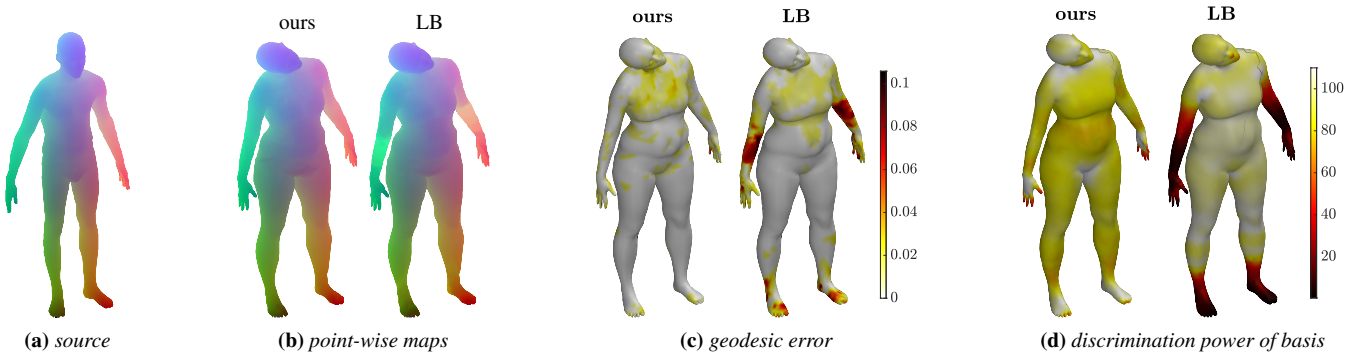
STAG: Smart Tools and Applications in Graphics (2022)
D. Cabiddu, T. Schneider, G. Cherchi, and R. Scateni (Editors)

PC-GAU: PCA Basis of Scattered Gaussians for Shape Matching via Functional Maps

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

[1]DEIB, Politecnico di Milano [2] University of Milano-Bicocca



(a) *source* **(b)** *point-wise maps* **(c)** *geodesic error* **(d)** *discrimination power of basis*

Figure 1: *Example of shape matching between the source man (a) and the target women (b) comparing* PC-GAU *(ours) 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 share the same color). (c) the respective geodesic errors encoded by the colormap (white means 0 error while dark colors are larger ones). (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.*

Abstract

Shape matching is a central problem in geometry processing applications, ranging from texture transfer to statistical shape analysis. The functional maps framework provides a compact representation of correspondences between discrete surfaces, which is then converted into point-wise maps required by real-world applications. The vast majority of methods based on functional maps involve the eigenfunctions of the Laplace-Beltrami Operator (LB) as the functional basis. A primary drawback of the LB basis is that its energy does not uniformly cover the surface. This fact gives rise to regions where the estimated correspondences are inaccurate, typically at tiny parts and protrusions. For this reason, state-of-the-art procedures to convert the functional maps (represented in the LB basis) into point-wise correspondences are often error-prone. We propose PC-GAU, *a new functional basis whose energy spreads on the whole shape more evenly than LB. As such,* PC-GAU *can replace the LB basis in existing shape matching pipelines.* PC-GAU *consists of the principal vectors obtained by applying Principal Component Analysis (PCA) to a dictionary of sparse Gaussian functions scattered on the surfaces. Through experimental evaluation of established benchmarks, we show that our basis produces more accurate point-wise maps — compared to LB — when employed in the same shape-matching pipeline.*

CCS Concepts

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

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 [PLPZ12, SP04], object retrieval [GBP08], and statistical shape analysis [BRLB14]. The problem is particularly challenging when a non-rigid deformation occurs between the shapes in the pair, such as a pose variation

DOI: 10.2312/stag.20221253

in human bodies. In this case, the space of possible point-wise correspondences is exponential in size.

Functional maps, the seminal work of Ovsjanikov *et al.* [OBCS*12], 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, such that a small matrix C can compactly encode the functional correspondence. The matrix C, which describes a linear mapping between the functional subspaces, has dimensions equal to the number of basis functions in each shape, and it is estimated by solving an optimization problem exploiting linear constraints. Therefore, functional maps provide 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 dense point-wise map [EBC17], for instance, when we want to collect data variations in statistical shape analysis.

Recovering a point-wise map from a functional map is not straightforward, and many works, starting from the seminal paper [OBCS*12] itself, 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 [NO17, RPWO18, MRR*19], or the algorithm applied to convert functional maps into point-wise maps [RMC15, EBC17, RPWO18].

As mentioned before, any method involving functional maps requires a basis for a functional space defined on a surface. The choice of the basis is a critical aspect, which heavily affects the final result. The vast majority of methods, starting from [OBCS*12] itself, 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 [VL08] for non-Euclidean surfaces, and their subset associated with the eigenvalues having the smallest absolute values are optimal for approximating smooth functions with limited variations [ABK15]. Nonetheless, LB presents a significant limitation for the point-wise conversion: these functions do not uniformly cover the mesh. In particular, the energy concentrates on the flat regions at the expense of narrower extremities. 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 as previous works [KBB*13, NVT*14, MRCB18], we propose a new basis for the functional space defined on surfaces designed for functional map pipelines. A primary characteristic of our basis is that it is evenly distributed on the surface, which we recognize as a limitation of LB. We construct our basis by generating a collection of Gaussian functions scattered on the shape, and then applying Principal Component Analysis (PCA) to obtain a compact set of orthogonal generators. The atoms of our basis are the Principal Components of a dictionary made of GAUssians, thus the name PC-GAU. The core idea of PC-GAU is that by minimizing the reconstruction error on the samples, PCA produces a basis of a subspace that reflects the distribution of the Gaussians in the initial dictionary and, as we will show, it is easy to enforce the uniform distribution of such Gaussians.

We quantitatively show that the embedding space generated by our basis provides a good representation for all the points of the surface. More specifically we assess the quality of the representation in the embedding space as the following properties: (1) the capability of discriminating between different vertices and (2) the preservation of vertices locality. At the same time, our basis inherits good properties from LB: it is orthonormal and invariant to isometries by construction, and empirically exhibits a weak frequency order, 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 different aspects of the functional map framework and achieving significant benefits.

Through numerous experiments on established datasets, we show that the properties of our basis actually improve the quality of the final point-wise map. The similarity between our basis and LB enables us to compare them in the same setting directly. Thus, we test PC-GAU and LB when injected in different shape-matching pipelines [NO17, MRR*19] 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`.

2. Related work

In this Section, we briefly overview the existing shape matching methods that fall, similarly to ours, into the functional map framework [OBCS*12]. For other approaches to shape matching, refer to the survey [VKZHCO11].

The functional map framework was proposed by Ovsjanikov *et al.* in [OBCS*12]. 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 represented compactly as a matrix C, which is found by imposing the preservation of linear functional constraints. These constraints express the correspondence between the two shapes of landmarks, segments, and descriptors. The most used descriptors in this context are invariant to near-isometric deformations, like heat diffusion [SOG09] or quantum mechanical properties [ASC11].

Additional constraints can improve the estimation of the matrix C. The commutativity with the LB operator, which forces the functional map to represent isometries, was firstly proposed in [OBCS*12] and then refined in [RPWO19]. In [NO17], the authors show that imposing the preservation of point-wise products is an additional, beneficial, constraint. In shape with symmetries, the preferred maps are those that do not mix up symmetries. This property has been promoted by introducing a new term in the optimization [RPWO18], or considering the complex counterpart of the functional map [DCMO22]. ZoomOut [MRR*19] is an iterative method for extending the size of an initial functional map. It works by alternating conversions to point-wise maps and back to functional maps of increased size. Many different alternatives build upon the ZoomOut procedure, such as [HRWO20, RMOW20,

PRM*21, RMWO21, PKO]. An efficient method for converting a functional map into a point-wise map coupled with the ICP-like refinement were proposed in the original framework [OBCS*12]. Subsequent works have proposed alternative methods for extracting better point-wise maps from functional maps [RMC15], by considering a probabilistic model [RMC17], by introducing a smoothness prior on the point-wise map [EBC17], or by devising a complex refinement scheme to promote continuity, coverage and bijectivity of the obtained map [RPWO18]. 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 method to extract the point-wise map from C. They make the implicit choice, which is ubiquitous since [OBCS*12], of using eigenfunctions of the Laplace-Beltrami operator [Lev06, VL08] as the functional basis. However, they are agnostic to the basis adopted. This fact makes these methods *complementary* to our proposal, as they can be used in combination with our basis.

Other works followed an approach more similar to ours and proposed new bases for the functional space of a mesh. Some of them target specific tasks, like the transfer of tessellation structure [MMM*20] or step functions [Mel19] from one mesh to another. The method in [MMM*20] 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 [NVT*14], which proposes a basis whose atoms have local support, thus promoting the sparsity of C, but without improving the point-wise accuracy with respect to the standard LB. Kovnatsky *et al.* [KBB*13] 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 [KBB*13] also depends on the other mesh in the pair and the landmarks used. With a purpose more similar to ours, Melzi *et al.* [MRCB18] proposed a basis that can extend LB and improve its expressive powers in specific mesh regions. Their approach is, however, different from ours: the atoms of their bases localize in predefined regions, which must be provided as input. This is not a trivial information, as it requires the knowledge of at least coarse correspondences between the two shapes. Our basis, on the contrary, does not require any input other than the mesh itself.

Finally, [NMR*18] and [MMO*21] showed that it is possible to extend a functional basis by adding point-wise products of atoms to the original basis. This method provides great benefits to the obtained point-wise maps and can be applied, in principle, to any basis, including ours.

3. Background

In this section, we introduce the notation and some background notions used in the paper.

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 [dC92] for the notions 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 $j, \forall x, j \in V_\mathcal{M}$.

3.2. Shape matching

The input of shape matching are usually two discretized meshes $\mathcal{M}$ and $\mathcal{N}$, having sets of vertices $V_\mathcal{M}$ and $V_\mathcal{N}$. 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 $y \in V_\mathcal{N}$ 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 $y \in V_\mathcal{N}$ as:

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

where $\text{GeoDist}_\mathcal{M}$ 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 $(y \in V_\mathcal{N}, x \in V_\mathcal{M})$ in known correspondence, namely $T(y) = x$. In some of the experiments in Section 5, we will refer to landmarks, which are not required to build the bases but only for the specific matching pipelines.

3.3. Functional maps

When we discretize a bi-dimensional surface $\mathcal{M}$, a real-valued function f on $\mathcal{M}$ is given by a vector that associates to each vertex $x \in V_\mathcal{M}$ 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 $T : V_\mathcal{N} \to V_\mathcal{M}$, the functional map framework [OBCS*12], instead of estimating T directly, searches for a functional correspondence between $\mathcal{M}$ and $\mathcal{N}$. Then, once the functional correspondence has been estimated, it extracts the corresponding point-wise map.

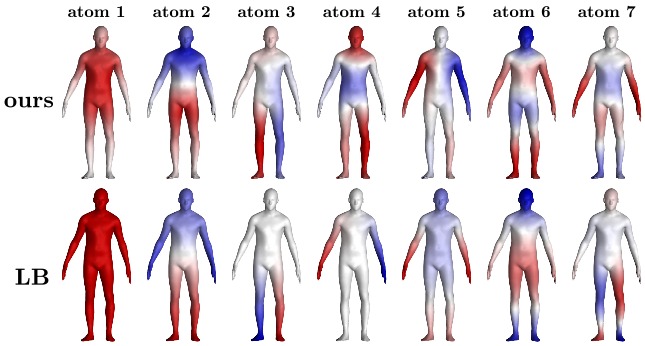


Figure 2: *Atoms of our basis (top) compared to LB basis (bottom) atoms. Each element of the bases is a function defined on the mesh, represented here through color. Positive values are red, negative values are blue, and white corresponds to zero.*

Functional maps builds upon the observation that a point-wise map T induces a *linear* operator $T_F : \mathcal{F}(\mathcal{M},\mathbb{R}) \to \mathcal{F}(\mathcal{N},\mathbb{R})$ 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 $\mathcal{F}(\mathcal{M},\mathbb{R})$ 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 consists in estimating 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 these functional constraints (in the least square sense) introduced by these matches. Note that, in principle, we can truncate the basis of $\mathcal{M}$ and $\mathcal{N}$ at a different number of atoms $k_\mathcal{M}$ and $k_\mathcal{N}$, thus producing a rectangular $C \in \mathbb{R}^{k_\mathcal{N} \times k_\mathcal{M}}$. However, for the sake of simplicity, we consider $k_\mathcal{M} = k_\mathcal{N} = k$ 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 the vector of values assumed by all the basis functions in x: $\mathrm{Emb}(x) = [\phi_i(x)] \in \mathbb{R}^k$. $\mathrm{Emb}(x)$ 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

We now consider the problem of converting a functional map C from $\mathcal{M}$ to $\mathcal{N}$ into a point-wise map $\bar{T} : V_\mathcal{N} \to V_\mathcal{M}$. A simple and efficient method, proposed in [OBCS*12], consists in finding, for each column of $\Phi_\mathcal{N}^T$, the nearest neighbor in the columns of $C\Phi_\mathcal{M}^T$. This procedure corresponds to transferring 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 [OBCS*12], 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, you can find a visual representation of the complete procedure, focusing on step (1) since it is the one that our method addresses.

3.5. Standard basis

The Laplace-Beltrami operator $\Delta_\mathcal{M} : \mathcal{F}(\mathcal{M},\mathbb{R}) \to \mathcal{F}(\mathcal{M},\mathbb{R})$ 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 [MDSB03, PP93]. $\Delta_\mathcal{M}$ admits an eigendecomposition: $\Delta_\mathcal{M}\phi_i = \lambda_i\phi_i$, where $\Phi = \{\phi_i\}$ are its eigenfunctions with corresponding real eigenvalues $\Lambda = \{\lambda_1 \le \lambda_2 \le \dots\}$. 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 [Lev06, VL08]. By selecting only the first k atoms we obtain an optimal [ABK15] low-pass filter approximation of functions in $\mathcal{F}(\mathcal{M},\mathbb{R})$. From [OBCS*12] 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 4.3, 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 areas of the mesh and better overall.

Algorithm 1 Construction of PC-GAU

1: **input:** $\mathcal{M}, V_{\mathcal{M}}, A_{\mathcal{M}}, \sigma, q$, normalize, k
2: $Q = \text{FPS}(\mathcal{M}, q)$
3: **for** $i \in V_{\mathcal{M}}$, $j \in Q$ **do**
4: $\quad D_{ij} = \text{GeoDist}_{\mathcal{M}}(i, j)$
5: $\quad G_{ij} = \exp\left(-D_{ij}^2/\sigma\right)$
6: **end for**
7: **if** normalize **then**
8: $\quad$ **for** $i \in V_{\mathcal{M}}$, $j \in Q$ **do**
9: $\quad\quad G_{ij} = G_{ij}/\sqrt{(G_{*j})^T A_{\mathcal{M}} G_{*j}}$
10: $\quad$ **end for**
11: **end if**
12: $P = \text{PCA}\left(G^T, \text{variableWeights} = A_{\mathcal{M}}\right)$
13: $P_k = P(:, 1:k)$
14: **output:** P_k

4. Proposed Solution: PC-GAU

In this section, we present the core contribution of this paper: a new basis for the space of functions defined on a mesh. We design such a basis to fit the role of truncated basis in functional maps pipelines for shape matching. PC-GAU works as a replacement of LB to obtain more accurate point-wise maps.

We first illustrate the procedure to build the basis. Then, we analyze the properties that directly arise from its construction. In Section 4.3, we analyze the main properties of PC-GAU, with reference to the *desiderata* expressed in Section 3.6.

4.1. Building procedure

The proposed method to construct the basis derives from signal processing [DPBR14]. In particular, it consists in building a dictionary of Gaussian functions scattered on the mesh and then perform dimensionality reduction through Principal Component Analysis (PCA). PCA produces an orthogonal set of generators for the functional space spanned by the Gaussians. Generators are ordered according to their capability to approximate the initial dictionary of Gaussians. 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. In the following, we present the procedure to build our basis on a mesh $\mathcal{M}$ with n vertices. We provide in Figure 3 and in Algorithm 1 a visual and a compact description of the proposed method, respectively, and we refer each step to the corresponding line in Algorithm 1.

Subset of vertices We start by selecting a subset Q of q vertices on the mesh with Farthest Point Sampling [MD03] (line 2), using Euclidean distance for efficiency reasons. This selection is not the only possible, as discussed in Section 5.1. The orange box in Figure 3 shows that the sampled points evenly cover the mesh surface.

Dictionary of Gaussians 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 $D_{ij} = \text{GeoDist}_{\mathcal{M}}(i, j)$ to any other vertex $i \in V_{\mathcal{M}}$ (line 4) and then define the value of the Gaussian G_{ij} as $G_{ij} = \exp\left(-(D_{ij}^2/\sigma)\right)$ (line 5). We approximate the geodesic distance as the length of the shortest path on the edges of the mesh [MMP87]. The parameter σ is arbitrarily chosen and sets the amplitude of the Gaussians. We store the Gaussian functions as columns of the matrix $G = [G_{ij}]$, of size $n \times q$. In lines 7-11, we can also normalize each column G_{*j} of G by division with its norm, computed as $\|G_{*j}\|_{\mathcal{M}} = \sqrt{G_{*j}^T A_{\mathcal{M}} G_{*j}}$, where $A_{\mathcal{M}}$ is the mass matrix of $\mathcal{M}$. In all the experiments we use $q = 1000$, $\sigma = 0.05$ and we apply normalization. In the green box in Figure 3, we report some examples of the resulting Gaussian functions.

Dimensionality reduction We then compute the PCA of G^T (line 12), meaning that each Gaussian function is considered a sample and each vertex of the mesh a variable. We do not center the variables, but we weigh them for the element of the area associated with each vertex. The result of PCA is a set of q vectors of size n, called Principal Components (PCs). We can interpret these vectors as the q generators of the functional space S spanned by the Gaussians. Since $q < n$, S is a proper subspace of $\mathcal{F}(\mathcal{M}, \mathbb{R})$ and, assuming the Gaussians 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 13). 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 [AK17]:

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

where G_{*i} is the i-th column of G, namely the i-th Gaussian function. Note that orthonormality and projection in (3) are expressed with respect to the inner product defined on the mesh. Providing a good approximation of the Gaussian functions, and thus of S, and satisfying (3), the truncated basis P_k inherits implicitly the uniform distribution owned by the Gaussian functions.

4.2. 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.

4.2.1. 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

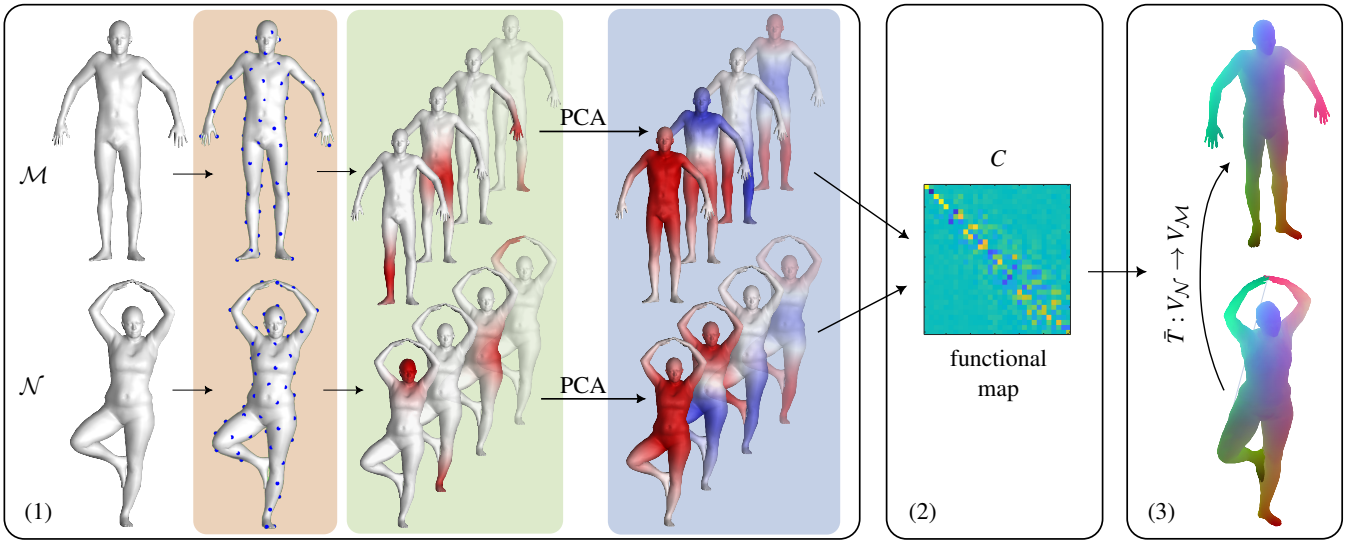


Figure 3: *Complete shape matching pipeline with functional maps: (1) definition of a functional basis, (2) estimation of C and (3) conversion to a point-wise map. Here, step (1) shows the building procedure of* PC-GAU*: selection of a subset of vertices (orange box), construction of the dictionary of Gaussian functions (green box) and dimensionality reduction through PCA (blue box).*

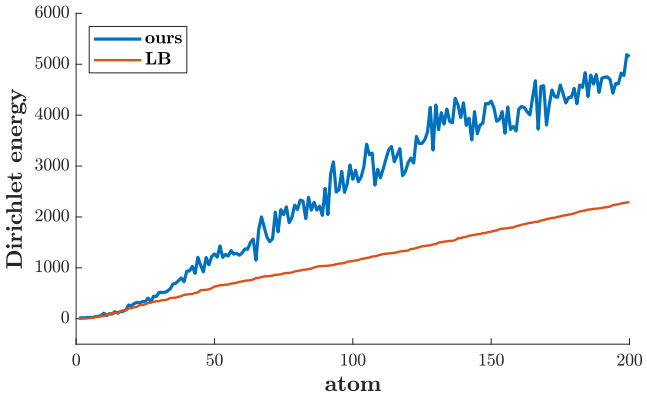


Figure 4: *Comparison between the Dirichlet energy (frequency) of atoms of our basis and LB, computed for an example mesh from FAUST. Atoms of* PC-GAU *are approximately ordered by increasing frequency.*

approximation when we project a function f on our truncated basis. We consider the possibility of sorting our atoms with respect to their frequency as an interesting direction. Still, we believe that their natural order is sufficient for our goal, and we leave its explicit constraint as future work.

4.2.2. Orthonormality

PC-GAU, like LB, is an orthonormal basis. We say that a basis Φ is orthonormal according to the inner product of a mesh $\mathcal{M}$ if $\Phi^T A_{\mathcal{M}} \Phi = I_k$, where $A_{\mathcal{M}}$, the mass matrix, is a diagonal matrix having as entries the area elements associated to each vertex, 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:

$$C = \Phi_{\mathcal{N}}^T A_{\mathcal{M}} \Pi \Phi_{\mathcal{M}}. \tag{4}$$

4.2.3. 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 $\mathcal{N}$:

$$\text{GeoDist}_{\mathcal{M}}(T(x), T(y)) = \text{GeoDist}_{\mathcal{N}}(x, y) \quad \forall x, y \in V_{\mathcal{N}}. \tag{5}$$

Since we construct our basis purely on geodesic distances, if an isometry occurs between $\mathcal{N}$ and $\mathcal{M}$, then our basis, under the assumption of equal distribution of the sample points, will be isometry-invariant. As for the LB, the functional map is a diagonal matrix with $+1$ or -1 on the diagonal when shapes are isometric. In more general cases, the energy of C is funnel-shaped, approaching diagonal as the relation between $\mathcal{M}$ and $\mathcal{N}$ gets closer to isometry. See Figure 3 (2) for an example.

4.3. 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.

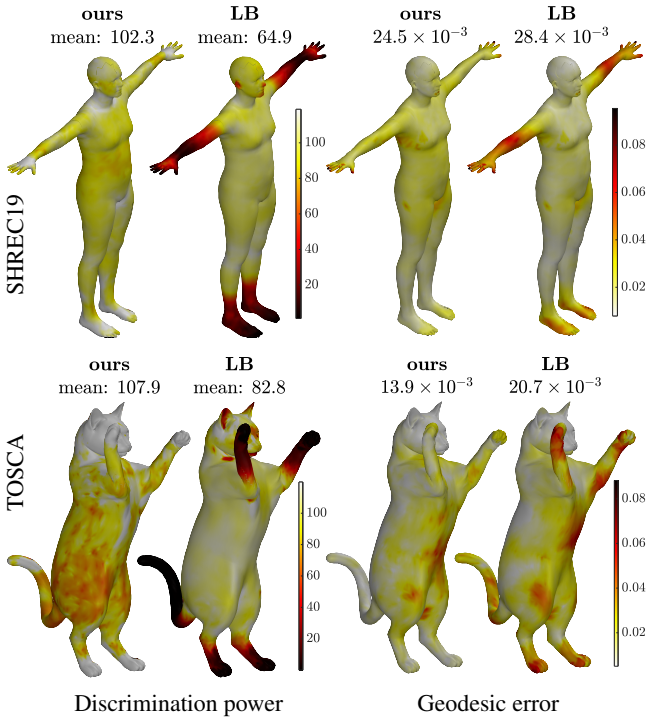


Figure 5: *Spatial distribution of discrimination power of the basis and geodesic error. Comparison between* ours *and LB. We visualize the average results across 40 human meshes from SHREC19 (top) and three cat meshes from TOSCA (bottom). Darker is worse in all cases. The similarity between distributions suggests a correlation between the error and the local quality of the embedding space.*

4.3.1. 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:

$$\mathrm{Dis}(x) = \frac{\|\mathrm{Emb}(x) - \mathrm{Emb}(y_x)\|_2}{\mathrm{GeoDist}_{\mathcal{M}}(x, y_x)} \tag{6}$$

where $y_x = \mathrm{argmin}_{z \in V_{\mathcal{M}} \setminus \{x\}} \left\{ \|emb(z) - emb(x)\|_2 \right\}$ is the vertex having the closest embedding to $\mathrm{Emb}(x)$ 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-GAU embedding with respect to LB, we compare their distribution of 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 5, we report the average distribution of $\mathrm{Dis}(x)$ on meshes obtained across 40 human meshes from SHREC19 (top row) and three cat meshes from TOSCA (bottom row). On the right of Figure 5, we visualize the average distribution of the geodesic error on the corresponding datasets matched using LB and PC-GAU as described in Section 5.1. 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.

4.3.2. 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 $\mathrm{Emb}(x)$, and estimate the correlation between the Euclidean distances of the embeddings and their geodesic distances from x:

$$\mathrm{EGDC}(x) = \mathrm{corr}\left(\mathrm{GeoDist}_{\mathcal{M}}(x, y), \|\mathrm{Emb}(x) - \mathrm{Emb}(y)\|_2 \right)_{y \in S}$$

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 $\mathrm{EGDC}(x)$ 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 $\mathrm{Emb}(x)$, 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 $\bar{R}$:

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

The more the embedding preserves the distance of the geodesic neighbor, the lower the value of $\mathrm{MGD}(x)$, which approaches 1. Therefore, the lower $\mathrm{MGD}(x)$ 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 6 shows the value of these metrics averaged on the same shapes from SHREC19 (top) and TOSCA (bottom) involved in Figure 5. The spatial distribution measured with EGDC and MGD for LB closely mimics the distribution of discrimination power (see Figure 5, first column). These three similar results support the choice of these measures that accurately encode the embedding space quality across different mesh areas. In Table 1, we present 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

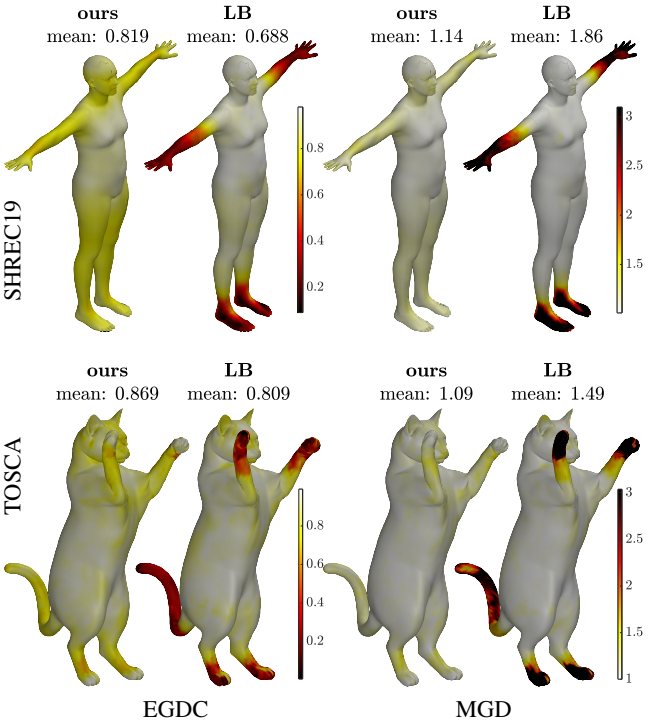


Figure 6: *Spatial distribution of EGDC and MGD. Comparison between* ours *and LB on average on human shapes from SHREC19 (top) and cats from TOSCA (bottom). Darker is worse in all cases. LB presents low locality preservation on the extremities (arms, feet, paws, and tail).*

the one we propose, increases the overall quality of the embedding space.

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.*

5. Experimental Evaluation

The purpose of PC-GAU 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 PC-GAU and LB in different settings. Since PC-GAU 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 $\bar{T} : V_{\mathcal{N}} \to V_{\mathcal{M}}$ by the average geodesic error:

$$\mathrm{AGE}(\bar{T}) = \operatorname*{Avg}_{x \in V_{\mathcal{N}}} \{e(x)\}, \tag{7}$$

where $e(x) = \mathrm{GeoDist}_{\mathcal{M}}(\bar{T}(x), T(x))$ 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 also consider the Relative Error of PC-GAU with respect to LB to make the accuracy gain more explicit:

$$\mathrm{RE}(\bar{T}_{\text{ours}}, \bar{T}_{\text{LB}}) = \frac{\mathrm{AGE}(\bar{T}_{\text{ours}}) - \mathrm{AGE}(\bar{T}_{\text{LB}})}{\mathrm{AGE}(\bar{T}_{\text{LB}})}. \tag{8}$$

To evaluate the overall performance on a dataset, we 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, PC-GAU 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 We used random pairs of meshes taken from standard datasets for our tests. We normalize all the meshes to the unitary area, which is a standard practice to compare the errors among different datasets.

- **FAUST** [BRLB14] is a recent dataset composed of 10 human subjects in 10 poses each. The dataset has been remeshed to ensure that no additional knowledge about the meshes is implicitly used. We used 200 random pairs.
- **MWG** contains 25 meshes representing a man, a woman, and 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** [BBK08] 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, we perform the qualitative evaluation on three different poses from the cat class.
- **SHREC19** [MMR*19] 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). For all the other datasets, we instead have a complete ground truth correspondence, namely there is always one point on the target that is corresponding to each vertex of the source shape.

Implementation We implemented the procedure to build PC-GAU and the entire experimental setting in MATLAB. We adopt the code available online for ZoomOut [MRR*19] and [NO17] provided by the authors. In all the following experiments, we fix the number of atoms $k = 60$, and we generated our basis with $q = 1000$, $\sigma = 0.05$, and using normalization. See Section 4.1 for the meaning of these parameters.

dataset	GT			NO17			ZoomOut		
	ours $\times 10^{-3}$	**LB** $\times 10^{-3}$	**MRE** $\times 10^{-2}$	**ours** $\times 10^{-3}$	**LB** $\times 10^{-3}$	**MRE** $\times 10^{-2}$	**ours** $\times 10^{-3}$	**LB** $\times 10^{-3}$	**MRE** $\times 10^{-2}$
FAUST	**15,7**	19,7	**-20,3**	**28,0**	30,6	**-4,9**	**24,6**	26,2	**-5,7**
MWG	**20,8**	24,9	**-20,2**	60,8	**60,4**	**-8,4**	**49,6**	69,6	**-28,7**
MWG iso	**13,6**	17,3	**-25,5**	**25,9**	27,6	**-10,1**	**18,6**	27,2	**-28,3**
TOSCA	**7,7**	12,3	**-39,6**	**12,7**	19,8	**-39,3**	**10,7**	17,8	**-44,2**
SHREC19	**24,5**	28,4	**-14,0**	**43,4**	65,9	**-15,3**	**35,7**	39,1	**-6,9**

Table 2: *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 [NO17], and estimated with ZoomOut [MRR*19].*

5.1. *C* computed from ground-truth correspondence

To evaluate our basis independently of the quality of functional mapping, we start by estimating C from the ground truth correspondence provided by the dataset. In this case, C is computed as in Equation 4 and can be considered the best possible functional correspondence between $\Phi_{\mathcal{M}}$ and $\Phi_{\mathcal{N}}$. For the conversion to point-wise map, we adopt the original algorithm proposed in [OBCS*12] and presented in Section 3.4. The first three columns of Table 2 show the results on the different datasets: we observe that PC-GAU provides substantially more accurate conversions. Results in terms of MRE are consistent with the average geodesic error. The right part of Figure 5 shows the average spatial distribution of the error on the considered pairs from SHREC19 and cats from TOSCA.

Following the protocol proposed in [KLF11], Figure 7 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.

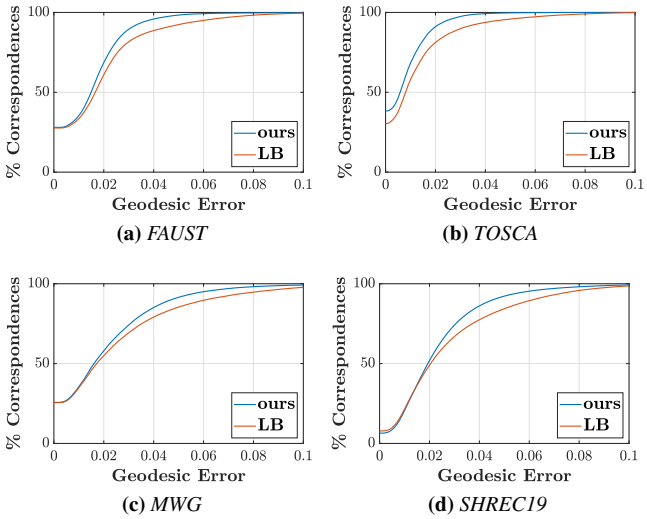
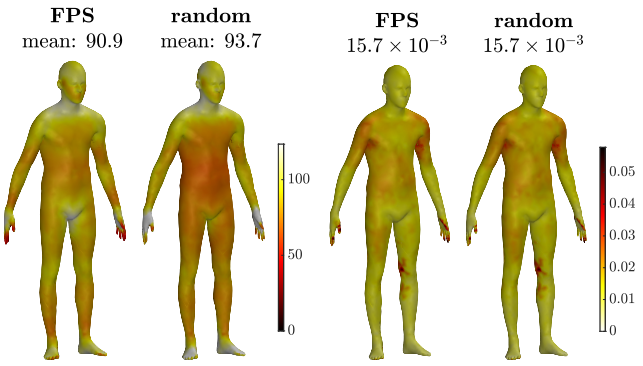


Figure 7: *Curves of the cumulative geodesic error of point-wise maps converted from ground-truth C for the four datasets. For each curve point, the y-axis represents the percentage of vertices with a geodesic error lower than the error threshold corresponding to the x-axis: the higher the curve, the more precise the estimated correspondence.*

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 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 [MD03], as described in 4.1. Figure 8 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 used for selecting Q. We still prefer using FPS over random sampling, because this makes the basis less dependent of vertex density and ensures uniform scattering also when q is low.

5.2. *C* estimated with product preservation

In a real world setting, we cannot compute C from the ground-truth correspondence, but we need to estimate it. Columns 4,5,6 of Table 2 present the results when C is estimated using product preservation as functional constraints [NO17]. In these tests, we used functional constraints based on six landmarks (feet, hands, head, and chest) and the WKS descriptor [ASC11]. 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, **MRE**, which is less sensitive to the single pair challenges, is always negative, confirming that our basis outperforms LB.



(a) *discrimination power* **(b)** *geodesic error*

Figure 8: *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.*

5.3. *C* estimated with ZoomOut

Finally, we tested our basis with the iterative refinement introduced in [MRR*19]. The last three columns of Table 2 contain these results. This technique heavily takes advantage of the conversions from functional to point-wise maps and vice-versa. The results show that PC-GAU outperforms LB. In these experiments we used an initial map of size $k_{\text{ini}} = 16$, estimated with [NO17] with the same constraints as in Section 5.2. We then adopt the same parameters from [MRR*19] 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 PC-GAU 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.

6. Conclusions

We presented the procedure to build PC-GAU, a new basis for the space of real-valued functions defined on a mesh. Compared to the eigenfunctions of the Laplace-Beltrami operator, which are the standard basis for functional maps, the energy of our basis is distributed more uniformly on the mesh. The resulting embedding space for the vertices is overall more amenable to point-wise conversion. PC-GAU can replace LB at no cost in virtually any functional map pipeline. We showed experimentally that this replacement leads to superior results in the accuracy of the obtained point-wise maps on different datasets and with different methods to estimate the functional maps.

Although the results are pretty impressive, we experienced that PC-GAU 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 consider a different composition of the initial dictionary of functions other than Gaussians. In the same spirit, we believe that an adaptive sampling of the Gaussians, whose parameters might be based on the geometric features of the shape, could provide a refined representation and improve mapping accuracy. PC-GAU depends on some parameters, and in the future, we aim to evaluate its dependencies better and improve its performance by fine-tuning. Moreover, we will test the ability of our basis to represent different signals defined on the surfaces as done in [MRR*19] to evaluate if the functional representation can also benefit from the more uniform distribution of the energy of our basis. Finally, we would investigate how it is possible to integrate PC-GAU in data-driven procedures inspired by the functional maps framework, such as [LRR*17, DSO20, MRMO20].

References

[ABK15] AFLALO Y., BREZIS H., KIMMEL R.: On the optimality of shape and data representation in the spectral domain. *SIAM J. Imaging Sci. 8*, 2 (2015), 1141–1160.

[AK17] AFLALO Y., KIMMEL R.: Regularized principal component analysis. *Chinese Annals of Mathematics, Series B 38* (2017), 1–12.

[ASC11] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on* (2011), IEEE, pp. 1626–1633.

[BBK08] BRONSTEIN A., BRONSTEIN M., KIMMEL R.: *Numerical Geometry of Non-Rigid Shapes*. Springer, New York, NY, 2008.

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and evaluation for 3D mesh registration. In *Proc. CVPR* (Columbus, Ohio, 2014), IEEE, pp. 3794–3801.

[dC92] DO CARMO M. P.: Riemannian geometry. *Bikhausen, Boston* (1992).

[DCMO22] DONATI N., CORMAN E., MELZI S., OVSJANIKOV M.: Complex functional maps: A conformal link between tangent bundles. *Computer Graphics Forum 41*, 1 (2022), 317–334.

[DPBR14] DA POIAN G., BERNARDINI R., RINALDO R.: 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* (2014), pp. 80–85.

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

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

[GBP08] GIORGI D., BIASOTTI S., PARABOSCHI L.: Shape retrieval contest 2007: Watertight models track. *SHREC competition 8* (07 2008).

[HRWO20] HUANG R., REN J., WONKA P., OVSJANIKOV M.: Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum 39*, 5 (2020), 265–278.

[KBB*13] KOVNATSKY A., BRONSTEIN M., BRONSTEIN A., GLASHOFF K., KIMMEL R.: Coupled quasi-harmonic bases. *Computer Graphics Forum 32*, 2pt4 (2013), 439–448.

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 79.

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

[LRR*17] LITANY O., REMEZ T., RODOLÀ E., BRONSTEIN A., BRONSTEIN M.: Deep functional maps: Structured prediction for dense shape correspondence. pp. 5660–5668. doi:10.1109/ICCV.2017.603.

[MD03] MOENNING C., DODGSON N. A.: *Fast Marching farthest point sampling*. Tech. Rep. UCAM-CL-TR-562, University of Cambridge, Computer Laboratory, Apr. 2003.

[MDSB03] MEYER M., DESBRUN M., SCHRÖDER P., BARR A. H.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*. Springer, 2003, pp. 35–57.

[Mel19] MELZI S.: Sparse representation of step functions on manifolds. *Computers & Graphics 82* (2019), 117–128.

[MMM*20] MELZI S., MARIN R., MUSONI P., BARDON F., TARINI M., CASTELLANI U.: Intrinsic/extrinsic embedding for functional remeshing of 3d shapes. *Computers & Graphics 88* (2020), 1–12.

[MMO*21] MAGGIOLI F., MELZI S., OVSJANIKOV M., BRONSTEIN M., RODOLÀ E.: Orthogonalized fourier polynomials for signal approximation and transfer. In *Proceedings of Eurographics 2021* (2021).

[MMP87] MITCHELL J. S. B., MOUNT D. M., PAPADIMITRIOU C. H.: The discrete geodesic problem. *SIAM Journal on Computing 16*, 4 (1987), 647–668.

[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), The Eurographics Association.

[MRCB18] MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M.: Localized manifold harmonics for spectral shape analysis. *Computer Graphics Forum 37*, 6 (2018), 20–34.

[MRMO20] MARIN R., RAKOTOSAONA M.-J., MELZI S., OVSJANIKOV M.: Correspondence learning via linearly-invariant embedding, 2020. arXiv:2010.13136.

[MRR*19] MELZI S., REN J., RODOLÀ E., SHARMA A., WONKA P., OVSJANIKOV M.: Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG) 38*, 6 (Nov. 2019), 155:1–155:14.

[NMR*18] NOGNENG D., MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M., OVSJANIKOV M.: Improved functional mappings via product preservation. *Computer Graphics Forum 37*, 2 (2018), 179–190.

[NO17] NOGNENG D., OVSJANIKOV M.: Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum 36*, 2 (2017), 259–267.

[NVT*14] NEUMANN T., VARANASI K., THEOBALT C., MAGNOR M., WACKER M.: Compressed manifold modes for mesh processing. *Computer Graphics Forum 33*, 5 (2014), 35–44.

[OBCS*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG) 31*, 4 (2012), 30:1–30:11.

[PKO] PANINE M., KIRGO M., OVSJANIKOV M.: Non-isometric shape matching via functional maps on landmark-adapted bases. *Computer Graphics Forum n/a*, n/a.

[PLPZ12] PANOZZO D., LIPMAN Y., PUPPO E., ZORIN D.: Fields on symmetric surfaces. *ACM Trans. Graph. 31*, 4 (jul 2012).

[PP93] PINKALL U., POLTHIER K.: Computing Discrete Minimal Surfaces and their Conjugates. *Experimental mathematics 2*, 1 (1993), 15–36.

[PRM*21] PAI G., REN J., MELZI S., WONKA P., OVSJANIKOV M.: 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)* (Los Alamitos, CA, USA, jun 2021), IEEE Computer Society, pp. 384–393.

[RMC15] RODOLÀ E., MOELLER M., CREMERS D.: Point-wise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)* (2015).

[RMC17] RODOLÀ E., MÖLLER M., CREMERS D.: Regularized pointwise map recovery from functional correspondence. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 700–711.

[RMOW20] REN J., MELZI S., OVSJANIKOV M., WONKA P.: Maptree: Recovering multiple solutions in the space of maps. *ACM Trans. Graph. 39*, 6 (nov 2020).

[RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete optimization for shape matching. *Computer Graphics Forum 40*, 5 (2021), 81–96.

[RPWO18] REN J., POULENARD A., WONKA P., OVSJANIKOV M.: Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG) 37*, 6 (2018).

[RPWO19] REN J., PANINE M., WONKA P., OVSJANIKOV M.: Structured regularization of functional map computations. In *Computer Graphics Forum* (2019), vol. 38, Wiley Online Library, pp. 39–53.

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. *Computer graphics forum 28*, 5 (2009), 1383–1392.

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation transfer for triangle meshes. In *ACM Transactions on Graphics (TOG)* (2004), vol. 23, ACM, pp. 399–405.

[VKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum 30*, 6 (2011), 1681–1707.

[VL08] VALLET B., LÉVY B.: Spectral Geometry Processing with Manifold Harmonics. *Computer Graphics Forum 27*, 2 (2008), 251–260.