi}^{\mathcal{M}}$ (resp. $\mathbf{\Psi}^{\mathcal{N}}$), $\mathbf{\Pi}$ 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 [MRR*19] allows to recover 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 [HRWO20] and optimize various energies [RMWO21] 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 [MRR*19]. However, as we discuss below, these do not come with approximation guarantees and indeed can fail to converge in the limit of complete information.

4.3. Eigenbasis approximation

To improve the scalability of spectral methods, recent works [NBH18, XLHH21] have tried to develop approximations of the Laplace Beltrami eigenbasis, via the reduction of the search space. Specifically, in [NBH18], 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:

$$\tilde{u}_j : x \in \mathcal{M} \mapsto \chi_\rho\left(d^{\mathcal{M}}(x, v_j)\right) \in \mathbb{R} \tag{3}$$

where $d^{\mathcal{M}}$ is the geodesic distance on shape $\mathcal{M}$ and $\chi_\rho : \mathbb{R}_+ \to \mathbb{R}$ 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 A. Finally, local functions u_j are defined to satisfy the partition of the unity by using:

$$u_j(x) = \frac{\tilde{u}_j(x)}{\sum_k \tilde{u}_k(x)} \quad \forall\, x \in \mathcal{M} \tag{4}$$

Now only considering functions that lie in the Span $\{u_1, \dots, u_p\}$, the original eigendecomposition system in Eq. (1) reduces to a generalized eigenproblem of size $p \times p$:

$$\overline{\mathbf{W}}\, \bar{\phi}_i = \bar{\lambda}_i \overline{\mathbf{A}}\, \bar{\phi}_i \tag{5}$$

with $\overline{\mathbf{W}} = \mathbf{U}^{\top}\mathbf{W}\mathbf{U}$ and $\overline{\mathbf{A}} = \mathbf{U}^{\top}\mathbf{A}\mathbf{U}$ 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 $\{u_j\}$, which define the approximated eigenvectors as $\overline{\psi}_i = \mathbf{U}\bar{\phi}_i$. Note that since $\bar{\phi}_i$ are orthonormal with respect to $\overline{\mathbf{A}}$ this implies that $\overline{\psi}_i$ are orthonormal with respect to $\mathbf{A}$.

While the original work [NBH18] focused on global per-shape applications such as filtering and Shape-DNA [RWP06] computation, we build on and modify this pipeline in order to obtain reliable functions to perform dense *shape correspondence*.

5. Our approach

In this section, we first present a functional map definition using the basis approximation strategy from of [NBH18], and provide theoretical bounds on the approximation error (Secs. 5.1 and 5.2 respectively). Based by these results, we then introduce our modification of [NBH18] in Section 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 5.4 and 5.5.

5.1. Approximate Functional Map

As mentioned in Sec. 4.3 the eigenfunctions computed using the approach in [NBH18] are, by construction, orthonormal with respect to the area matrix $\mathbf{A}$. Thus, they can be used to compute a functional map following Eq. (2). This leads to the following definition:

Definition 5.1 Given two shapes $\mathcal{M}$ and $\mathcal{N}$ with approximated eigenfunctions $\left(\Psi_i^{\mathcal{M}}\right)_i$ stacked as columns of matrix $\overline{\mathbf{\Psi}}^{\mathcal{M}}$ (resp. with $\mathcal{N}$), the *reduced* functional map associated to a pointwise map $\mathbf{\Pi} : \mathcal{N} \to \mathcal{M}$ is defined as:

$$\overline{\mathbf{C}} = \left(\overline{\mathbf{\Psi}}^{\mathcal{N}}\right)^{\top} \mathbf{A}^{\mathcal{N}} \mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}} \tag{6}$$

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 $\overline{\mathbf{C}}$ that only relies on maps between samples, independently from the original number of points:

Definition 5.2 Using the same setting as in Definition 5.1, with eigenfunctions arising from Eq. (5), $\left(\bar{\phi}_i^{\mathcal{M}}\right)_i$ (resp. with $\mathcal{N}$) being stacked as columns of a matrix $\overline{\mathbf{\Phi}}^{\mathcal{M}}$ (resp. with $\mathcal{N}$), given a pointwise map $\overline{\mathbf{\Pi}} : \mathcal{S}^{\mathcal{N}} \to \mathcal{S}^{\mathcal{M}}$, our *restricted* functional map is defined as:

$$\widehat{\mathbf{C}} = \left(\overline{\mathbf{\Phi}}^{\mathcal{N}}\right)^{\top} \overline{\mathbf{A}}^{\mathcal{N}} \overline{\mathbf{\Pi}}\, \overline{\mathbf{\Phi}}^{\mathcal{M}} \tag{7}$$

Recall that, as mentioned in Sec 4.3 $\mathcal{S}$ denotes the sparse set of samples on each shape. Therefore, in order to define $\widehat{\mathbf{C}}$, we only need to have access to a pointwise map between *the sample points* on the two shapes. This restricted functional map $\widehat{\mathbf{C}}$ is a pull-back operator associated to the reduced spaces Span$\left\{\bar{\phi}_k^{\mathcal{M}}\right\}_k$ and Span$\left\{\bar{\phi}_k^{\mathcal{N}}\right\}_k$ since both families are orthonormal with respect

to $\overline{\mathbf{A}}$. Furthermore, using the factorization $\overline{\mathbf{\Psi}} = \mathbf{U}\overline{\mathbf{\Phi}}$ on each shape in (6) as well as the definition of $\overline{\mathbf{A}}$, we remark that going from Eq. (6) to (7) only requires the approximation $\mathbf{\Pi}\mathbf{U}^{\mathcal{M}} \simeq \mathbf{U}^{\mathcal{N}}\overline{\mathbf{\Pi}}$, for which we will later on derive an upper bound in Proposition 5.2. Note that one might want to replace $\overline{\mathbf{\Phi}}^{\mathcal{M}}$ by $\overline{\mathbf{\Psi}}^{\mathcal{M}}$ in Eq. (7) so that the map $\overline{\mathbf{\Pi}}$ actually transports pointwise values rather than coefficients. In practice as evaluated in Appendix B, we did not observe any improvement using this modification.

The first benefit of the approximated functional map in Eq. (7) compared to the exact one in Eq. (6) is the exclusive use of small-sized matrices. Observe that functions $\left(\overline{\phi}_i\right)_i$, are associated with the area and stiffness matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{W}}$, 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 $\widehat{\mathbf{C}}$ and $\overline{\mathbf{C}}$, as described later in Section 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 [MRR*19] (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 $\overline{\mathbf{C}}$ using

$$\mathbf{C}_{\text{F-ZO}} = \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{Q}^{\mathcal{N}}\mathbf{\Psi}^{\mathcal{N}}\mathbf{X} - \mathbf{\Pi}\mathbf{Q}^{\mathcal{M}}\mathbf{\Psi}^{\mathcal{M}}\|_F^2 \tag{8}$$

where $\mathbf{Q}^{\mathcal{N}} \in \{0,1\}^{q \times n^{\mathcal{N}}}$ with $\mathbf{Q}^{\mathcal{N}}_{ij} = 1$ 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 [NBH18] to reduce these errors and obtain a more accurate and principled correspondence approach.

5.2. Approximation Errors

Most expressions above involve a given pointwise map $\mathbf{\Pi}$ 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 5.1 Let $\mathcal{M}$ and $\mathcal{N}$ be compact surfaces and $T : \mathcal{N} \to \mathcal{M}$ a diffeomorphism. Then there exists $B_T \in \mathbb{R}$ so that:

$$\|f \circ T\|_{\mathcal{N}} \leq B_T \|f\|_{\mathcal{M}} \quad \forall\, f \in L^2(\mathcal{M}) \tag{9}$$

the proof of which can be found in [HCO18] (Proposition 3.3).

Our overall goal is to use the newly designed functional map $\overline{\mathbf{C}}$ 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 $\overline{\Psi}_k$ approximate well the true ones Ψ_k. The following proposition bounds the error between the two functional maps:

Proposition 5.1 Let $\overline{\mathbf{\Psi}}^{\mathcal{N}}$ (resp. $\overline{\mathbf{\Psi}}^{\mathcal{M}}$) and $\mathbf{\Psi}^{\mathcal{N}}$ (resp. $\mathbf{\Psi}^{\mathcal{M}}$) the approximated and true first K eigenvectors of the Laplacian on $\mathcal{N}$ (resp. $\mathcal{M}$). Let $\mathbf{C}$ and $\overline{\mathbf{C}}$ be the original and reduced (see Eq. (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 5.1. If there exists $\varepsilon \in \mathbb{R}_+^*$ so that for any $j \in \{1,\dots,K\}$:

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

Then:

$$\frac{1}{K}\left\|\mathbf{C} - \overline{\mathbf{C}}\right\|_2^2 \leq \varepsilon^2\left(1 + B_T^2\right) \tag{10}$$

The proof can be found in Appendix C. 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 $\overline{\mathbf{C}}$ from Def. 5.1 and $\widehat{\mathbf{C}}$ from Def. 5.2. As mentioned above, the estimation relies on the identification $\mathbf{\Pi}\overline{\mathbf{\Psi}}^{\mathcal{M}} \simeq \mathbf{U}^{\mathcal{N}}\overline{\mathbf{\Pi}}\,\overline{\mathbf{\Phi}}^{\mathcal{M}}$, where $\overline{\mathbf{\Pi}}$ is a map between the two sets of samples $\mathcal{S}^{\mathcal{N}}$ and $\mathcal{S}^{\mathcal{M}}$, which we expect to be similar to $\mathbf{\Pi}$ on these spaces. This approximation treats equivalently the two following procedures: 1) interpolating between values on $\mathcal{S}^{\mathcal{M}}$ then transferring using the map $\mathbf{\Pi}$, 2) transferring values on $\mathcal{S}^{\mathcal{M}}$ to values on $\mathcal{S}^{\mathcal{N}}$ using $\overline{\mathbf{\Pi}}$ and then interpolating on $\mathcal{N}$. The following proposition bounds the error of this approximation:

Proposition 5.2 Let $T : \mathcal{N} \to \mathcal{M}$ be a pointwise map between the shapes represented by $\mathbf{\Pi}$, and let B_T be the bound given by Lemma 5.1. Suppose that $T_{|\mathcal{S}^{\mathcal{N}}} : \mathcal{S}^{\mathcal{N}} \to \mathcal{S}^{\mathcal{M}}$ is represented by $\overline{\mathbf{\Pi}}$. 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 $x, y \in \mathcal{S}^{\mathcal{M}}$:

$$d^{\mathcal{M}}(x,y) \leq \rho^{\mathcal{M}} \Rightarrow |\overline{\Psi}_k^{\mathcal{M}}(x) - \overline{\Psi}_k^{\mathcal{M}}(y)| \leq \varepsilon \tag{11}$$

and

$$d^{\mathcal{M}}(x,y) \leq \rho^{\mathcal{M}} \Rightarrow |\overline{\Phi}_k^{\mathcal{M}}(x) - \overline{\Phi}_k^{\mathcal{M}}(y)| \leq \varepsilon. \tag{12}$$

Then

$$\frac{1}{K}\left\|\mathbf{\Pi}\overline{\mathbf{\Psi}}^{\mathcal{M}} - \mathbf{U}^{\mathcal{N}}\overline{\mathbf{\Pi}}\,\overline{\mathbf{\Phi}}^{\mathcal{M}}\right\|_{\mathcal{N}}^2 \leq \varepsilon^2(1-\alpha) + \varepsilon^2 B_T^2 \tag{13}$$

The proof is given in Appendix D. 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 basis coefficients with function values. A discussion on the numerical values of the quantities used in Proposition 5.2 are provided in Appendix E.

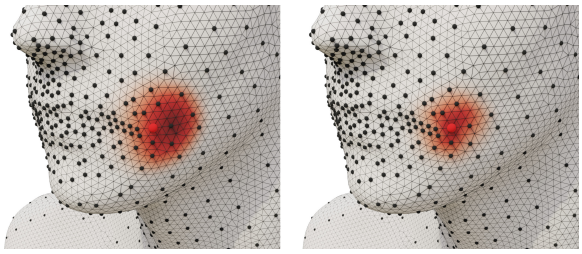

Figure 3: *Example of a local function u_j (red color) centered on v_j (red vertex), visualized without (Left) and with (Right) our adaptive radius strategy. Other samples v_k are displayed in black.*

In the following, we will therefore seek to modify the basis approximation [NBH18] 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.

5.3. Improved Eigenbasis Approximation

In this section, we propose a modification of the approach from [NBH18], 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 Prop. 5.2, high self weights allow to stabilize our functional map approximation. Interestingly with the construction in [NBH18], the value $u_j(v_j)$ only depends on the geodesic distance between v_j and other sampled points v_i for $i \neq j$:

$$u_j(v_j) = \frac{1}{1+\sum_{i\neq j} \tilde{u}_i(v_j)}. \tag{14}$$

where $\tilde{u}_i$ are the un-normalized local functions. We modify the pipeline from [NBH18] 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 Eq. (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, this does not require additional local geodesic distance computations. More details are provided in Sec. 5.5, and the algorithm to compute these new functions is displayed in Algorithm 1. We observe that the adaptive radius strategy generates better local functions than those introduced in [NBH18], especially for non-uniform sampling, as can be seen on a surface from the DFaust dataset [BRPB17] in Figure 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 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.

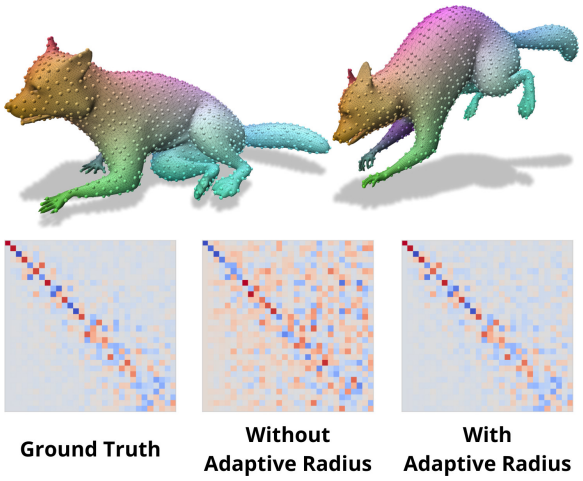

Figure 4: *Effect of the adaptive radius on functional map approximation. Top row displays a pointwise map T from the right mesh to the left mesh using color transfer. Bottom row displays $\overline{\mathbf{C}}$ (Left), $\widehat{\mathbf{C}}$ when using the pipeline from [NBH18] (Middle) and our functional map $\widehat{\mathbf{C}}$ (Right).*

Algorithm 1 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 : (3), (4)
3: Add sample points if necessary
4: **while** some k with $u_k(v_k) < \varepsilon$ **do**
5: $\quad j \leftarrow \underset{i\neq k}{\operatorname{argmax}}\, u_i(v_k)$
6: $\quad \rho_j \leftarrow \rho_j/2$
7: $\quad$ update all u using Eq. (3), (4)
8: **end while**
9: Add unseen vertices in the sample

The positive effect of our adaptive radius algorithm for functional map estimation is further visualized in Figure 4, where given a single pointwise map T, we display the exact functional map on the approximated spaces $\overline{\mathbf{C}}$, and two approximated functional maps $\widehat{\mathbf{C}}$, one being computed with a shared radius [NBH18] and the other with our adaptive radius scheme.We highlight that the ground truth functional map actually differ for each approximation $\widehat{\mathbf{C}}$ 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 F), which is why we only display one in Figure 4. Note that using the adaptive radius strategy then generates a sparsity pattern on matrix $\widehat{\mathbf{C}}$ very close to the ground truth one.

5.4. Scalable ZoomOut

In light of the previous discussions and theoretical analysis, we now describe how to use the approximated functional map $\widehat{\mathbf{C}}$ within a standard ZoomOut pipeline [MRR*19]. Our complete pipeline is summarized in Algorithm 2, where the notation $\overline{\mathbf{\Phi}}_{1:k}$ indicates that we only use the first k column of matrix $\overline{\mathbf{\Phi}}_{1:k}$.

Algorithm 2 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. 1
3: Approximate eigenvectors $\overline{\mathbf{\Phi}}^{\mathcal{M}}$ and $\overline{\mathbf{\Phi}}^{\mathcal{M}}$ solving (5)
4: Set $\overline{\mathbf{\Psi}}^{\mathcal{M}} = \mathbf{U}^{\mathcal{M}}\overline{\mathbf{\Phi}}^{\mathcal{M}}$ and $\overline{\mathbf{\Psi}}^{\mathcal{N}} = \mathbf{U}^{\mathcal{N}}\overline{\mathbf{\Phi}}^{\mathcal{N}}$
5: Obtain $\overline{\mathbf{\Pi}}$ between *samples* using the initial map
6: **for** $k = k_{\text{init}}{:}k_{\text{final}}$ **do**
7: $\quad\widehat{\mathbf{C}} = \left(\overline{\mathbf{\Phi}}^{\mathcal{N}}_{1:k}\right)^{\top}\overline{\mathbf{A}}^{\mathcal{N}}\overline{\mathbf{\Pi}}\,\overline{\mathbf{\Phi}}^{\mathcal{M}}_{1:k}$
8: $\quad\overline{\mathbf{\Pi}} = \text{NNsearch}\left(\overline{\mathbf{\Phi}}^{\mathcal{M}}_{1:k}, \overline{\mathbf{\Phi}}^{\mathcal{N}}_{1:k}\widehat{\mathbf{C}}\right)$ potentially using (16)
9: **end for**
10: $\mathbf{\Pi} = \text{NNsearch}\left(\overline{\mathbf{\Psi}}^{\mathcal{M}}_{1:k}, \overline{\mathbf{\Psi}}^{\mathcal{N}}_{1:k}\widehat{\mathbf{C}}\right)$
11: **Return** $\mathbf{\Pi}$

As mentioned earlier, using $\widehat{\mathbf{C}}$ and matrices $\overline{\mathbf{A}}$ and $\overline{\mathbf{W}}$ allows to apply 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 $\widehat{\mathbf{C}}^*$ and a refined pointwise map *between samples* $\overline{\mathbf{\Pi}}^*$. The last remaining non-trivial task consists in converting the refined functional map into a global pointwise map $\mathbf{\Pi}^*$ 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 $\widehat{\mathbf{C}}$ and $\overline{\mathbf{C}}$, which then allows us to compute the pointwise map $\mathbf{\Pi}^*$ by solving the standard least square problem:

$$\mathbf{\Pi}^* = \underset{\mathbf{\Pi}}{\text{argmin}} \|\overline{\mathbf{\Psi}}^{\mathcal{N}}\widehat{\mathbf{C}}^* - \mathbf{\Pi}\overline{\mathbf{\Psi}}^{\mathcal{M}}\|^2_{\mathbf{A}^{\mathcal{N}}}. \tag{15}$$

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 $\overline{\mathbf{\Pi}}$ 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 $\overline{\mathbf{\Pi}}$, that is

$$\mathcal{I}(x) = \{y \mid \exists j \in N(x),\, u_{\bar{T}(j)}(y) > 0\} \tag{16}$$

where $\bar{T}$ is the function representation of $\overline{\mathbf{\Pi}}$. 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.

Table 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	***Preprocess***	***LBO***	***ZoomOut***	***Conversion***	***Total (s)***
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

5.5. Implementation

We implement the complete algorithm in Python and provide the code at https://github.com/RobinMagnet/Scalable_FM.

Following [NBH18], 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 5.3 does not require additional geodesic distance computations. Furthermore finding the set of potential images for a point as described in Section 5.4 simply reduces to checking non-zero indices in a sparse matrix. More details are provided in Appendix G.

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 all steps of the pipeline - that is preprocessing 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.

6.1. Timings

The method introduced in [NBH18] aimed at approximating the spectrum of the Laplace Beltrami Operator efficiently. As mentioned above, the additional building blocks we introduced in Section 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 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

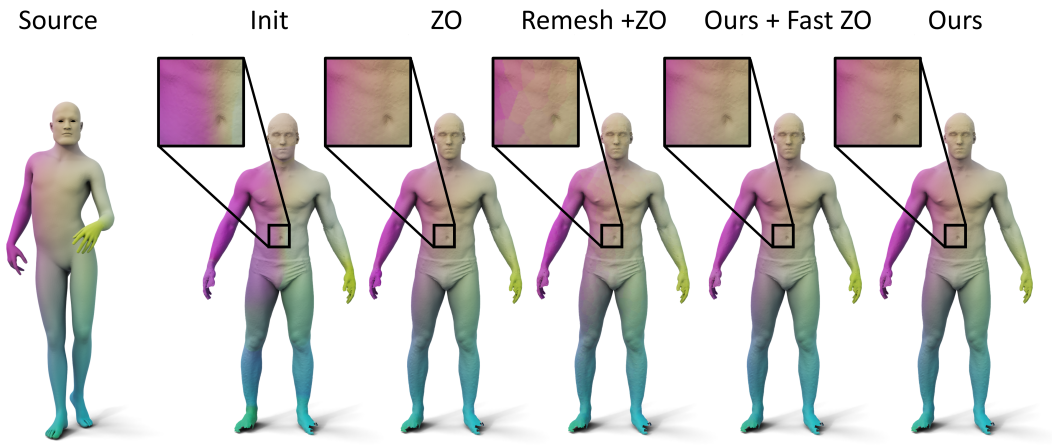

Figure 5: *Qualitative results on the SHREC19 dataset. Although processing time differ heavily, there is no significant difference between our method and results from ZoomOut. However, remeshing the surface before ZoomOut results in locally constant correspondences.*

meshes compared to Poisson sampling, which explains the similar preprocessing timings between our method and Fast ZoomOut.

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 [MMR*19] 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 [MRR*19] in only a fraction of the required time.

Metrics We evaluate different methods using standard metrics [RPWO19] for dense shape correspondence, that is accuracy, coverage and smoothness. The accuracy of a computed dense map $T : \mathcal{N} \to \mathcal{M}$ 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. 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 [MRR*19] 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 [ELC20, ESB19] 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 (8) on the approximated functional space $\overline{\mathcal{F}}$ introduced in section 5.1 (Ours + Fast ZO). Table 2 shows the values of the evaluation metrics on the SHREC19 dataset where the accuracy curves can be found on Figure 6, and Figure 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 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.

Table 2: *Evaluation of different methods on the complete SHREC19 dataset. Blue highlights the best two methods.*

methods	***Accuracy***	***Coverage***	***Smoothness***
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**

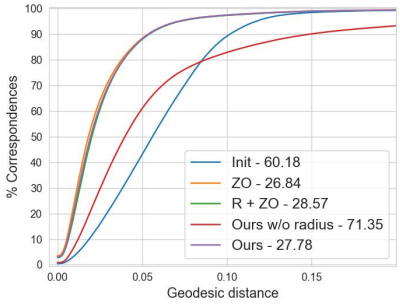

Figure 6: *Accuracy curves for different methods presented in Table 2. Numbers in the legend provide the average geodesic error ($\times 10^3$).*

Sub-sample accuracy One Figure 7, 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 [ROA*13]. 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 8 and in Appendix H, which provides further details on the shapes.

Adaptive Radius While results on Table 2 highlight the efficiency of the adaptive radius scheme, we additionally evaluate how this heuristic allows to improve the estimation $\Delta = \|\overline{\mathbf{C}} - \widehat{\mathbf{C}}\|$ presented in section 5. For this we simply compute $\overline{\mathbf{C}}$ and $\widehat{\mathbf{C}}$ 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 3. In this experiment we notice our method improves the baseline by two orders of magnitude.

Table 3: *Norm of the estimation error Δ with and without adaptive radius on the SHREC19 dataset*

	w/o radius	***Ours***
Δ ($\times 10$)	1.486	0.018

7. Conclusion, Limitations and Future Work

In this paper 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 [NBH18] 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 [NBH18] and demonstrate how better behaved local functions can be obtained with very little additional effort. We use this construction 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 [MRR*19], 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.

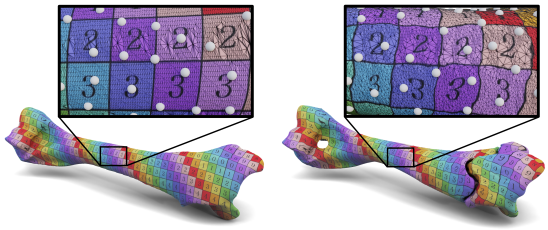

Figure 7: *Texture transfer using our scalable version of ZoomOut. Samples used in the pipeline are shown as white dots.*

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 [NBH18], 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 10's of millions to billions of vertices, while possible in principle, would be an excellent practical future extension of our approach.

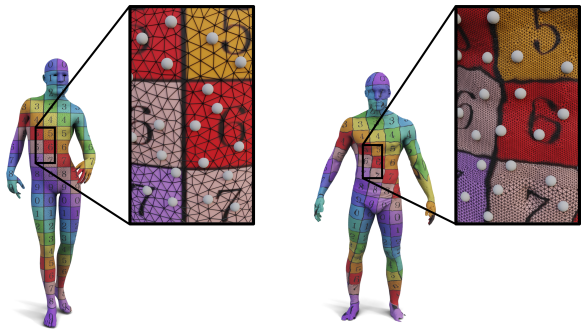

Figure 8: *Texture transfer using our scalable version of ZoomOut on a pair of the SHREC19 dataset. Samples used in the pipeline are shown as white dots.*

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.

References

[AK13] AFLALO Y., KIMMEL R.: Spectral multidimensional scaling. *Proceedings of the National Academy of Sciences 110*, 45 (Nov. 2013),

18052–18057. 2

[BCBB16] BIASOTTI S., CERRI A., BRONSTEIN A., BRONSTEIN M.: Recent trends, applications, and perspectives in 3d shape similarity assessment. In *Computer graphics forum* (2016), vol. 35, Wiley Online Library, pp. 87–119. 2

[BDK17] BURGHARD O., DIECKMANN A., KLEIN R.: Embedding shapes with green's functions for global shape matching. *Computers & Graphics 68* (2017), 1–10. 2

[BRPB17] BOGO F., ROMERO J., PONS-MOLL G., BLACK M. J.: Dynamic FAUST: Registering Human Bodies in Motion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Honolulu, HI, July 2017), IEEE, pp. 5573–5582. 6

[CLJL20] CHEN H., LIU H.-T. D., JACOBSON A., LEVIN D. I. W.: Chordal decomposition for spectral coarsening. *ACM Trans. Graph. 39*, 6 (nov 2020). 2

[DLR*20] DYKE R. M., LAI Y.-K., ROSIN P. L., ZAPPALÀ S., DYKES S., GUO D., LI K., MARIN R., MELZI S., YANG J.: Shrec'20: Shape correspondence with non-isometric deformations. *Computers & Graphics 92* (2020), 28–43. 1

[DYDZ22] DENG B., YAO Y., DYKE R. M., ZHANG J.: A survey of non-rigid 3d registration. *arXiv preprint arXiv:2203.07858* (2022). 1, 2

[EB17] EZUZ D., BEN-CHEN M.: Deblurring and Denoising of Maps between Shapes. *Computer Graphics Forum 36*, 5 (Aug. 2017), 165–174. 2

[ELC20] EISENBERGER M., LAHNER Z., CREMERS D.: Smooth Shells: Multi-Scale Shape Registration With Functional Maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 12265–12274. 2, 8

[ERGB16] EYNARD D., RODOLA E., GLASHOFF K., BRONSTEIN M. M.: Coupled functional maps. In *2016 Fourth International Conference on 3D Vision (3DV)* (2016), IEEE, pp. 399–407. 2

[ESB19] EZUZ D., SOLOMON J., BEN-CHEN M.: Reversible Harmonic Maps between Discrete Surfaces. *ACM Transactions on Graphics 38*, 2 (Mar. 2019), 15:1–15:12. 8

[HCO18] HUANG R., CHAZAL F., OVSJANIKOV M.: On the Stability of Functional Maps and Shape Difference Operators. *Computer Graphics Forum 37*, 1 (Feb. 2018), 145–158. 5

[HRWO20] HUANG R., REN J., WONKA P., OVSJANIKOV M.: Consistent ZoomOut: Efficient Spectral Map Synchronization. *Computer Graphics Forum 39*, 5 (Aug. 2020), 265–278. 2, 4

[JZVK07] JAIN V., ZHANG H., VAN KAICK O.: Non-rigid spectral correspondence of triangle meshes. *International Journal of Shape Modeling 13*, 01 (2007), 101–124. 2

[KBB*12] KOVNATSKY A., BRONSTEIN M. M., BRONSTEIN A. M., GLASHOFF K., KIMMEL R.: Coupled quasi-harmonic bases. *arXiv:1210.0026 [cs]* (Sept. 2012). arXiv:1210.0026. 2

[KGB16] KOVNATSKY A., GLASHOFF K., BRONSTEIN M. M.: Madmm: a generic algorithm for non-smooth optimization on manifolds. In *European Conference on Computer Vision* (2016), Springer, pp. 680–696. 2

[LJO19] LIU H.-T. D., JACOBSON A., OVSJANIKOV M.: Spectral coarsening of geometric operators. *ACM Transactions on Graphics 38*, 4 (July 2019), 1–13. 2

[LLT*20] LESCOAT T., LIU H.-T. D., THIERY J.-M., JACOBSON A., BOUBEKEUR T., OVSJANIKOV M.: Spectral Mesh Simplification. *Computer Graphics Forum 39*, 2 (May 2020), 315–324. 2

[LRBB17] LITANY O., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.: Fully Spectral Partial Shape Matching. *Computer Graphics Forum 36*, 2 (May 2017), 247–258. 2

[LTT*21] LI Y., TAKEHARA H., TAKETOMI T., ZHENG B., NIESSNER M.: 4dcomplete: Non-rigid motion estimation beyond the observable surface. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 12706–12716. 1

[MDK*16] MARON H., DYM N., KEZURER I., KOVALSKY S., LIPMAN Y.: Point registration via efficient convex relaxation. *ACM Transactions on Graphics 35*, 4 (July 2016), 1–12. 2

[MHK*08] MATEUS D., HORAUD R., KNOSSOW D., CUZZOLIN F., BOYER E.: Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (2008), IEEE, pp. 1–8. 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). 1, 8

[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 38*, 6 (Nov. 2019), 1–14. 1, 2, 3, 4, 5, 7, 8, 9

[NBH18] NASIKUN A., BRANDT C., HILDEBRANDT K.: Fast Approximation of Laplace-Beltrami Eigenproblems. *Computer Graphics Forum 37*, 5 (Aug. 2018), 121–134. 2, 3, 4, 5, 6, 7, 9, 11

[NH22] NASIKUN A., HILDEBRANDT K.: The hierarchical subspace iteration method for laplace–beltrami eigenproblems. *ACM Transactions on Graphics (TOG) 41*, 2 (2022), 1–14. 2, 3

[NO17] NOGNENG D., OVSJANIKOV M.: Informative Descriptor Preservation via Commutativity for Shape Matching. *Computer Graphics Forum 36*, 2 (May 2017), 259–267. 2

[OBS*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 31*, 4 (Aug. 2012), 1–11. 1, 2, 3

[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* (New York, NY, USA, July 2017), SIGGRAPH '17, Association for Computing Machinery, pp. 1–62. 2

[OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L.: One point isometric matching with the heat kernel. In *Computer Graphics Forum* (2010), vol. 29, Wiley Online Library, pp. 1555–1564. 2

[OMPG13] OVSJANIKOV M., MÉRIGOT Q., PĂTRĂUCEAN V., GUIBAS L.: Shape Matching via Quotient Spaces. *Computer Graphics Forum 32*, 5 (Aug. 2013), 1–11. 2

[RCB*17] RODOLÀ E., COSMO L., BRONSTEIN M. M., TORSELLO A., CREMERS D.: Partial Functional Correspondence: Partial Functional Correspondence. *Computer Graphics Forum 36*, 1 (Jan. 2017), 222–236. 2

[RMOW20] REN J., MELZI S., OVSJANIKOV M., WONKA P.: MapTree: Recovering multiple solutions in the space of maps. *ACM Transactions on Graphics 39*, 6 (Nov. 2020), 264:1–264:17. 2, 3

[RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete Optimization for Shape Matching. *Computer Graphics Forum 40*, 5 (Aug. 2021), 81–96. 2, 3, 4

[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 32*, 4 (July 2013), 1. 2, 9

[RPWO19] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics 37*, 6 (Jan. 2019), 1–16. 1, 2, 8

[RWHO20] REN J., WONKA P., HARIHARA G., OVSJANIKOV M.: Geometric analysis of shape variability of lower jaws of prehistoric humans. *L'Anthropologie 124*, 5 (Dec. 2020), 102808. 3

[RWP06] REUTER M., WOLTER F.-E., PEINECKE N.: Laplace–beltrami spectra as 'shape-dna' of surfaces and solids. *Computer-Aided Design 38*, 4 (2006), 342–366. 2, 4

[Sah20] SAHILLIOĞLU Y.: Recent advances in shape correspondence. *The Visual Computer 36*, 8 (Aug. 2020), 1705–1721. 1, 2

[SH10] SHARMA A., HORAUD R.: 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), IEEE, pp. 29–36. 2

[SVBC19] SHOHAM M., VAXMAN A., BEN-CHEN M.: Hierarchical functional maps between subdivision surfaces. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 55–73. 2

[TCL*12] TAM G. K., CHENG Z.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics 19*, 7 (2012), 1199–1217. 2

[VBCG10] VAXMAN A., BEN-CHEN M., GOTSMAN C.: A multi-resolution approach to heat kernels on discrete surfaces. In *ACM SIGGRAPH 2010 papers*. 2010, pp. 1–10. 2

[VKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. In *Computer graphics forum* (2011), vol. 30, Wiley Online Library, pp. 1681–1707. 2

[XLHH21] XU C., LIN H., HU H., HE Y.: Fast calculation of Laplace-Beltrami eigenproblems via subdivision linear subspace. *Computers & Graphics 97* (June 2021), 236–247. 4

[XLZ21] XIANG R., LAI R., ZHAO H.: A Dual Iterative Refinement Method for Non-Rigid Shape Matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021), pp. 15930–15939. 2

Appendix A: Function χ

The function $\chi : \mathbb{R}_+ \to [0,1]$, differentiable with $\chi(0) = 1$ and $\chi(x) = 0$ for $x \leq 1$ can be defined two ways. Following [NBH18], 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 9.

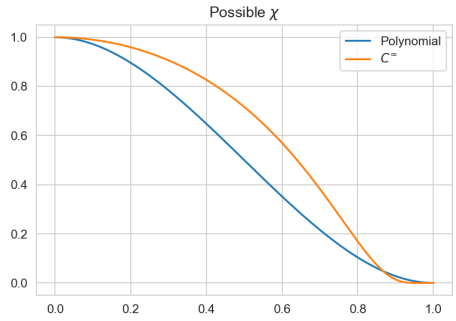

Figure 9: *Possible choices for function* χ

Appendix B: Coefficient weighting

Table 4 compares our algorithm (Ours) with a similar one (Ours + reweight), where we replace $\overline{\mathbf{\Phi}}^{\mathcal{M}}$ by $\overline{\mathbf{\Psi}}^{\mathcal{M}}$ in Eq. (7) so that the map $\overline{\mathbf{\Pi}}$ actually transports pointwise values rather than coefficient, as mentioned in section 5.1. This amounts to reweigthing local coefficients to actually become function values. We notice this method does not improve our pipeline on the SHREC19 dataset.

Table 4: *Evaluation of the reweighting scheme on the SHREC19 dataset.*

methods	***Accuracy***	***Coverage***	***Smoothness***
Init	60.18	26.5 %	9.5
Ours + Reweight	28.1	54.6 %	6.3
Ours	27.78	56.7 %	5.6

Appendix C: Proof of Proposition 5.1

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

$$\mathbf{C}_{i,j} = \langle \psi_j^{\mathcal{N}}, \psi_i^{\mathcal{M}} \circ T \rangle_{\mathcal{N}} \tag{17}$$

$$\overline{\mathbf{C}}_{i,j} = \langle \overline{\psi}_j^{\mathcal{N}}, \overline{\psi}_i^{\mathcal{M}} \circ T \rangle_{\mathcal{N}} \tag{18}$$

Furthermore, given f_1, g_1, f_2, g_2 functions on $\mathcal{N}$.
Then for any $x \in \mathcal{N}$

$$\begin{aligned} f_1(x)g_1(x) - f_2(x)g_2(x) =& f_1(x)\left(g_1(x) - g_2(x)\right) \\ &+ g_2(x)\left(f_1(x) - f_2(x)\right) \end{aligned} \tag{19}$$

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

$$\begin{aligned} \|f_2 - f_1\|_\infty &\leq \varepsilon \\ \|g_2 - g_1\|_\infty &\leq \varepsilon \\ \|g_2\|_{\mathcal{N}} &\leq B_T \end{aligned} \tag{20}$$

Therefore

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

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

Appendix D: Proof of Proposition 5.2

This 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 Sec 4.3.
Given $f : \mathcal{M} \to \mathbb{R}$, 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:

$$|\tilde{f}(x) - f(x)| \leq \varepsilon \quad \forall x \in \mathcal{M} \tag{21}$$

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

$$|\tilde{f}(v_j) - f(v_j)| \leq \varepsilon \left(1 - u_j(v_j)\right) \tag{22}$$

Proof of Prop. D.1
Let $f : \mathcal{M} \to \mathbb{R}$, $\mathcal{S} = \{v_1, \dots, v_p\}$ 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

$$f(x) = \sum_{j=1}^{p} u_j(x) f(x) \tag{23}$$

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

$$\begin{aligned} f(x) - \tilde{f}(x) &= \sum_{j=1}^{p} u_j(x)(f(x) - f(v_j)) \\ &= \sum_{j,\, d(v_j,x)<\rho} u_j(x)(f(x) - f(v_j)) \end{aligned} \tag{24}$$

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

$$\begin{aligned} |f(x) - \tilde{f}(x)|^2 &\leq \sum_{j,\, d(v_j,x)<\rho} u_j(x)^2 |f(x) - f(v_j)|^2 \\ &\leq \sum_{j,\, d(v_j,x)<\rho} u_j(x) |f(x) - f(v_j)|^2 \end{aligned} \tag{25}$$

Which gives $|f(x) - \tilde{f}(x)|^2 \leq \varepsilon$ 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

$$\begin{aligned} |f(x) - \tilde{f}(x)|^2 &\leq \sum_{j \neq k,\, d(v_j,x)<\rho} u_j(v_k)^2 |f(v_k) - f(v_j)|^2 \\ &\leq \varepsilon^2 \sum_{j \neq k,\, d(v_j,x)<\rho} u_j(v_k) \\ &\leq \varepsilon^2 (1 - u_k(v_k)) \end{aligned} \tag{26}$$

□

Proof of Prop. 5.2
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 $\left\| \mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}} - \mathbf{U}^{\mathcal{N}} \overline{\mathbf{\Pi}}\, \overline{\mathbf{\Phi}}^{\mathcal{M}} \right\|_{\mathcal{N}}$:

$$\begin{aligned} \left\| \mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}} - \mathbf{U}^{\mathcal{N}} \overline{\mathbf{\Pi}}\, \overline{\mathbf{\Phi}}^{\mathcal{M}} \right\|_{\mathcal{N}}^2 \leq & \left\| \mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}} - \widetilde{\mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}}} \right\|_{\mathcal{N}}^2 \\ & + \left\| \widetilde{\mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}}} - \mathbf{U}^{\mathcal{N}} \overline{\mathbf{\Pi}}\, \overline{\mathbf{\Phi}}^{\mathcal{M}} \right\|_{\mathcal{N}}^2 \end{aligned} \tag{27}$$

The first term can be decomposed as a sum of the norms of its K columns, where each term is in the form $\|\overline{\Psi}_j^{\mathcal{M}} \circ T - \widetilde{\overline{\Psi}_j^{\mathcal{M}}} \circ T\|_{\mathcal{N}}^2$, and can be controlled by applying the bound on interpolation error from proposition D.1 associated with the bounded distortion lemma, that is

$$\left\| \mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}} - \widetilde{\mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}}} \right\|_{\mathcal{N}}^2 \leq K B_T^2 \varepsilon^2 \tag{28}$$

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 $\left(\overline{\Psi}_j^{\mathcal{M}} \circ T(v_k) - \overline{\Phi}_j^{\mathcal{M}} \circ T_{|\mathcal{S}^{\mathcal{N}}}(v_k)\right)_k$, 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 $\overline{\Psi}_j^{\mathcal{M}}$, for all $x \in \mathcal{M}$ we have $\overline{\Psi}_j^{\mathcal{M}}(x) = \sum_{k=1}^{p^{\mathcal{M}}} \overline{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}}) u_k^{\mathcal{M}}(x)$
Therefore, denoting $\Delta_j(i) = \overline{\Psi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) - \overline{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})$

$$\Delta_j(i) = \sum_{k=1}^{p^{\mathcal{M}}} \overline{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}}) u_k^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) - \overline{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) \tag{29}$$

$$= \sum_{k=1}^{p^{\mathcal{M}}} u_k^{\mathcal{M}}(v_{i_0}^{\mathcal{M}}) \left(\overline{\phi}_j^{\mathcal{M}}(v_k^{\mathcal{M}}) - \overline{\phi}_j^{\mathcal{M}}(v_{i_0}^{\mathcal{M}})\right) \tag{30}$$

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

$$\left\| \widetilde{\mathbf{\Pi} \overline{\mathbf{\Psi}}^{\mathcal{M}}} - \mathbf{U}^{\mathcal{N}} \overline{\mathbf{\Pi}}\, \overline{\mathbf{\Phi}}^{\mathcal{M}} \right\|_{\mathcal{N}}^2 \leq K\varepsilon(1 - \alpha) \tag{31}$$

Summing terms from (28) and (31) produce the upper bound of proposition 5.2.
□

Appendix E: Values of theoretical quantities

We here provide values for the named values from Proposition 5.2. We again highlight the proposed bounds are not tight and only serves as guidance to select parameters.

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

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 .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

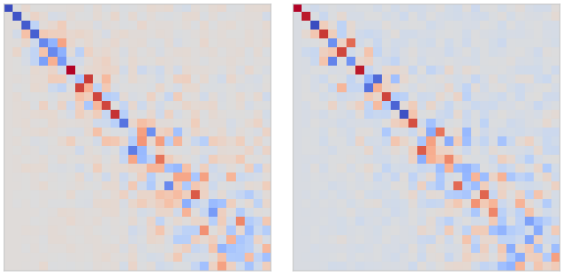

Figure 10: *Ground truth functional map $\overline{\mathbf{C}}$ using the functional space $\overline{\mathcal{F}}$ without (Left) and with (Right) adaptive radius. Notice that up to a change of sign, both functional maps look similar*

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.

Appendix F: Functional Map approximation

We here display on Figure 10 images of the ground truth functional maps for Figure 4

Appendix G: 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 or 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 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 $\left(\tilde{u}_j\right)_j$ 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 (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 $\widetilde{U}$ 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.

Appendix H: 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 11. Here the leftmost shape contains 50 000 vertices and rightmost 200 000, but we only use nearly 1500 samples to obtain such correpondences. 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.

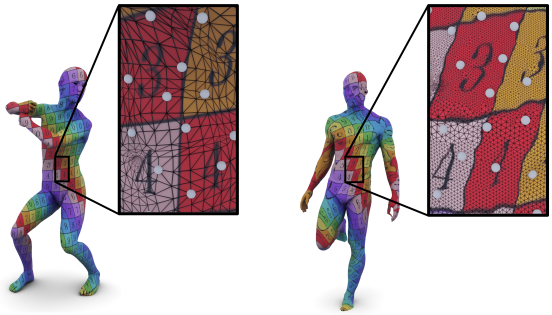

Figure 11: *Texture transfer using our scalable version of ZoomOut on a pair of the SHREC19 dataset. Samples used in the pipeline are shown as white dots.*