Improved Functional Mappings via Product Preservation

D Nogneng, S Melzi, E Rodolà, U Castellani, M Bronstein, Maks Ovsjanikov

EUROGRAPHICS 2018 / D. Gutierrez and A. Sheffer
(Guest Editors)

Volume 37 (2018), Number 2

Improved Functional Mappings via Product Preservation

D. Nogneng[1†], S. Melzi[2†], E. Rodolà[3], U. Castellani[2], M. Bronstein[4,5,6] and M. Ovsjanikov[1]

[1]LIX, École Polytechnique, UMR CNRS [2]University of Verona [3]Sapienza University of Rome
[4]USI Lugano [5]Tel Aviv University [6]Intel Perceptual Computing

Abstract

In this paper, we consider the problem of information transfer across shapes and propose an extension to the widely used functional map representation. Our main observation is that in addition to the vector space *structure of the functional spaces, which has been heavily exploited in the functional map framework, the functional* algebra *(i.e., the ability to take pointwise products of functions) can significantly extend the power of this framework. Equipped with this observation, we show how to improve one of the key applications of functional maps, namely transferring real-valued functions without conversion to point-to-point correspondences. We demonstrate through extensive experiments that by decomposing a given function into a linear combination consisting not only of basis functions but also of their pointwise products, both the representation power and the quality of the function transfer can be improved significantly. Our modification, while computationally simple, allows us to achieve higher transfer accuracy while keeping the size of the basis and the functional map fixed. We also analyze the computational complexity of* optimally *representing functions through linear combinations of products in a given basis and prove NP-completeness in some general cases. Finally, we argue that the use of function products can have a wide-reaching effect in extending the power of functional maps in a variety of applications, in particular by enabling the transfer of high-frequency functions without changing the representation size or complexity.*

CCS Concepts

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

1. Introduction

Shape correspondence is one of the key problems in digital geometry processing with applications in fields ranging from manufacturing to shape morphing [KMP07], statistical shape analysis [HSS*09, BRLB14], texture mapping, animation and deformation transfer [SP04] among a wide variety of others. In all of these applications, the key requirement of shape matching algorithms is to enable *information transfer* across two or more shapes.

Over the past several decades, a large number of computational techniques has been developed for addressing the shape matching problem. While most early methods have concentrated on rigid shape matching, more recently numerous approaches have also been proposed in the more general context of finding correspondences between non-rigid shapes, such as humans in arbitrary poses [VKZHCO11]. One of the major challenges arising in this setting is that the space of possible correspondences between points on a pair of shapes is exponential, which gives rise to difficult optimization problems when establishing reliable point-to-point maps.

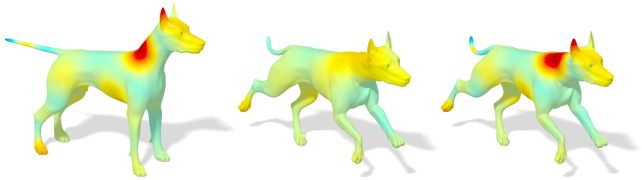

Figure 1: *Transferring a real-valued function from a source shape (left) onto the target using a fixed functional map with the standard approach (center) vs. our extended method (right).*

To address this challenge, recent works have focused either on defining a consistent parameterization for a pair of shapes, often with the use of landmarks (e.g., [LF09, APL15, AL16]), or by choosing a different representation for shape correspondences, which is more amenable to direct optimization and manipulation. These include either soft maps or measure couplings, which can benefit from computational techniques in optimal transport [SNB*12, SPKS16], and a class of techniques based on the recently introduced functional map representation [OBCS*12, OCB*17]. This latter is based on the idea of using correspondences between real-valued function rather than points on the shapes. Since certain (e.g., square integrable) function spaces admit a vector space

† Equal Contribution

structure and therefore can be endowed with a multi-scale functional basis (e.g., the Laplacian eigenbasis), a common approach is to restrict the search for an optimal correspondence to a subspace spanned by a small number of basis functions. This implies that the optimal functional map can be represented using a moderate-sized matrix, independent of the number of points on a pair of shapes.

This restriction to functional maps between small subspaces significantly reduces the computational complexity of the shape correspondence problem as it leads to simpler optimization problems with fewer unknowns (in the simplest setting, the shape correspondednce boils down to a least-squares problem [OCB*17]). At the same time, it also somewhat reduces the utility of the computed functional map especially since converting a functional to a point-to-point map can be a challenging problem in itself [RMC15, VLB*17]. Motivated by this fact, several applications have argued for using functional maps directly, by exploiting their ability to transfer real-valued functions across shapes without recovering the point-to-point map. Examples of applications include segmentation transfer, demonstrated in the original article [OBCS*12], tangent-vector field design [ABCCO13], image and shape co-segmentation [WHOG14, HWG14] and even more recently, consistent mesh quadrangulation [ACBCO17]. In all of these works, a given function on the source shape is represented as a linear combination of the basis functions which are then transferred using the functional map onto the target. Since in most cases, the basis consists of the first few eigenfunctions of the Laplace-Beltrami operator, in practice functional maps can only transfer sufficiently smooth functions, leaving out high-frequency details, which severely limits the applicability of the entire functional maps framework. As a possible remedy, several recent works considered alternatives to the Laplace-Beltrami basis [KBB*13, NVT*14, MRCB17].

In this work, we argue that more accurate function transfer can be achieved within the functional maps framework without changing the basis or increasing its size. In particular, we propose to use the algebraic structure of function spaces [HGK04], which means that in addition to defining a vector space through scalar multiplication and addition, functions can also naturally be multiplied pointwise. Moreover, most natural functional maps (i.e., those arising from point-to-point ones) must preserve this algebraic structure. Putting these two properties together, our key observation is that *the ability to transfer basis functions also gives us the ability to transfer their pointwise products*. This means that if a given function on a source shape can be represented via not just the elements of the basis, but also their pointwise products, a given functional map can be used to transfer it accurately onto the target, as shown in Figure 1.

We demonstrate through extensive experiments that this observation can lead to a significant improvement in the function transfer quality without changing the functional map representation. Remarkably, it allows us to map even high frequency information by using only a low-frequency basis without converting a functional map to a point-to-point one. As a result, our approach can have a direct impact on all applications of the functional map framework.

2. Related Work

As mentioned above, relating information across different shapes is among the most basic problems in shape analysis. Since our approach is designed to better exploit a *given functional map* and is such more closely related to *representing* correspondences, rather than *computing* them, in the following we primarily give an overview of the most common representations for maps between shape, and refer an interested reader to recent surveys on shape matching [VKZHCO11, TCL*13, BCBB16] for a more in-depth discussion of correspondence problems.

Most early models of correspondences between non-rigid shapes are based on the standard notion of pointwise mappings, e.g., [BBK06, LF09, OMMG10, KLF11]. A particular instance of this class of methods is based on spectral embeddings defined by the Laplace-Beltrami eigenfunctions, followed by a correspondence procedure in the embedding space, e.g. [MHK*08, DK10]. Shtern and Kimmel proposed constructing spectral embedding using pointwise products of Laplace-Beltrami eigenfunctions [SK14] and the triple products of their gradients with the surface normal [SK15] as a means of capturing additional information. The main drawback of pointwise correspondence models is that such frameworks can often be unstable and lead to difficult non-linear non-convex optimization problems. As a result, several authors have proposed to consider more general notions of mappings, which are more amenable to direct optimization. In the most basic setting, such generalized mappings arise as relaxations of pointwise correspondence problems, as e.g., in [LH05, BBM05]. In these scenarios, however, such representations are typically used as intermediate steps, aimed at facilitating pointwise map recovery.

A more principled approach to soft mappings is provided by the notion of measure couplings, that have been used for both representing and finding correspondences [Mém07, Mém11, SNB*12] and that are intimately related to the formalism of optimal transport. These representations have a probabilistic interpretation and benefit from computational advances in efficiently solving certain optimal transport problems [Cut13, SDGP*15]. Consequently, they have recently gained prominence in addressing shape matching and alignment tasks [SPKS16, MCSK*17]. Nevertheless, the complexity of these representations is directly related to the sampling density of the shapes, and can quickly become prohibitive, often requiring heuristics and multi-resolution schemes.

A different formalism that our work directly builds on is provided by the functional maps framework [OBCS*12]. This representation is based on representing correspondences through their action (via pull-back) of real-valued functions. Since the pull-back of functions is linear and since functional spaces can be endowed with a multi-scale basis, this leads to a representation of correspondences as moderately-sized matrices, which can be directly manipulated and optimized for. Although initially introduced as a tool for shape matching, functional maps have been used for relating tangent vector fields [ABCCO13], extending the Generalized Multi-Dimensional Scaling to the spectral domain [AK13], computing maps between symmetric [OMPG13] and partial [RCB*17, LRB*16, LRBB17] shapes, coupled bases [KBB*13] and even consistent quadrangulation [ACBCO17] among others. Most recently, functional maps were integrated as differentiable layers into intrinsic deep learning architectures [LRR*17].

All of these applications benefit from the properties of the functional representation, including its compactness, which often trans-

lates into relatively simple optimization problems, and its capacity for information transfer. Indeed, although the original article [OBCS*12] proposed an approach for recovering a point-to-point map from a functional one, follow-up works, have observed that, in many scenarios functional maps can be used directly for transferring information such as tangent vector fields [ABCCO13] or segmentations in shape collections [HWG14] among others. In these works, the information being transferred is represented by using the *vector space* structure of functional spaces. At the same time, as we argue in this paper, real-valued functions also have a natural *algebra* defined through pointwise products, which can be used directly to improve the quality of function transfer without relying on point-to-point maps. Remarkably, we show how this property can be exploited to transfer high-frequency functions even in the presence of a functional map relating only low-frequency bases.

Perhaps most closely related to ours is a recent work by Nogneng and Ovsjanikov [NO17], in which the authors showed that by representing descriptors as linear operators acting on functions through *pointwise multiplication*, it is possible to obtain a significant improvement in the quality of the recovered functional map. Similarly to theirs, our work is also motivated by the classical result that a non-trivial functional map acts as a point-to-point map if and only if it preserves pointwise function products [SM93]. However, rather than trying to improve the quality of an optimized functional map through better use of descriptors, our emphasis is on showing how *a given* functional map can be used more effectively by allowing the transfer of not just basis functions but also their (possibly high-order) point-wise products. More concretely, unlike previous works, including [NO17], which have always restricted the functional subspaces to linear combinations of basis functions, we show that a much richer space can be constructed and used without sacrificing the computational and storage efficiency of functional maps, by exploiting pointwise products of basis functions.

3. Motivation and Overview

Suppose we are given a pair of shapes $\mathcal{M}$ and $\mathcal{N}$ consisting of $n_{\mathcal{M}}$ and $n_{\mathcal{N}}$ points.

A method for constructing the so-called "functional maps" between them was proposed in [OBCS*12] and then significantly extended in the follow-up works (see [OCB*17] for an overview) and consists of the following general steps:

1. Construct a set of basis functions on both $\mathcal{M}$ and $\mathcal{N}$. Store them as columns of matrices $\mathbf{\Phi}^{\mathcal{M}} = (\phi_1^{\mathcal{M}}, \dots, \phi_{k_{\mathcal{M}}}^{\mathcal{M}})$ and $\mathbf{\Phi}^{\mathcal{N}} = (\phi_1^{\mathcal{N}}, \dots, \phi_{k_{\mathcal{N}}}^{\mathcal{N}})$ for some small $k_{\mathcal{M}}, k_{\mathcal{N}}$. The most commonly used basis consists in taking the eigenfunctions corresponding to the smallest eigenvalues of the Laplace-Beltrami operator, although the ideas presented below are not tied to this choice.
2. Compute q descriptor functions (e.g., Wave Kernel Signature [ASC11]) on the source and target shapes. Express them in the corresponding bases and store the expansion coefficients as columns of matrices $\mathbf{A}$ and $\mathbf{B}$, of size $k_{\mathcal{M}} \times q$ and $k_{\mathcal{N}} \times q$ respectively.
3. Compute the optimal *functional map* matrix $\mathbf{C}$ of size $k_{\mathcal{N}} \times k_{\mathcal{M}}$. The most basic approach, introduced in [OBCS*12] consists in finding a matrix that as best as possible aligns the descriptor functions and also commutes with the Laplace-Beltrami operators on the two shapes. Formally this can be written as:
$$\mathbf{C} = \arg\min_{\mathbf{X}} \|\mathbf{X}\mathbf{A} - \mathbf{B}\| + \varepsilon \|\mathbf{\Lambda}^{\mathcal{N}}\mathbf{X} - \mathbf{X}\mathbf{\Lambda}^{\mathcal{M}}\|,$$
where $\mathbf{\Lambda}^{\mathcal{N}}, \mathbf{\Lambda}^{\mathcal{M}}$ are diagonal matrices of Laplace-Beltrami eigenvalues on the two shapes, and ε is a small regularizer.
4. Convert the functional map $\mathbf{C}$ to a point-to-point map between the shapes.

One of the main advantages of the functional map representation is that the key optimization step 3. above can be solved efficiently using standard numerical linear algebra tools, and in the most basic case reduces to solving a simple linear system of equations. This step has been extended considerably, by both employing manifold constraints, robust optimization methods [KGB16], and by formulating better descriptor preservation, commutativity, and regularization terms [LRB*16, NO17]. On the other hand, as has been observed in several follow-up works (see, e.g. [RMC15]), converting a functional map to a point-to-point one in step 4. of the pipeline can be a challenging and error-prone step in itself. At the same time, as argued in the original article [OBCS*12] (Section 8.3), the knowledge of a point-to-point map might not be required in certain applications. Indeed, the functional map matrix $\mathbf{C}$ can be used, e.g. to transfer real-valued functions across shapes, which can be directly used, e.g. to transport segmentations across shapes [OBCS*12] or images [WHOG14] as well as other information such as tangent vector fields [ABCCO13] or cross-fields [ACBCO17]. Thus, given a real-valued function $\mathbf{f}$, its image under the functional map C can be computed as:
$$\mathbf{g} = \mathbf{T}\mathbf{f} = \mathbf{\Phi}^{\mathcal{N}}\mathbf{C}(\mathbf{\Phi}^{\mathcal{M}})^{\dagger}\mathbf{f},$$
where † denotes the Moore-Penrose pseudo-inverse. Note that the large $n_{\mathcal{N}} \times n_{\mathcal{M}}$ matrix $\mathbf{T}$ representing the functional map in the 'spatial' domain is never actually constructed explicitly.

While very simple and intuitive, this procedure also has a severe limitation: it only allows to transfer functions (or their projections) that lie within the vector space spanned by $\mathbf{\Phi}^{\mathcal{M}}$. When the basis is given by the first $k_{\mathcal{M}}$ eigenfunctions of the Laplace-Beltrami operator, which is the most choice in practice, this implies that only *sufficiently smooth* or low-frequency functions can be transferred using the map $\mathbf{C}$.

Functional Algebra In this paper, we argue that this is an unnecessary restriction, which can be, at least partially lifted in many settings. For this we propose using the algebraic structure of the functional spaces on the shapes, which has so far not been exploited fully. The key property we consider is that in addition to defining a *vector space* via inner products, real-valued (square integrable) functions also have a well-defined point-wise product operation: $f_1 \odot f_2 \to f_3$, where $f_3(x) = f_1(x) f_2(x)$ at every point $x \in \mathcal{M}$.

This point-wise product operation is compatible with functional maps: it is well-known [SM93] that a non-trivial linear functional map corresponds to a point-to-point one if and only if it preserves the functional algebra. I.e. the distributivity of the functional map over point-wise product:
$$\mathbf{T}(f_1 \odot f_2) = \mathbf{T}(f_1) \odot \mathbf{T}(f_2) \quad \forall\, f_1, f_2. \tag{1}$$

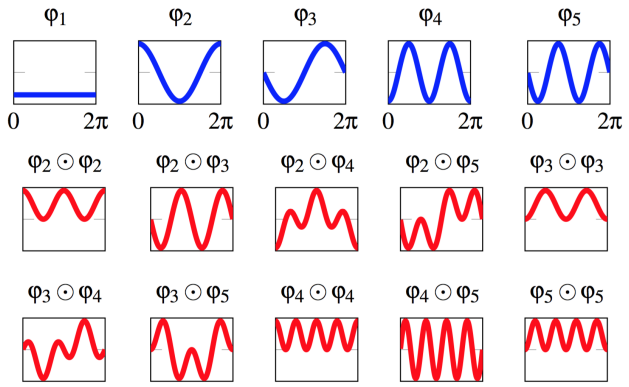

Figure 2: *The first 5 standard 1D Fourier basis functions (in blue) and their pairwise products (in red) on a periodic domain. Above each function we report the product from which it is generated. Note how the products exhibit higher frequencies compared to the original functions. Note also some linear dependencies among some of these functions (e.g.:* $\varphi_2 \odot \varphi_2 + \varphi_3 \odot \varphi_3 + \lambda\varphi_1 = 0$ *for a scalar* λ*).*

Intuitively, if a functional map corresponds to a point-to-point one, then product preservation follows directly from the definition of composition. Conversely, if a functional map preserves pointwise products, then $\mathbf{T}(f^2) = \mathbf{T}(f) \odot \mathbf{T}(f)$, which implies that indicator functions should be mapped to indicator functions.

Remark: It is interesting to measure how Eq. (1) changes when the functional map deviates from a pointwise map. A simple computation shows that if $\tilde{\mathbf{T}} = \mathbf{T} + \delta$, where $\mathbf{T}$ satisfies Eq. (1), then:

$$\begin{aligned}\tilde{\mathbf{T}}(f_1 \odot f_2) =& \tilde{\mathbf{T}}(f_1) \odot \tilde{\mathbf{T}}(f_2) + \delta(f_1 \odot f_2) - \delta(f_1) \odot \delta(f_2) \\ & - \delta(f_1) \odot \tilde{\mathbf{T}}(f_2) - \tilde{\mathbf{T}}(f_1) \odot \delta(f_2)\end{aligned}$$

Note that the result involves the cross-terms $\tilde{\mathbf{T}}(.) \odot \delta(.)$. Therefore, to bound the deviation, one needs not only a bound on the error δ but also potentially a bound on the functional map itself. Some analysis of such bounds was presented in [HCO17] (Condition 3.1 and Proposition 3.3), and we leave the precise analysis of the failure of product preservation as interesting future work.

Now, suppose that a functional map is expected to be of sufficiently "high-quality" to satisfy the product preservation property. For example, the functional map should approximate some *unknown* point-to-point map. Then, we can use Eq. (1) explicitly without computing that point-to-point map. Namely, if we would like to transfer a given real-valued function, we can decompose it into a linear combination of the basis functions *and of their pointwise products*, and then transfer the coefficients by combining the linearity of the map with Eq. (1) to compute the image of products of basis functions, and thus of the given function.

Our main motivation for using this construction is that it allows us to extend the space of functions that can be transferred by a given functional map $\mathbf{C}$ without changing the basis and without converting it to a point-to-point map. For example, consider the standard Fourier basis functions and their pairwise products shown in Figure 2. Note that some products exhibit higher frequency behavior. As a result, they allow a significantly more accurate function reconstruction, shown in Figure 3 for different values of $k_{\mathcal{M}}$. Finally, since Eq. (1) allows us to transfer coefficients of products of basis functions, given the knowledge of images of basis functions themselves, this enables a more accurate function transfer, which can have a direct effect on all applications that use this aspect of functional maps, e.g., [ABCCO13, WHOG14, ACBCO17].

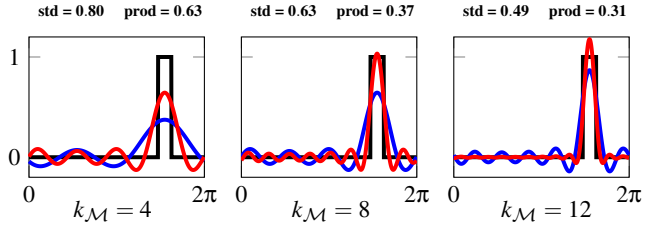

Figure 3: *A 1D indicator function defined on the standard periodic domain (in black), its reconstruction using the standard first five Laplacian basis functions (in blue) and using all the products of the first five eigenfunctions (in red). With an increasing value of k from left to right* $k_{\mathcal{M}} = 4$, $k_{\mathcal{M}} = 8$ *and* $k_{\mathcal{M}} = 12$*. The bases used are the ones from Figure 2. Note the significant improvement in reconstruction error shown above for all the different values of k.*

4. Method Description

More concretely, suppose that we are given a pair of discrete shapes $\mathcal{M}$ and $\mathcal{N}$, represented as triangle meshes containing $n_{\mathcal{M}}$ and $n_{\mathcal{N}}$ points respectively. Moreover, suppose both shapes are endowed with a reduced functional basis, stored as columns of matrices $\Phi_{\mathcal{M}}, \Phi_N$ of size $n_{\mathcal{M}} \times k_{\mathcal{M}}, n_{\mathcal{N}} \times k_{\mathcal{N}}$ for some small $k_{\mathcal{M}}, k_{\mathcal{N}}$. In practice we use the eigenfunctions of the Laplace-Beltrami operators to construct the basis, although the ideas presented below are not tied to this choice.

We also suppose that we are given a functional map, represented in the reduced basis as a matrix $\mathbf{C}$ of size $k_{\mathcal{N}} \times k_{\mathcal{M}}$, which is either induced by a point-to-point map or is close to such a map. For example, if $\mathbf{C}$ comes from the optimization pipeline outlined in Section 3, then we would expect it to be close to satisfying this property, as this pipeline is intended to recover a point-to-point map. Such a functional map should therefore satisfy the distributive relation of Eq. (1). Finally, we are also given a function f, stored simply as a vector $\mathbf{f}$ of size $n_{\mathcal{M}}$, that we would like to transfer from $\mathcal{M}$ to $\mathcal{N}$. Our goal then is to use the distributive property of multiplication, to extend the definition of our map to a space larger than that spanned by the original basis functions.

4.1. Function Representation

To achieve higher accuracy transfer, our approach will be to first represent a given function f as a linear combination of the basis functions $\varphi_1, \ldots, \varphi_k$ *and* of their pointwise products. I.e., we look for the best coefficients a_i and b_j to approximate f as

$$f \approx \sum_{i=1}^{k} a_i \varphi_i + \sum_{j=1}^{P} b_j \prod_{l=1}^{r_j} \varphi_{i_{jl}}, \tag{2}$$

where i_{jl} are indices drawn from $\{1, \ldots, k\}$ (possibly with repetitions) and r_j is the number of terms used in the j^{th} product. Note

that a single basis function can appear multiple times in the same product, which allows representing higher order powers of basis functions. Also note that, in principle, both the number of products P and the number of terms r_j in each product is arbitrary.

Once we find such an approximation, we can transfer f by using the given functional map $\mathbf{C}$ and exploiting Eq. (1).

One difficulty with this approach is that if the functional map is approximate and does not satisfy Eq. (1) exactly, then terms involving function products can amplify the noise present in the map. In order to avoid this effect, we should look for a way to make as few products as possible to approximate f. However, the following theorem (proved in the supplementary material) shows that this problem is NP-hard:

Theorem 1 We consider the following problem (APPROX):
INPUT: A positive integer $n_{\mathcal{M}}$, K basis functions $\varphi_1,\dots,\varphi_K : \{1,\dots,n_{\mathcal{M}}\} \to \mathbb{R}$, a "target" function $f : \{1,\dots,n_{\mathcal{M}}\} \to \mathbb{R}$, and $\varepsilon > 0$
OUTPUT: Minimum cost $c \in \mathbb{N}$, $g_{1,1},\dots,g_{1,r_1},\dots,g_{P,1},\dots,g_{P,r_P} \in \{1,\dots,k\}$, $\alpha_1,\dots,\alpha_P \in \mathbb{R}$ such that:

- $\|f - \sum_{i=1}^{P} \alpha_i \cdot G_i\|_\infty < \varepsilon$, where $G_i = \prod_{j=1}^{r_i} \varphi_{g_{i,j}}$
- $\sum_{i=1}^{P} (r_i - 1) \leq c$

(APPROX) is NP-hard.

Note that c corresponds to the cost that we would like to minimize, since it represents the number of pointwise products used in the approximation. Intuitively, this theorem is related to the optimal sparse (L_0 norm) function approximation, which is also known to be NP-hard [Nat95]. However, due to the special structure of our problem, which involves pointwise products of basis functions, our proof is independent and uses a reduction of 3-SAT directly.

4.2. Extended Functional Basis

Due to Theorem 1 and also because we have observed that using high order products usually brings little improvement in practice, we focus on only using point-wise products between *pairs* of basis functions $\varphi_i^{\mathcal{M}} \odot \varphi_j^{\mathcal{M}}$, where potentially $i = j$. Therefore, we look for the best coefficients a_i and $b_{i,j}$ to approximate f:

$$f \approx \sum_i a_i \varphi_i^{\mathcal{M}} + \sum_{i,j} b_{ij} \varphi_i^{\mathcal{M}} \odot \varphi_j^{\mathcal{M}}. \tag{3}$$

For this, we define an *extended basis* on $\mathcal{M}$ as $\mathbf{B}^{\mathcal{M}} = (\mathbf{\Phi}^{\mathcal{M}}, \widetilde{\mathbf{\Phi}}^{\mathcal{M}})$, where each column of the matrix of $\widetilde{\mathbf{\Phi}}^{\mathcal{M}}$ is of the form $\widetilde{\varphi}_i = \varphi_l^{\mathcal{M}} \odot \varphi_m^{\mathcal{M}}$ for some unique pair $l,m \in \{1,\dots,k_{\mathcal{M}}\}$. Note that we exclude products with the constant function from $\widetilde{\mathbf{\Phi}}^{\mathcal{M}}$, but include both ordered pairs (l,m) and (m,l). We include both ordered pairs only for simplicity of the computation and the formulas involved, although they carry the same information.

Given the extended basis, our goal then is to approximate a given function f by computing a vector of coefficients $\mathbf{a}$ that would minimize $\|\mathbf{B}^{\mathcal{M}}\mathbf{a} - \mathbf{f}\|$. Note, however, that the matrix $\mathbf{B}^{\mathcal{M}}$ is not full rank, and, therefore the best approximation of f may involve a large amplification of noise.

To handle this issue, we use two solutions:

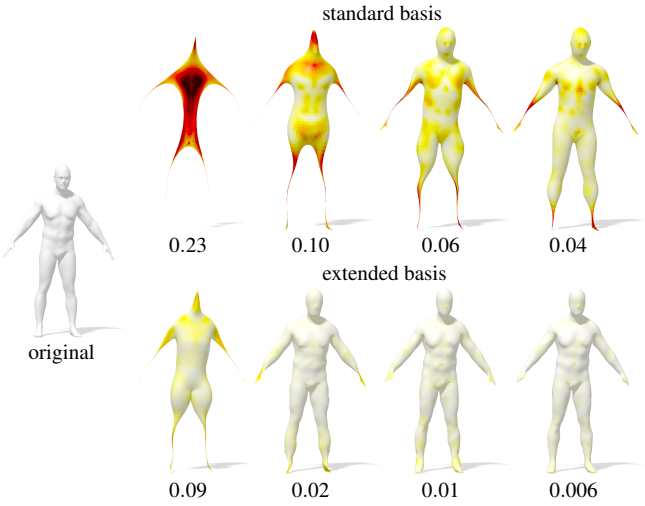

Figure 4: *Surface reconstruction via approximation of the* 3*-coordinates functions. On the left the original shape. On the right, for different values of* k $(9,29,49,69)$*, some surface reconstruction results. On the top using the first* $k+1$ *Laplacian eigenfunctions, bottom adding also pairwise products. Under each shape is reported the error. The colormap encodes reconstruction error, decreasing from dark red to white.*

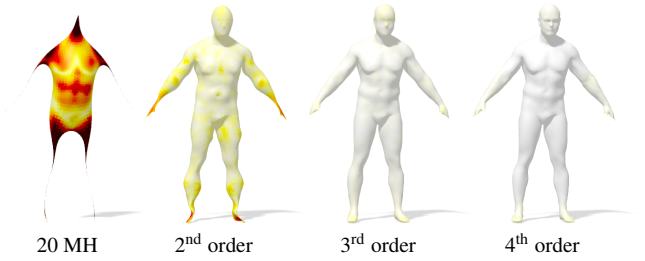

Figure 5: *Surface reconstruction via approximation of the* 3*-coordinates functions using different orders of products. From left to right using* 20 *eigenfunctions, their second order products, third order and finally fourth order products.*

- **Approach A**: compute the optimal coefficients $\mathbf{a}$ by solving the Lasso-type problem:

$$\arg\min_{\mathbf{a}} \|\mathbf{B}^{\mathcal{M}}\mathbf{a} - \mathbf{f}\|_2 + \varepsilon\|\mathbf{a}\|_p, \tag{4}$$

  where we use $p = 1$ or $p = 2$, which respectively promote sparsity and penalize large coefficients, and ε is a small regularizer. To solve Equation 4 with $p = 1$ we use a toolbox for non-smooth convex optimization [Pey] which implements the FISTA method.

- **Approach B**: compute the singular value decomposition (SVD) of $\sqrt{\mathbf{A}^{\mathcal{M}}}\mathbf{B}^{\mathcal{M}}$ as $\sqrt{\mathbf{A}^{\mathcal{M}}}\mathbf{B}^{\mathcal{M}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{A}^{\mathcal{M}}$ is the area matrix associated with the triangle mesh $\mathcal{M}$. We then set all singular values in $\mathbf{\Sigma}$ that are below 0.1% of the maximal value to zero and compute:

$$\mathbf{a} = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^\top\sqrt{\mathbf{A}^{\mathcal{M}}}\mathbf{f}, \tag{5}$$

  Note that this approach is nearly identical to using the pseudo-

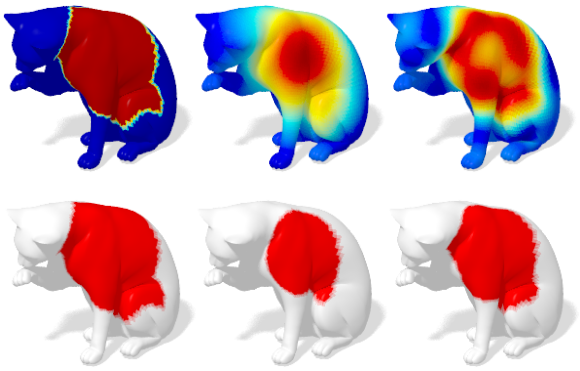

Figure 6: *Reconstruction of an indicator function of a region, using the standard and extended bases. On top we plot the original indicator function and its transfer using standard basis (middle) and extended basis (right). The second row shows the regions detected by thresholding the functions above using the same fixed value, the corresponding error shown below each plot.*

inverse of $\mathbf{B}^{\mathcal{M}}$ to compute $\mathbf{a} = (\mathbf{B}^{\mathcal{M}})^{\dagger}\mathbf{f}$ with the thresholding parameter $0.001\sigma_{\max}$. However we use the area matrix to properly scale the basis with respect to the area elements on the mesh.

The first approach has the advantage of continuously depending on ε, while in the second case computing the optimal coefficients can be done effectively once the matrix $\mathbf{V}\mathbf{\Sigma}^{\dagger}\mathbf{U}^{\top}$ is pre-computed, which allows us to use it to transfer many functions. For these reasons, unless specified, we adopt the **Approach B** below.

We show an example of the reconstruction quality of the coordinate functions using the standard and extended basis in Figure 4 and the same for an indicator function of a region in Figure 6. Note that the use of the extended basis allows to capture higher frequency details even for the same size of the original basis. Therefore, intuitively, we would expect that it would allow more accurate function transfer across shapes.

4.3. Extended Function Transfer

Once the optimal coefficients $\mathbf{a}$ of the function $\mathbf{f}$ in the extended basis $\mathbf{B}^{\mathcal{M}}$ are computed, we use the given functional map $\mathbf{C}$ to transfer $\mathbf{f}$ onto shape $\mathcal{N}$. For this, we first construct an extended basis $\mathbf{B}^{\mathcal{N}}$ using the same procedure as described in Section 4.2 above. We then construct an extended transfer matrix:

$$\widetilde{\mathbf{C}}(\mathbf{C}) = \begin{bmatrix} \mathbf{C} & \mathcal{R}(\mathbf{C}) \\ 0 & \mathbf{C}(1:k,1:k) \otimes \mathbf{C}(1:k,1:k) \end{bmatrix}$$

with

$$\mathcal{R}(\mathbf{C}) = \begin{bmatrix} \phi_0 \mathbf{C}(0,1:k) \otimes \mathbf{C}(0:k,1:k) \\ +\phi_0 \begin{bmatrix} 0 \dots 0 \\ \mathbf{C}(1:k,1:k) \end{bmatrix} \otimes \mathbf{C}(0,1:k) \end{bmatrix}$$

where $\otimes$ is the Kronecker product of matrices, $\mathbf{C}(0,1:k)$ means that we consider the row of index 0 and the columns of indices going from 1 to k, and $\phi_0 = \Phi_{\mathcal{N}}(0,0)$ is the constant value taken by the constant eigenfunction on any point.

The following result (proved in the appendix) shows that $\widetilde{\mathbf{C}}$ allows us to transfer functions expressed in the extended basis

Lemma 2 The image of a function $\mathbf{f} = \mathbf{B}^{\mathcal{M}}\mathbf{a}$ on shape $\mathcal{N}$ is given by $\mathbf{B}^{\mathcal{N}}\widetilde{\mathbf{C}}\mathbf{a}$

This lemma shows that transferring functions in the extended basis can simply be done by matrix-vector multiplication, as is the case in the standard basis. Note that the transfer *is not linear* in the original functional map $\mathbf{C}$, since the construction of $\widetilde{\mathbf{C}}$ involves Kronecker products. This implies that it is not straightforward to include the extended basis in the pipeline for *computing* functional maps, since this would involve terms with products of the unknown map, which could significantly increase the optimization complexity. Nevertheless, once the map $\mathbf{C}$ is computed, using it for function transfer with the extended basis can be done in closed form.

4.4. Function Comparison and Pointwise Map Recovery

We note that in some applications (e.g., converting a functional map to a point-to-point one) it is important to compare functions by simply comparing their coefficients in a reduced basisWhen considering the extended basis $\mathbf{B}_{\mathcal{N}}$ for example, this can lead to problems because the basis is not orthonormal, and furthermore not necessarily full-rank. This means, in particular, that a single function can be represented using multiple different coefficients in the extended basis. To alleviate this issue, similarly to the procedure described in Section 4.2 we compute the SVD of $\sqrt{\mathbf{A}^{\mathcal{N}}}\mathbf{B}^{\mathcal{N}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$, and for every function $\mathbf{f}$ with coefficients $\mathbf{a}$ in the extended basis, such that $\mathbf{f} = \mathbf{B}^{\mathcal{N}}\mathbf{a}$ we define its *canonical* coefficients using $\tilde{\mathbf{a}} = \mathbf{\Sigma}\mathbf{V}^{\top}\mathbf{a}$. Note that in this case

$$\|\mathbf{f}\|^2_{\mathbf{A}} = \mathbf{f}^{\top}\mathbf{A}^{\mathcal{N}}\mathbf{f} = \mathbf{a}^{\top}(\mathbf{B}^{\mathcal{N}})^{\top}\mathbf{A}^{\mathcal{N}}\mathbf{B}^{\mathcal{N}}\mathbf{a} = \mathbf{a}^{\top}\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\top}\mathbf{a}^{\top} = \tilde{\mathbf{a}}^{\top}\tilde{\mathbf{a}},$$

where the second and third equations hold because $\mathbf{f} = \mathbf{B}^{\mathcal{N}}\mathbf{a}$ by assumption and because the area matrix and the diagonal matrix $\mathbf{\Sigma}$ are symmetric, such that $(\mathbf{B}^{\mathcal{N}})^{\top}\sqrt{\mathbf{A}^{\mathcal{N}}}^{\top}\sqrt{\mathbf{A}^{\mathcal{N}}}\mathbf{B}^{\mathcal{N}} = \mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^{\top}$.

It follows that given two functions $\mathbf{f}_1, \mathbf{f}_2$ with coefficients $\mathbf{a}_1, \mathbf{a}_2$ in the extended basis, we can compute the norm of their difference as simply the L_2-norm of the difference $\|\tilde{\mathbf{a}}_1 - \tilde{\mathbf{a}}_2\|_2$.

Pointwise map recovery With this procedure in place, we can exploit Eq. (1) and the extended basis for recovering a point-to-point map from a given functional map $\mathbf{C}$. Note that the input functional map is always given in the original basis, and we only use extended basis for more accurate function transfer. For converting $\mathbf{C}$ to a pointwise map, we first compute the canonical coefficients of Dirac δ functions at each point y on shape $\mathcal{N}$ in the extended basis. We then compute the image of each δ function of points on shape $\mathcal{M}$ and compute its canonical coefficients. Finally, following the procedure described in [OBCS*12] we construct a point-to-point map by looking at the nearest neighbors in the coefficient space. This procedure has the advantage of being efficient, since functions are represented in the space of dimension at most $\tilde{k}_{\mathcal{N}} = k_{\mathcal{N}} + \frac{k_{\mathcal{N}}^2 + k_{\mathcal{N}}}{2}$, where the second terms represents the number of distinct pairwise products of basis functions. For small values of $k_{\mathcal{N}}$ the value $\tilde{k}_{\mathcal{N}}$ is still significantly lower than the number of vertices. In practice we use (squared) heat kernel instead of Dirac δ functions for representing points, as described in Section 5.3 below.

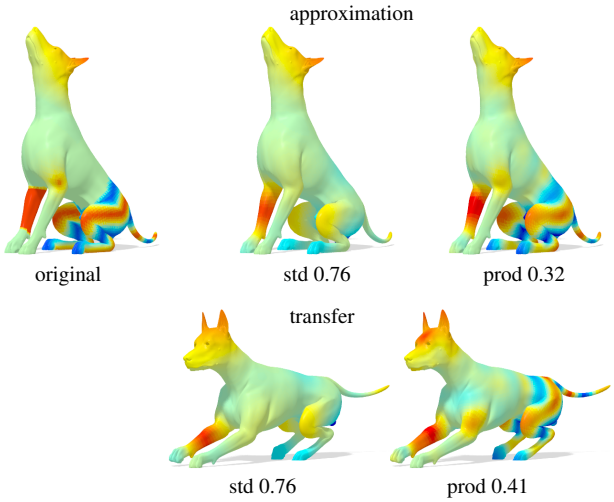

Figure 7: *Approximation (top) and transfer (bottom) of a real-valued function from a source shape onto a target shape using a fixed functional map with the standard approach (center) vs. our extended method (right).*

5. Results

In the following experiments we use two standard datasets: FAUST [BRLB14] an TOSCA [BBK08]. The former consists of 100 shapes, with 10 subjects in 10 poses, represented as triangle meshes with the same connectivity and with ground truth point-wise correspondences. TOSCA high resolution dataset contains synthetic models in 7 different shape classes with ground truth correspondences given in each class.

5.1. Function approximation and transfer

In our first application, we evaluate the utility of our function transfer procedure using both ground truth (arising from known point-wise maps) and computed functional maps. Namely, given a set of pairs of (source, target) shapes and a collection of different functions on each source, we evaluate: i) the approximation of each function on the source shape, and ii) the transfer of a function between the source and the target shape. Figure 7 shows a qualitative example of approximation and transfer of a real-valued function, with the original function shown on the left. This function is generated as a combination of an indicator function (on the left leg) a gaussian around a point (top of the right leg) a sine function of the y coordinate (on the tail) and a continuously increasing function (on the ears). We compare the standard approach (std) vs. our extended method (prod), for both function approximation and transfer onto a different pose of the same shape. Note the improvement with our approach (Approach B) as captured by the mean squared error reported under each plot.

In our quantitative experiments we consider the following families of functions:

- hk k, hk K: the heat kernel functions $h_t(x,\cdot)$ between a random point x and the rest of the shape approximated using $k_{\mathcal{M}}+1$ eigenfunctions (for hk k) and using 200 eigenfunctions (for hk K). Note that in the former case, the function is contained in the span of the original basis, while in the latter $200 > k_{\mathcal{M}}$.
- HKS, WKS: the Heat and Wave Kernel Signatures [SOG09, ASC11] for 10 randomly time and energy values respectively.
- Random: the function obtained as a linear combination of the extended basis using a random set of coefficients.
- XYZ: the X, Y, Z, coordinates of vertices.
- Indicator: the binary indicator function of a random region.
- SHOT, AWFT: the SHOT [TSDS10] and AWFT [MRCB16] descriptors for 10 randomly chosen dimensions.

ground truth	**approx**		**transfer**			
function	**std**	**our l1**	**std**	**our l1**	**our l2**	**our B**
hk k	**0.00**	0.01	0.06	0.06	0.06	**0.05**
hk K	0.91	**0.86**	0.91	0.87	0.92	**0.86**
HKS	0.54	**0.00**	0.54	**0.15**	**0.15**	**0.15**
WKS	0.21	**0.00**	0.21	**0.04**	**0.04**	**0.04**
Random	0.38	**0.03**	0.38	0.07	**0.06**	**0.06**
XYZ	0.29	**0.21**	0.29	**0.23**	0.27	**0.23**
Indicator	0.36	**0.28**	0.37	**0.28**	0.32	**0.28**
SHOT	0.84	**0.82**	0.84	**0.82**	0.84	**0.82**
AWFT	0.31	**0.26**	0.31	**0.27**	0.30	**0.27**

Table 1: *Approximation and transfer quality results using a ground truth functional map with $k_{\mathcal{M}} = k_{\mathcal{N}} = 9$ of various functions on* 20 *shape pairs from the FAUST dataset. Note that for all functions except hk k, which lies in the span of the original basis, our approach produces a significant improvement.*

We compare the function approximation and transfer using the standard method (**std**) with the two approaches described in Section 4.2. For Approach A we have two different regularizations using the L_1-norm and the L_2-norm (**our l1** and **our l2**). We denote our Approach B with **our B**. For the L_1 regularization we used the *Sparse Optimization Toolbox* for Matlab [Pey].

Table 1 shows the results for function approximation and transfer using ground truth functional maps on a set of shape pairs from the FAUST dataset [BRLB14]. Here and below, the errors are computed as the normalized integral of the difference between the function f and the ground truth g as $err = \sqrt{\int_{\mathcal{M}}(f-g)^2 dx} \Big/ \sqrt{\int_{\mathcal{M}}(g)^2 dx}$, where, dx is the area element induced by the metric. Note that all of our approaches result in a significant improvement for approximation and transfer. Note also that although the L_1 regularization produces good results, it performs similarly to other approaches. At the same time, as it is significantly more time-consuming, and requires solving a convex optimization problem for every function to be transferred, we omit it from further evaluation and rely on the L_2 regularization for Approach A.

In Table 2 we show a similar evaluation but on a *computed* functional map, using a recent approach of [NO17], which estimates a functional map by exploiting a set of descriptors on each shape, where we use the WKS for some energy values. Here again, our approaches show a significant improvement with respect to the baseline standard method for all functions. Throughout our experiments we observed that when increasing the basis size $k_{\mathcal{M}}$ on the source,

computed	approx			transfer		
function	**std**	**our A**	**our B**	**std**	**our A**	**our B**
hk k	**0.00**	0.03	0.01	0.13	0.13	0.13
hk K	0.82	0.54	**0.49**	0.82	**0.57**	0.58
HKS	0.55	0.14	**0.00**	0.55	**0.22**	**0.22**
WKS	0.14	0.03	**0.00**	0.14	0.07	**0.06**
Random	0.48	0.04	**0.01**	0.49	**0.15**	**0.15**
XYZ	0.12	0.09	**0.06**	0.13	**0.11**	**0.11**
Indicator	0.28	0.19	**0.17**	0.28	0.20	**0.19**
SHOT	0.81	0.76	**0.74**	0.81	**0.77**	0.78
AWFT	0.24	0.19	**0.17**	0.25	**0.20**	**0.20**

Table 2: *Approximation and transfer quality using a* computed *functional map with $k_{\mathcal{M}} = 29, k_{\mathcal{N}} = 39$ using the method from [NO17] of various functions on* 20 *shape pairs from the FAUST dataset. Note in particular that our transfer approach produces a significant improvement.*

computed	$k_{\mathcal{M}} = 6, k_{\mathcal{N}} = 7$		$k_{\mathcal{M}} = 27, k_{\mathcal{N}} = 35$	
function	**std**	**our B**	**std**	**our B**
hk k	0.77	**0.60**	0.08	**0.06**
hk K	0.92	**0.87**	0.79	**0.58**
HKS	0.66	**0.61**	0.55	**0.17**
WKS	0.47	**0.25**	0.17	**0.04**
Random	0.75	**0.69**	0.46	**0.09**
Coordinates	0.34	**0.24**	0.12	**0.10**
Indicator	0.58	**0.34**	0.28	**0.19**
SHOT	0.90	**0.88**	0.84	**0.81**
AWFT	0.35	**0.29**	0.25	**0.21**

Table 3: *Transfer quality comparison using a ground truth functional map with $k_{\mathcal{M}} = 6$ and $k_{\mathcal{N}} = 7$ and $k_{\mathcal{M}} = 27$ and $k_{\mathcal{N}} = 35$, average on* 20 *pairs from FAUST dataset. Note in particular that our transfer approach produces a significant improvement as the number of basis functions used increases.*

it is often beneficial to have $k_{\mathcal{N}} > k_{\mathcal{M}}$ since this allows to better represent the transfer of basis functions.

Table 3 shows how the transfer using standard and our approach benefit from the increase in the basis size. In the first two columns we use a ground truth functional map represented by a matrix $\mathbf{C}$ with $k_{\mathcal{M}} = 6$ and $k_{\mathcal{N}} = 7$. In the last two columns we use $k_{\mathcal{M}} = 27$ and $k_{\mathcal{N}} = 35$ non-constant basis functions. These dimensions are chosen since e.g., starting with k basis functions, the number of different functions obtained by adding the pairwise products is at most: $k + \frac{k(k+1)}{2}$. The transferred functions that depend on the basis dimension (e.g., the HKS) are computed using the larger value of $k_{\mathcal{M}}$ in all cases. As can be seen in Table 3, both the standard and our approach benefit from the increase in the basis size, while our approach is always better for a given basis size. Observe that the results obtained with the standard approach using more basis function are close to those obtained with our approach and fewer basis functions. This is remarkable, given the optimality of the Laplacian eigenfunctions for representing functions with bounded variation, as shown by Aflalo et al. [ABK15]. Furthermore increasing the number of basis functions in the standard approach requires

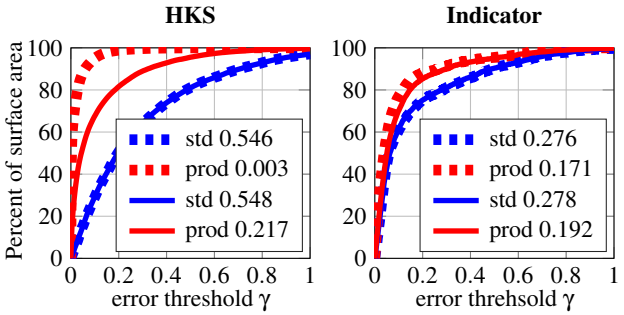

Figure 8: *Plots of the percentage of the surface area for which the function approximation and transfer is below some threshold for the HKS (left) and the indicator function of a region (right), using computed functional maps on* 20 *pairs of shapes from the FAUST dataset. Dashed and solid lines represent function approximation and transfer respectively.*

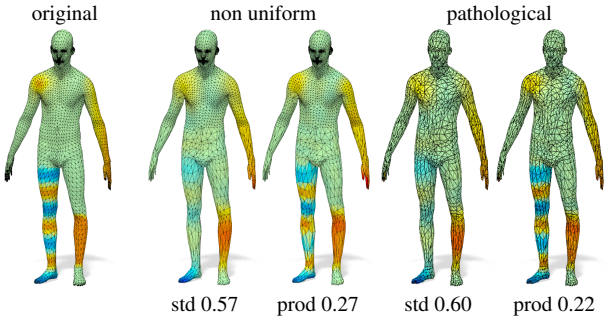

Figure 9: *An example of function transfer from a source shape with a regular mesh (left), to two different meshes with different triangulations. From left to right: a non uniform resampling (fewer vertices on the legs), a mesh with pathological triangles (around the 60% are very thin triangles).*

solving an eigen-problem and the estimation of new coefficients in the functional map. Conversely our approach improves the representation and the transfer of functions without any new estimation of coefficients. Please see the Appendix for results using functional maps of various sizes. We also compared our method with first converting the functional map to a point-to-point one and using the latter for function transfer. This approach produces even more error than the standard method, as it introduces an additional source of noise.

In Figure 8 we show a different representation of the results from Table 2 for the HKS and the indicator functions. Namely, we plot the fraction of the surface area for which the difference between the approximated and the ground truth functions is below a threshold $\gamma \in [0,1]$. I.e., the x-axis represents the error threshold γ, whereas the y axis shows the surface area of all points x such that $(f(x) - g(x))^2 < \gamma\sqrt{\int g(x)^2 dx}$. Figure 9 shows some results on the stability of transfer between different meshes. Starting from an original mesh and a function (left), we compare the transfer using the standard basis and our extended basis. The two meshes on which we compare the transfer have a different connectivity: a mesh with non uniform resampling (fewer vertices on the bottom of the shape), a mesh with around 60% of pathological triangles (very

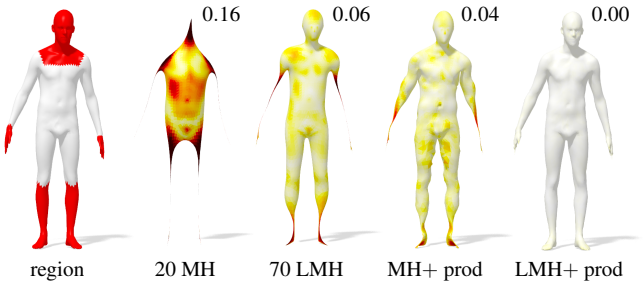

Figure 10: *Surface reconstruction via approximation of the* 3*-coordinates functions using two different bases: the standard manifold harmonics (MH) and localized manifold harmonics (LMH). On the left we show the original shape and the region (in red) in which are localized the LMH. On the right of each shape we report the reconstruction error. The colormap encodes reconstruction error, decreasing from dark red to white.*

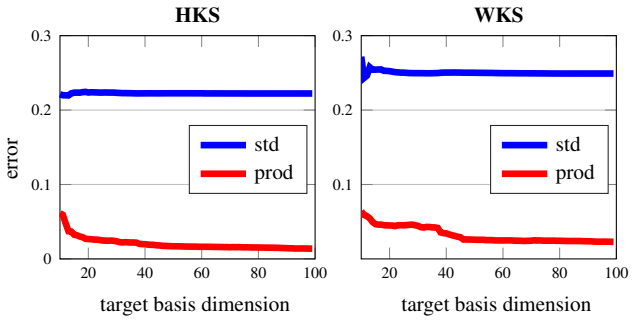

Figure 11: *Comparison on the transfer for HKS and WKS, average on 5 pairs from FAUST dataset [BRLB14]. Starting with* $k+1=10$ *basis functions on the source shape we compute HKS and WKS for* 1 *scale, and their coefficients in the* $k+1$ *fixed basis and its products extension. Varying the basis dimension on the target shape from* 10 *to* 100*, we define the* $\boldsymbol{C}$ *and the* $\widetilde{\boldsymbol{C}}$ *and compute the tranfer of HKS and WKS on the target shape. Here we plot the transfer error (y-axis) for HKS (left) and WKS (right) varying the basis dimension on the target shape (x-axis).As can be seen the error with products is smaller and decreases with increasing basis dimension while without products error stops decreasing.*

thin triangles). As can be seen, our approach gives better results and is stable with respect to changes in mesh connectivity.

Approximation in other bases As mentioned above, our extended basis construction is not tied to the Laplacian eigenfunctions. To illustrate this, in Figure 10 we show how the pointwise products improve the functional space representation in a different basis. Namely, we perform the same test that is in Figure 4, but using different basis manifold harmonics (MH) and the recently proposed localized manifold harmonics (LMH) [MRCB17]. As can be seen, for both of these bases pairwise products allow a more accurate representation of the surface.

5.2. HKS and WKS approximation and transfer

One interesting observation related to our method is that both the Heat Kernel Signature and the Wave Kernel Signature functions are

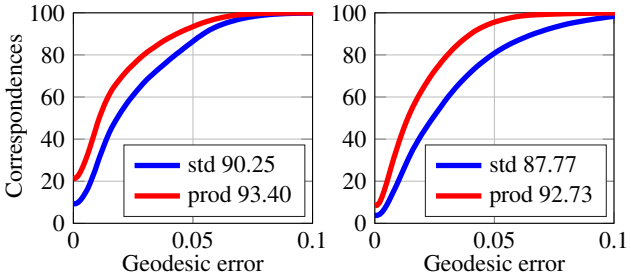

Figure 12: *Quality of correspondences obtained from a given ground truth functional map of dimension* 40×30*, between all the possible pairs of shapes for one subject from FAUST dataset (left), and for all the pairs of shapes for the centaur from TOSCA high resolution dataset (right).*

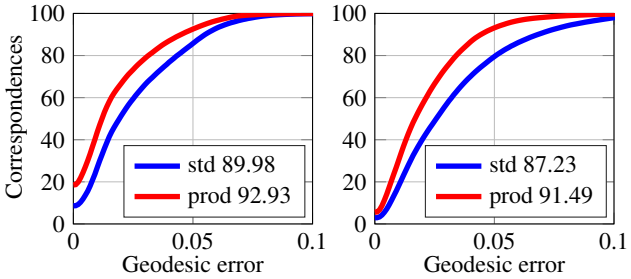

Figure 13: *Quality of correspondences obtained from a* computed functional map *of size* 40×30*, between all the possible pairs of shapes for one subject from FAUST dataset (left), and all pairs of shapes for the centaur from TOSCA high resolution dataset (right).*

of the form $h_t(x,x)=\sum_{i=1}^{k_{\mathcal{M}}}\alpha_i\phi_i^2(x)$, for some scalar coefficients α_i. In other words, they are constructed explicitly using squares of eigenfunctions and therefore it must be possible to represent and transfer them *exactly* in our extended basis. Therefore, they provide a good test for the correctness of both our extended function approximation and transfer methods. To demonstrate the difference between the transfer of these functions using the standard and the extended basis we show in Figure 11 the transfer error using the ground truth functional map of size $k_{\mathcal{N}}\times k_{\mathcal{M}}$ for increasing $k_{\mathcal{N}}$. Note that for large values of $k_{\mathcal{N}}$ we can approximate the transfer of all $k_{\mathcal{M}}$ basis functions from the source well, which means that we would expect our extended transfer method to produce progressively better results. This can clearly be seen in Figure 11 where our method converges to a very small value while the standard technique does not improve. This is because the functions ϕ_i^2 cannot be well-approximated in the basis of the source shape, which means that regardless of the basis size on the target, we cannot achieve small error.

5.3. Point-to-point map recovery

We also demonstrate the utility of our approach for *converting* functional maps to point-to-point ones. As mentioned in Section 4.4, our extended basis can also be used to embed points in a coefficient space where function comparison can be done directly, which can be useful for point-to-point map recovery. In particular, we

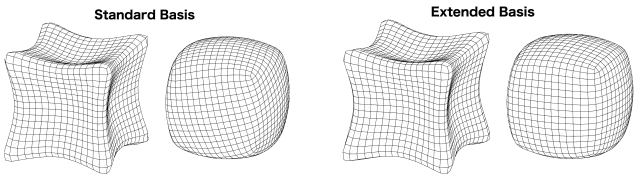

Figure 14: *Joint quadrangulation of two triangle meshes using the approach of [ACBCO17] with the standard Laplacian $k = 5$ eigenfunctions (left) and our extended (right) basis. Note that the presence of products of basis functions allows us to transport higher frequency information, which leads to a more consistent result.*

represent each point on the shape as the square of the heat kernel for a small time scale centered at that point. We then convert a functional map to a point-to-point staring with a ground truth map and also a computed one using the method of [NO17] on a set of pairs from the FAUST [BRLB14] TOSCA high resolution datasets [BBK08]. We plot the conversion results in Figures 12 and 13 using the ground truth and estimated functional maps respectively, both of size 40×30. Note that as explained in Section 5.1 it is beneficial to rectangular matrices $\mathbf{C}$ in order to take full advantage of the use of the products. As can be seen in Figures 12 and 13 the extended basis and our improved function transfer also contribute to better point-to-point map recovery.

5.4. Joint quadrangulation

We also used our approach in the context of joint quadrangulation of triangle meshes by adapting the recent technique presented in [ACBCO17]. This method is based on using functional maps for constructing consistent cross fields on a pair of surfaces, which are in turn used for designing approximately consistent quad-meshes. Moreover, the pipeline presented in [ACBCO17] only relies on the ability of functional maps to transfer real-valued functions and does not require point-wise correspondences. We adapted this method to take advantage of the extended basis by simply enabling the transfer of pointwise products of basis functions, without any other modification, using the code provided by the authorss. Figure 14 presents a result on a pair of shapes using the standard Laplacian basis with $k = 5$ eigenfunctions and using our extended basis. Note that the presence of pointwise products in the extended basis allows us to transfer higher frequency information which leads to a more consistent result overall. In this application we used a functional map arising from a ground truth pointwise correspondence. Although preliminary, this result suggests the utility of our extended function transfer for joint quadrangulation, and we leave a more in-depth exploration of this application as future work.

6. Conclusion, Limitations and Future Work

In this paper, we presented a novel method for function transfer with functional maps, by exploiting the algebraic structure of function spaces. We showed that by extending the functional basis to include *pointwise products* of basis functions, we can significantly improve both the reconstruction and the transfer quality of functions, while maintaining the computational complexity of the original functional map. Our approach has direct consequences on all applications of functional maps, and in particular shows how high-frequency information can be transferred even in the presence of only low frequency basis functions.

Our main limitation is that in the current formulation we only use products of pairs of basis functions. Although we have observed that higher-order products do not often bring significant improvement, a more in-depth analysis of the scenarios in which they can be useful is necessary. In the future, we also plan to work on more scalable methods for function approximation, by removing the need to explicitly compute the entire extended basis, which can be prohibitive for large bases.

Acknowledgements The authors would like to thank Omri Azencot for help with the joint quadrangulation application. Parts of this work were supported by the chaire Jean Marjoulet from Ecole Polytechnique, FUI project TANDEM 2, a Google Focused Research Award, ERC Starting Grant No. 758800 (EXPROTEA), ERC Consolidator Grant No. 724228 (LEMAN), and Rudolf Diesel fellowship from the TUM Institute for Advanced Study.

References

[ABCCO13] AZENCOT O., BEN-CHEN M., CHAZAL F., OVSJANIKOV M.: An operator approach to tangent vector field processing. In *Computer Graphics Forum* (2013), vol. 32, pp. 73–82. 2, 3, 4

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

[ACBCO17] AZENCOT O., CORMAN E., BEN-CHEN M., OVSJANIKOV M.: Consistent functional cross field design for mesh quadrangulation. *ACM Trans. Graph. 36*, 4 (July 2017), 92:1–92:13. 2, 3, 4, 10

[AK13] AFLALO Y., KIMMEL R.: Spectral multidimensional scaling. *PNAS 110*, 45 (2013), 18052–18057. 2

[AL16] AIGERMAN N., LIPMAN Y.: Hyperbolic orbifold tutte embeddings. *ACM TOG 35*, 6 (2016). 1

[APL15] AIGERMAN N., PORANNE R., LIPMAN Y.: Seamless Surface Mappings. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 72. 1

[ASC11] AUBRY M., SCHLICKEWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops* (2011), IEEE, pp. 1626–1633. 3, 7

[BBK06] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS 103*, 5 (2006), 1168–1172. 2

[BBK08] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: *Numerical Geometry of Non-Rigid Shapes*. Springer Science & Business Media, 2008. 7, 10

[BBM05] BERG A. C., BERG T. L., MALIK J.: Shape matching and object recognition using low distortion correspondences. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 1, pp. 26–33. 2

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

[BRLB14] BOGO F., ROMERO J., LOPER M., BLACK M. J.: FAUST: Dataset and Evaluation for 3d Mesh Registration. In *Proc. CVPR* (2014), pp. 3794–3801. 1, 7, 9, 10

[Cut13] CUTURI M.: Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems* (2013), pp. 2292–2300. 2

[DK10] DUBROVINA A., KIMMEL R.: Matching shapes by eigendecomposition of the Laplace-Beltrami operator. *Proc. 3DPVT 2*, 3 (2010). 2

[HCO17] HUANG R., CHAZAL F., OVSJANIKOV M.: On the stability of functional maps and shape difference operators. *Computer Graphics Forum* (2017). 4

[HGK04] HAZEWINKEL M., GUBARENI N., KIRICHENKO V.: *Algebras, rings and modules*. Kluver Academic Publishers, 2004. 2

[HSS*09] HASLER N., STOLL C., SUNKEL M., ROSENHAHN B., SEIDEL H.-P.: A statistical model of human pose and body shape. In *Computer Graphics Forum* (2009), vol. 28, pp. 337–346. 1

[HWG14] HUANG Q., WANG F., GUIBAS L.: Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 36. 2, 3

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

[KGB16] KOVNATSKY A., GLASHOFF K., BRONSTEIN M. M.: MADMM: a generic algorithm for non-smooth optimization on manifolds. In *Proc. ECCV* (2016), Springer, pp. 680–696. 3

[KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended Intrinsic Maps. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 79. 2

[KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric Modeling in Shape Space. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, p. 64. 1

[LF09] LIPMAN Y., FUNKHOUSER T.: Möbius Voting for Surface Correspondence. In *ACM Transactions on Graphics (TOG)* (2009), vol. 28, ACM, p. 72. 1, 2

[LH05] LEORDEANU M., HEBERT M.: A spectral technique for correspondence problems using pairwise constraints. In *Proc. ICCV* (2005), vol. 2, IEEE, pp. 1482–1489. 2

[LRB*16] LITANY O., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M., CREMERS D.: Non-rigid puzzles. In *Computer Graphics Forum* (2016), vol. 35, pp. 135–143. 2, 3

[LRBB17] LITANY O., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.: Fully spectral partial shape matching. *Computer Graphics Forum 36*, 2 (2017), 247–258. 2

[LRR*17] LITANY O., REMEZ T., RODOLÀ E., BRONSTEIN A. M., BRONSTEIN M. M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proc. ICCV* (2017). 2

[MCSK*17] MANDAD M., COHEN-STEINER D., KOBBELT L., ALLIEZ P., DESBRUN M.: Variance-minimizing transport plans for inter-surface mapping. *ACM Transactions on Graphics 36* (2017), 14. 2

[Mém07] MÉMOLI F.: On the use of Gromov-Hausdorff distances for shape comparison. *Point-Based Graphics* (2007). 2

[Mém11] MÉMOLI F.: Gromov–Wasserstein Distances and the Metric Approach to Object Matching. *Foundations of computational mathematics 11*, 4 (2011), 417–487. 2

[MHK*08] MATEUS D., HORAUD R., KNOSSOW D., CUZZOLIN F., BOYER E.: Articulated shape matching using laplacian eigenfunctions and unsupervised point registration. In *Proc. CVPR* (2008), IEEE, pp. 1–8. 2

[MRCB16] MELZI S., RODOLA E., CASTELLANI U., BRONSTEIN M.: Shape analysis with anisotropic windowed fourier transform. In *International Conference on 3D Vision (3DV)* (2016). 7

[MRCB17] MELZI S., RODOLÀ E., CASTELLANI U., BRONSTEIN M. M.: Localized manifold harmonics for spectral shape analysis. *arXiv:1707.02596* (2017). 2, 9

[Nat95] NATARAJAN B. K.: Sparse approximate solutions to linear systems. *SIAM Journal on Computing 24*, 2 (1995), 227–234. 5

[NO17] NOGNENG D., OVSJANIKOV M.: Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum 36*, 2 (2017), 259–267. 3, 7, 8, 10

[NVT*14] NEUMANN T., VARANASI K., THEOBALT C., MAGNOR M., WACKER M.: Compressed manifold modes for mesh processing. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 35–44. 2

[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, 2, 3, 6

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

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

[OMPG13] OVSJANIKOV M., MÉRIGOT Q., PĂTRĂUCEAN V., GUIBAS L.: Shape matching via quotient spaces. In *Computer Graphics Forum* (2013), vol. 32, pp. 1–11. 2

[Pey] PEYRÉ G.: Toolbox sparse optimization. https://www.mathworks.com/matlabcentral/fileexchange/16204. Accessed: 2017-10-10. 5, 7

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

[RMC15] RODOLÀ E., MOELLER M., CREMERS D.: Point-wise map recovery and refinement from functional correspondence. In *Proc. Vision, Modeling and Visualization (VMV)* (2015). 2, 3

[SDGP*15] SOLOMON J., DE GOES F., PEYRÉ G., CUTURI M., BUTSCHER A., NGUYEN A., DU T., GUIBAS L.: Convolutional wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 66. 2

[SK14] SHTERN A., KIMMEL R.: Matching the lbo eigenspace of non-rigid shapes via high order statistics. *Axioms 3*, 3 (2014), 300–319. 2

[SK15] SHTERN A., KIMMEL R.: Spectral gradient fields embedding for nonrigid shape matching. *Computer Vision and Image Understanding 140* (2015), 21–29. 2

[SM93] SINGH R. K., MANHAS J. S.: *Composition Operators on Function Spaces*, vol. 179. Elsevier, 1993. 3

[SNB*12] SOLOMON J., NGUYEN A., BUTSCHER A., BEN-CHEN M., GUIBAS L.: Soft maps between surfaces. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1617–1626. 1, 2

[SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion. In *Computer graphics forum* (2009), vol. 28, pp. 1383–1392. 7

[SP04] SUMNER R. W., POPOVIĆ J.: Deformation Transfer for Triangle Meshes. *ACM TOG 23*, 3 (2004), 399–405. 1

[SPKS16] SOLOMON J., PEYRÉ G., KIM V. G., SRA S.: Entropic metric alignment for correspondence problems. *ACM Transactions on Graphics (TOG) 35*, 4 (2016), 72. 1, 2

[TCL*13] TAM G. K., CHENG Z.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.: Registration of 3D point clouds and meshes: a survey from rigid to nonrigid. *IEEE TVCG 19*, 7 (2013), 1199–1217. 2

[TSDS10] TOMBARI F., SALTI S., DI STEFANO L.: Unique signatures of histograms for local surface description. In *International Conference on Computer Vision (ICCV)* (2010), pp. 356–369. 7

[VKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. In *Computer Graphics Forum* (2011), vol. 30, pp. 1681–1707. 1, 2

[VLB*17] VESTNER M., LÄHNER Z., BOYARSKI A., LITANY O., SLOSSBERG R., REMEZ T., RODOLA E., BRONSTEIN A., BRONSTEIN M., KIMMEL R., CREMERS D.: Efficient deformable shape correspondence via kernel matching. In *Proc. 3DV* (2017). 2

computed	approx			transfer		
function	**std**	**our A**	**our B**	**std**	**our A**	**our B**
hk k	**0.00**	0.01	0.01	0.05	0.05	0.05
hk K	0.88	**0.75**	0.78	0.88	0.82	**0.79**
HKS	0.54	**0.00**	**0.00**	0.54	**0.11**	**0.11**
WKS	0.22	**0.00**	0.01	0.22	**0.04**	**0.04**
Random	0.38	**0.00**	**0.00**	0.38	**0.05**	**0.05**
XYZ	0.28	**0.18**	0.20	0.28	0.25	**0.23**
Indicator	0.40	**0.28**	0.29	0.41	0.33	**0.31**
SHOT	0.84	**0.80**	0.81	0.84	0.83	**0.82**
AWFT	0.31	**0.24**	0.25	0.32	0.29	**0.27**

Table 4: $k_{\mathcal{M}} = 9$ *and* $k_{\mathcal{N}} = 9$*, average on* 20 *pairs from FAUST dataset. Computed functional map*

ground truth	approx			transfer		
function	**std**	**our A**	**our B**	**std**	**our A**	**our B**
hk k	**0.00**	0.03	0.01	0.26	**0.20**	**0.20**
hk K	0.79	0.52	**0.45**	0.80	**0.57**	0.75
HKS	0.57	0.14	**0.00**	0.57	**0.43**	0.54
WKS	0.12	0.03	**0.00**	0.12	0.07	**0.05**
Random	0.47	0.04	**0.01**	0.48	**0.17**	0.18
XYZ	0.12	0.09	**0.06**	**0.12**	0.13	0.16
Indicator	0.27	0.18	**0.17**	0.27	**0.20**	0.25
SHOT	0.82	0.76	**0.75**	0.82	**0.78**	0.88
AWFT	0.24	0.19	**0.17**	0.25	**0.22**	0.27

Table 5: $k_{\mathcal{M}} = 29$ *and* $k_{\mathcal{N}} = 29$*, average on* 20 *pairs from FAUST dataset. Given a ground truth functional map*

ground truth	approx			transfer		
function	**std**	**our A**	**our B**	**std**	**our A**	**our B**
hk k	**0.00**	0.03	0.01	0.09	0.09	**0.08**
hk K	0.79	0.54	**0.49**	0.79	**0.57**	0.61
HKS	0.55	0.14	**0.00**	0.56	0.18	**0.14**
WKS	0.13	0.03	**0.00**	0.13	0.05	**0.04**
Random	0.46	0.04	**0.01**	0.46	**0.09**	**0.09**
XYZ	0.12	0.09	**0.06**	0.12	0.10	**0.09**
Indicator	0.28	0.19	**0.18**	0.28	0.20	**0.19**
SHOT	0.82	0.76	**0.75**	0.82	**0.77**	0.78
AWFT	0.24	0.19	**0.17**	0.25	**0.20**	**0.20**

Table 6: $k_{\mathcal{M}} = 29$ *and* $k_{\mathcal{N}} = 39$*, average on* 20 *pairs from FAUST dataset. Given a ground truth functional map*

[WHOG14] WANG F., HUANG Q., OVSJANIKOV M., GUIBAS L. J.: Unsupervised multi-class joint image segmentation. In *Proc. CVPR* (2014), pp. 3142–3149. 2, 3, 4

7. Appendix

7.1. Additional Results

Tables 4, 5 and 6 show approximation and transfer quality of different functions using ground truth and computed functional maps of different sizes.

7.2. Proof of Theorem 1

Please see the supplementary material.

7.3. Proof of Lemma 2:

The image of a function $f = B_{\mathcal{M}}\mathbf{f}$ on shape $\mathcal{N}$ is given by $B_{\mathcal{N}}\widetilde{\mathbf{C}}\mathbf{f}$. where

$$\widetilde{\mathbf{C}} = \begin{bmatrix} & \phi_0\mathbf{C}(0,1:k)\otimes\mathbf{C}(0:k,1:k) \\ \mathbf{C} & +\phi_0\begin{bmatrix}0\dots0\\ \mathbf{C}(1:k,1:k)\end{bmatrix}\otimes\mathbf{C}(0,1:k) \\ 0 & \mathbf{C}(1:k,1:k)\otimes\mathbf{C}(1:k,1:k)\end{bmatrix} \tag{6}$$

Proof Let $K_{\mathcal{M}} = 1 + k_{\mathcal{M}} + {k_{\mathcal{M}}}^2$ be the number of columns in $B_{\mathcal{M}}$. By linearity, it is sufficient to prove the statement for each $e_i^{(K_{\mathcal{M}})} = (0,\dots,0,1,0,\dots,0) \in \mathbb{R}^{K_{\mathcal{M}}}$ (the 1 is at position i).

- if $0 \le i \le k_{\mathcal{M}}$:
  By definition of $\widetilde{\mathbf{C}}$, $B_{\mathcal{N}}\widetilde{\mathbf{C}}\mathbf{f} = \Phi_{\mathcal{N}}\mathbf{C}e_i^{(k_{\mathcal{M}}+1)}$ which is the image of $B_{\mathcal{M}}e_i^{(k_{\mathcal{M}}+1)} = f$.
- if $i > k_{\mathcal{M}}$:
  we write $i = k_{\mathcal{M}} + k_{\mathcal{M}}(i_1-1) + i_2$, which represents the pair of indices $1 \le i_1, i_2 \le k_{\mathcal{M}}$.

$$\begin{aligned}
B_{\mathcal{N}}\widetilde{\mathbf{C}}\mathbf{f} &= \Phi_{\mathcal{N}}(\phi_0(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_0(Ce_{i_2}^{(k_{\mathcal{M}}+1)})) \\
&\quad + \phi_0(Ce_{i_2}^{(k_{\mathcal{M}}+1)})_0(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_{1:k_{\mathcal{M}}}) \\
&\quad + (\Phi_{\mathcal{N}}\otimes\Phi_{\mathcal{N}})((Ce_{i_1}^{(k_{\mathcal{M}}+1)})_{1:k_{\mathcal{M}}}\otimes(Ce_{i_2}^{(k_{\mathcal{M}}+1)})_{1:k_{\mathcal{M}}}) \\
&= (\Phi_{\mathcal{N}}Ce_{i_2}^{(k_{\mathcal{M}}+1)})\odot(\Phi_{\mathcal{N}}(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_0) \\
&\quad + (\Phi_{\mathcal{N}}(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_{1:k})\odot(\Phi_{\mathcal{N}}(Ce_{i_2}^{(k_{\mathcal{M}}+1)})_0) \\
&\quad + (\Phi_{\mathcal{N}}(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_{1:k})\odot(\Phi_{\mathcal{N}}(Ce_{i_2}^{(k_{\mathcal{M}}+1)})_{1:k}) \\
&= (\Phi_{\mathcal{N}}Ce_{i_2}^{(k_{\mathcal{M}}+1)})\odot(\Phi_{\mathcal{N}}(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_0) \\
&\quad + (\Phi_{\mathcal{N}}(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_{1:k})\odot\big((\Phi_{\mathcal{N}}(Ce_{i_2}^{(k_{\mathcal{M}}+1)})_0) \\
&\qquad + (\Phi_{\mathcal{N}}(Ce_{i_2}^{(k_{\mathcal{M}}+1)})_{1:k})\big) \\
&= (\Phi_{\mathcal{N}}Ce_{i_2}^{(k_{\mathcal{M}}+1)})\odot(\Phi_{\mathcal{N}}(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_0) \\
&\quad + (\Phi_{\mathcal{N}}(Ce_{i_1}^{(k_{\mathcal{M}}+1)})_{1:k})\odot(\Phi_{\mathcal{N}}Ce_{i_2}^{(k_{\mathcal{M}}+1)}) \\
&= (\Phi_{\mathcal{N}}Ce_{i_2}^{(k_{\mathcal{M}}+1)})\odot(\Phi_{\mathcal{N}}Ce_{i_1}^{(k_{\mathcal{M}}+1)}),
\end{aligned}$$

which is the point-wise product of the images of $B_{\mathcal{M}}e_{i_1}^{k_{\mathcal{M}}+1}$ and $B_{\mathcal{M}}e_{i_2}^{(k_{\mathcal{M}}+1)}$, which is, by Eq. (1), the image of $B_{\mathcal{M}}e_{i_1}^{(k_{\mathcal{M}}+1)}\odot B_{\mathcal{M}}e_{i_2}^{(k_{\mathcal{M}}+1)} = f$.

□