Extracting a functional representation from a dictionary for non-rigid shape matching

Michele Colombo[1], Giacomo Boracchi[1], Simone Melzi[2]

[1]Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

[2]Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, Italy

Abstract

Shape matching is a fundamental problem in computer graphics with many applications. Functional maps translate the point-wise shape-matching problem into its functional counterpart and have inspired numerous solutions over the last decade. Nearly all the solutions based on functional maps rely on the eigenfunctions of the Laplace-Beltrami Operator (LB) to describe the functional spaces defined on the surfaces and then convert the functional correspondences into point-wise correspondences. However, this final step is often error-prone and inaccurate in tiny regions and protrusions, where the energy of LB does not uniformly cover the surface. We propose a new functional basis Principal Components of a Dictionary (PCD) to address such intrinsic limitation. PCD constructs an orthonormal basis from the Principal Component Analysis (PCA) of a dictionary of functions defined over the shape. These dictionaries can target specific properties of the final basis, such as achieving an even spreading of energy. Our experimental evaluation compares seven different dictionaries on established benchmarks, showing that PCD is suited to target different shape-matching scenarios, resulting in more accurate point-wise maps than the LB basis when used in the same pipeline. This evidence provides a promising alternative for improving correspondence estimation, confirming the power and flexibility of functional maps.

Keywords: Geometry Processing, Shape Matching, Functional Maps

1. Introduction

Shape matching is a key problem in Computer Graphics and Geometry Processing. From an intuitive point of view, its goal is to estimate correspondences between points on a pair of 3D shapes, as shown in Figure 1. Shape matching has a wide range of applications, including texture and deformation transfer [37, 46], object retrieval [14], and statistical shape analysis [4]. The problem is particularly challenging when a non-rigid deformation occurs between the shapes in the pair, such as a pose variation in human bodies. In this case, the point-wise correspondence estimation is an NP-hard quadratic combinatorial problem in the number of points of the shapes.

Functional maps, the seminal work of Ovs-janikov *et al.* [34], represented a breakthrough in the field. The key intuition behind functional maps is that it is easier to establish a correspondence among functions defined on the surfaces than directly finding a point-wise map between shapes. More specifically, functional maps builds upon a basis spanning a subspace of the functional space of each 3D shape. Then, functional correspondences can be described by a linear basis transformation, thus by a small matrix C, having dimensions equal to the number of basis functions in each shape. The matrix C is estimated by solving an optimization problem exploiting linear constraints. Therefore, functional maps provides a compact representation for shape correspondence and, at the same time, a convenient tool to transfer functions from one shape to another. However, for many applications, it is necessary to recover

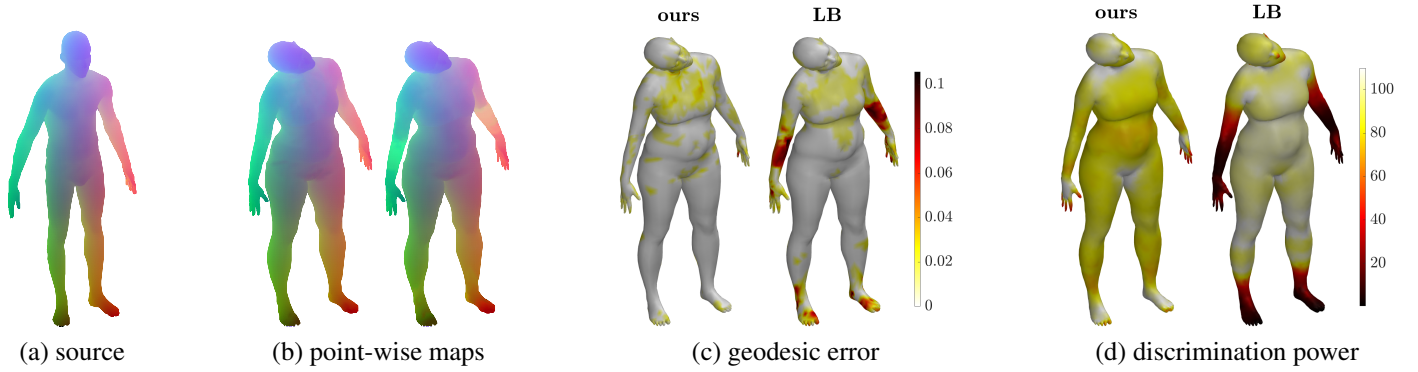| (a) source | (b) point-wise maps | (c) geodesic error | (d) discrimination power |

Figure 1: Example of shape matching between the source man (a) and the target women (b) comparing PCD (ours with a dictionary of Gaussian) and the standard basis Laplace Beltrami eigenfunctions (LB). (b) the obtained point-wise maps between the woman and the man, encoded through color coding (correspondent points are depicted with the same color). (c) the respective geodesic errors encoded by the colormap (white means perfect match while dark colors are larger errors). (d) shows the spatial distribution of the energy of the basis. The energy of our basis is uniformly distributed on the mesh, which reflects in the error distribution.

a dense point-wise map [13], for instance, when we want to collect data variations in statistical shape analysis.

Recovering a point-wise map from a functional map is far from being straightforward, and many works, starting from the seminal paper [34], have proposed different methods to accomplish this task. Recently, a few works tried to improve point-wise accuracy, focusing mainly on improving the estimation of matrix C [33, 42, 26], or the algorithm applied to convert functional maps into point-wise maps [43, 13, 42, 35].

As a matter of fact, the choice of the basis turns to be key in this point-wise conversion as it heavily affects the final result. The vast majority of methods, starting from [34], have adopted the eigenfunctions of the Laplace-Beltrami operator (LB for brevity) to define the functional bases. These eigenfunctions are the equivalent of the harmonic basis [48] for non-Euclidean surfaces. Therefore, the subset associated with the eigenvalues having the smallest absolute values is optimal for approximating smooth functions with limited variations [1]. Nonetheless, LB presents a significant limitation for the point-wise conversion as LB functions do not uniformly cover the mesh. In particular, the energy of LB functions concentrates on the flat regions at the expense of narrower extremities and tiny protrusions. In other words, LB's capability of discriminating between vertices and providing them with a meaningful

representation is uneven across the surface.

In the same spirit of previous works [18, 31, 27], we propose an innovative framework to produce new bases for the functional spaces used in functional map pipelines. More specifically, we first generate a dictionary of *informative* functions on the shape and then apply Principal Component Analysis (PCA) to obtain a compact set of orthonormal generators. The atoms of our basis are the Principal Components of a dictionary D, thus the name PCD. The rationale behind PCD is that by minimizing the reconstruction error on the atoms in D, the PCA produces a basis of a compact subspace that reflects the distribution of these functions in the initial dictionary. With PCD, we can promote the resulting basis to be, for example, evenly distributed on the surface, which is difficult to achieve with LB. At the same time, PCD is a compact set of orthonormal functions ready to use in the functional maps framework.

We show that, for some of the dictionaries we propose, our framework can generate a rich functional representation for all the points of the surfaces. Specifically, we assess the quality of such representation by the following properties: (1) the capability of discriminating between different vertices and (2) the preservation of vertex locality. We provide a quantitative evaluation of such properties for the most effective dictionaries we have investigated and LB, showing a favorable comparison in this regard. At the same time, independently from the dictionary

2

adopted, our basis shares good properties with LB: it is orthonormal and invariant to isometries by construction and empirically exhibits a weak frequency order in their components, thus being a suitable and direct replacement for LB. These properties make our representation ready to be combined with other approaches proposed to improve the results achieved by the standard LB, tackling other aspects of the functional map framework and achieving significant benefits.

Through numerous experiments on established datasets, we show that the properties of our basis improve the quality of the final point-wise map. The similarity between our basis and LB enables us to directly compare them in the same setting. Thus, we test PCD and LB when injected in different shape-matching pipelines [33, 26] and also using functional maps computed from ground-truth correspondence. In these experiments, we see that our basis reaches significantly higher values of point-wise accuracy in all the analyzed settings. Our code is publicly available at `https://github.com/michele-colombo/PC-Gau_STAG2022`.

This work extends the idea proposed in *"PC-GAU: PCA Basis of Scattered Gaussians for Shape Matching via Functional Maps"* [7], where this procedure was applied only to a dictionary consisting of a collection of Gaussian functions scattered on the surface. Our major extensions with respect to [7] consists in generalizing the procedure and investigating different definitions of the initial dictionary (see Section 6). Moreover, we assess the role and the actual dependence of PCD from its more relevant parameters (Section 7) and we provide additional quantitative evaluation by expanding the test sets and scenarios of our experiments.

2. Related work

In this Section, we briefly overview the existing shape matching methods that fall, similarly to ours, into the functional map framework [34]. For other approaches to shape matching, refer to the survey [49, 45].

The functional map framework was proposed by Ovsjanikov *et al.* in [34]. Its core idea is first to estimate the correspondence between functional spaces defined on the meshes and then extract a point-wise map from it. Given a basis for the functional space, the functional correspondence is compactly represented as a matrix C, which is estimated by imposing linear functional constraints. These constraints express correspondent landmarks, segments, and descriptors between the two shapes. The most used descriptors in this context are invariant to near-isometric deformations, like heat diffusion [47] or quantum mechanical properties [3].

Additional constraints can improve the estimation of the matrix C. Commutativity of the functional map with the LB operator, which forces the matrix C to represent isometries, was firstly proposed in [34] and then refined in [41]. In [33], the authors show that imposing the preservation of point-wise products is an additional, beneficial constraint. In shape with symmetries, the preferred maps do not mix up symmetries. This property has been promoted by introducing a new term in the optimization [42] or considering the complex counterpart of the functional map [11]. An efficient method for converting a functional map into a point-wise map by exploiting an iterative closest point (ICP) as a form of refinement was proposed in the original framework [34]. Subsequent works have proposed alternative solutions for extracting better point-wise maps from functional maps [43], by considering a probabilistic model [44], by introducing a smoothness prior on the point-wise map [13], or by devising a complex refinement scheme to promote continuity, coverage and bijectivity of the obtained map [42]. ZoomOut [26] is an iterative method for extending the size of an initial functional map estimated with few eigenvectors, while improving the quality of the estimated correspondence. Many recent algorithms build upon the ZoomOut procedure [15, 39, 35, 40, 36], which alternates conversions to point-wise maps and back to functional maps of increased size. Despite pursuing the general goal of improving the quality of shape-matching via functional maps, all these methods differ radically from our approach since they focus either on the estimation of C or on the process of extracting the point-wise map from C. They all make the implicit choice, ubiquitous since [34], of using eigenfunctions of the Laplace-Beltrami operator [19, 48] as the functional basis, even though

3

they are in principle agnostic to the basis adopted. This fact makes our proposal *complementary* to most methods in the literature, as they can be used in combination with our basis. In perfect agreement with our intuition, [32] and [21] showed that extending the LB basis by adding point-wise products of atoms provides excellent benefits to the resulting point-wise maps. In principle, the same procedure can be applied to any basis, including ours.

A few other works followed an approach similar to ours, proposing a new bases for the functional space of a mesh. Some of them target specific tasks, like the transfer of tessellation structure [24] or step functions [23] from one mesh to another. The method in [24] has also been used for shape matching, showing specific improvements in parts of the human body such as hands and feet. This solution, however, requires the meshes in the pair to be in similar poses, severely limiting its use in the general context. A different solution is pursued in [31], which proposes a basis whose atoms have local support, thus promoting the sparsity of C, but without improving the point-wise accuracy of the standard LB. Kovnatsky *et al.* [18] built a coupled pair of bases by joint diagonalization to overcome the instability of LB in non-isometric pairs. Since the atoms of these coupled bases approximately diagonalize the Laplace-Beltrami operator, their energy will distribute similarly to the one of LB. Furthermore, a functional basis obtained with [18] also depends on the other mesh in the pair and the landmarks used. With a purpose more similar to ours, Melzi *et al.* [27] proposed a basis that can extend LB and improve its expressive powers in specific mesh regions. However, their approach differs from ours: the atoms of their bases localize in predefined areas, which must be provided as input. This is a non-trivial information to be provided, as it requires the knowledge of at least coarse correspondences between the two shapes. On the contrary, our framework generally does not require any input other than the mesh itself.

This work extends [7], which proposes to extract a new basis by PCA from a dictionary of Gaussian functions scattered uniformly on the shape. In this extended version, we improve the proposed procedure by considering a variety of dictionaries featuring different properties. Furthermore, we extend the analysis of the parameters of the method and evaluate the pipeline by considering additional test sets. Our novel contributions highlight the potential of extracting basis by PCA from a redundant dictionary in general. This paper aims to pave the way for further exploration of novel bases to improve the accuracy of the functional map framework.

3. Background

In this section, we introduce 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 surfaces embedded in $\mathbb{R}^3$. In the continuous setting, we can represent these surfaces as a compact and connected smooth 2-dimensional Riemannian manifold $\mathcal{M} \subset \mathbb{R}^3$. We refer to [10] for the definitions of differential geometry that are necessary to deal with this continuous representation of $\mathcal{M}$. In the discrete setting we represent $\mathcal{M}$ as a triangular mesh $\mathcal{M} = (V_\mathcal{M}, E_\mathcal{M})$, where $V_\mathcal{M}$ is the set of n vertices and $E_\mathcal{M}$ is the list of edges which means that $e_{xj} \in E_\mathcal{M} \iff$ exists an edge that connect x and $j, \forall x, j \in V_\mathcal{M}$. 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 shape matching is usually composed of two discrete meshes $\mathcal{M}$ and $\mathcal{N}$, having sets of vertices $V_\mathcal{M}$ and $V_\mathcal{N}$, respectively. Here we consider $|V_\mathcal{M}| = |V_\mathcal{N}| = n$ for simplicity of notation, but this is not a necessary assumption. We assume that an unknown correspondence $T : V_\mathcal{N} \to V_\mathcal{M}$ between $\mathcal{M}$ and $\mathcal{N}$ exists without specific requirements on the function T. For instance, $\mathcal{M}$ can be a non-rigid deformation of $\mathcal{N}$. We will refer to T as the ground truth point-wise map and we represent it either as a vector of vertex indices of size n or as an $n \times n$ matrix Π such that $\Pi_{ij} = 1$ if $T(i) = j$ and 0 otherwise, $\forall i \in V_\mathcal{N}$ and $\forall j \in V_\mathcal{M}$.

The goal of shape matching is, given $\mathcal{M}$ and $\mathcal{N}$, to estimate the unknown map T. The estimated map $\bar{T}$ has to be as close as possible to T, which means

that $\bar{T}$ should assign to each vertex $y \in V_N$ a vertex on $\mathcal{M}$ that is geodesically close, ideally coincident, to the one associated by T. When the ground truth map T is available, we assess the mapping error of each vertex $y \in V_N$ as:

$$e(y) = \text{GD}_{\mathcal{M}}(\bar{T}(y), T(y)), \tag{1}$$

where $\text{GD}_{\mathcal{M}}$ is the geodesic distance on the surface $\mathcal{M}$. Figure 1b presents an example of two point-wise maps rendered through color correspondence, where we depicted correspondent points with the same color. Moreover, in Figure 1c, we visualize their respective geodesic errors, encoded by the colormap where 0 error corresponds to white while dark colors denote larger ones.

Some shape matching pipelines require a set of input landmarks. A landmark is a couple of points $(y \in V_N, x \in V_M)$ in known correspondence, namely $T(y) = x$. In some of the experiments in Section 7, we will refer to landmarks, which are not required to build the bases but only for the specific matching pipelines.

3.3. Functional maps

In the discrete setting, a real-valued function f on $\mathcal{M}$ is given by a vector that associates to each vertex $x \in V_M$ a value $f(x) \in \mathbb{R}$. We call $\mathcal{F}(\mathcal{M}, \mathbb{R})$ the space of such functions. A basis for $\mathcal{F}(\mathcal{M}, \mathbb{R})$ is a set of orthonormal functions belonging to $\mathcal{F}(\mathcal{M}, \mathbb{R})$, an example of basis is shown in Figure 2. Given a pair of discrete surfaces $\mathcal{M}$ and $\mathcal{N}$ related by a ground truth point-wise map $T : V_N \rightarrow V_M$, the functional map framework [34], instead of estimating T directly, searches for a correspondence among functions defined on $\mathcal{M}$ and $\mathcal{N}$. Then, once the *functional correspondence* has been estimated, it extracts the corresponding point-wise map.

Functional maps builds 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:

$$T_F(f) = f \circ T \qquad \forall f \in \mathcal{F}(\mathcal{M}, \mathbb{R}) \tag{2}$$

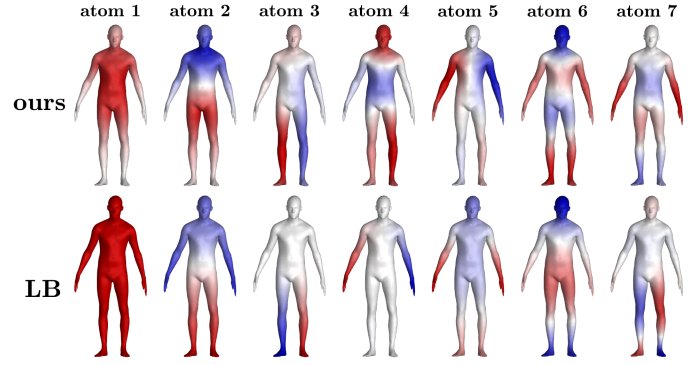Given a pair of bases $\Phi = \{\phi\}_i$ and $\Psi = \{\psi\}_j$ for

Figure 2: Atoms of our basis (top) compared to LB basis (bottom) atoms. Each element of the bases is a function defined on the mesh, represented here through color. Positive values are red, negative values are blue, and white corresponds to zero.

$\mathcal{F}(\mathcal{M}, \mathbb{R})$ and $\mathcal{F}(\mathcal{N}, \mathbb{R})$, respectively, we can write

$$g = T_F(f) = T_F\Big(\sum_i a_i \phi_i\Big) = \sum_i a_i T_F(\phi_i) =$$

$$= \sum_i a_i \sum_j c_{ji} \psi_j = \sum_{ji} a_i c_{ji} \psi_j = \sum_j b_j \psi_j$$

where $\mathbf{a} = [a_i]$ and $\mathbf{b} = [b_j]$ are the projections of f and g on Φ and Ψ respectively. c_{ji} is the projection of $T_F(\phi_i)$ on ψ_j and depends only on T_F and the two bases. Therefore T_F is compactly represented by the matrix $C = [c_{ij}]$ and $\mathbf{b} = C \cdot \mathbf{a}$.

In practice, we consider only the first k atoms of the bases, truncating the previous series after the first k coefficients. As a matter of fact, k is independent of the number of vertices n of the meshes and usually $k \ll n$. Therefore, matching two shapes in the functional map framework boils down to the estimation of a matrix C of size $k \times k$. We can represent landmarks, corresponding segments, and descriptors as functions defined on $\mathcal{M}$ and $\mathcal{N}$ and find the functional map C that best preserves the functional constraints (in the least square sense) introduced by these corresponding functions. Note that, in principle, we can truncate the basis of $\mathcal{M}$ and $\mathcal{N}$ at a different number of atoms k_M and k_N, thus producing a rectangular $C \in \mathbb{R}^{k_N \times k_M}$. However, for the sake of simplicity, we consider $k_M = k_N = k$ in this paper.

Embedding. Once we truncate a basis for $\mathcal{F}(\mathcal{M}, \mathbb{R})$ to size k, we can store it in a matrix $\Phi_{\mathcal{M}}$, where each column is a basis atom represented as a vector of real values. $\Phi_{\mathcal{M}}$ has thus size $n \times k$, where

n is the number of vertices and k is the number of basis atoms considered. We call *spectral embedding* (or simply *embedding*) of a vertex $x \in \mathcal{M}$, the vector of values assumed by all the basis functions in x: $\text{Emb}(x) = [\phi_i(x)] \in \mathbb{R}^k$. $\text{Emb}(x)$ also corresponds to the coefficients, in the basis $\Phi_{\mathcal{M}}$, of a Delta function centered in the vertex x. Thus $\Phi_{\mathcal{M}}^T$ contains the coefficients of all the Delta functions of $\mathcal{M}$ (one for each vertex) as column vectors.

3.4. Conversion to point-wise map

Functional maps estimates a mapping between functional spaces built over $\mathcal{M}$ and $\mathcal{N}$, and this needs to be converted into a point-wise map $\bar{T} : V_{\mathcal{N}} \rightarrow V_{\mathcal{M}}$. A simple solution, proposed in [34], consists in finding, for each column of $\Phi_{\mathcal{N}}^T$, the nearest neighbour in the columns of $C\Phi_{\mathcal{M}}^T$. This procedure corresponds to transfer Delta functions, and thus embeddings of vertices, from $\mathcal{M}$ to $\mathcal{N}$ through C and putting similar points in the embedding space in correspondence.

Summarizing. The complete functional-map pipeline for shape matching, as detailed in [34], is: (1) find a truncated basis of size k on each mesh, (2) find the $C \in \mathbb{R}^{k \times k}$ that best preserves some functional constraints, and (3) convert the functional map C to a dense point-wise map $\bar{T}$. In Figure 3, we report an illustration of the entire procedure, focusing on step (1) since it is the one that our method addresses.

3.5. Standard basis from LB Operator

The Laplace-Beltrami operator $\Delta_{\mathcal{M}} : \mathcal{F}(\mathcal{M}, \mathbb{R}) \rightarrow \mathcal{F}(\mathcal{M}, \mathbb{R})$ associates to each function $f \in \mathcal{F}(\mathcal{M}, \mathbb{R})$ another function that is the divergence of the gradient of f. For discrete meshes, this operator corresponds to a $n \times n$ matrix $\Delta_{\mathcal{M}}$ and is usually computed using the cotangent scheme [28, 38]. $\Delta_{\mathcal{M}}$ admits an eigendecomposition: $\Delta_{\mathcal{M}}\phi_i = \lambda_i\phi_i$, where $\Phi = \{\phi_1, \phi_2, \ldots\}$ are its eigenfunctions with corresponding real eigenvalues $\Lambda = \{\lambda_1 \leq \lambda_2 \leq \ldots\}$. Φ forms an orthonormal basis for $\mathcal{F}(\mathcal{M}, \mathbb{R})$, namely LB. LB atoms are ordered in increasing frequencies which are encoded in the corresponding eigenvalue. We depict this order qualitatively in Figure 2 (bottom row) and quantitatively in Figure 4. Here, by frequency of a function

$f \in \mathcal{F}(\mathcal{M}, \mathbb{R})$, we intend the Dirichlet energy of f, which is a measure of its smoothness. LB is considered as the mesh equivalent of the harmonic basis [19, 48]. By selecting only the first k atoms we obtain an optimal [1] low-pass filter approximation of functions in $\mathcal{F}(\mathcal{M}, \mathbb{R})$. From [34] on, LB has been the standard choice of the functional basis on meshes.

3.6. Motivation

Despite its many strengths, LB presents a significant limitation: the basis energy is not evenly distributed on the mesh surface but is concentrated in massive areas, leaving narrower extremities less covered. To define the energy of a basis, we consider two properties in particular:

– discrimination power between different vertices,

– locality preservation.

As we demonstrate through our experiments in Section 5.3, the lack of these properties in some areas of the mesh produces bad assignments in the point-wise map, affecting its overall accuracy. Instead, the shape matching task requires a functional basis whose energy is evenly distributed on the mesh, i.e., the value of the previous two properties is similar across different mesh areas and better overall.

4. Proposed method

This section presents our core contribution PCD, namely Principal Component of a Dictionary: a new framework to build a basis for the functional space defined on a triangular mesh. The produced basis plays the role of the truncated basis in functional maps pipelines for shape matching. In other words, the output of PCD can be a replacement for LB to obtain more accurate point-wise maps.

We first illustrate the proposed procedure to build the basis starting from a given dictionary of functions D. Then, in Section 5, we describe the procedure to construct the dictionary exploited in [7] and analyze the properties that directly arise from its construction, paying particular attention to the *desiderata* expressed in Section 3.6. Finally, in Section 6, we propose six alternative dictionaries that can be compared to LB and to the solution from [7].

Algorithm 1 The PCD framework

1: **input:** D, $A_\mathcal{M}$, k, *normalize*
2: $n = size(D, 1)$ = number of vertices
3: $q = size(D, 2)$ = number of atoms
4: **if** *normalize* **then**
5:     **for** $i = \{1, \ldots, n\}$, $j = \{1, \ldots q\}$ **do**
6:         $D_{ij} = D_{ij} / \sqrt{(D_{*j})^T A_\mathcal{M} D_{*j}}$
7:     **end for**
8: **end if**
9: $P = \text{PCA}\left(D^T, \text{variableWeights} = A_\mathcal{M}\right)$
10: $P_k = P(:, 1:k)$
11: **output:** P_k

4.1. Building procedure

Given a dictionary D composed of $q \in \mathbb{N}$ functions, we perform dimensionality reduction through Principal Component Analysis (PCA), producing an orthogonal set of generators for the functional space spanned by the atoms in D. Generators from PCA are ordered according to their capability to approximate the initial dictionary, therefore we select the only the first components. In the following, we present the pipeline to build our basis for any triangular mesh, which is summarized in Algorithm 1. In Figure 3 we report a scheme illustrating the entire pipeline, which includes the construction of the dictionary of Gaussians proposed in [7].

Input dictionary and first steps. Given a Mesh $\mathcal{M}$ with n vertices, the input of our pipeline comprises two main ingredients and two parameters. The first ingredient is a dictionary $D \in \mathbb{R}^{n \times q}$, composed of q functions defined on $\mathcal{M}$. The second one is $A_\mathcal{M} \in \mathbb{R}^{n \times n}$, the mass matrix of $\mathcal{M}$, which is a diagonal matrix whose entry $A_\mathcal{M}(i, i)$ corresponds to the area elements of the surface associated to the vertex i. The necessary parameters are an integer k which sets the dimension of the generated basis, and *normalize*, a boolean variable which activates the normalization step. D is a fundamental, and almost the only choice in our pipeline. For this reason, we dedicate two sections to its definition, in which we show some possibilities and their properties. For now, let us consider a given, generic dictionary D, leaving deeper discussion on its choice to the following sections.

The algorithm starts by extracting the dimensionality of the mesh, n (line 2), and of the dictionary, q (line3), from D. In lines 4-8, if the variable *normalize* is set to true, we normalize each column D_{*j} of D by division with its norm, computed as $\|D_{*j}\|_\mathcal{M} = \sqrt{D_{*j}^T A_\mathcal{M} D_{*j}}$, where $A_\mathcal{M}$ defines the inner product on the functional space $\langle f, g \rangle_{\mathcal{F}(\mathcal{M}, \mathbb{R})} = f^\top A_\mathcal{M} g$, $\forall f, g \in \mathcal{F}(\mathcal{M}, \mathbb{R})$.

Dimensionality reduction. We then compute the PCA of D^T (line 9), meaning that each atom in the dictionary D is considered a sample and each vertex of the mesh a variable. We do not centre the variables, but we weigh them for $A_\mathcal{M}(i, i)$, the element of area associated with each vertex. The result of PCA is P, a set of q vectors of size n, called Principal Components (PCs). We can interpret these vectors as q generators of the functional space S spanned by the atoms in D. Since $q < n$, S is a proper subspace of $\mathcal{F}(\mathcal{M}, \mathbb{R})$ and, if the atoms are linearly independent, S has dimension q.

Finally, we select the first k PCs to form our basis, and we store them in a matrix P_k of size $n \times k$ (line 10). The columns of this matrix are a truncated basis of $\mathcal{F}(\mathcal{M}, \mathbb{R})$ or, equivalently, a basis of a k-dimensional subspace $R \subset S \subset \mathcal{F}(\mathcal{M}, \mathbb{R})$. In particular, PCA transform guarantees that the first k PCs form the set of the k orthonormal generators with the lowest approximation error on the initial samples [2]:

$$P_k = \underset{P_k \in \mathbb{R}^{n \times k}}{\arg\min} \left\{ \sum_{i=1}^{q} \left\| D_{*i} - P_k P_k^T A_\mathcal{M} D_{*i} \right\|_2^2 \right\} \quad (3)$$
$$\text{s.t.} \quad P_k^T A_\mathcal{M} P_k = I_k$$

where D_{*i} is the i-th column of D, namely the i-th atoms of the dictionary, and I_k is the identity matrix of size k. Orthonormality and projection in (3) are expressed with respect to the inner product defined on the mesh $\langle \cdot, \cdot \rangle_{\mathcal{F}(\mathcal{M}, \mathbb{R})}$. We remark that the truncated basis P_k satisfies Equation (3), thus provides a good approximation of D by the spanned subspace S.

5. The dictionary of Gaussians

In [7], we consider a single choice of dictionary D, which we construct by taking inspiration from the signal processing literature [9]. More
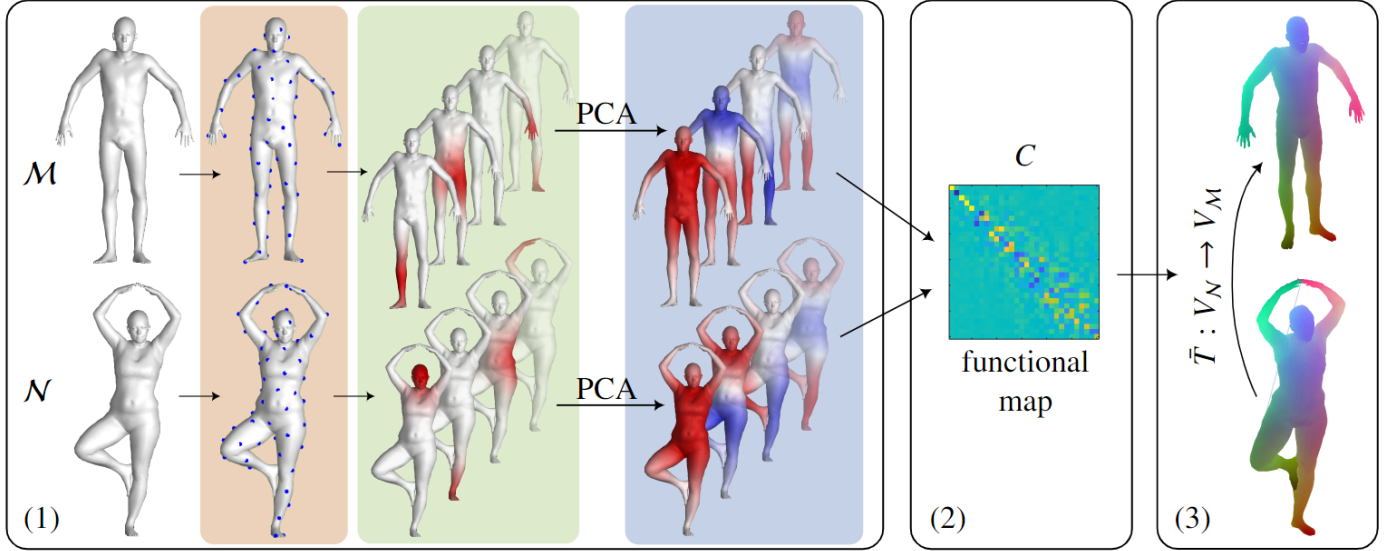
Figure 3: Complete shape matching pipeline with functional maps: (1) definition of a functional basis, (2) estimation of *C* and (3) conversion to a point-wise map. Here, step (1) shows the building procedure proposed in [7]: selection of a subset of vertices (orange box), construction of the dictionary of Gaussian functions (green box) and dimensionality reduction through PCA (blue box).

specifically, we consider a dictionary D composed of a collection of Gaussian functions centred in a subset of vertices that uniformly cover the surface of the input mesh. Despite its simplicity, this basis enjoys many desirable properties for devising accurate shape correspondences via functional maps. Indeed, we show through a broad set of quantitative evaluations that the obtained basis is evenly distributed on the mesh surface and guarantees excellent discrimination power and locality preservation. In this section, we describe the procedure to define the dictionary of Gaussians and then analyze the properties that give rise to its superior accuracy in shape matching.

5.1. Design of the dictionary of Gaussians

In the following, we present the steps required to construct the dictionary of Gaussians for a mesh $\mathcal{M} = (V_{\mathcal{M}}, E_{\mathcal{M}})$ with n vertices. In Algorithm 2, we report in a compact format the proposed procedure which we represent visually on the left of Figure 3 (1).

Subset of vertices. We start by selecting a subset $Q \subseteq V_{\mathcal{M}}$ of q vertices on the mesh with Farthest Point Sampling [30] (line 2), using Euclidean distance for efficiency reasons. Thus is not the only viable critera, as discussed in Section 7.1. The orange box in

---

Algorithm 2 Construction of the dictionary of Gaussians

1: **input:** $\mathcal{M}$, $V_{\mathcal{M}}$, σ, q
2: $Q = \mathrm{FPS}(\mathcal{M}, q)$
3: **for** $i \in V_{\mathcal{M}}, j \in Q$ **do**
4:     $G_{ij} = \mathrm{GD}_{\mathcal{M}}(i, j)$
5:     $D_{ij} = \exp\left(-G_{ij}^2/\sigma\right)$
6: **end for**
7: **output:** D

Figure 3 shows that the sampled points evenly cover the mesh surface. The idea is to obtain an even distribution of basis energy by controlling the uniformity of scattering of the Gaussians in the dictionary, since uniform sampling is rather easy to enforce.

Gaussian computation. We compute q Gaussian functions, each one centered in a vertex of Q. To do so, for each vertex $j \in Q$ (line 3) we compute the geodesic distance $G_{ij} = \mathrm{GD}_{\mathcal{M}}(i, j)$ to any other vertex $i \in V_{\mathcal{M}}$ (line 4) and then define the value of the Gaussian D_{ij} (line 5) as:

$$D_{ij} = \exp\left(\frac{-G_{ij}^2}{\sigma}\right). \qquad (4)$$

8

We approximate the geodesic distance as the length of the shortest path on the edges of the mesh [29]. The parameter σ is arbitrarily chosen and sets the amplitude of the Gaussians. We store the q Gaussian functions as columns of the matrix $D = [D_{*1}, D_{*2}, \ldots, D_{*q}]$, of size $n \times q$. In the green box in Figure 3, we report some examples of the resulting Gaussian functions. In all the experiments we use $q = 1000$, $\sigma = 0.05$ and we apply normalization. In Section 7.5, we analyze the impact of these two parameters on the accuracy of the point-to-point correspondence computed by our complete pipeline. These experiments were not included in [7] and validate our choices. Once we have the dictionary D, we can apply the steps described in Section 4 to obtain a basis of dimension k for the functional space $\mathcal{F}(\mathcal{M}, \mathbb{R})$. Coherently with [7], we refer to this basis as PC-Gau or *ours*.

5.2. Properties of PC-Gau

The basis obtained in such a way shares many of the good properties of LB, which makes PC-Gau a suitable replacement for LB in existing functional map pipelines.
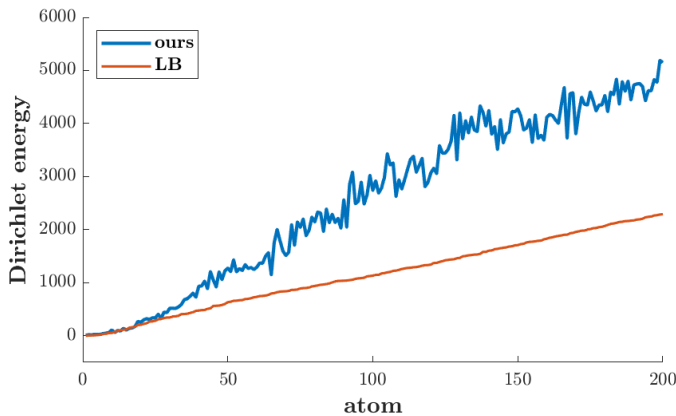

Figure 4: Comparison between the Dirichlet energy (frequency) of atoms of our basis and LB, computed for an example mesh from FAUST. Atoms of PC-Gau are approximately ordered by increasing frequency.

5.2.1. Frequency ordering

Figure 4 compares the Dirichlet energy of the atoms of PC-Gau and LB. The Dirichlet energy measures the smoothness of a function and can be interpreted as its frequency. We observe that, although

not perfectly, atoms of *ours* are approximately ordered in frequency. Even if not explicitly imposed, this order arises naturally in our basis as we empirically observe. We can also evaluate this qualitatively in the example atoms shown in Figure 2. Frequency ordering means that, similarly to LB, we operate a low-pass filter approximation when we project a function on our truncated basis. We consider sorting our atoms with respect to their frequency as an interesting research direction. Still, we believe that their natural order is sufficient for our goal, and we leave its explicit constraint as future work.

5.2.2. Orthonormality

PC-Gau, like LB, is an orthonormal basis. We say that a basis Φ is orthonormal according to the inner product of the functional space $\mathcal{F}(\mathcal{M}, \mathbb{R})$ if $\Phi^T A_{\mathcal{M}} \Phi = I_k$, where $A_{\mathcal{M}}$ is the mass matrix and I_k is the identity matrix of size k. We can project a function f on an orthonormal basis $\Phi_{\mathcal{M}}$ simply by matrix multiplication: $\mathbf{a} = \Phi_{\mathcal{M}}^T A_{\mathcal{M}} f$. Similarly, we can recover the function from its projection through $f = \Phi \mathbf{a}$. This is useful, in particular, when converting a given point-wise map $\Pi : \mathcal{N} \to \mathcal{M}$ (represented here as a matrix) to a functional map C, which boils down to:

$$C = \Phi_{\mathcal{N}}^T A_{\mathcal{M}} \Pi \Phi_{\mathcal{M}}. \tag{5}$$

5.2.3. Isometry-invariance

Two meshes $\mathcal{M}$ and $\mathcal{N}$ are isometric if the underlying correspondence $T : V_{\mathcal{N}} \to V_{\mathcal{M}}$ is an isometry, which means that it preserves the geodesic distance of any pair of vertices on $\mathcal{N}$:

$$\mathrm{GD}_{\mathcal{M}}(T(x), T(y)) = \mathrm{GD}_{\mathcal{N}}(x, y) \quad \forall x, y \in V_{\mathcal{N}}. \tag{6}$$

Since we construct our basis purely on geodesic distances, under the assumption of equal distribution of the sample points, then our basis will be isometry-invariant. As for the LB, when shapes are isometric, the functional map is a diagonal matrix with $+1$ or -1 on the diagonal. Most often, the energy of C is funnel-shaped, approaching diagonal as the relation between $\mathcal{M}$ and $\mathcal{N}$ gets closer to isometry. In Figure 5, we show a qualitative comparison of the structure of the functional map computed with LB (left) and with PC-Gau (right) for a near-isometric (top row: two poses of the same human subject) and
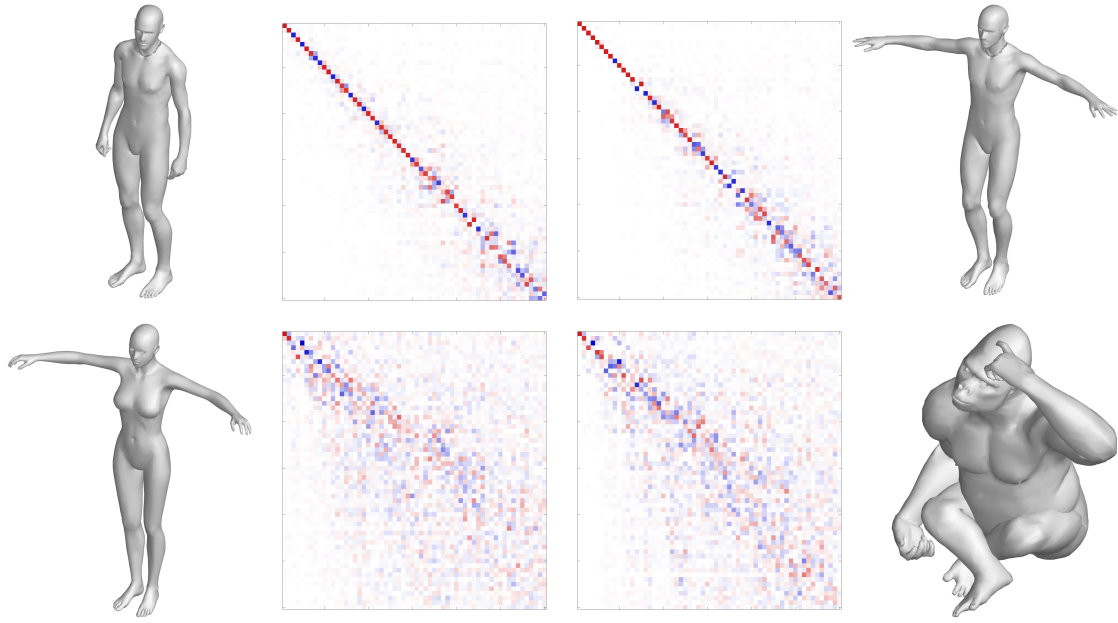
Figure 5: Example of functional maps in case of a near-isometric pair (top) and a non-isometric pair (bottom), computed using the ground-truth correspondence. By comparing the structure of the matrix for PC-Gᴀᴜ (left) and LB (right), we can see that their behaviour with respect to isometry is similar.

a non-isometric pair (bottom row: a woman and a gorilla in different poses). We remark that both bases in the two settings generate a matrix C with a similar shape: close to diagonal for the near-isometric pair and a funnel shape for the non-isometric one. Their similarity further support our claim that the PC-Gᴀᴜ exhibits an invariance to isometry similar to the one of LB.

5.3. Spatial distribution of basis energy

We now consider the spatial distribution of the energy of the basis on the mesh surface. Intuitively, by basis energy in a vertex x, we mean the expressive power and the quality of the embedding space in a neighborhood of x. More precisely, as we stated in Section 3.6, we introduce two quantities to assess the quality of the embedding space: (1) the discrimination power and (2) the preservation of locality.

5.3.1. Discrimination power

We define the *discrimination power* as the capability of a basis to assign sufficiently different embeddings to different vertices. For each vertex $x \in$

$V_{\mathcal{M}}$, we measure it with the following metric:

$$\text{Dis}(x) = \frac{\|\text{Emb}(x) - \text{Emb}(y_x)\|_2}{\text{GD}_{\mathcal{M}}(x, y_x)}, \quad (7)$$

$$\text{where } y_x = \operatorname*{argmin}_{z \in V_{\mathcal{M}} \setminus \{x\}} (\|\text{Emb}(z) - \text{Emb}(x)\|_2) \quad (8)$$

Here, y_x is the vertex having the closest embedding to $\text{Emb}(x)$ measured through Euclidean distance. The normalization makes the metric independent of vertex density and rewards geodesic proximity between x and y_x. According to the continuous nature of the surface, it is reasonable to require that geodesically close points have similar embeddings. For this reason, we penalize a vertex having as the nearest point in the embedding a vertex that is geodesically far.

To assess the quality of PC-Gᴀᴜ embedding with respect to LB, we compare the distribution of their discrimination power over the mesh. Moreover, we investigate the relation between discrimination power and the average geodesic error, when the mesh is matched to another. On the left of Figure 6, we report the average distribution of $\text{Dis}(x)$ on meshes obtained across 40 human meshes from SHREC19 (top row) and three cat meshes from TOSCA (bottom row). On the right of Figure 6, we visualize the
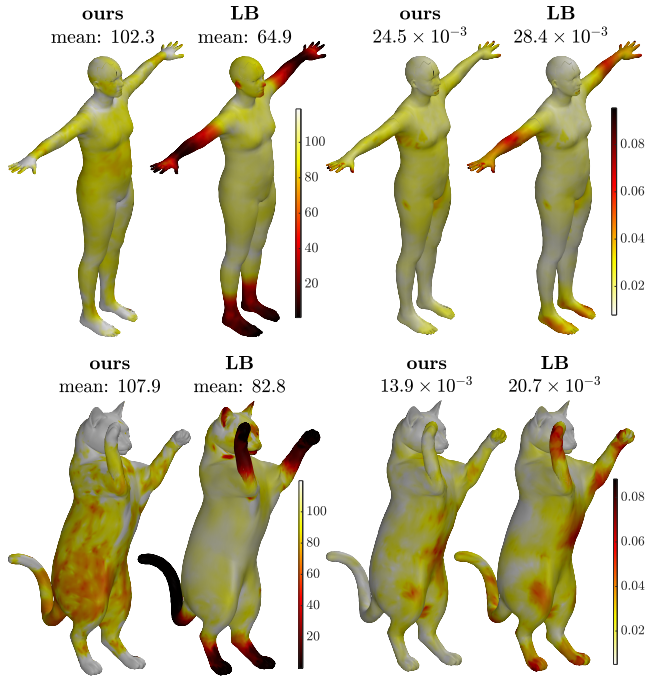
Figure 6: Spatial distribution of discrimination power of the basis and geodesic error. Comparison between *ours* and LB. We visualize the average results across 40 human meshes from SHREC19 (top) and three cat meshes from TOSCA (bottom). Darker is worse in all cases. The similarity between distributions suggests a correlation between the error and the local quality of the embedding space.

average distribution of the geodesic error on the corresponding datasets matched using LB and PC-Gau as described in Section 7.1. We observe that:

1. LB has a much lower discrimination power on narrow extremities (arms, feet, paws, and tail), while *ours* presents a uniform discrimination power similar to the best level achieved by LB.

2. The error for LB is coherently localized in narrow extremities, while our diffuses uniformly on the surface.

These observations support our claim that the errors in point-wise correspondence localize in the areas where the basis provides a less discriminative embedding. This result suggests that through PC-Gau, we can improve such correspondences by making the quality of the embedding space more uniform on the mesh.

5.3.2. Locality preservation

We define the following two metrics to quantitatively asses locality preservation of a given embedding space.

Embedding/Geodesic Distance Correlation (EGDC). For each vertex x, we select the set S of $s \in \mathbb{N}$ vertices having the embedding closest to $\text{Emb}(x)$, and estimate the correlation (ρ) between the Euclidean distances of the embeddings and their geodesic distances from x:

$$\text{EGDC}(x) = \rho \left(\text{GD}_{\mathcal{M}}(x, y), \|\text{Emb}(x) - \text{Emb}(y)\|_2 \right)_{y \in S}$$
(9)

This measure evaluates how much an embedding space preserves the neighborhood of x in the geodesic sense. The higher EGDC, the better. In our experiments we assess $\text{EGDC}(x)$ using $s = 80$ vertices.

Mean Geodesic Distance (MGD). For each vertex x, we define the set R of the $t \in \mathbb{N}$ vertices having the lowest embedding distance to $\text{Emb}(x)$, and the set $\bar{R}$ of the t vertices having the lowest geodesic distance from x. Then, we compute the ratio between the mean geodesic distance of the vertices in R and $\bar{R}$:

$$\text{MGD}(x) = \frac{\text{Avg}_{y \in R} \{\text{GD}_{\mathcal{M}}(x, y)\}}{\text{Avg}_{z \in \bar{R}} \{\text{GD}_{\mathcal{M}}(x, z)\}}$$
(10)

The more the embedding preserves the distance of the geodesic neighbor, the lower the value of $\text{MGD}(x)$, which approaches 1. Therefore, the lower $\text{MGD}(x)$ the better. In the evaluation we used $t = 10$.

We analyze the locality preservation of PC-Gau, compared to LB exploiting EGDC and MGD. We claim that point-wise embedding designed for shape matching should encode the neighbor information, preserving the relation of nearest and potentially similar points. Figure 7 shows the value of these metrics averaged on the same shapes from SHREC19 (top) and TOSCA (bottom) involved in Figure 6. The spatial distribution measured with EGDC and MGD for LB closely mimics the distribution of discrimination power (see Figure 6, first column). The three similar distributions observed in these results support the choice of these measures to evaluate the quality of the embedding in different regions of the surfaces. In Table 1, we present
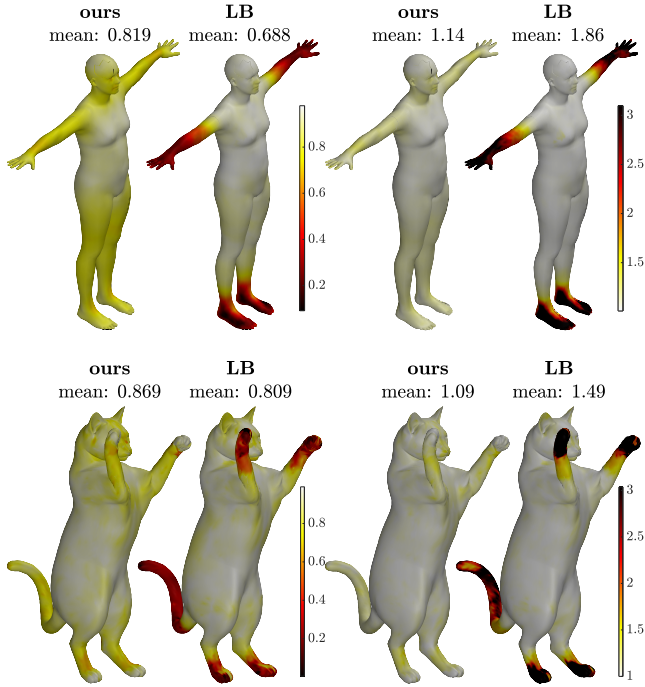
11

Figure 7: Spatial distribution of EGDC and MGD. Comparison between *ours* and LB on average on human shapes from SHREC19 (top) and cats from TOSCA (bottom). Darker is worse in all cases. LB presents low locality preservation on the extremities (arms, feet, paws, and tail).

dataset	EGDC		MGD	
	ours $\times 10^{-2}$	**LB** $\times 10^{-2}$	**ours**	**LB**
FAUST	**79,7**	73,8	**1,11**	1,34
MWG	**83,8**	78,1	**1,12**	1,36
TOSCA	**84,1**	79,0	**1,12**	1,40
SHREC19	**81,9**	68,8	**1,14**	1,86

Table 1: Overall values of EGDC and MGD averaged on the meshes of different datasets.

previous section, that we showed to be relevant to improve the estimation of point-to-point correspondences. We briefly introduce each of them and refer the interested reader to the related literature for more details on each derivation.

*Adaptative Gaussians (***ADAPT***).* To define the Gaussian functions for PC-GAU, we fix the value of σ and define for all the samples in Q a Gaussian having the same covariance. Due to the diverse local structure of different regions on the surface, this choice can be suboptimal. For this reason, we define a procedure to vary the value of σ on the surface to adapt the definition of each Gaussian to the local geometry in the neighborhood of its vertex.

We proceed as follows:

1. We compute the error of reconstruction of the geometry (*RECerr*) provided by an LB basis composed of 60 eigenfunctions that is:

$$RECerr = \|X_{\mathcal{M}} - \Phi_{\mathcal{M}}\Phi_{\mathcal{M}}^{\dagger}X_{\mathcal{M}}\|_2, \qquad (11)$$

where $X_{\mathcal{M}}$ are the vectors containing the 3D coordinates of the vertices of $\mathcal{M}$. *RECerr* is a function that encodes, for each vertex, the error in the reconstruction given by the selected LB basis. The larger the value of *RECerr*, the larger the error in the reconstruction.

2. We normalize *RECerr* within the interval [0, 1].

3. We define and adaptive $\widetilde{\sigma}(x)$ at each sample $x \in \mathcal{M}$ as:

$$\widetilde{\sigma}(x) = \sigma(1 - (\frac{2}{5}RECerr)). \qquad (12)$$

the overall value for EGDC and MGD (computed as the average on the vertices of each mesh) for different datasets that we introduce in the next section. These results support our claim: a more even distribution of the energy of a basis, as exhibited by the one we propose, increases the overall quality of the embedding space. PC-GAU inherits these properties from the Gaussian dictionary we propose. Providing a good approximation of the dictionary, and thus of the subspace S, and satisfying (3), our truncated basis P_k implicitly preserves the uniform distribution owned by the Gaussian functions.

6. Alternative Dictionaries

Even if the PC-GAU basis from [7] already achieves the desired performance as a replacement for the LB in the functional map framework, the new PCD framework is more general and applies to any dictionary D constructed on a triangular mesh. In this section, we propose six alternative dictionaries, motivated both by the previous literature in geometry processing and by the properties analyzed in the

12

This formula restricts the value of the adaptive $\widetilde{\sigma}$ in the interval $[\frac{3}{5}\sigma, \sigma]$. More specifically, $\widetilde{\sigma}(x)$ is smaller when the reconstruction error is larger. So we want to reduce the amplitude of the Gaussian in the regions where the representation provided by LB is worse. As for all our experiments, we set $\sigma = 0.05$ and refer to this dictionary as **ADAPT**. This is not the only viable solution to produce an adaptive dictionary. In our procedure, we have tested different definitions varying the formula to compute *RECerr*, the size of the LB basis, and the value of $\frac{2}{5}$ in the last formula. A comprehensive exploration of the adaptive dictionary in our pipeline is out of the scope of our analysis, and we leave this task to future exploration.

Heat Gaussians (HEAT). We consider the heat diffusion and, in particular, the heat kernel, which encodes for each pair of points $x, y \in \mathcal{M}$ the amount of heat transferred from x to y in a time t given a unit heat source in x. This quantity is a function of the LB eigenvalues and eigenfunctions, and thus, it is fast to compute as described in [47]:

$$h_t(x, y) = \sum_{\ell=1}^{k} e^{-t\lambda_\ell} \phi_\ell(x) \phi_\ell(y). \qquad (13)$$

With this equation, we can define a dictionary D where the j-th atom, $\forall j \in Q$, is equal to $h_t(j, *)$. $h_t(j, *)$ is a function that $\forall y \in \mathcal{M}$ has value $h_t(j, y)$, i.e. the amount of heat transferred from j to y in time t starting from a unit heat source in j at time 0. We set $t = 1 \times 10^{-2}$ to obtain a heat diffusion with an amplitude similar to the one of the Gaussians in PC-GAU. Being fully determined by the LB, the heat kernel is an isometric-invariant quantity and is strictly related to the geodesic distance [8]. The dictionary we build from the heat kernel, namely **HEAT**, inherits all these properties and the ones given by the uniform sampling of the Q subset of vertices we use to define it.

Spectral Gaussians (SPEC). We take inspiration from [5] and [23] for a different definition of the dictionary of Gaussian functions. We define a Gaussian filter on the frequency domain and interpret them as a set of coefficients to linearly combine the LB eigenfunctions obtaining a Gaussian function in the space

domain (i.e. on the surface). Following [23], we can define these coefficients as:

$$\hat{g}(\lambda_\ell) = e^{-\alpha\lambda_\ell}, \ \forall \ell \in \{1, \dots, k\}, \qquad (14)$$

where $\alpha \in \mathbb{R}$ controls the amplitude of the Gaussians (we fix the parameter $\alpha = 1 \times 10^{-4}$). Then, to reconstruct the value $g_{\alpha,j}(y), \forall y \in \mathcal{M}$ of the Gaussian centered at $j \in Q \subseteq \mathcal{M}$ we compute:

$$g_{\alpha,j}(y) = \sum_{\ell=1}^{k} \hat{g}(\lambda_\ell) \phi_\ell(j) \phi_\ell(y). \qquad (15)$$

We refer to this dictionary in which atoms correspond to $g_{\alpha,j} \forall j \in Q$, as **SPEC**. Being linear combinations of the LB eigenfunctions, the atoms of **SPEC** are invariant to isometries. Furthermore, the generated basis respects the same prior given by the even distribution of the samples in Q.

Wave Kernel Signature (WKS). As described in Section 3, it is possible to optimize a functional map by imposing functional constraints to preserve some point-wise descriptor. In our experiments, we adopt the method NO17 [33], which exploits the *wave kernel signature* [3] (WKS) to produce such constraints. WKS is a multiscale descriptor based on the solution of the wave equation on a non-Euclidean domain. As described in [33], as for the heat diffusion, it is possible to approximate the wave diffusion through the eigenfunctions and the eigenvalues of the LB. This approximation is fast to compute and makes this descriptor invariant to isometries. For these reasons, it is reasonable to consider the different scales of WKS as atoms of a dictionary. We compute WKS with the same parameters from [33], and we refer to [3] for more details. With this choice, we force the orthonormal bases produced by our method to improve the representation of the functions which define the optimization constraints. We refer to the dictionary composed by only the WKS as **WKS**. Additionally, we consider a dictionary, which contains both Gaussian functions, computed as in Section 5, and the wave kernel signature. We refer to this dictionary as **WKS+PC-GAU**. Note that in this case we are considering a dictionary composed by functions from different families. This is perfectly possible, as our framework does not pose any kind of constraint on the composition of the initial dictionary.

	dataset ↓	PC-Gau	ADAPT	HEAT	SPEC	WKS	WKS+PC-Gau	WAVE	LB
GT	FAUST 1:1	**8,4**	**7,8**	15,0	19,6	42,6	**7,2**	27,1	13,6
	FAUST	**15,7**	**15,3**	20,2	23,6	56,2	**14,5**	28,1	19,7
	MWG	**20,8**	**20,1**	24,7	31,7	94,2	**18,4**	30,4	24,7
	MWG iso	**13,5**	**13,1**	17,3	25,8	99,0	**11,7**	24,7	17,3
	TOSCA	**6,3**	**6,3**	9,8	13,0	92,1	**5,8**	16,6	9,9
	TOSCA cat	**12,8**	**12,5**	19,7	22,9	41,0	**12,0**	21,3	19,8
	SHREC19	**24,5**	**24,0**	28,6	31,2	88,6	36,7	34,3	**28,4**
NO17	FAUST 1:1	**16,4**	**15,9**	28,3	313,8	451,6	**14,9**	38,4	29,9
	FAUST	**28,0**	**28,8**	29,8	346,4	387,6	**26,2**	41,3	30,6
	MWG	61,3	61,2	**59,7**	295,6	473,1	**55,2**	87,4	**60,6**
	MWG iso	**26,0**	27,7	27,9	202,1	471,6	**18,8**	44,8	**27,5**
	TOSCA	**12,9**	**13,1**	18,1	409,8	462,4	**13,9**	46,7	19,9
	TOSCA cat	**29,2**	**29,8**	35,4	295,0	374,4	**28,6**	60,6	34,7
	SHREC19	**43,3**	**44,5**	66,6	404,0	486,0	**47,8**	118,7	65,9
ZoomOut	FAUST 1:1	**13,8**	**13,5**	24,0	36,3	235,2	**26,3**	33,7	31,3
	FAUST	**24,6**	**24,4**	26,5	30,5	241,8	**23,1**	33,5	26,0
	MWG	**49,6**	**49,5**	70,0	72,6	311,4	**59,7**	85,2	67,9
	MWG iso	**18,6**	**19,9**	30,7	34,6	281,7	**22,9**	33,2	27,4
	TOSCA	**10,9**	**11,1**	18,8	19,7	215,4	**12,5**	22,3	18,0
	TOSCA cat	**26,1**	**25,5**	35,1	36,3	164,8	**28,8**	39,7	35,7
	SHREC19	**35,7**	**35,1**	38,0	44,9	312,9	39,5	42,8	**39,2**

Table 2: Comparison on the Average Geodesic Error (AGE) of point-wise maps estimated from a functional map and representing the functional spaces with LB or with the proposed approach considering seven different input dictionaries (one for each column). We divide the rows of the table into three parts, which correspond to different computations of the functional maps, from top to bottom: computed from ground-truth correspondence, estimated with [33], and estimated with ZoomOut [26]. In each row, we consider a different test set. For each dataset (row), we depict the AGE value of the best, second best and third best, respectively, in **red**, **green** and **blue** to highlight the distribution of the more accurate solutions. All the results reported in the table must be multiplied by a factor equal to 10^{-3}.

*Wavelets (**WAVE**).* Taking inspiration from the signal processing literature, we replace Gaussians by considering wavelet functions centered in the same set of sampled vertices $Q \subset V_M$ as atoms of our dictionary. The main motivation is that wavelets should provide a more informative representation of the region around each sample. To define the wavelets, we adopt the procedure proposed in [17] for its efficiency and relation with heat diffusion and we exploit the code provided in [17]. We refer to the obtained dictionary as **WAVE**.

7. Experimental Evaluation

The purpose of PCD is to improve the quality of the estimated point-wise map when employed in a functional maps pipeline for shape-matching.

Therefore, the primary evaluation criteria consists in assessing the accuracy of point-wise maps between pairs of meshes. In the following, we compare the results between PCD and LB in different settings and for different choices of the dictionary D. Since PCD and LB can be interchangeably used in the functional maps framework, it is easy to set up the same pipeline and compare the final results obtained in the same conditions.

Metrics. We evaluate the overall accuracy of a point-wise map $\bar{T} : V_N \to V_M$ by the average geodesic error:

$$\text{AGE}(\bar{T}) = \underset{x \in V_N}{\text{Avg}}\{e(x)\}, \quad (16)$$

where $e(x) = \text{GD}_M(\bar{T}(x), T(x))$ is computed with respect to the ground-truth correspondence T provided

by the dataset for evaluation. Since our goal is to improve the performance compared to LB, we can also consider the Relative Error of PCD with respect to LB to make the accuracy gain more explicit:

$$\text{RE}(\bar{T}_{\text{ours}}, \bar{T}_{\text{LB}}) = \frac{\text{AGE}(\bar{T}_{\text{ours}}) - \text{AGE}(\bar{T}_{\text{LB}})}{\text{AGE}(\bar{T}_{\text{LB}})}. \quad (17)$$

To evaluate the overall performance on a dataset, we can compute the mean of AGE and RE over all the pairs considered in the dataset. A negative value of MRE (Mean Relative Error) indicates that, on average, PCD is performing better than LB on the given dataset. Note that MRE does not coincide with the relative difference of the mean AGE, and the assessment provided by RE is less dependent on the absolute complexity of the pairs considered.

Datasets. As test sets, we generate random pairs of meshes taken from standard datasets. We normalize all the meshes to the unitary area, which is a standard unsupervised practice to compare the errors among different datasets.

- **FAUST 1:1** [4] is a dataset composed of 10 human subjects in 10 poses each. All the shapes in this dataset share a fixed template that provides a 1:1 correspondence among their vertices. We additionally consider a version of this dataset, namely **FAUST**, in which all the shapes have been independently remeshed to avoid the implicit use of additional knowledge about the meshes. We used 200 random pairs for both **FAUST 1:1** and FAUST.

- **MWG** contains 25 meshes representing a man, a woman, or a gorilla, in different poses and with different connectivities. We indicate with **MWG iso** the dataset containing only the man and woman meshes. We used 200 random pairs form MWG and 120 from MWG iso.

- **TOSCA** [6] is a benchmark of 3D shapes divided in different classes. Meshes are high resolution (∼50K vertices), isometric, and with vertices in 1:1 correspondence in each class. We limit our quantitative analysis to 20 pairs involving five meshes from the class Michael (one of

the human subjects) in different poses. Moreover, in quantitative and qualitative evaluations, we consider pairs composed only of different poses from the cat class. We refer to this test set as **TOSCA cat**.

- **SHREC19** [25] contains 40 human bodies. These meshes largely differ in the number of vertices, type of tessellation, and model style, making it a challenging dataset. We used 200 random pairs.

In MWG we only have a sparse set of ground truth correspondences (around 1K correspondences). We have a complete ground truth correspondence for all the other datasets. In other words, there is always one point on the target that corresponds to each vertex of the source shape.

Implementation. We implemented the procedure to build PCD and the entire experimental setting in MATLAB. We adopt the code available online for ZoomOut [26] and [33] provided by the authors. In all the following experiments, we fix the number of atoms $k = 60$ and the specific parameters required by each dictionary selected as described in the previous Sections.

7.1. Functional maps computed from ground-truth correspondence

To evaluate our basis independently of the quality of the estimation of the functional map, we start by computing C from the ground truth correspondence provided by each dataset. In this case, C is computed as in Equation 5 and can be considered the best possible functional correspondence in the given bases $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$. For the conversion to pointwise map, we adopt the original algorithm proposed in [34] and presented in Section 3.4. The first seven rows of Table 2 show the results on the different datasets. In this Table, for each dataset (row), we depict the AGE value of the best, second best and third best, respectively, in red, green and blue to highlight the distribution of the more accurate solutions. We observe that PCD provides substantially more accurate conversions. Among the different dictionaries, **WKS+PC-Gau** achieves the best result in six of the seven test sets failing only on the SHREC19 dataset.
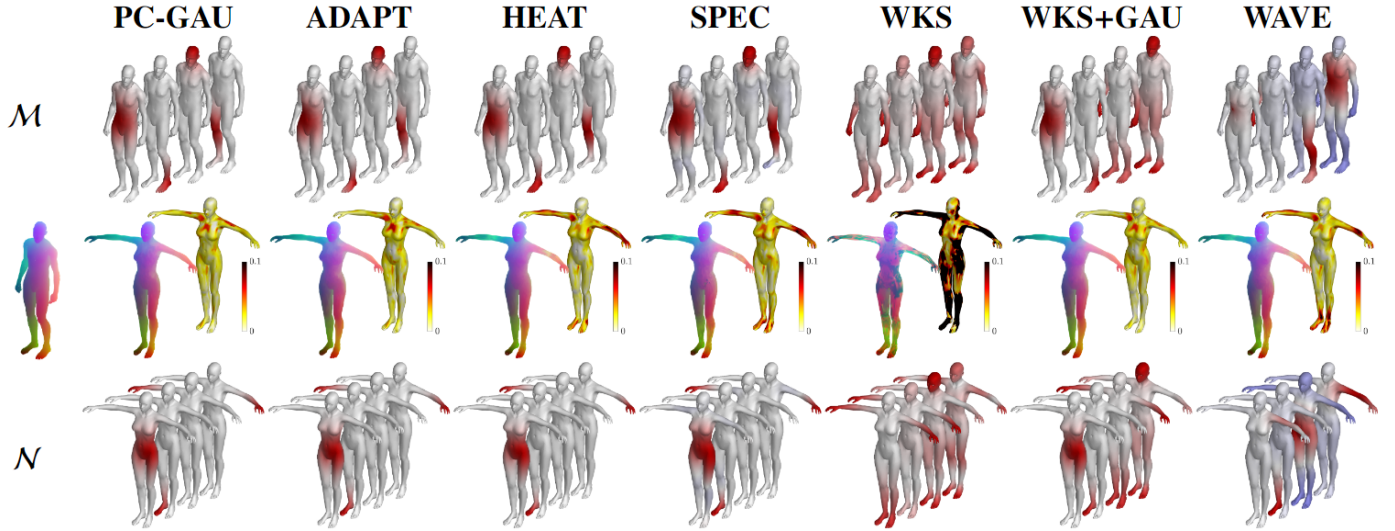
Figure 8: An example of four atoms for each considered dictionary, represented on a random pair $\mathcal{M}$ (top), $\mathcal{N}$ (bottom) taken from MWG. In the middle row, obtained point-wise maps and the corresponding geodesic error are shown. The map is computed from a ground-truth C. Note that, even if atoms with the same indices between $\mathcal{M}$ and $\mathcal{N}$ are visualized, their localization on the mesh is not the same. This is because the points in $Q_\mathcal{M}$ and $Q_\mathcal{N}$ are selected independently.

Even if not so accurate, **PC-Gau** and **ADAPT** are competitive and outperform the LB. The other four solutions are pretty far from the best results. In Figure 8, we show a qualitative comparison for a random pair from the MWG dataset among the different dictionaries in shape matching estimated through the functional framework from the ground truth correspondence. For each dictionary, we show some of the atoms that compose the dictionary on the two shapes (respectively on top and bottom). In the middle, we report the color coding of the matching and the geodesic error represented by the colors. White points have a 0 error, while the darker the color, the larger the error).

7.2. Functional maps estimated with product preservation

In a real-world setting, we cannot compute C from the ground-truth correspondence, but we need to estimate it. The second block of seven rows in Table 2 presents the results when C is estimated using product preservation as functional constraints [33], namely NO17. In these tests, we used functional constraints based on six landmarks (placed on the extremities like feet, hands, head, and chest for human) and the WKS descriptor [3]. PCD produces better results than LB. Only on MWG and MWG iso LB ob-

tain the third best score. The comparison among the different dictionaries is similar to the ground-truth case, with **WKS+PC-Gau** being the best one, but presenting some difficulties in targeting SHREC19. The difference among **WKS+PC-Gau**, **PC-Gau** and **WKS** reveals that having WKS in the dictionary is useful but only if supported by an evenly distributed set of atoms like the Gaussians from **PC-Gau**.

7.3. Functional maps estimated with ZoomOut

Finally, we tested our basis with the iterative refinement introduced in [26]. The last block of rows in Table 2 contains these results. This technique heavily takes advantage of the conversions from functional to point-wise maps and vice-versa. The results show that PCD outperforms LB. In these experiments, we used an initial map of size $k_{\text{ini}} = 16$, estimated with [33] with the same constraints as in Section 7.2. We then adopt the same parameters from [26], increasing the size of the functional map to $k_{\text{final}} = 60$ with step equal to 2. The great performance achieved by our basis, when combined with ZoomOut, confirms that PCD is better suited to represent point-wise correspondences in the functional paradigm. Moreover, with ZoomOut, our method seems more stable also in the presence of non-isometries. The comparison among the differ-

16

ent dictionaries behaves similarly to the other cases even if with ZoomOut **PC-Gau** and **ADAPT** perform better than **WKS+PC-Gau**, with a little gap which favours **ADAPT**.

A comprehensive assessment of the results in Table 2 highlights that **PC-Gau**, **ADAPT** and **WKS+PC-Gau** are not only comparable to LB's performance but even superior. The other dictionaries, which do not involve Gaussians defined through geodesic distances, give rise to less accurate correspondences. **WKS**, the only solution not based on the sampling strategy and the generation of localized functions, is the worst dictionary and confirms the goodness of the sampling strategy adopted in [7]. Moreover, the atoms of **WKS** do not discern the symmetric points because, being entirely intrinsic, they are quasi-symmetric. Due to this property, the representation obtained from **WKS** with our pipeline can give rise to some errors in symmetric shapes. On the contrary, **HEAT** and **SPEC**, which arise from the sampling strategy and the definition of local Gaussians, do not perform well compared to the solutions that exploit the geodesic distances. The spectral Gaussians and the heat kernel are more sensitive to the slight variation in the intrinsic geometry. They do not provide a point representation that permits the correct matching of the corresponding points. The **WAVE** dictionary, which produces less precise maps, does not adopt the geodesic distances to define its atoms. Furthermore, for the **WAVE** solution, we select the efficient and recent solution from [17], maintaining the parameters provided by the authors. A more accurate analysis to select the parameters that define the wave functions is necessary to build a more informative dictionary of waves.

7.4. Direct comparison between PC-Gau and LB

The previous quantitative comparison does not establish a clear champion among the candidate dictionaries, even if it clearly shows that **PC-Gau**, **ADAPT** and **WKS+PC-Gau** well exceed the accuracy of LB. **WKS+PC-Gau** is probably the best candidate with both ground-truth and NO17 estimation, but it does not perform well with the ZoomOut refinement that is currently the most selected method to estimate functional map. **ADAPT** works well with ZoomOut, but its performance is similar to the one of

PC-Gau, which does not require additional computation to produce an adaptive σ. For these reasons, we consider **PC-Gau** as our representative choice and leave further exploration to strengthen PCD by defining more appropriate dictionaries as an intriguing future direction.

Focusing on **PC-Gau**, we perform a direct comparison to LB that we report in Table 3, adding the RE for each test set. We consider again the same three estimations of the functional maps, now organized as three blocks on the columns of Table 3. In the first three columns, we observe that **PC-Gau** provides substantially more accurate conversions with the ground-truth functional maps. Results in terms of MRE are consistent with the average geodesic error. The right part of Figure 6 shows the average spatial distribution of the error on the considered pairs from SHREC19 and cats from TOSCA.

Following the protocol proposed in [16], Figure 9 shows the curves of the cumulative geodesic error for each dataset (more details in the caption). Note that the percentage of vertices with a low geodesic error is similar between **PC-Gau** and LB, but **PC-Gau** has a considerably lower percentage of vertices with a large error. This result is consistent with our analysis: the embedding space of LB is not poor in general but only in specific areas of the mesh. The vertices in the weakly-represented regions have a large geodesic error when using LB, while the quality of point-wise maps is similar between **PC-Gau** and LB in the well-covered areas.

With NO17 [33] (columns 4,5, and 6), **PC-Gau** still gets better results than LB, except in MWG. Removing the gorilla meshes from the dataset brings the advantage back (see row MWG iso), suggesting that **PC-Gau** is more susceptible to the error raised by non-isometries when we compute the C adopting NO17 and WKS as descriptors. In all the cases, even for all the tests with the Zoomout refinement, **MRE**, which is less sensitive to the single pair challenges, is always negative, confirming that our basis outperforms LB.

Random sampling of Q. The quality of our basis does not depend on the exact composition of the subset of points where the Gaussians are initially computed, as far as the scattering of vertices in Q is

dataset	GT			NO17			ZoomOut		
	ours	**LB**	**MRE**	**ours**	**LB**	**MRE**	**ours**	**LB**	**MRE**
	$\times 10^{-3}$	$\times 10^{-3}$	$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-3}$	$\times 10^{-2}$	$\times 10^{-3}$	$\times 10^{-3}$	$\times 10^{-2}$
FAUST 1:1	**8,4**	13,6	**-37,8**	**16,4**	29,9	**-33,0**	**13,8**	31,3	**-43,5**
FAUST	**15,7**	19,7	**-20,3**	**28,0**	30,6	**-4,9**	**24,6**	26,0	**-5,4**
MWG	**20,8**	24,7	**-19,8**	61,3	**60,6**	**-4,9**	**49,6**	67,9	**-27,6**
MWG iso	**13,5**	17,3	**-25,6**	**26,0**	27,5	**-9,4**	**18,6**	27,4	**-29,7**
TOSCA	**6,3**	9,9	**-39,1**	**12,9**	19,9	**-39,0**	**10,9**	18,0	**-43,7**
TOSCA cat	**12,8**	19,8	**-35,9**	**29,2**	34,7	**-17,5**	**26,1**	35,7	**-28,8**
SHREC19	**24,5**	28,4	**-14,0**	**43,3**	65,9	**-15,4**	**35,7**	39,2	**-7,1**

Table 3: Average Geodesic Error (both absolute and relative) of point-wise maps converted from a C (from left to right): computed from ground-truth correspondence, estimated with [33], and estimated with ZoomOut [26].

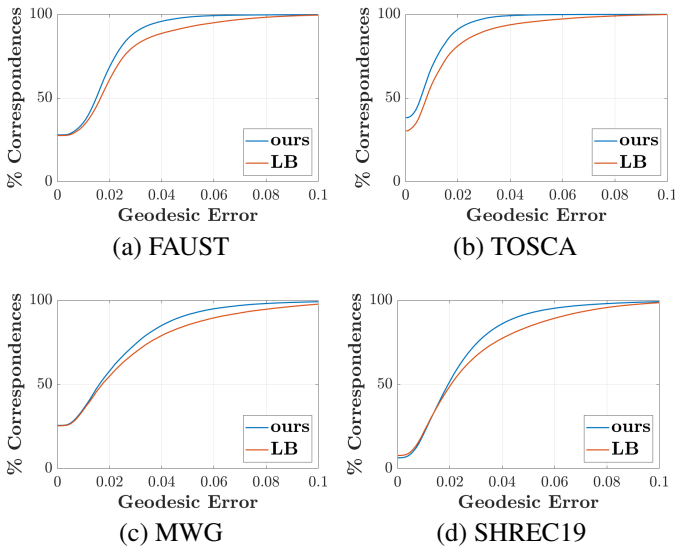

(a) FAUST

(b) TOSCA

(c) MWG

(d) SHREC19

Figure 9: Curves of the cumulative geodesic error of point-wise maps converted from ground-truth C for the four datasets. For each curve point, the y-axis represents the percentage of vertices with a geodesic error lower than the error threshold corresponding to the x-axis: the higher the curve, the more precise the estimated correspondence.

sufficiently uniform on the mesh surface. To verify this claim, we build **PC-GAU** by randomly selecting Q (without replacement) instead of using Farthest Point Sampling [30], as described in 5. Figure 10 compares the bases obtained with the two methods regarding both discrimination power and accuracy of the final point-wise map. Discrimination power shows a similar distribution between the two, with **PC-GAU** computed from FPS being slightly more uniform over the surface (look at the difference in value between chest and feet, for instance). This difference, however, does not impact the point-wise maps obtained, which show almost identical geodesic errors. Therefore, **PC-GAU** is robust to the algorithm for selecting Q. We still prefer using FPS over random sampling, because this makes the basis less dependent on vertex density and ensures uniform scattering also when q is low.

7.5. Parameter selection

As we showed in Section 5, **PC-GAU** computation requires two parameters, namely the number q and the amplitude σ of the Gaussian functions. In this Section, we analyze the impact of such parameters on the accuracy of the final point-wise map, and we show that we can exploit the metrics presented in Section 5.3.1 and 5.3.2 to select appropriate values for σ and q, without any knowledge of the ground-truth correspondence.

7.5.1. Amplitude of Gaussian functions σ

As shown in Section 5, the parameter σ sets the amplitude of the Gaussian functions: the lower
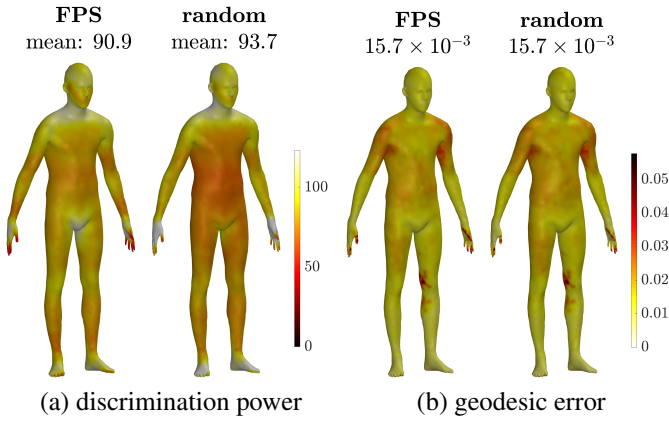
(a) discrimination power   (b) geodesic error

Figure 10: Comparison between random sampling and FPS for the selection of Q on the FAUST dataset. (a) shows the average distribution of the discrimination power of the obtained bases. (b) shows the distribution and the overall value of the geodesic error for point-wise maps obtained from the two bases. We compute C exploiting the provided ground-truth correspondence.
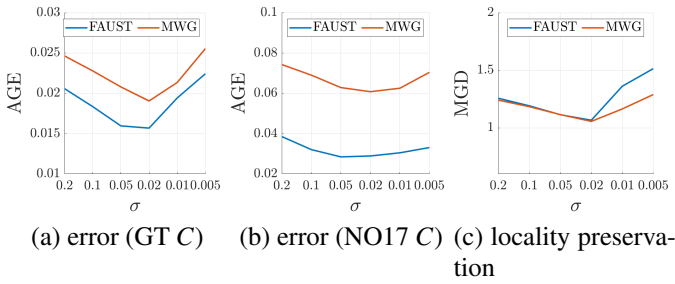

(a) error (GT C)   (b) error (NO17 C)   (c) locality preservation

Figure 11: Comparison of point-wise error (a) (ground-truth), (b) (NO17), and locality preservation (c) for different values of σ. MGD is an indicator of the error at test time, thus, we can exploit it to select the best value of σ.

the value of σ and the more the Gaussian function is localized around its center point.

Figure 11a and 11b present the average geodesic error of point-wise maps (y-axis) obtained using different values of σ (x-axis). For each value of σ **PC-Gᴀᴜ** is computed and then tested both estimating C using the ground-truth correspondence, as done in Section 7.1, and using product preservation method [33], as done in Section 7.2. The plots show that values of σ that are too large and too small result in a sensible decrease in accuracy in the maps obtained. Therefore it is important to choose a value of σ near the mean. Figure 11c shows the value of MGD obtained by our basis computed with different values of σ. We can see that the value of σ that

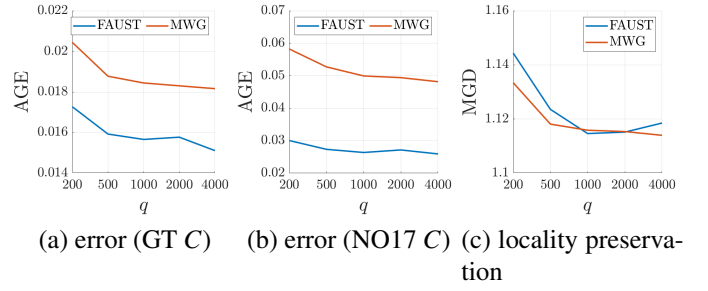(a) error (GT C)   (b) error (NO17 C)   (c) locality preservation

Figure 12: Comparison of point-wise error (a) (ground-truth), (b) (NO17), and locality preservation (c) among different values of q. Average on 30 pairs from FAUST and MWG datasets.

minimizes MGD is also optimal or nearly optimal for the point-wise error. Since we compute MGD for every single mesh and, thus, without any knowledge about the ground truth correspondence, we can adopt this analysis in real scenarios where the correspondence between different meshes is unknown. Here, we only consider MGD, but we obtain similar results with EGDC and Dis, which behave similarly to MGD.

7.5.2. Number of Gaussian functions q

The parameter q determines the number of points selected on the mesh surface and, thus, the number of Gaussian functions. Similarly to the previous paragraph, Figure 12 shows both the point-wise error and the value of MGD for **PC-Gᴀᴜ** (y-axis) computed using different values of q (x-axis). We can still observe a strong relationship between the metric and the error obtained, thus allowing us to use MGD also for this parameter. We remark that the error decreases as q increases, with less noticeable effects after $q = 1000$. However, the time needed to compute our basis increases more than linearly with q. Therefore, $q = 1000$, i.e. the value used in our experiments, represents a good trade-off between accuracy and efficiency.

Note, at last, that the predictive power of the metrics defined in Section 5.3.1 and 5.3.2 on the point-wise error is an additional confirmation of their relevance. This relationship means that they assess the properties of the embedding space relevant to the conversion to point-wise maps.

8. Conclusions

We presented the procedure to build a new basis for the space of real-valued functions defined on a mesh. Our framework, namely PCD, starts by selecting a dictionary of informative functions for the mesh and then extracts an orthonormal basis by applying the principal component analysis. The output of PCD can replace LB at no cost in virtually any functional map pipeline to target the shape matching task. In particular, we focused on PC-Gau, the basis obtained by selecting a dictionary of Gaussians, which are easy to compute and give rise to desirable properties. Compared to the eigenfunctions of the Laplace-Beltrami operator, which are the standard basis for functional maps, the energy of PC-Gau is distributed more uniformly on the mesh. The resulting embedding space for the vertices is overall more amenable to point-wise conversion. We extended our analyses to different compositions of the initial dictionary, evaluating their performances on several established datasets and using different methods for estimating the functional map. We showed that substituting LB with PCD, for some specific choices of the initial dictionary, leads to far superior results in the accuracy of the obtained point-wise map. We also analyzed the dependency of the matching accuracy on the value of the parameters required by PC-Gau, showing a simple method to choose them in real-case scenarios.

Although the results are pretty impressive, we experienced that PCD has some limitations: *i*) it suffers when there are significant errors in the estimation of the functional map; *ii*) the entire framework has been defined for meshes, while we did not explore its applicability to point clouds, neither considering meshes extracted from dense point clouds nor directly working on 3D points without connectivity; *iii*) we believe that even if the embedding from PC-Gau is uniform, this is not optimal and could be improved by explicitly enforcing this property.

A first interesting direction to explore in the future is to test the ability of our basis to represent different signals defined on the surfaces as done in [26], to evaluate if the functional representation can also benefit from the more uniform distribution of the energy of our basis. A second direction we

only partially addressed is to evaluate other possible dictionary compositions as inputs of our procedure. We saw that some alternatives could provide better results compared to PC-Gau, even if only on some test sets. Finally, we would investigate how it is possible to integrate PCD in data-driven procedures inspired by the functional maps framework, such as [20, 12, 22].

Acknowledgements

References

[1] Y. Aflalo, H. Brezis, and R. Kimmel. On the optimality of shape and data representation in the spectral domain. *SIAM J. Imaging Sci.*, 8(2):1141–1160, 2015.

[2] Y. Aflalo and R. Kimmel. Regularized principal component analysis. *Chinese Annals of Mathematics, Series B*, 38:1–12, 2017.

[3] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[4] F. Bogo, J. Romero, M. Loper, and M. J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR*, pages 3794–3801, Columbus, Ohio, 2014. IEEE.

[5] D. Boscaini, J. Masci, S. Melzi, M. M. Bronstein, U. Castellani, and P. Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Computer Graphics Forum*, 34(5):13–23, 2015.

[6] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical Geometry of Non-Rigid Shapes*. Springer, New York, NY, 2008.

[7] M. Colombo, G. Boracchi, and S. Melzi. PC-GAU: PCA Basis of Scattered Gaussians for Shape Matching

via Functional Maps. In D. Cabiddu, T. Schneider, D. Allegra, C. E. Catalano, G. Cherchi, and R. Scateni, editors, *Smart Tools and Applications in Graphics - Eurographics Italian Chapter Conference*. The Eurographics Association, 2022.

[8] K. Crane, C. Weischedel, and M. Wardetzky. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph.*, 32(5), oct 2013.

[9] G. Da Poian, R. Bernardini, and R. Rinaldo. Gaussian dictionary for compressive sensing of the ECG signal. In *2014 IEEE Workshop on Biometric Measurements and Systems for Security and Medical Applications (BIOMS) Proceedings*, pages 80–85, 2014.

[10] M. P. do Carmo. Riemannian geometry. *Bikhausen, Boston*, 1992.

[11] N. Donati, E. Corman, S. Melzi, and M. Ovsjanikov. Complex functional maps: A conformal link between tangent bundles. *Computer Graphics Forum*, 41(1):317–334, 2022.

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

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

[14] D. Giorgi, S. Biasotti, and L. Paraboschi. Shape retrieval contest 2007: Watertight models track. *SHREC competition*, 8, 07 2008.

[15] R. Huang, J. Ren, P. Wonka, and M. Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum*, 39(5):265–278, 2020.

[16] V. G. Kim, Y. Lipman, and T. Funkhouser. Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)*, volume 30, page 79. ACM, 2011.

[17] M. Kirgo, S. Melzi, G. Patanè, E. Rodolà, and M. Ovsjanikov. Wavelet-based heat kernel derivatives: Towards informative localized shape analysis. *Computer Graphics Forum*, 40(1):165–179, 2021.

[18] A. Kovnatsky, M. Bronstein, A. Bronstein, K. Glashoff, and R. Kimmel. Coupled quasi-harmonic bases. *Computer Graphics Forum*, 32(2pt4):439–448, 2013.

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

[20] O. Litany, T. Remez, E. Rodolà, A. Bronstein, and M. Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5659–5667. IEEE, 2017.

[21] F. Maggioli, S. Melzi, M. Ovsjanikov, M. Bronstein, and E. Rodolà. Orthogonalized fourier polynomials for signal approximation and transfer. In *Proceedings of Eurographics 2021*, 2021.

[22] R. Marin, M.-J. Rakotosaona, S. Melzi, and M. Ovsjanikov. Correspondence learning via linearly-invariant embedding. In *Advances in Neural Information Processing Systems*, volume 33, pages 1608–1620. Curran Associates, Inc., 2020.

[23] S. Melzi. Sparse representation of step functions on manifolds. *Computers & Graphics*, 82:117–128, 2019.

[24] S. Melzi, R. Marin, P. Musoni, F. Bardon, M. Tarini, and U. Castellani. Intrinsic/extrinsic embedding for functional remeshing of 3d shapes. *Computers & Graphics*, 88:1–12, 2020.

[25] S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. SHREC 2019: Matching Humans with Different Connectivity. In *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019.

[26] S. Melzi, J. Ren, E. Rodolà, A. Sharma, P. Wonka, and M. Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155:1–155:14, Nov. 2019.

[27] S. Melzi, E. Rodolà, U. Castellani, and M. Bronstein. Localized manifold harmonics for spectral shape analysis. *Computer Graphics Forum*, 37(6):20–34, 2018.

[28] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.

[29] J. S. B. Mitchell, D. M. Mount, and C. H. Papadimitriou. The discrete geodesic problem. *SIAM Journal on Computing*, 16(4):647–668, 1987.

[30] C. Moenning and N. A. Dodgson. Fast Marching farthest point sampling. Technical Report UCAM-CL-TR-562, University of Cambridge, Computer Laboratory, Apr. 2003.

[31] T. Neumann, K. Varanasi, C. Theobalt, M. Magnor, and M. Wacker. Compressed manifold modes for mesh processing. *Computer Graphics Forum*, 33(5):35–44, 2014.

[32] D. Nogneng, S. Melzi, E. Rodolà, U. Castellani, M. Bronstein, and M. Ovsjanikov. Improved functional mappings via product preservation. *Computer Graphics Forum*, 37(2):179–190, 2018.

[33] D. Nogneng and M. Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36(2):259–267, 2017.

[34] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012.

[35] G. Pai, J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov. Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 384–393, Los Alamitos, CA, USA, jun 2021. IEEE Computer Society.

[36] M. Panine, M. Kirgo, and M. Ovsjanikov. Non-isometric shape matching via functional maps on landmark-adapted

bases. *Computer Graphics Forum*, 41(6):394–417, 2022.

[37] D. Panozzo, Y. Lipman, E. Puppo, and D. Zorin. Fields on symmetric surfaces. *ACM Trans. Graph.*, 31(4), jul 2012.

[38] U. Pinkall and K. Polthier. Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics*, 2(1):15–36, 1993.

[39] J. Ren, S. Melzi, M. Ovsjanikov, and P. Wonka. Maptree: Recovering multiple solutions in the space of maps. *ACM Trans. Graph.*, 39(6), nov 2020.

[40] J. Ren, S. Melzi, P. Wonka, and M. Ovsjanikov. Discrete optimization for shape matching. *Computer Graphics Forum*, 40(5):81–96, 2021.

[41] J. Ren, M. Panine, P. Wonka, and M. Ovsjanikov. Structured regularization of functional map computations. In *Computer Graphics Forum*, volume 38, pages 39–53. Wiley Online Library, 2019.

[42] J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6), 2018.

[43] E. Rodolà, M. Moeller, and D. Cremers. Point-wise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)*, 2015.

[44] E. Rodolà, M. Möller, and D. Cremers. Regularized point-wise map recovery from functional correspondence. In *Computer Graphics Forum*, volume 36, pages 700–711. Wiley Online Library, 2017.

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

[46] R. W. Sumner and J. Popović. Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 399–405. ACM, 2004.

[47] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer graphics forum*, 28(5):1383–1392, 2009.

[48] B. Vallet and B. Lévy. Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum*, 27(2):251–260, 2008.

[49] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.