



**HAL**  
open science

# Complex Functional Maps : a Conformal Link Between Tangent Bundles

Nicolas Donati, Etienne Corman, Simone Melzi, Maks Ovsjanikov

► **To cite this version:**

Nicolas Donati, Etienne Corman, Simone Melzi, Maks Ovsjanikov. Complex Functional Maps : a Conformal Link Between Tangent Bundles. Computer Graphics Forum, 2022, 41 (1), pp.317-334. 10.1111/cgf.14437 . hal-03480605

**HAL Id: hal-03480605**

**<https://inria.hal.science/hal-03480605>**

Submitted on 17 Dec 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Complex Functional Maps : a Conformal Link Between Tangent Bundles

Nicolas Donati<sup>1</sup>, Etienne Corman<sup>2</sup>, Simone Melzi<sup>3</sup>, Maks Ovsjanikov<sup>1</sup>

<sup>1</sup>LIX, Ecole Polytechnique, IP Paris, <sup>2</sup>Université de Lorraine, CNRS, Inria, LORIA, France, <sup>3</sup>Sapienza, University of Rome

## Abstract

In this paper, we introduce complex functional maps, which extend the functional map framework to conformal maps between tangent vector fields on surfaces. A key property of these maps is their orientation awareness. More specifically, we demonstrate that unlike regular functional maps that link functional spaces of two manifolds, our complex functional maps establish a link between oriented tangent bundles, thus permitting robust and efficient transfer of tangent vector fields. By first endowing and then exploiting the tangent bundle of each shape with a complex structure, the resulting operations become naturally orientation-aware, thus favoring orientation and angle preserving correspondence across shapes, without relying on descriptors or extra regularization. Finally, and perhaps more importantly, we demonstrate how these objects enable several practical applications within the functional map framework. We show that functional maps and their complex counterparts can be estimated jointly to promote orientation preservation, regularizing pipelines that previously suffered from orientation-reversing symmetry errors.

## CCS Concepts

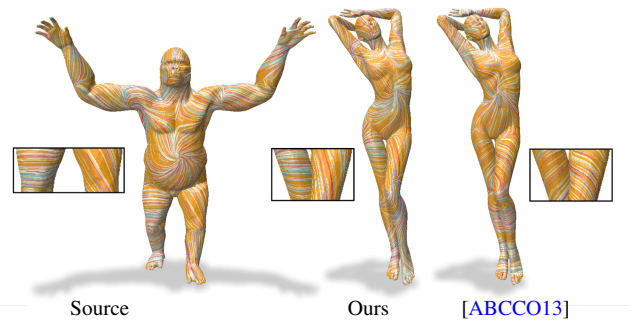
•Computing methodologies → Shape analysis; •Theory of computation → Computational geometry;

## 1. Introduction

Non-rigid shape matching is a well-established challenge in computer graphics, geometry processing and related fields [VKZHCO11, Sah20], with applications ranging from medical imaging to statistical shape analysis, to name a few.

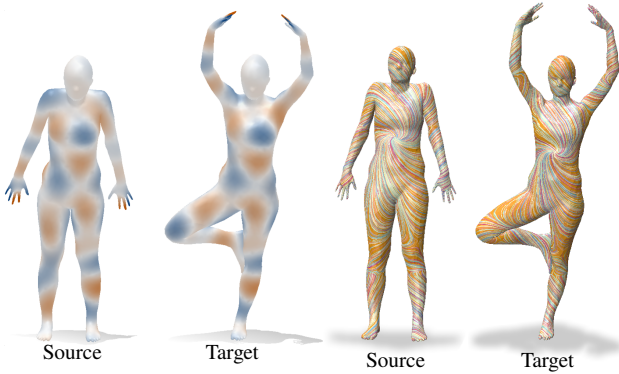
One prominent direction for addressing this problem is given by the functional map framework [OBCS\*12]. This framework is based on representing correspondences as linear transformations between function spaces, and encoding them as matrices using a reduced basis. A key advantage of this construction is that it allows to both optimize for and to manipulate mappings by solving small-scale optimization problems, whose complexity is largely independent of the size of the underlying meshes. Furthermore, the continuous nature of this representation enables the use of differentiable optimization techniques, which has recently proven useful in learning pipelines, e.g., [LRR\*17, HLR\*19, GR20].

Despite the flexibility and simplicity of the functional map representation, it has several key limitations: first, while functional maps encode correspondences between points, they do not immediately provide access to maps between derived quantities such as the surface metric or tangent vector fields that require the notion of a map differential. Several attempts have been made to recover differential information in the functional map framework, e.g., [ABCCO13, CSBC\*17]. However, these approaches often lead to non-trivial optimization problems and, as we demonstrate be-



**Figure 1:** A comparison to [ABCCO13] in a vector field transfer application. The transfer is done with a functional map mixing symmetries (see Section 5.1). Unlike the method of [ABCCO13], our approach is robust to this type of noise and transfers the vector field correctly, without breaking its asymmetry.

low, can be prone to error especially when faced with approximate maps in the reduced basis (see Figure 1). Perhaps even more importantly, the functional map representation does not encode information about the *surface orientation*, which means that standard functional map optimization energies can easily lead to orientation-reversing correspondences that may arise, e.g., due to intrinsic symmetries. Existing methods try to tackle this challenge through a range of solutions including by using landmarks [NO17, MMRC20], injecting orientation into descriptor-based energies [RPWO18] or alignment in the ambient space [RMOW20] among others. Unfortu-



**Figure 2:** A visualization of function transfer (left) and vector field transfer (right) obtained respectively via a functional map and our complex functional map. The values of the functions on the left are encoded by a colormap, while the vector fields are visualized using line integral convolution. Both functions and vector fields can be decomposed in a Laplacian basis and be transferred using small matrices.

nately, these solutions often require additional user input and careful parameter tuning or incur significant computational cost.

In this work, we introduce a novel construction that helps to address these challenges directly, without relying on user input or post-processing. Our key idea is to build a representation for correspondences that *only allows* orientation and angle-preserving (conformal) maps. Such a representation is, by its nature, more restrictive than the functional map representation, since only a subset of maps is allowed. However, as we demonstrate below, using this representation and especially the link with standard functional maps helps to regularize map computation and to improve accuracy in several applications without sacrificing expressive power.

To achieve this goal, we first observe that in the case of conformal maps, the pushforward (also called “map differential”) is a *complex linear* operator between tangent bundles. As a result, it can be encoded as a small-sized complex-valued matrix, given a choice of basis for the tangent vector fields on each shape. Since a single vector field can be represented as a *complex function*, and the pushforward allows to transfer vector fields across shapes, we call these operators *complex functional maps*. As we highlight below, complex functional maps have several properties that are complementary to the standard functional map representation. Specifically, they provide a simple and robust tool for transferring vector fields; furthermore, they allow to disambiguate intrinsic symmetries and help to promote more accurate, smooth, orientation-preserving point-to-point correspondences.

**Contributions** To summarize, our main contributions include:

- We introduce a novel complex-linear operator acting on tangent vector fields, used as relaxation of the map differential.
- We demonstrate that this operator is naturally orientation-aware, and show how it can be used to regularize functional map estimation, especially with respect to symmetry errors.
- We highlight the utility of our construction in a range of applications, from robust vector field transfer to orientation-aware map refinement, leading to consistent improvement in accuracy.

## 2. Related Work

Computing maps or correspondences between non-rigid 3D shapes is a key challenge in geometry processing and computer vision. Below, we review some approaches in this area and focus especially on methods that we either build upon or use as baselines, and refer to recent surveys [VKZHC011, BCBB16, Sah20] for a more in-depth discussion.

**Functional maps framework** Our method heavily relies on the functional map representation, which was originally introduced in [OBSC\*12] as a tool for non-rigid shape matching. The key idea of this representation is to represent point-to-point maps as small matrices, encoded in a reduced basis, which greatly simplifies subsequent optimization problems. The original work used only a basic set of linear constraints on functional maps, which have been extended significantly in, e.g., [KBB\*13, AK13, HWG14, ERGB16, BDK17, RCB\*17, NO17, HO17, NMR\*18, RPWO18, MRR\*19] among many other works (see [OCB\*17]). These approaches heavily exploit the compact and continuous nature of the functional map representation and have been used to improve the accuracy, speed and robustness of the resulting shape matching pipeline. However, the functional representation itself has two major limitations that limit its applicability, as we review below.

**Functional representations of differential quantities** As mentioned in the introduction, functional maps do not naturally provide information about derived quantities such as the map differential. Several attempts have been made to recover differential information in the functional map framework. This includes operator representations for tangent vector fields [ABCCO13] and cross fields [ACBCO17] as well as covariant derivatives and parallel transport [AOCBC15]. These operator representations enable tasks such as computing vector field flow efficiently, which has been used both for matching functions on surfaces [AVBC16] and even for recovering continuous maps between shapes [COC15]. Furthermore, functional representations have been also proposed for extrinsic “deformation fields” [CO19] and for capturing the intrinsic shape metric [ROA\*13, CSBC\*17], among others (see also Chapter 6 in [OCB\*17] and [BCA19] for an overview of some recent approaches).

These constructions significantly extend the power and flexibility of the functional maps framework. At the same time, the basic concept of the map differential or pushforward, and thus reliable mapping of *tangent vector fields* is still cumbersome to define and use within this formalism. As we demonstrate below, in the special case of orientation-preserving conformal maps, however, this differential has a particularly convenient representation, which provides complementary information to standard functional maps.

**Orientation preservation** Another common challenge to using the functional map representation is that it does not encode surface orientation. This implies that typically-used optimization energies, e.g., based on preservation of intrinsic descriptors, such as the HKS or WKS [SOG09, ASC11] can lead to undesirable orientation-reversing correspondences (also known as symmetry flips). One can tackle this issue by using extrinsic descriptors such as SHOT [TSDS10], but such descriptors can be very sensitive to discretization [PO18, DSO20], which can have dramatic effects

on robustness. Existing methods have tried to address this challenge by using either segment [OBCS\*12, KO18] or pointwise landmarks [NO17, MMRC20], injecting orientation information with descriptor-based energies [RPWO18], factoring the functional space using symmetry information [OMPG13], alignment in the ambient space [ELC20] or, most recently, using map space exploration strategies [RMOW20]. However, since the functional map representation itself does not encode orientation information, these solutions only address the problem indirectly, and, e.g., in [RMOW20] orientation-preserving maps are selected *a posteriori* among the set of candidates, using a set of filtering criteria.

In contrast, we demonstrate that by first endowing the tangent bundle with the complex algebra using the outward normals, and then defining the derived *complex functional maps* it becomes possible to directly promote orientation-preserving correspondences without any additional descriptor preservation constraints or post-processing.

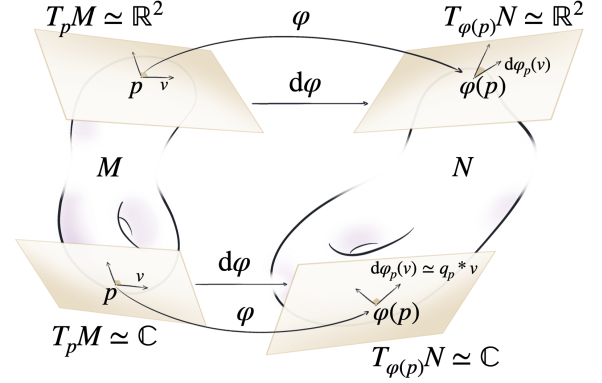
**Vector Field Map Representation** Closest to our construction is the work of [WLZT18], where the authors extend the functional map representation to differential forms on manifolds. Similarly to our approach, that method encodes the pushforward as a linear operator acting on vector fields, while imposing orthogonality, which is a necessary condition to arise from a conformal map. However, crucially, the authors of [WLZT18] use  $\mathbb{R}$ -linear operators to encode the pushforward, whereas we use the outward normals to construct and exploit the *complex algebra* on the tangent space, leading to  $\mathbb{C}$ -linear operators. This difference is fundamental, as it implies that the representation in [WLZT18] *cannot* distinguish orientation-preserving from orientation-reversing maps. As we demonstrate in Section 5, this severely limits the scope of applications of that representation, which are, in contrast, enabled by our approach thanks, in particular, to its orientation-aware nature.

### 3. Tangent Bundle Map as Operator

In this section, we describe the theoretical aspects of our complex functional maps. At a high level, we follow the motivation behind the original functional map framework, by ultimately providing a linear relaxation of a particular geometric concept. While functional maps aim at representing diffeomorphisms as linear operators acting on functions, our goal is to represent the pushforward of a conformal map as a  $\mathbb{C}$ -linear operator acting on complex fields.

#### 3.1. Notation

From now on, we consider a pair of compact Riemannian surfaces  $M, N$  embedded in  $\mathbb{R}^3$ . We use  $T_p M$  to denote the tangent plane at a point  $p \in M$ , while the *tangent bundle*  $TM := \cup_{p \in M} T_p M$ , is the disjoint union of all the tangent planes of  $M$ . We equip this space with a proper inner product (i.e. the Riemannian metric)  $\langle \cdot, \cdot \rangle_{T_p M} : T_p M \times T_p M \rightarrow \mathbb{R}$ , which depends smoothly on  $p \in M$ . For every pair of real-valued functions  $f, g : M \rightarrow \mathbb{R}$ , we use another,  $L^2$  inner product, which is defined as  $\langle f, g \rangle_M = \int_M f(p)g(p)d\mu_M(p)$ , where  $d\mu_M$  is the area element on the surface  $M$ . With respect to this inner product, we define  $L^2(M) = \{f : M \rightarrow \mathbb{R} \text{ s.t. } \langle f, f \rangle_M < +\infty\}$ , as the space of square-integrable real-valued function defined over  $M$ . Similarly



**Figure 3:** The pushforward  $d\varphi$  takes a tangent vector  $v \in T_p M$  and transports it to the plane tangent to  $\varphi(p) \in N$ . In all generality (top) surface tangent planes identify to  $\mathbb{R}^2$  and the pushforward is locally a  $\mathbb{R}^2$ -linear map. If  $\varphi$  is conformal (bottom), tangent planes identify to the complex plane  $\mathbb{C}$  and  $d\varphi$  is locally a multiplication by a complex number  $q_p \in \mathbb{C}$ .

for every pair of vector fields  $X, Y : M \rightarrow TM$  we consider their inner product  $\langle X, Y \rangle_{TM} = \int_M \langle X(p), Y(p) \rangle_{T_p M} d\mu_M(p)$ . Finally we denote by  $L_M : TM \rightarrow TM$  the connection Laplacian acting on vector fields.

#### 3.2. Pushforward in the smooth setting

A diffeomorphism  $\varphi : M \rightarrow N$  bijectively maps points on a surface  $M$  to points on a surface  $N$ . The pushforward  $d\varphi : T_p M \rightarrow T_{\varphi(p)}N$  associated to  $\varphi$  maps the tangent space at point  $p \in M$ , denoted as  $T_p M$ , to the tangent space  $T_{\varphi(p)}N$  and can be understood as the best linear approximation of the map at the given point  $p$  (see Figure 3 top). Thus, the pushforward contains two pieces of information: 1) which tangent plane of  $N$  corresponds to a given tangent plane of  $M$  and 2) *how* a tangent plane is deformed by the diffeomorphism  $\varphi$ . While the first is already contained in  $\varphi$ , the second is of very different nature, and is especially difficult to recover in the discrete setting as it requires to numerically differentiate the mapping.

For surfaces, tangent planes can be identified to the vector space  $\mathbb{R}^2$ . Given an arbitrary diffeomorphism, the pushforward acts *linearly* on the tangent vectors (see [GP10], Chapter 2) so at that each point  $p \in M$ ,  $(d\varphi)_p$  can be represented as a linear map between two Euclidean spaces. This map can be represented by a  $2 \times 2$  matrix if each space is endowed with a local 2D basis formed by two linearly independent tangent vectors. For a more in-depth discussion of map differentials, we refer to [GP10, Lee13].

#### 3.3. Pushforwards of conformal maps

In this paper, we are especially interested in the pushforward of a conformal map. Conformal maps have been widely used in computer graphics notably for texture mapping [RL03, SSP08], parametrization [KSS06, MTAD08, SC17] and shape matching [LF09, KLF11] among others. Their success is mostly due to their simple structure-preserving property: a conformal map preserves angles between tangent vectors. This means that, by definition, the pushforward of

a conformal map is a *similarity transformation* (i.e., a combination of rotation and uniform scaling) of the tangent space at every point.

Each tangent plane on an orientable surface can also be naturally identified with the complex plane  $\mathbb{C}$  by identifying an arbitrary fixed direction with the real axis, and using the outward normal to determine the  $90^\circ$  counter-clockwise rotations, associated with multiplication by  $\iota$ . Moreover, observe that for any fixed complex number  $q$ , the transformation  $S : \mathbb{C} \rightarrow \mathbb{C}$  given by complex multiplication by  $q$ , and defined as  $S(v) = q * v$ , is a *similarity transformation* of  $\mathbb{R}^2$ . To see this, observe that in polar coordinates complex multiplication simply adds the arguments (angles) and multiplies the magnitudes. Conversely, it is easy to see that for any similarity transformation  $S : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ , there exists a unique complex number  $q$  such that  $S(v) = q * v$ , where we implicitly identify  $\mathbb{R}^2$  with  $\mathbb{C}$  on both sides of the equality (see Figure 3 bottom). These observations can be therefore summarized in the following lemma:

**Lemma 3.1** Given a smooth map  $\varphi : M \rightarrow N$  between surfaces, if each tangent plane  $T_p$  and  $T_{\varphi(p)}$  is identified with the complex plane, then the map  $\varphi$  is conformal if and only if for every point  $p \in M$  there exists a complex number  $q_p$  such that the pushforward  $d\varphi_p(v) = v * q_p$ .

Stated differently, a map is conformal if and only if its pushforward, which can be thought of as a mapping between (two copies of) the complex plane, can be represented, at every point, as multiplication by some fixed complex number.

In general the pushforward is a real-linear map between tangent spaces identified with  $\mathbb{R}^2$ . However, in the special case of *conformal* mappings, the pushforward is a  $\mathbb{C}$ -linear map. Crucially, while any  $\mathbb{C}$ -linear map is also real-linear, the converse, of course, does not hold. Remark also that up to technical conditions of differentiability, Lemma 3.1 is equivalent to the well-known *Cauchy-Riemann equations*, which can be stated compactly as saying that a mapping is conformal if and only if the associated map differential is  $\mathbb{C}$ -linear (see [Rem91], p. 51 for a discussion).

### 3.4. Complex functional maps

Following a similar reasoning as in the functional map framework [OBCS\*12], in which a functional map is *any* linear transformation between functions spaces, we now consider the space of *all* complex linear maps between *tangent bundles*. As mentioned above, we implicitly assume that the tangent bundles are endowed with the complex structure given by some reference direction at every tangent space, and using the outward normal to define counter-clockwise rotation.

Thus we call a *complex functional map* any  $\mathbb{C}$ -linear operator  $Q$  that maps a complex field  $X \in TM \simeq \cup_{p \in M} \mathbb{C}$  to a complex field  $Q(X) \in TN \simeq \cup_{p \in N} \mathbb{C}$  on  $N$ , where we use  $TM \simeq \cup_{p \in M} \mathbb{C}$  to denote the identification between tangent spaces and complex planes.

Remark that the complex structure allows us to represent any tangent vector field as a complex function, and, as a result, the operator  $Q$  enables the transfer of tangent vector fields defined on  $M$  to those defined on  $N$ . Crucially, by Lemma 3.1 we have that the pushforward of a conformal map must be a  $\mathbb{C}$ -linear operator and thus a

complex functional map. The converse, however, does not hold, as not all  $\mathbb{C}$ -linear operators on the tangent bundle come from conformal pushforwards, in a similar way that not all linear functional mappings arise from pullbacks of pointwise correspondences.

In the remainder of the section we list the key properties of complex functional maps. In Section 3.5 we exhibit a necessary and sufficient condition for a complex functional map to represent the pushforward of a conformal mapping. Section 3.6 studies additional useful properties, notably a weaker necessary condition for  $Q$  to uniformly scale tangent planes and a characterization of isometric pushforwards as a subset of conformal maps.

### 3.5. Pushforwards vs. Complex-Linear Maps

As introduced above, a complex functional map  $Q : TM \rightarrow TN$  is *any*  $\mathbb{C}$ -linear operator between tangent bundles. Our goal is to obtain a necessary and sufficient condition for such an operator to represent the pushforward (differential) of a conformal mapping. For this, we will use the following two key properties.

**Connection to pullbacks** First, suppose that  $\varphi : M \rightarrow N$  is any diffeomorphism between two smooth surfaces. It is well-known that the pushforward  $d\varphi : T_p M \rightarrow T_{\varphi(p)} N$  is the unique linear map between tangent spaces that satisfies:

$$\langle X, \nabla(f \circ \varphi) \rangle_{T_p M} = \langle d\varphi(X), \nabla f \rangle_{T_{\varphi(p)} N}, \quad (1)$$

for all  $X \in TM, f \in L^2(N)$  (see [Lee13], Chapter 3).

**Orientability** Second, consider any  $\mathbb{C}$ -linear mapping  $Q$  between tangent bundles. By linearity, for any complex number  $q \in \mathbb{C}$  and any two complex fields  $X, Y \in TM \simeq \cup_{p \in M} \mathbb{C}$ , we have:

$$X = qY \iff Q(X) = qQ(Y).$$

For instance, if  $q = \iota$ , the map  $Q$  preserves  $90^\circ$  rotation. As rotations are defined relative to the local basis of the tangent plane, they carry the information of the *orientation* of the manifold. Therefore **the operator  $Q$  cannot change the orientation of the tangent bundle**. This property of the complex functional maps is key in our shape matching applications: it forces the maps to remain orientation-preserving. Combining these observations, we conclude that complex functional maps can only represent the differential of an *orientation preserving* conformal map. Put differently, we have:

**Theorem 3.1** The complex-linear map  $Q$  is a pushforward if and only if there exists an *orientation-preserving* and conformal diffeomorphism  $\varphi : M \rightarrow N$  satisfying:

$$\langle X, \nabla(f \circ \varphi) \rangle_{T_p M} = \langle Q(X), \nabla f \rangle_{T_{\varphi(p)} N},$$

for all  $X \in TM, f \in L^2(N)$ .

Thm. 3.1 is a direct consequence of Lemma 3.1 and the uniqueness property of the map differential in Eq. (1). For completeness we provide the full argument in Appendix A.

This theorem highlights the importance of Eq. (1) to link the complex-linear map  $Q$  with the underlying pointwise map  $\varphi$ . Note that importantly, Eq. (1) only depends on the knowledge of the *pullback* associated with  $\varphi$ , and thus provides a natural link between  $Q$  and the standard functional map representation, which encodes

pullbacks. In Section 4.2, we show how this property can be used to relate our complex and standard functional maps in practice.

### 3.6. Properties of complex functional maps

In this section we provide two additional structural properties of complex functional maps.

**Orthogonality.** First, we show a *necessary* condition for  $Q$  to represent the differential of a conformal map:  $Q$  must be an orthogonal operator. Interestingly, this is different from standard functional maps, which are orthonormal if and only if the underlying correspondence is locally area-preserving [OBCS\*12, ROA\*13].

**Theorem 3.2** If  $Q$  represents the pushforward of a conformal map  $\varphi$  between surfaces, then:

$$Q^*Q = I,$$

where  $Q^*$  is the adjoint operator uniquely defined by  $\langle QX, Y \rangle_{TN} = \langle X, Q^*Y \rangle_{TM}$ .

Intuitively, this theorem comes from the fact the change of metric under the conformal map is given by some scaling factor at each tangent plane. When *integrating* the inner products on the surface, this scaling factor cancels out with the change of area measure. A similar result was shown in [ROA\*13] for gradients of functions. We provide the complete proof in Appendix B.

Importantly, Theorem 3.2 only establishes a necessary condition: indeed, even if  $Q$  is an *orthonormal*  $\mathbb{C}$ -linear operator it is only guaranteed to represent the pushforward of a conformal map, if it satisfies Eq. (1).

**Isometries.** Secondly, a pushforward is isometric if and only if it commutes with the Levi-Civita connection [Car92] (p.181). A similar statement can be made about the connection Laplacian recently used in geometry processing [SSC19] to compute transport of vector fields along geodesic paths.

**Theorem 3.3** Let  $L$  be the connection Laplacian. If a complex functional map  $Q : TM \rightarrow TN$  represents the pushforward  $d\varphi$  of a conformal map, then it satisfies:

$$L_N \circ Q = Q \circ L_M$$

if and only if  $\varphi$  is an isometry.

Theorem 3.3, proved in detail in Appendix C, is very similar to that of functional maps [OBCS\*12].

### 3.7. Operators in a reduced basis

In order to improve the efficiency of our algorithms it is often desirable to consider operators acting on a subspace of smooth vector fields. Since tangent planes can be identified to complex planes, we will use the *complex inner product* defined as:

$$\langle X, Y \rangle_{\mathbb{C}_p M} := \langle X, Y \rangle_{T_p M} + i \langle \mathcal{J}X, Y \rangle_{T_p M} \in \mathbb{C},$$

where  $\mathcal{J}$  is the  $90^\circ$  rotation around the normal. Note that for two vectors  $x, y$  in the same complex plane this inner product is equivalent to the standard  $\bar{x}y$ , where  $\bar{x}$  is the complex conjugate of  $x$ .

Suppose that  $M$  is equipped with the family of vector fields  $\{\Psi_i^M\}$

orthonormal with respect to the inner product  $\langle \cdot, \cdot \rangle_{\mathbb{C}M}$ . Then [RS80] (Thm II.6), any vector field  $X$  can be written as a linear combination:

$$X = \sum_i a_i \Psi_i^M,$$

where  $a_i = \langle X, \Psi_i^M \rangle_{\mathbb{C}M} \in \mathbb{C}$ . The transferred complex field  $Q(X)$  can be decomposed in the basis of  $N$ :

$$\begin{aligned} Q(X) &= \sum_j \langle Q(X), \Psi_j^N \rangle_{\mathbb{C}N} \Psi_j^N \\ &= \sum_j \Psi_j^N \sum_i \langle Q(\Psi_i^M), \Psi_j^N \rangle_{\mathbb{C}N} a_i, \end{aligned}$$

where the second equality follows from the  $\mathbb{C}$ -linearity of  $Q$ . Thus the complex functional map can be understood as a matrix with coefficients  $Q_{ji} = \langle \Psi_j^N, Q(\Psi_i^M) \rangle_{\mathbb{C}N}$ , matching coefficients in basis on  $M$  to coefficients of the basis on  $N$ . Figure 2 provides a visual comparison between a classical functional map transferring functions (*left*) and a complex functional map transporting tangent vector fields (*right*), both using a reduced basis of size 30.

## 4. Discrete Setting

In this section, we introduce complex functional maps in the discrete setting. Throughout, we consider oriented manifold triangle meshes  $(V, E, F)$ . Our overall strategy is based on representing tangent vector fields as complex-valued pointwise functions, so that  $X_i \in \mathbb{C}$  per vertex  $i \in V$ . This choice of a point-based representation for tangent vector fields (and not face-based, as in e.g. [ABCCO13]) is motivated by our main application: disambiguating symmetries in non-rigid shape correspondence problems. Matching vertices between surface meshes is convenient as it directly allows to transfer texture coordinates or deformations.

We thus represent discrete complex functional maps  $Q$  as complex-valued matrices mapping complex-valued pointwise functions on  $M$  to complex-valued pointwise functions on  $N$ . In a similar spirit to functional maps, we improve computational efficiency by representing the operator  $Q$  as a small matrix in a reduced basis of vector fields. As a basis, we use the first eigenvectors of the *connection* Laplacian, discretized as in [SSC19].

Below we introduce Laplacian operators and the necessary local complex structure, then we construct our complex functional map and translate each continuous property in the discrete setting.

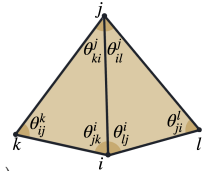
### 4.1. Laplacian operators

**Cotan-Laplacian** The standard Laplacian operator  $W \in \mathbb{R}^{|V| \times |V|}$  for a piecewise linear function  $f \in \mathbb{R}^{|V|}$  is obtained by the well-known cotan-weight formula [BKP\*10]:

$$(Wf)_i := \frac{1}{2} \sum_{(ij) \in E} \left( \cot \theta_{ij}^k + \cot \theta_{ji}^l \right) (f_i - f_j),$$

where the index notation is defined in the inset figure. We also define the diagonal  $|V| \times |V|$  lumped mass matrix  $A$ :

$$A_{ii} := \frac{1}{3} \sum_{(ijk) \in T} a_{ijk},$$



where  $a_{ijk}$  are the areas of triangles adjacent to  $i$ . This matrix defines the scalar product in the space of piecewise linear functions. Namely if  $f, g \in \mathbb{R}^{|V|}$  are two piecewise linear real-valued functions then:

$$\langle f, g \rangle_{L^2} := f^T A g.$$

**Connection Laplacian** Our discretization of tangent vector fields using complex numbers follows [KCPS13, SSC19]. Namely, we assign to each vertex an arbitrary reference unit vector orthogonal to the vertex normal. This reference direction represents the tangent vector, associated with the complex number  $1 + \iota 0$ . The outward normal dictates the orientation of the tangent planes, providing the additional axis  $\iota$ . Given this reference frame, any tangent vector can be represented as a complex number, and a tangent vector field as a complex-valued function.

In this context, the mass matrix  $A$  defines a complex scalar product in the space of tangent fields, represented as complex functions:

$$\langle X, Y \rangle_{CM} := X^* A Y,$$

where  $*$  represents the conjugate-transpose operation.

As mentioned above, in our applications we use a family of smooth vector fields given by the first  $k$  first eigenfunctions of the discrete *connection Laplacian*, as defined in [SSC19]. The connection Laplacian is the matrix  $L \in \mathbb{C}^{|V| \times |V|}$  uniquely defined by the Dirichlet energy:

$$X^* L X = \frac{1}{2} \sum_{(ij) \in E} (\cot \theta_{ij}^k + \cot \theta_{ji}^l) |X_i - r_{ij} X_j|^2,$$

where the unit complex number  $r_{ij}$  are the rotations necessary to compare vectors expressed in two different bases.

By definition,  $L$  is a complex Hermitian matrix, and similarly to the cotangent Laplacian, on Delaunay meshes has only real non-negative eigenvalues. Therefore,  $L$  admits the generalized eigendecomposition:

$$L \Psi = A \Psi \Lambda,$$

where  $\Psi^* A \Psi = I$ ,  $\Lambda$  the diagonal matrix of eigenvalues, and  $\Psi$  is a set of complex-valued eigenfunctions. In particular, any tangent vector field  $U$  can be expressed as a linear combination *with complex coefficients* of the family  $\{\Psi_i, i = 1, \dots, |V|\}$  [HJ12] (Thm. 4.1.5), i.e.:  $U = \sum_i c_i \Psi_i$ , where  $c_i \in \mathbb{C}$ . In practice we truncate this sum and typically use a fixed number  $k$  of the complex eigenfunctions, associated with the eigenvalues of smallest modulus.

## 4.2. Discrete complex functional map

In order to define a discrete equivalent of the pushforward, we simply discretize the continuous definition in Eq. (1). For meshes of same connectivity we are able to derive a closed-form expression for  $Q$  and a consistent notion of discrete conformality. For meshes with different connectivity or when the deformation is not exactly conformal we enforce this equation in the least-squares sense.

To discretize Eq. (1), we will need the pullback operator  $C$ , represented by a functional map [OBSC\*12], and the operator  $D_X$

often encountered in differential geometry to define tangent vectors [Mor01] and introduced in geometry processing for vector field design in [ABCCO13].

To define the discrete pullback, recall that given a map  $\varphi : M \rightarrow N$ , the associated functional map  $C : L^2(N) \rightarrow L^2(M)$  can be discretized in the full, “hat”, basis as a binary matrix  $C_{NM} = \Pi_{MN}$ , where  $\Pi_{MN}(i, j) = 1$  if and only if  $\varphi(i) = j$ , while in the reduced basis we have  $C_{NM} = (\Phi^M)^\top A^M \Pi_{MN} \Phi^N$  where  $\Phi^M, \Phi^N$  are matrices that store, as columns, the basis functions on the two shapes.

**Vector field operator** In addition, we will use the linear functional operator  $D_X : L^2(M) \rightarrow L^2(M)$  describing the action of a vector field  $X \in TM$  on a function  $f \in L^2(M)$ :

$$D_X(f)_p := \langle X, \nabla f \rangle_{T_p M}. \quad (2)$$

This operator uniquely characterizes a tangent vector field  $X$  on a manifold [Mor01], and will allow us to write Eq. (1) as an equality between matrices. Azencot *et al.* [ABCCO13] proposed a discretization for face-based vector fields, however, as previously stated, we use a different vertex-based discretization, that we describe in detail below.

We discretize the operator  $D_X \in \mathbb{R}^{|V| \times |V|}$  as a vertex-wise scalar product at tangent planes:

$$(D_X f)_i := \langle X_i, \nabla f_i \rangle_{T_i M}, \quad (3)$$

where  $\nabla f_i$  evaluates the gradient of a piece-wise linear function at vertex  $i$  (rather than at a face). In practice, we store the operator  $D_X$  into a complex sparse matrix.

Locally, the gradient of a function represents the best  $\mathbb{R}^2$ -linear approximation of this function in the tangent plane. The directional derivative of  $f$  along an edge vector  $e_{ij}$  is simply  $(f_i - f_j)/|e_{ij}|$ . We therefore ask  $\nabla f$  to be the best approximation of all directional derivatives, namely:

$$\nabla f_i := \arg \min_{X \in \mathbb{R}^2} \sum_{(ij) \in E} \|e_{ij}^\top X - (f_i - f_j)\|^2. \quad (4)$$

This least squares optimization problem amounts to pseudo-inverting a  $d_i \times 2$  matrix per vertex, where  $d_i$  is the degree of vertex  $i$ . This can be done exactly in pre-processing. As the gradient at vertex  $i$  only depends on its neighbors, it can be encoded as a complex *sparse* matrix whose non-zero coefficients per-line are equal to the degree of the vertex plus one.

**Discrete definition of a pusforward** By combining these tools, we obtain a simple discretization of the definition of the pushforward in Eq. (1), as a composition of functional operators, represented in the discrete setting as matrix multiplication:

$$D_X^M C_{NM} = C_{NM} D_{QX}^N, \quad \forall X \in \mathbb{C}^{|V_M|}. \quad (5)$$

Note that in this expression, consistently with Eq. (1), the pushforward  $Q$  maps vector fields in the opposite direction from the pullback  $C$ . In practice, we have found it simpler to work with an expression that assumes that the two operators map in the same direction, which leads to:

$$C_{MN} D_X^M = D_{QX}^N C_{MN}, \quad \forall X \in \mathbb{C}^{|V_M|}. \quad (6)$$

This expression can be obtained simply by pre- and post multiplying Eq. (5) by  $C_{MN}$ , and assuming an invertible mapping.

In practice, mappings are usually not conformal, so we cannot hope to satisfy Eq. (6) exactly. However, we can define the energy  $E_{cq}$  measuring how close  $Q : TM \rightarrow TN$  is to be the differential of the operator  $C_{MN} : L^2(M) \rightarrow L^2(N)$  and evaluating the constraint in the least squares sense:

$$E_{cq}(C, Q) = \sum_i \|CD_{X_i}^M - D_{QX_i}^N C\|_F^2, \quad (7)$$

where  $\{X_i\}$  is a family of vector fields in  $TM$ . Remark that Eq. (6) is linear with respect to  $X$ . Consequently, minimizing the energy of Eq. (7) will ensure that Eq. (6) is satisfied as well as possible on the subspace of  $TM$  generated by  $\{X_i\}$ . As above, we use the first eigenfunctions of the connection Laplacian operator for this family, and thus the energy in Eq. (7) ensures that Eq. (6) is satisfied as well as possible for *smooth tangent vector fields*.

Our overall strategy thus consists in recovering  $Q$  given an arbitrary functional map  $C$  by solving the problem:

$$\min_Q E_{cq}(C, Q). \quad (8)$$

Eq. (8) defines a simple least squares system which can be efficiently optimized by solving a linear system of equations. We describe how to do in it more details in Appendix F. The solution of this problem is the best approximation of the map differential by an *orientation-preserving* conformal pushforward. The energy  $E_{cq}$  is fundamental in our experiments as it allows us to extract orientation information from any given functional map.

### 4.3. A closed-form expression for $Q$

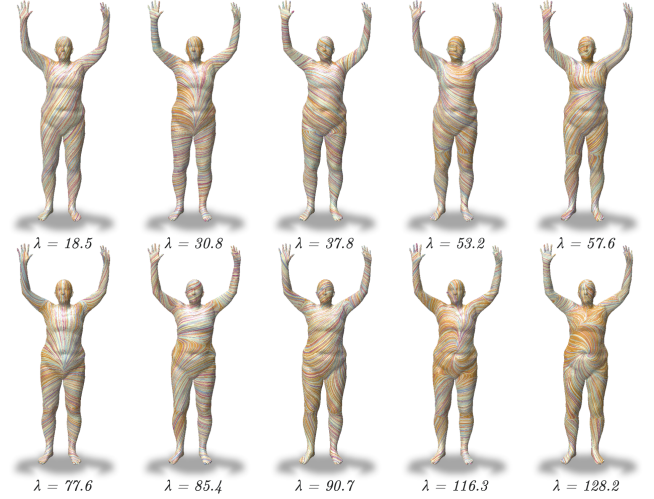
While Eq. (7) plays a fundamental role in our experiments, we remark that in the case of meshes with the same connectivity, it is possible to obtain an intuitive closed-form expression for  $Q$ . Recall that a conformal pushforward is given by 1) an assignment between points and 2) a similarity transformation between matching tangent planes. Therefore, it is expected that the discrete  $d\varphi$ , the pushforward between piecewise linear complex fields, is simply the composition of a matrix  $\Pi_{MN}$ , assigning vertices of  $M$  to those on  $N$ , and a multiplication by a complex field  $q : N \rightarrow \mathbb{C}$  performing the tangent plane deformation at each vertex. Using our discrete definition of the pushforward (Eq. (6)), we can recover this property when the two meshes have same the connectivity.

**Theorem 4.1** Given two meshes with same connectivity and given the permutation matrix  $\Pi$  describing the vertex-to-vertex correspondence, the solution of Eq. (8) has the form:

$$Q = D(q)\Pi,$$

where  $D(q)$  is a diagonal matrix with complex coefficients  $q_i$ . The complex numbers  $q_i$  are the best conformal alignment of the tangent planes.

The proof of Thm. 4.1 can be found in Appendix D.



**Figure 4:** LIC visualization of the first 10 eigenvectors of the connection Laplacian (from left to right, top to bottom). We see that a higher eigenvalue  $\lambda$  gives an eigen tangent vector field whose frequency is higher, resulting in a less smooth flow. Nevertheless, all these tangent vector fields represent the generators of the smoothest vector fields over the shape. It is in this basis that we will decompose tangent vector fields, as described in Section 4.5.

### 4.4. Constraints on complex functional maps

**Operator orthogonality** While the discrete version of Thm. 3.2 holds exactly in our discrete setting, it is not very informative, since the discrete notion of conformality induced by Thm. 4.1 is too rigid, since the only deformations of a mesh that preserve angles at every triangle exactly are isometries with possible global scaling.

However, orthonormality of  $Q$  is still a valuable constraint in the reduced basis and forces  $Q$  to be the approximation of a pushforward. Furthermore, the use of a reduced basis helps to avoid both the reliance on an exact mesh structure and the rigidity of exactly conformal maps. In our experiments this constraint proved to be very effective.

In practice, we therefore solve the following Procrustes problem:

$$\min_{Q^*Q=I} E_{cq}(C, Q). \quad (9)$$

This is a standard problem that can be solved exactly using a Singular Value Decomposition. For completeness we provide the details in Appendix F.

**Commutativity with the connection Laplacian** The isometric constraint of Thm. 3.3 is still valid in the discrete setting. Interestingly, the isometry condition by enforcing the commutativity with the Laplacian is identical to the standard functional map framework. In particular, as remarked in [OBCS\*12], it implies the more isometric a mapping is, the more diagonal our matrix  $Q$  will be, when expressed in the reduced basis.

**Theorem 4.2** The conformal pushforward  $Q$  commutes with the connection Laplacian:

$$L_N Q = Q L_M,$$



Random noise			
Method / level of noise	$s = 0$	$s = 0.2$	$s = 0.5$
Wang <i>et al.</i> [WLZT18]	<b>0.40</b>	2.2	5.5
Azencot <i>et al.</i> [ABCCO13]	6.2	4.6	3.2
Ours	<b>0.40</b>	<b>0.43</b>	<b>0.74</b>
Symmetric noise			
Method / level of noise	$a = 0.3$	$a = 0.5$	$a = 0.6$
Wang <i>et al.</i> [WLZT18]	0.75	1.1	1.3
Azencot <i>et al.</i> [ABCCO13]	4.6	1.8	4.2
Ours	<b>0.40</b>	<b>0.51</b>	<b>0.81</b>

**Table 1:** Average accuracy of the three vector field transfer algorithms on 20 random pairs of FAUST [BRLB14] for two types of noise, three noise levels. We use  $k = 50$  eigenvectors for both real and complex Laplacian operators. For noisy input functional maps, our method is always the most accurate. For completeness, we report more results for  $k = 30, 70, 150$  in Appendix G.

if and only if it represents an isometric map.

The proof is deferred to Appendix E.

#### 4.5. Discrete operators in a reduced basis

In order to improve the efficiency of our algorithms we will not consider all piecewise linear tangent vector fields but only those spanned by a small number  $k$  of smooth vector fields stored in a complex  $|V| \times k$ -matrix  $\Psi$ .

For a known deformation across compatible meshes, the closed-form expression of  $Q$  is exhibited in Thm. 4.1 and can easily be rewritten in a reduced basis:

$$Q_{NM} = (\Psi^M)^* A^M D(q) \Pi_{MN} \Psi^N. \quad (10)$$

Note that Eq. (10) is very similar to the expression of discrete functional map  $C$  introduced in [OBCS\*12]. For the function bases  $\Phi^M$  and  $\Phi^N$  a functional map reads:

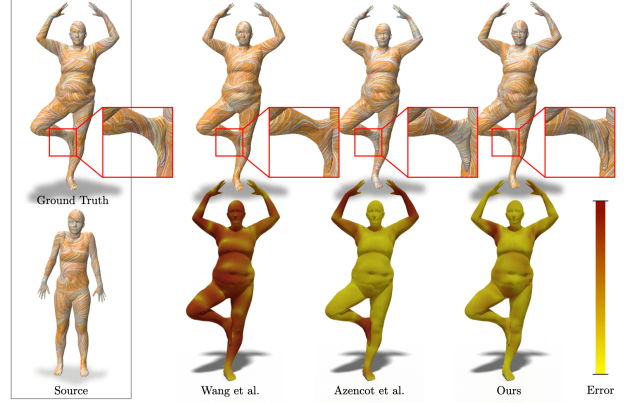
$$C_{NM} = (\Phi^M)^T A^M \Pi_{MN} \Phi^N. \quad (11)$$

All the constraints on  $Q$  in the “hat” basis can be simply rewritten by replacing each term by an operator projected in the reduced basis.

In theory any orthonormal basis could be considered. For the purpose of non-rigid 3D shape matching a basis smooth and stable under nearly-isometric deformations leads to better results. In our experiments, we use the  $k$  first eigenvectors of the connection Laplacian for complex functional maps, that we visualize via Line Integral Convolution in Figure 4. Indeed, as proved in Thm. 3.3 this operator is invariant under isometric deformations and moreover its discretization is easily implemented.

#### 4.6. Point-to-point map conversion

In the functional map pipeline a key step is the conversion from a functional map to a vertex-to-vertex map. As originally described in [OBCS\*12] and extended in [PRM\*21], one just needs to transfer Dirac functions on  $M$  using the adjoint of the functional



**Figure 5:** Comparison of three tangent vector field transfer methods. Top row: we display a LIC visualisation of the transferred vector field on the target shape. Bottom row: we show the transfer error, computed as the difference with the ground-truth transfer. For this transfer, we used a  $50 \times 50$  ground-truth functional map blurred with a random noise of magnitude 0.1. Quantitative results can be found in Table 1.

map, and compute the closest Dirac function on  $N$  using a nearest neighbor search algorithm. Namely, the operation performed is  $\Pi_{MN} = \text{NNsearch}(\Phi^N, \Phi^M C_{NM})$ , where  $\Phi^M$  and  $\Phi^N$  are the eigenfunctions from the standard Laplace-Beltrami operator on  $M$  and  $N$ .

In our case, extending this algorithm to “Dirac vector fields” is not straightforward as a single vector is not isotropic. Instead, we propose to use the divergence operator to convert the vector field basis to functions and then use the standard functional point-to-point conversion scheme via nearest neighbor search. When  $Q$  is expressed in a reduced basis, this simply amounts to computing:

$$\Pi_{MN} = \text{NNsearch}(\text{div}_N \Psi_N, \text{div}_M \Psi_M Q_{NM}). \quad (12)$$

Where we define the discrete divergence to be the adjoint of the gradient operator matrix defined in Eq. (4).

This solution is not fully satisfying as it relies on the commutativity of the pushforward with the divergence operator, and thus is geared towards near isometries. This approach, however, proves to be sufficient for our shape matching applications in Section 5. Furthermore, in our current pipeline,  $Q$  is evaluated alongside  $C$  and thus, if needed, the conversion can be done using the standard functional map. We leave finding a robust and general conversion scheme for complex functional maps as interesting future work.

## 5. Results

In this section, we present several applications of our complex functional maps. We start by demonstrating that robust vector field transfer can be achieved with complex functional maps without requiring additional information or computation. Then, we demonstrate that the orientation-aware nature of complex functional maps can be used to eliminate the symmetry ambiguity in

Method (-/+ICP)	[NO17]	[RPWO18]
-	0.24 / 0.21	0.21 / 0.17
+ Ours	0.15 / 0.13	0.12 / 0.10
Method (-/+ICP)	[WLZT18]	[WLZT18] + [RPWO18]
-	0.31 / 0.23	0.23 / 0.13
+ Ours	0.16 / 0.13	<b>0.11 / 0.083</b>

**Table 2:** Average geodesic error on 190 FAUST remeshed shape pairs. Our complex functional map step always improves the correspondence quality for all four algorithms, even those already incorporating the orientation-aware operators from [RPWO18]. Detailed graph geodesic error vs. percentage of correspondences can be found in Fig. 8 of the Appendix.

non-rigid near-isometric shape matching problems. Complex functional maps are easily added to standard functional map pipelines like map estimation from descriptors [OBBS\*12, NO17, RPWO18] or refinement methods like ZOOMOUT [MRR\*19] and its follow-ups [RMOW20, RMWO21]. An implementation of our method can be found at: <https://github.com/nicolasdonati/QMaps>.

### 5.1. Vector field transfer

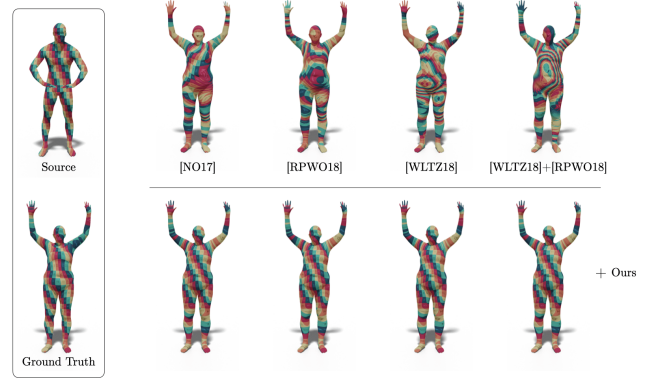
The first direct application of complex functional maps is tangent vector field transfer. We compare three methods for vector field transfer using as only input an approximate functional map  $C$ . We demonstrate below that using our approach is more accurate, compared to existing methods [WLZT18, ABCCO13] especially in the presence of noise.

**Vector field transfer with complex functional maps** Our representation allows for efficient and easy-to-use tangent vector field transfer. Indeed, given a functional map  $C_{MN}$ , we can easily recover a complex functional map  $Q$  by solving the Procrustes problem in Eq. (9). Then, transferring a vector field  $X$  defined on shape  $M$  to shape  $N$ , boils down to : 1) projecting  $X$  in the complex spectral basis  $X \approx \Psi^M x$  (see Section 4.1), 2) transferring the spectral coefficients using:  $y = Qx$  and 3) recovering the output as linear combination of the target basis:  $Y = \Psi^M y$ .

To assess the accuracy of our method, we compare it with two standard baselines: the transfer using Hodge decomposition from Wang *et al.* [WLZT18] and the transfer using the vector field operator proposed by Azencot *et al.* [ABCCO13].

**Hodge decomposition transfer** On a shape  $M$  with a sphere-topology, any tangent vector field  $X \in TM$  can be decomposed as the sum of a gradient and a rotated gradient. As remarked in [WLZT18], transferring vector fields with a conformal map can be done using only the functional map  $C$  by: 1) computing the Hodge decomposition, *i.e.* finding the functions  $f, g \in \mathbb{R}^{|V|}$  such that  $X = \nabla f + \iota \nabla g$ , 2) transferring the functions  $f, g$  using  $C$  and 3) computing the gradient and rotated gradient on  $N$ .

**Vector field operator** Azencot *et al.* [ABCCO13] use the representation of a tangent vector field  $X \in TM$  via the associated functional



**Figure 6:** We visualize the influence of complex functional map step on some descriptor-based functional map pipelines. The quality of the computed maps is displayed with texture transfer on a pair of FAUST re-meshed shapes (91 as source, 89 as target). We see that for every of the considered pipeline, our method helped with continuity and left-right symmetry errors. Quantitative results can be found in Table 2.

operator  $f \mapsto D_X(f)$ , defined in Eq. (2) above. The method boils down to transferring  $X$  by solving :

$$\min_Y \|D_Y C - C D_X\|_F^2.$$

This method is similar to ours as their energy is also inspired by Equation (1). However, there are two key differences. Firstly, our method estimates the transfer for all low-frequency tangent vector fields of the source eigenbasis simultaneously by estimating  $Q$ , whereas the approach of Azencot *et al.* is limited to one vector field transfer at a time. Secondly, the approach in [ABCCO13] is not limited to conformal deformations, making it more flexible but also more sensitive to noisy input functional maps.

**Results** Table 1 reports a quantitative comparison between the three methods described above on 20 random pairs taken from the original FAUST dataset [BRLB14]. These meshes are in one-to-one vertex correspondence, allowing us to compute the ground-truth pushforward. To perform the comparison, we generate a smooth vector fields on the source, transfer it and then compute the  $L^2$ -distance with the exact transfer normalized by the norm of the input vector field. We use  $k = 50$  eigenvectors for both the functional space and the vector field space. We assess the robustness of the three functional-based vector field transfer methods by adding two types of noise to the input ground-truth  $C_{gt}$  functional map:

- **Random noise:** the input ground-truth  $C_{gt}$  functional map is corrupted by adding a random matrix  $N$  whose entries are taken uniformly at random between  $-s$  and  $s$ ,  $s$  being a given threshold. We transfer tangent vector fields of the form  $X = \nabla f + \iota \nabla g$  by randomizing the spectral coordinates of  $f, g$  such that they decrease in intensity as frequency goes up. The results are reported in the first half of Table 1.
- **Symmetric noise:** The input ground-truth map  $C_{gt}$  is mixed with an orientation-reversing functional map  $C_{sym}$  by linearly

Method / stats	Avg.	Med.	Min.
ZO	0.520	0.523	0.226
ZO + Ours	<b>0.320</b>	<b>0.328</b>	<b>0.037</b>
ZO + bij	0.508	0.447	0.106
ZO + bij + Ours	<b>0.367</b>	<b>0.382</b>	<b>0.040</b>
ZO + bij + conf	0.47	0.45	<b>0.025</b>
ZO + bij + conf + Ours	<b>0.225</b>	<b>0.078</b>	0.029
ZO + bij + iso	0.450	0.433	<b>0.025</b>
ZO + bij + iso + Ours	<b>0.198</b>	<b>0.081</b>	0.029

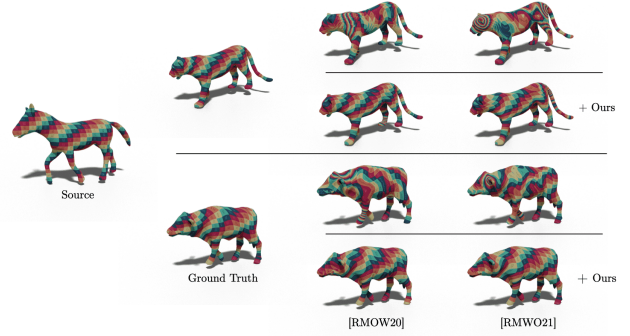
**Table 3:** Adding our complex functional map step in the pipeline of Ren *et al.* [RMWO21] always improves map accuracy. We report the average, median and minimal geodesic distance error on 50 shape pairs of the SMAL dataset. Detailed graph geodesic error vs. percentage of correspondences can be found in Fig. 9 of the Appendix.

interpolating between the two maps  $C = aC_{gt} + (1-a)C_{sym}$ . This kind of noise often arises when estimating maps from descriptors with the original pipeline introduced in [OBCS\*12]. We transfer a tangent vector field of the form  $X = \nabla f$ , with  $f$  the extrinsic coordinate of the left-right axis. This results in an antisymmetric vector field that will only be transferred correctly if the method is robust to noise. The results are reported in the second half of Table 1.

In presence of noise, our method always outperforms the baselines. This is due to the fact that we first compute the pushforward closest to  $C$  in the least square sense, making it robust to random noise. By construction, our pushforward is *orientation-preserving*, so it is resistant to symmetric noise and is able to recover a well-oriented transfer as shown in Fig. 1. In comparison, the other two methods directly rely on the functional map and fail if  $C$  does not exactly represent a pointwise map. Moreover, if  $C$  is a mix of a direct and orientation-reversing map, it is not easy to recover the underlying orientation-preserving map and thus to be robust to such noise.

In Figure 5 we provide qualitative illustrations of vector field transfer obtained using our method compared to baselines. Note that even a small amount of noise in the input functional map can compromise the quality of the transfer performed by [WLZT18]. Our method only exhibits minor errors, even though it is only designed to handle conformal deformation. In fact, the only visible mis-transfer happens at the shoulder joint where the deformation is far from conformal. The approach of Azencot *et al.* [ABCCO13] is clearly under-performing and always achieves the worst accuracy of our comparisons. However this method is more general and would give the best results in presence of perfect information *and in the full basis*, even with strongly non-isometric deformations.

In conclusion, complex functional maps can help to alleviate the errors in map orientation and allow to accurately transfer vector fields between near isometric shapes even if the deformation is not exactly conformal.



**Figure 7:** Texture transfer on SMAL re-meshed dataset [ZKJB17] illustrating maps obtained with two methods based on ZoomOut: bijective ZoomOut [RMOW20] (third column) and bijective ZoomOut with isometric energy [RMWO21] (last column). Adding our  $Q$ -step (bottom sub-rows) considerably improves the accuracy of the map. For quantitative results see Table 3.

## 5.2. Disambiguating symmetry in functional maps computation

In our next application, we show that complex functional maps can be used within the standard descriptor-based functional map pipeline to significantly improve robustness and accuracy. The key issue that we consider is that, as remarked in prior works [OMPG13, COC15] intrinsic descriptors [SOG09, ASC11] are often *symmetric*, which can lead to poor correspondences, where a point is arbitrarily matched to the correct target point or its symmetric counterpart [RMOW20]. As we demonstrate below, injecting our orientation-preserving complex maps into the pipeline can help to resolve this issue efficiently.

To achieve this, we propose to *project* a given functional map into the space of orientation-preserving maps by using  $Q$  as an intermediary. This projection is done in two steps. First, we approximate the associated map differential  $Q$  by solving the Procrustes problem in Eq. (9). Secondly, we extract from  $Q$  the underlying point-to-point mapping using the algorithm described in Section 4.6. Since, by construction,  $Q$  is orientation preserving, the projection removes the orientation reversing component of the input map, and thus the resulting point-to-point mapping should be orientation preserving.

In a third optional step, one can reconstruct a new functional map from the point-to-point map using Eq. (11). This allows us to improve the mapping using post-processing technique like the spectral ICP refinement introduced in [OBCS\*12].

We compare this approach to four alternative algorithms for computing functional maps solely from 50 WKS [ASC11] descriptors:

- The algorithm introduced by Nogneng *et al.* [NO17] where descriptors are converted into operators that must commute with the functional map.
- We add the orientation-promoting operators of [RPWO18] to the pipeline of Nogneng *et al.* [NO17].
- The method proposed in [WLZT18], in theory closest to ours, which proposes to also transfer differential 1-forms with functional maps. It differs from our method mainly because they

rely on  $\mathbb{R}^2$  – *linearity* whereas our maps are  $\mathbb{C}$ -linear, and thus orientation-aware. In [WLZT18] the authors transform descriptor functions into tangent vector fields by taking the gradient.

- A combination of [WLZT18] with the orientation-aware operators of [RPWO18].

To assess the ability of  $Q$  to recover orientation information, we compare maps obtained by direct conversion of the functional map computed using each of approaches above, with maps obtained by conversion from  $Q$ , that has first been estimated from  $C$ .

The results are computed on 20 shapes (which corresponds to 190 pairs) from the FAUST dataset [BRLB14], which were re-meshed in [RPWO18] in order to remove the bias of identical triangulations. We then compare the mean geodesic distance error obtained by these different methods – for each source point, we compute the geodesic distance on the target between the point mapped by the obtained maps and the point mapped by ground-truth. The results are reported in Table 2. Additionally, we display in Figure 6 a qualitative result where our additional step was able to disambiguate symmetry whereas the standard pipelines could not.

This experiment shows that our  $Q$ -map projection step significantly improves the accuracy of the correspondence *with all algorithms*, even in the presence of orientation-aware operators.

### 5.3. Orientation-preserving ZOOMOUT

Finally, our complex functional maps can also be used to improve more recent functional map algorithms based on spectral upsampling, inspired by ZOOMOUT [MRR\*19]. The ZOOMOUT algorithm starts with an initial pointwise map  $\Pi_0$  and alternates between two steps: estimating a new functional map  $C_n$  from  $\Pi_n$  and recovering the new pointwise map  $\Pi_{n+1}$  from  $C_n$ . In order to increase the map precision, the size of the spectral basis increases at each iteration.

Interestingly, we can modify the basic ZOOMOUT approach to incorporate complex functional maps and thus promote discovery of orientation preserving maps. Our new algorithm basically boils down to 3 steps instead of 2: 1) estimate the functional map  $C_n$  from  $\Pi_n$ . 2) [*Q-step*] Estimate the complex functional map  $Q_n$  from  $C_n$ , using Eq. (9). 3) Estimate the new pointwise map  $\Pi_{n+1}$  from  $Q_n$  (instead of  $C_n$  like in classic ZOOMOUT) using Eq. (12). The pseudo-code can be found in Algorithm 1 where the lines that describe our modification are highlighted in bold.

We remark that this “*Q-step*” can easily be added to other algorithms built on top of ZOOMOUT. MapTree [RMOW20], which uses a tree structure to explore the space of maps, modifies ZOOMOUT to promote bijectivity. This idea was later extended to other properties like conformality or isometry by Ren *et al.* [RMWO21]. Like ZOOMOUT, these algorithms are based on spectral upscaling and conversions between spectral mappings and vertex-to-vertex maps. We describe their modification in Appendix I.

We demonstrate the beneficial effect of this *Q-step* by refining *random* functional maps with four versions of ZOOMOUT with and without our modification. We upscale the maps from 4 eigenfunctions up to 50, with a step of 1, and 10 inner loops per step. We perform this experiment on 50 shape pairs of SMAL [ZKJB17] re-meshed. We report the obtained quantitative results in Table 3. For

---

#### Algorithm 1 Complex ZOOMOUT

---

- 1: **Input:** Manifold meshes  $M$  and  $N$
  - 2: Initial pointwise maps  $\Pi_{MN}$
  - 3: **Output:** Refined maps  $\Pi_{MN}^{ref}$
  - 4: **Parameters:** The number of refinement steps  $J$
  - 5: An array  $[k_j], j \in [1, J]$  with the (increasing) number of spectral coordinates to use at each refinement step
  - 6: **Preprocessing:** Compute the Laplace Beltrami eigenbases  $\Phi^M$  and  $\Phi^N$  (used for function in spectral basis)
  - 7: Compute the connection Laplacian complex eigenbases  $\Psi^M, \Psi^N$  (used for vector field spectral bases, see Section 4.5)
  - 8: Compute the differential operators  $D_{\Phi_i^M}$ , and  $D_{\Phi_i^N}$  for  $i \in [1, k_J]$  (used for estimating  $Q$  from  $C$ , see Section 4.2)
  - 9: Compute the reduced divergence operators  $\text{div}_M$  and  $\text{div}_N$  (used for conversion from  $Q$  to pointwise map, see Section 4.6)
  - 10: **for**  $k \in [k_1, \dots, k_J]$  **do**
  - 11:      $\Phi_M = \Phi_{[1,k]}^M, \Phi_N = \Phi_{[1,k]}^N$
  - 12:      $C_{NM} = \Phi_M^\dagger \Pi_{MN} \Phi_N$
  - 13:      $Q_{NM} = \arg \min_{Q \in O(k)} \sum_{i=1}^k \|C_{NM} D_{\Psi_i^N} - D_{Q\Psi_i^N} C_{NM}\|_F^2$
  - 14:      $\Pi_{MN} = \text{NNsearch}(\text{div}_N \Psi_N, \text{div}_M \Psi_M Q_{NM})$
  - 15: **end for**
- 

the baselines, we used the standard ZOOMOUT algorithm [MRR\*19], bijective ZOOMOUT [RMOW20], and the discrete optimization with first conformal (*conf*) and isometric energy (*iso*) [RMWO21]. For all of these baselines, we report both their overall geodesic error (mean, median and minimum error) and that of their modification with our method. We observe that our modification is always relevant in this case, resulting in a significant boost in overall accuracy on this dataset. More specifically, the discrete optimization approach [RMWO21] with complex functional maps performs really well (median error below 0.09) despite the fact it is randomly initialized, and does not use *any* descriptors.

In Figure 7 we provide qualitative results with two shape pairs with the same source, and report five maps per pair: the ground-truth, bijective ZOOMOUT [RMOW20], and bijective isometric ZOOMOUT [RMWO21], as well as their version with our modification. For both shape pairs and both methods, the original algorithms are affected by the left-right symmetry and converge to discontinuous maps.

In summary, in state-of-the-art refinement pipelines involving ZOOMOUT, our *Q-step* appears to promote orientation preservation and continuity, resulting in better convergence and more accurate results overall. This confirms that using complex functional maps in functional map pipelines is beneficial to estimate high quality correspondences from very low-frequency or extremely unreliable initialization.

## 6. Conclusion, Limitations & Future Work

In this paper, we have introduced a new functional operator resulting from a  $\mathbb{C}$ -linear relaxation of the space of pushforwards: the complex functional map. This operator allows to robustly transfer tangent vector fields between non-rigid surfaces in 3D. Further-

more, the most prominent property of this new tool is that it reflects the complex structure of the surfaces and is thus orientation-aware. In our experiments, we exploited the orientation-aware property of complex functional maps in several shape matching tasks. This contribution to the functional framework considerably increases the applicability of intrinsic shape matching methods, which are often hindered by the presence of orientation-reversing intrinsic symmetries. This is particularly relevant for functional map-based algorithms, since they rely on a linear relaxation of diffeomorphisms, which can linearly blend direct and orientation-reversing maps, potentially resulting in large discontinuities.

However our framework is naturally restricted to differentials of *conformal* orientation-preserving mappings. This constraint is only enforced in a least squares sense in our approach and thus helps to *promote* conformality which can be beneficial for shape pairs satisfying this assumption. Besides, like most algorithms derived from the functional framework, our construction is dependent on the choice of reduced basis. This can limit its applicability in more general non-isometric shape matching. As follow-up work, we would like to study other representations for orientation-preserving pushforwards and alleviate the dependency on the reduced basis.

In the future it would also be interesting to investigate the utility of complex functional maps in other applications that involve vector field transfer, including deformation or pose transfer, or synchronized convolution in the context of geometric deep learning. Indeed, restricting the search to well-oriented maps without additional supervision could lead to new efficient methods in unsupervised 3D deep learning.

**Acknowledgements** The authors would like to thank the anonymous reviewers for their helpful feedback and suggestions. Parts of this work were supported by the ERC Starting Grants No. 758800 (EXPROTEA) and No. 802554 (SPECGEO), and the ANR AI Chair AIGRETTE.

## References

- [ABCCO13] AZENCOT O., BEN-CHEN M., CHAZAL F., OVSJANIKOV M.: An operator approach to tangent vector field processing. In *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing* (2013), Eurographics Association, pp. 73–82. 1, 2, 5, 6, 8, 9, 10, 16
- [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). URL: <https://doi.org/10.1145/3072959.3073696>, doi:10.1145/3072959.3073696. 2
- [AK13] AFLALO Y., KIMMEL R.: Spectral multidimensional scaling. *PNAS* 110, 45 (2013), 18052–18057. 2
- [AOCBC15] AZENCOT O., OVSJANIKOV M., CHAZAL F., BEN-CHEN M.: Discrete derivatives of vector fields on surfaces—an operator approach. *ACM Transactions on Graphics (TOG)* 34, 3 (2015), 1–13. 2
- [ASC11] AUBRY M., SCHLICKWEI 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. 2, 10
- [AVBC16] AZENCOT O., VANTZOS O., BEN-CHEN M.: Advection-based function matching on surfaces. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 55–64. 2
- [BCA19] BEN-CHEN M., AZENCOT O.: Operator-based representations of discrete tangent vector fields. In *Handbook of Numerical Analysis*, vol. 20. Elsevier, 2019, pp. 117–147. 2
- [BCBB16] BIASOTTI S., CERRI A., BRONSTEIN A., BRONSTEIN M.: Recent trends, applications, and perspectives in 3d shape similarity assessment. *Computer Graphics Forum* 35, 6 (2016), 87–119. 2
- [BDK17] BURGHARD O., DIECKMANN A., KLEIN R.: Embedding shapes with Green’s functions for global shape matching. *Computers & Graphics* 68 (2017), 1–10. 2
- [BGV03] BERLINE N., GETZLER E., VERGNE M.: *Heat kernels and Dirac operators*. Springer Science & Business Media, 2003. 14
- [BKP\*10] BOTSCH M., KOBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon mesh processing*. CRC press, 2010. 5
- [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. 8, 9, 11, 16
- [Car92] CARMO M. P. D.: *Riemannian geometry*. Birkhäuser, 1992. 5, 14
- [CO19] CORMAN E., OVSJANIKOV M.: Functional characterization of deformation fields. *ACM Transactions on Graphics (TOG)* 38, 1 (2019), 1–19. 2
- [COC15] CORMAN E., OVSJANIKOV M., CHAMOLLE A.: Continuous matching via vector field flow. *Computer Graphics Forum* 34, 5 (2015), 129–139. 2, 10
- [CSBC\*17] CORMAN E., SOLOMON J., BEN-CHEN M., GUIBAS L., OVSJANIKOV M.: Functional characterization of intrinsic and extrinsic geometry. *ACM Transactions on Graphics (TOG)* 36, 2 (2017), 1–17. 1, 2
- [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. 2
- [ELC20] EISENBERGER M., LAHNER Z., CREMERS D.: Smooth shells: Multi-scale shape registration with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 12265–12274. 3
- [ERGB16] EYNARD D., RODOLÀ E., GLASHOFF K., BRONSTEIN M. M.: Coupled functional maps. In *2016 Fourth International Conference on 3D Vision (3DV)* (Oct 2016), pp. 399–407. 2
- [GGLZ10] GU X. D., GUO R., LUO F., ZENG W.: Discrete laplace-beltrami operator determines discrete riemannian metric. *arXiv preprint arXiv:1010.4070* (2010). 15
- [GP10] GUILLEMIN V., POLLACK A.: *Differential topology*, vol. 370. American Mathematical Soc., 2010. 3
- [GR20] GINZBURG D., RAVIV D.: Cyclic functional mapping: Self-supervised correspondence between non-isometric deformable shapes. In *European Conference on Computer Vision* (2020), Springer, pp. 36–52. 1
- [HJ12] HORN R. A., JOHNSON C. R.: *Matrix analysis*. Cambridge university press, 2012. 6
- [HLR\*19] HALIMI O., LITANY O., RODOLA E., BRONSTEIN A. M., KIMMEL R.: Unsupervised learning of dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019), pp. 4370–4379. 1
- [HO17] HUANG R., OVSJANIKOV M.: Adjoint map representation for shape analysis and matching. *Computer Graphics Forum* 36, 5 (2017), 151–163. 2
- [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
- [KBB\*13] KOVNATSKY A., BRONSTEIN M., BRONSTEIN A., GLASHOFF K., KIMMEL R.: Coupled quasi-harmonic bases. *Computer Graphics Forum* 32, 2pt4 (2013), 439–448. 2

- [KCPS13] KNÖPPEL F., CRANE K., PINKALL U., SCHRÖDER P.: Globally optimal direction fields. *ACM Transactions on Graphics (ToG)* 32, 4 (2013), 1–10. [6](#)
- [KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. In *ACM Transactions on Graphics (TOG)* (2011), vol. 30, ACM, p. 79. [3](#), [16](#)
- [KO18] KLEIMAN Y., OVSJANIKOV M.: Robust structure-based shape correspondence. In *Computer Graphics Forum* (2018), Wiley Online Library. [3](#)
- [KSS06] KHAREVYCH L., SPRINGBORN B., SCHRÖDER P.: Discrete conformal mappings via circle patterns. *ACM Transactions on Graphics (TOG)* 25, 2 (2006), 412–438. [3](#)
- [Lee13] LEE J. M.: Smooth manifolds. In *Introduction to Smooth Manifolds*. Springer, 2013, pp. 1–31. [3](#), [4](#), [13](#)
- [LF09] LIPMAN Y., FUNKHOUSER T.: Möbius voting for surface correspondence. *ACM Transactions on Graphics (TOG)* 28, 3 (2009), 1–12. [3](#)
- [LRR\*17] LITANY O., REMEZ T., RODOLÀ E., BRONSTEIN A., BRONSTEIN M.: Deep functional maps: Structured prediction for dense shape correspondence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), IEEE, pp. 5659–5667. [1](#)
- [MMRC20] MARIN R., MELZI S., RODOLÀ E., CASTELLANI U.: Farm: Functional automatic registration method for 3d human bodies. *Computer Graphics Forum* 39, 1 (2020), 160–173. [1](#), [3](#)
- [Mor01] MORITA S.: *Geometry of Differential Forms*. American Mathematical Society, 2001. [6](#)
- [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. [2](#), [9](#), [11](#)
- [MTAD08] MULLEN P., TONG Y., ALLIEZ P., DESBRUN M.: Spectral conformal parameterization. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 1487–1494. [3](#)
- [NMR\*18] NOGNGEN 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. [2](#)
- [NO17] NOGNGEN D., OVSJANIKOV M.: Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum* 36, 2 (2017), 259–267. [1](#), [2](#), [3](#), [9](#), [10](#)
- [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. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [9](#), [10](#)
- [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), SIGGRAPH '17, pp. 5:1–5:62. [2](#)
- [OMPG13] OVSJANIKOV M., MÉRIGOT Q., PĂTRĂUCEAN V., GUIBAS L.: Shape matching via quotient spaces. *Computer Graphics Forum* 32, 5 (2013), 1–11. [3](#), [10](#)
- [PO18] POULENARD A., OVSJANIKOV M.: Multi-directional geodesic neural networks via equivariant convolution. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–14. [2](#)
- [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 *CVPR* (2021). [8](#)
- [RCB\*17] RODOLÀ E., COSMO L., BRONSTEIN M., TORSSELLO A., CREMERS D.: Partial functional correspondence. *Computer Graphics Forum* 36, 1 (2017), 222–236. [2](#)
- [Rem91] REMMERT R.: *Theory of complex functions*, vol. 122. Springer Science & Business Media, 1991. [4](#)
- [RL03] RAY N., LÉVY B.: Hierarchical least squares conformal map. In *11th Pacific Conference on Computer Graphics and Applications, 2003. Proceedings.* (2003), IEEE, pp. 263–270. [3](#)
- [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). [1](#), [3](#), [9](#), [10](#), [11](#), [16](#)
- [RMWO21] REN J., MELZI S., WONKA P., OVSJANIKOV M.: Discrete optimization for shape matching. In *SGP* (2021). [9](#), [10](#), [11](#), [16](#)
- [ROA\*13] RUSTAMOV R. M., OVSJANIKOV M., AZENCOT O., BEN-CHEN M., CHAZAL F., GUIBAS L.: Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 72. [2](#), [5](#)
- [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). [1](#), [2](#), [3](#), [9](#), [10](#), [11](#)
- [RS80] REED M., SIMON B.: *Methods of Modern Mathematical Physics: Functional Analysis; Rev. ed.* Academic press, 1980. [5](#)
- [Sah20] SAHILIOĞLU Y.: Recent advances in shape correspondence. *The Visual Computer* 36, 8 (2020), 1705–1721. [1](#), [2](#)
- [SC17] SAWHNEY R., CRANE K.: Boundary first flattening. *ACM Transactions on Graphics (ToG)* 37, 1 (2017), 1–14. [3](#)
- [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. [2](#), [10](#), [14](#)
- [SSC19] SHARP N., SOLIMAN Y., CRANE K.: The vector heat method. *ACM Trans. Graph.* 38, 3 (2019). [5](#), [6](#), [14](#)
- [SSP08] SPRINGBORN B., SCHRÖDER P., PINKALL U.: Conformal equivalence of triangle meshes. In *ACM SIGGRAPH 2008 papers*. 2008, pp. 1–11. [3](#)
- [TSDS10] TOMBARI F., SALTI S., DI STEFANO L.: Unique signatures of histograms for local surface description. In *Proc. ECCV* (2010), Springer, pp. 356–369. [2](#)
- [VKZHC011] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707. [1](#), [2](#)
- [WLZT18] WANG Y., LIU B., ZHOU K., TONG Y.: Vector field map representation for near conformal surface correspondence. *Computer Graphics Forum* 37, 6 (2018), 72–83. [3](#), [8](#), [9](#), [10](#), [11](#), [16](#)
- [ZKJB17] ZUFFI S., KANAZAWA A., JACOBS D., BLACK M. J.: 3D menagerie: Modeling the 3D shape and pose of animals. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017* (Piscataway, NJ, USA, July 2017), IEEE, pp. 5524–5532. [10](#), [11](#)

## Appendix A: Proof of Theorem 3.1

*Necessary condition:* Let  $Q$  be the differential of an orientation-preserving conformal diffeomorphism  $\varphi$ . Then by Lemma 3.1  $Q$  must be  $\mathbb{C}$  linear and by virtue of being a differential,  $Q = d\varphi$  must satisfy Eq. (1), [Lee13].

*Sufficient condition:* Let  $Q$  be a  $\mathbb{C}$  linear operator and suppose that there exists a diffeomorphism  $\varphi : M \rightarrow N$  such that  $Q$  satisfies Eq. (1). Since the pushforward  $d\varphi : TM \rightarrow TN$  is the unique operator satisfying Eq. (1) (see [Lee13], Chapter 3), we must have  $Q = d\varphi$ . Now, since  $Q$  is a  $\mathbb{C}$ -linear pushforward, it preserves both the angles between the vectors and the orientation of the tangent bundle. Therefore, the map  $\varphi$  must then be an orientation-preserving conformal map.

**Algorithm 2** Complex Bijective ZOOMOUT

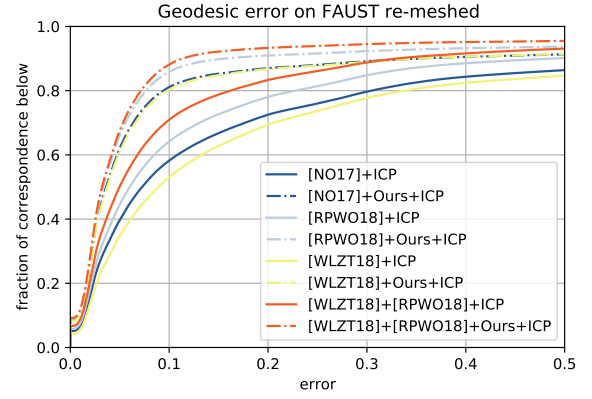
- 1: **Input:** Manifold meshes  $M$  and  $N$
- 2: Initial pointwise maps  $\Pi_{MN}$  and  $\Pi_{NM}$
- 3: **Output:** Refined maps  $\Pi_{MN}^{ref}$  and  $\Pi_{NM}^{ref}$
- 4: **Parameters:** The number of refinement steps  $J$
- 5: An array  $[k_j], j \in [1, J]$  with the (increasing) number of spectral coordinates to use at each refinement step
- 6: **Preprocessing:** Compute the Laplace Beltrami eigenbases  $\Phi^M$  and  $\Phi^N$  (used for function in spectral basis)
- 7: Compute the connection Laplacian complex eigenbases  $\Psi^M$  and  $\Psi^N$  (used for vector fields in spectral basis, see Section 4.5)
- 8: Compute the differential operators  $D_{\Phi_i^M}$  and  $D_{\Phi_i^N}$  for  $i \in [1, k_j]$  (used for estimating  $Q$  from  $C$ , see Section 4.2)
- 9: Compute the reduced divergence Operators  $\text{div}_M$  and  $\text{div}_N$  (used for conversion from  $Q$  to pointwise map, see Section 4.6)
- 10: **for**  $k \in [k_1, \dots, k_J]$  **do**
- 11:      $\Phi_M = \Phi_{[1,k]}^M, \Phi_N = \Phi_{[1,k]}^N$
- 12:      $C_{MN} = \Phi_N^\dagger \Pi_{NM} \Phi_M, C_{NM} = \Phi_M^\dagger \Pi_{MN} \Phi_N$
- 13:      $Q_{MN} = \arg \min_{Q \in \mathcal{O}(k)} \sum_{i=1}^k \|C_{MN} D_{\Psi_i^M} - D_{Q\Psi_i^M} C_{MN}\|_F^2$  -
- 14:      $Q_{NM} = \arg \min_{Q \in \mathcal{O}(k)} \sum_{i=1}^k \|C_{NM} D_{\Psi_i^N} - D_{Q\Psi_i^N} C_{NM}\|_F^2$  -
- 15:      $\Pi_{MN} = \text{NNsearch}(\text{div}_N \Psi_N, \text{div}_M \Psi_M Q_{MN})$
- 16:      $\Pi_{NM} = \text{NNsearch}(\text{div}_M \Psi_M, \text{div}_N \Psi_N Q_{NM})$
- 17:      $C_{MN} = \begin{pmatrix} \Phi_N \\ \Pi_{MN} \Phi_M \end{pmatrix}^\dagger \begin{pmatrix} \Pi_{NM} \Phi_M \\ \Phi_M \end{pmatrix}$
- 18:      $C_{NM} = \begin{pmatrix} \Phi_M \\ \Pi_{NM} \Phi_M \end{pmatrix}^\dagger \begin{pmatrix} \Pi_{MN} \Phi_N \\ \Phi_N \end{pmatrix}$
- 19:      $\Pi_{MN} = \text{NNsearch}((\Phi_N C_{NM} \ \Phi_N C_{MN}), (\Phi_M \ \Phi_M))$
- 20:      $\Pi_{NM} = \text{NNsearch}((\Phi_M C_{MN} \ \Phi_M C_{NM}), (\Phi_N \ \Phi_N))$
- 21: **end for**

**Appendix B:** Proof of Theorem 3.2

Let  $\varphi : M \rightarrow N$  be a conformal diffeomorphism and  $Q = d\varphi : TM \rightarrow TN$  its corresponding pushforward. By definition of conformality there exists a log-scale factor  $u : M \rightarrow \mathbb{R}$  relating the metric tensors of both surfaces  $\varphi^* \mathbf{g}^N = e^{2u} \mathbf{g}^M$  and their volume form  $\varphi^* d\mu_N = e^{2u} d\mu_M$ . Thus, the  $L^2$  scalar product between vector fields  $X, Y \in TM$  is preserved by the pushforward:

$$\begin{aligned} \int_N \mathbf{g}_p^N(Q(X), Q(Y)) d\mu_N(p) &= \int_N \mathbf{g}_p^N(d\varphi(X), d\varphi(Y)) d\mu_N(p) \\ &= \int_M \mathbf{g}_{\varphi(q)}^N(d\varphi(X), d\varphi(Y)) e^{-2u} d\mu_M(q) \\ &= \int_M (\varphi^* \mathbf{g}^N)_q(X, Y) e^{-2u} d\mu_M(q) \\ &= \int_M \mathbf{g}_q^M(X, Y) d\mu_M(q) \end{aligned}$$

This this holds for arbitrary  $X, Y$ , we obtain that  $Q^* Q = I$ .



**Figure 8:** Geodesic error of different methods with and without our  $Q$ -step, then refined with ICP, on 190 shape pairs of FAUST re-meshed.

**Appendix C:** Proof of Theorem 3.3

Let us assume that  $L_M = Q^{-1} \circ L_N \circ Q$ . As shown in [BGV03, SSC19], the diffusion kernel of the connection Laplacian is, at first order, the parallel transport along a geodesic and its magnitude is identical to the decay of the scalar heat kernel. Since the pushforward preserves the connection Laplacian, it also preserves the scalar heat kernel, therefore it must be an isometry [SOG09].

As proved in [Car92] (p.181) the pushforward of the Levi-Civita connection by a conformal mapping is itself the Levi-Civita connection if and only if the map is an isometry. Therefore, if  $\varphi$  is an isometry, the pushforward of the connection Laplacian  $Q^{-1} \circ L_N \circ Q$  is equal to the connection Laplacian on  $M$ .

**Appendix D:** Proof of Theorem 4.1

Let  $\varphi : M \rightarrow N$  be the permutation associated to the matrix  $\Pi$ .

The energy of Eq. (8) for vertex-based vector field must be evaluated for  $X$  a basis of complex field and  $f$  a basis of functions. We choose the ‘‘hat’’ basis for the fields  $X^i \in \mathbb{C}^{|V_M|}$  where  $X_j^i = z \delta_{ij}, z = 1$  or  $\iota$  and  $f^i \in \mathbb{R}^{|V_M|}$  where  $f_j^i = \delta_{ij}$ . The symbol  $\delta$  denotes the Kronecker delta. Explicitly writing the coefficients of the matrices, boils down to:

$$(CD_{X^i}^M)_{kj} = \begin{cases} \langle z, (\nabla f^j)_i \rangle, & k = \varphi(i) \\ 0, & k \neq \varphi(i) \end{cases}$$

$$(D_{QX^i}^N C)_{kj} = \begin{cases} \langle Q(X^i)_k, (\nabla f^{\varphi(j)})_k \rangle, & (k\varphi(j)) \in E_N \text{ or } k = \varphi(j) \\ 0, & (k\varphi(j)) \notin E_N \text{ and } k \neq \varphi(j) \end{cases}$$

An immediate conclusion is that  $Q$  must be zero everywhere except at  $Q_{\varphi(i),i}$  so there exists vector  $q \in \mathbb{C}^{|V_M|}$  such that:

$$Q = D(q)\Pi.$$

Now we can go back the least-squares problem and find each coefficient of  $q$  individually. Using the fact that  $z$  form a basis of

**Algorithm 3** Complex Discrete Optimisation

- 
- 1: **Input:** Manifold meshes  $M$  and  $N$
  - 2: Initial pointwise maps  $\Pi_{MN}$
  - 3: **Output:** Refined maps  $\Pi_{MN}^{ref}$
  - 4: **Parameters:** The number of refinement steps  $J$
  - 5: An array  $[k_j], j \in [1, J]$  with the (increasing) number of spectral coordinates to use at each refinement step
  - 6: **Preprocessing:** Compute the Laplace Beltrami eigenbases  $\Phi^M$  and  $\Phi^N$  (used for function in spectral basis)
  - 7: Compute the connection Laplacian complex eigenbases  $\Psi^M$  and  $\Psi^N$  (used for vector fields in spectral basis, see Section 4.5)
  - 8: Compute the differential operators  $D_{\Phi_i^M}$  and  $D_{\Phi_i^N}$  for  $i \in [1, k_j]$  (used for estimating  $Q$  from  $C$ , see Section 4.2)
  - 9: Compute the reduced divergence Operators  $\text{div}_M$  and  $\text{div}_N$  (used for conversion from  $Q$  to pointwise map, see Section 4.6)
  - 10: **for**  $k \in [k_1, \dots, k_J]$  **do**
  - 11:      $\Phi_M = \Phi_{[1,k]}^M, \Phi_N = \Phi_{[1,k]}^N$
  - 12:      $Q_{NM} = \arg \min_{Q \in \mathcal{O}(k)} \sum_{i=1}^k \|C_{NM} D_{\Psi_i^N} - D_{Q \Psi_i^N} C_{NM}\|_F^2$
  - 13:      $\Pi_{MN} = \text{NNsearch}(\text{div}_N \Psi_N, \text{div}_M \Psi_M Q_{NM})$
  - 14:      $C_{NM} = \Phi_M^\dagger \Pi_{MN} \Phi_N$
  - 15:      $\Pi_{MN} = \text{NNsearch}(\text{div}_N \Psi_N, \text{div}_M \Psi_M Q_{NM})$
  - 16: **end for**
- 

$\mathbb{C}$ , at a vertex  $i \in M$  the best conformal deformation of the tangent plane  $q_{\varphi(i)}$  is solution of:

$$q_{\varphi(i)} = \arg \min_{x \in \mathbb{C}} \sum_j \left| \left\langle x, (\nabla f^{\varphi(j)})_{\varphi(i)} \right\rangle - \left\langle 1, (\nabla f^j)_i \right\rangle \right|^2.$$

**Appendix E:** Proof of Theorem 4.2

- Suppose that  $Q$  is an isometric pushforward then the commutativity with the Laplacian immediately holds as the connection Laplacian, like the usual cotangent Laplacian, is preserved by isometric changes.
- Assume that  $Q$  commutes with the Laplacian then thanks to Thm. 4.1, the cotan-weights are preserved under the mapping. Therefore, the deformation is an isometry [GGLZ10].

**Appendix F:** Solving the least-squares problem to estimate  $Q$ 

Here we propose to complete Section 4.2 by giving one potential way to solve Eq. (8) and Eq. (9) explicitly. To that end we first proceed to rewrite Eq. (6), by switching from functional to vector field operator. More precisely, for a shape  $M$ , and  $f \in L^2(M)$ ,  $X \in TM$ , we define the linear operator  $D_f \in TM \rightarrow L^2(M)$ :

$$D_f(X)_p = \langle X, \nabla f \rangle_{T_p M}$$

With this operator we adopt the dual point of view from  $D_X$ . Indeed  $\forall f \in L^2(M)$ ,  $X \in TM$ ,  $D_f(X) = D_X(f)$ . Consequently, the discretization is almost identical, except that since this operator takes vector fields as input, we choose to encode it as a complex operator  $\mathbf{D}_f \in \mathbb{C}^{|V| \times |V|}$ . Namely,  $\mathbf{D}_f$  is a diagonal

matrix such that  $(\mathbf{D}_f)_{ii} = \overline{\nabla f_i}$ . One can then retrieve  $D_f(X)$  by taking the real part  $\text{Re}(\mathbf{D}_f X)$ . It can also be noted that  $\text{Im}(\mathbf{D}_f X) = \text{Re}(\mathbf{D}_f \cdot iX) = D_f \cdot iX$ , so that the complex matrix  $\mathbf{D}_f$  also stores information for rotated gradients.

Let  $M$  and  $N$  be two manifolds. Switching from  $D_X$  to  $D_f$  in Eq. (6), we get Eq. (13):

$$C_{MN} D_f^M = D_{C_{MN} f}^N Q_{MN}, \quad \forall f \in \mathbb{R}^{|V_M|}. \quad (13)$$

We then discuss how to minimize, for a fixed input functional map  $C$  and a family of smooth functions  $f_i$  the following energy. Similarly to Eq. (7),  $f_i$  is chosen to be the truncated eigenbasis of the Laplace-Beltrami operator.

$$E_q(Q) = \sum_i \|C D_{f_i}^M - D_{C f_i}^N Q\|_F^2, \quad (14)$$

**Regular problem** In this first paragraph we ignore the constraint  $QQ^* = I$ , which corresponds to Eq. (8).

Consider an input functional map  $C$ , and reference functions  $f_i, i \in [1, k_f]$ . The minimum to Eq. (14) can be written explicitly: by concatenating  $C D_{f_i}, i \in [1, k_f]$  in a big matrix  $A \in \mathbb{C}^{(|V_N| \times k_f) \times |V_M|}$ , and  $D_{C f_i}, i \in [1, k_f]$  in another matrix  $B \in \mathbb{C}^{(|V_N| \times k_f) \times |V_N|}$ , we can re-write Eq. (13) as  $Q = B^\dagger A$ , where  $B^\dagger$  is the Moore pseudo-inverse of  $B$ . In practice, we express these operators in a reduced spectral basis (See Section 4.5) before computing the pseudo-inverse, to improve computation time. Indeed in the reduced basis, operators  $A$  and  $B$  are respectively of size  $k_N k_f \times k_M$  and  $k_N k_f \times k_N$  if we choose to truncate the tangent vector field eigenbasis at  $k_M$  on  $M$  and at  $k_N$  on  $N$ . This results in a  $k_N \times k_M$  reduced complex operator for  $Q$ .

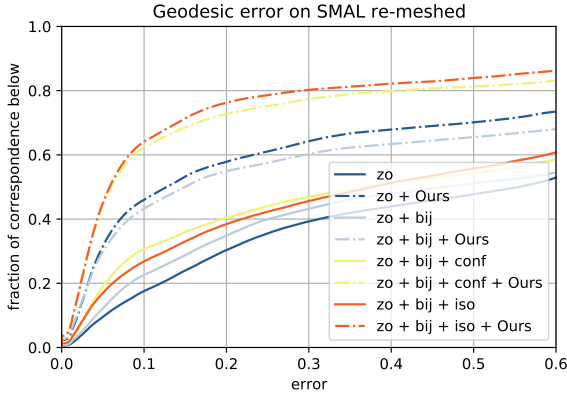
**Procrustes problem** Considering the same setting and keeping the last notations as in last paragraph, minimizing Eq. (14) with the constraint  $QQ^* = I$ , which corresponds to Eq. (9), is a Procrustes problem. As such, it boils down to a Singular Value Decomposition: writing  $\Omega = B^* A$ , we use its SVD  $\Omega = U \Sigma V$  to retrieve the orthogonal matrix  $Q = UV$  minimizing Eq. (9). As stated in the previous paragraph, one benefits from writing these equations in the reduced basis.

**Appendix G:** Vector Field transfer with more spectral values

In this appendix, we complete the vector field transfer experiment of Section 5.1 by showing the evolution of vector field transfer accuracy with the number of spectral coordinates involved in both the Laplace-Beltrami and the connection Laplacian operator (the latter is only used in Azencot *et al.* and our method). Table 4 reports this comparison in detail. Let us recall that only a number below 1 shows a result with reasonable accuracy, above this threshold the error is higher than the norm of the input vector field.

We observe that the transfer from Wang *et al.* is always very sensitive to noise, leading to high inaccuracies when noise corrupts the high frequencies. Notice also that the approach of Azencot *et*





**Figure 9:** Geodesic error of different versions of ZOOMOUT with and without our  $Q$ -step, on 50 shapes pairs of SMAL re-meshed.

$al$ . only starts to give reasonable transfer with high spectral values, namely  $k = 150$ . In comparison, our method almost doesn't suffer from noise, whether it is random or symmetric, and is accurate even for low spectral values.

Our method is the only one to recover from strong symmetric noise ( $a \geq 0.5$ ). Indeed the vector field we transfer for this second type of noise is antisymmetric as described in Section 5.1, but the input (blurred) functional map projects antisymmetric functions to 0. This results in baselines transferring the input vector field to a vector field close to 0 on the target shape, and thus errors above or close to 1.

#### Appendix H: Geodesic distance curves for Table 2 and 3

In Figure 8 and 9 we respectively display the results of Table 2 and 3 with more precision using the Princeton graphs first introduced in [KLF11]. With these curves it is easier to assert the finer quality of the correspondence computed with the help of complex functional maps.

#### Appendix I: Complex bijective ZOOMOUT algorithms

In this section we describe precisely how to implement the different versions of ZOOMOUT to which we add our  $Q$ -step. These algorithms are used to generate the correspondence whose quality are reported in Figure 7 and Table 3.

**Complex bijective ZoomOut** We first display the bijective ZOOMOUT algorithm used in [RMOW20], modified to include our  $Q$ -step. This results in Algorithm 2, where we bold the line numbers where change occurs. The idea behind bijective ZOOMOUT is to optimize for maps in both directions (from  $M$  to  $N$  and from  $N$  to  $M$ ). The energy to minimize is the so-called bijective energy  $E_{bij} = \|C_{MN}C_{NM} - I\|_F^2 + \|C_{NM}C_{MN} - I\|_F^2$ , and [RMOW20] proposes to optimize it with the following steps: Remark that  $C_{NM} = \Phi_M^\dagger \Pi_{MN} \Phi_N$ , we have  $\|\Phi_M\|_F^2 \|C_{NM}C_{MN} - I\|_F^2 = \|\Pi_{MN} \Phi_N C_{MN} - \Phi_M\|_F^2$ . Thus, knowing  $\Pi_{MN}$  and  $\Pi_{NM}$ , one

Random noise			
Method / level of noise	$s = 0$	$s = 0.2$	$s = 0.5$
$k = 30$			
Wang <i>et al.</i> [WLZT18]	<b>0.42</b>	5.3	14
Azencot <i>et al.</i> [ABCCO13]	11	12	12
Ours	0.54	<b>0.57</b>	<b>0.80</b>
$k = 70$			
Wang <i>et al.</i> [WLZT18]	<b>0.41</b>	14	33
Azencot <i>et al.</i> [ABCCO13]	2.6	2.4	2.4
Ours	0.46	<b>0.48</b>	<b>0.77</b>
$k = 150$			
Wang <i>et al.</i> [WLZT18]	<b>0.37</b>	27	68
Azencot <i>et al.</i> [ABCCO13]	0.68	0.81	1.0
Ours	0.44	<b>0.47</b>	<b>0.81</b>
Symmetric noise			
Method / level of noise	$a = 0.3$	$a = 0.5$	$a = 0.6$
$k = 30$			
Wang <i>et al.</i> [WLZT18]	0.87	1.13	1.29
Azencot <i>et al.</i> [ABCCO13]	10.19	6.84	10.59
Ours	<b>0.54</b>	<b>0.62</b>	<b>0.76</b>
$k = 70$			
Wang <i>et al.</i> [WLZT18]	0.81	1.09	1.26
Azencot <i>et al.</i> [ABCCO13]	2.93	1.99	2.55
Ours	<b>0.39</b>	<b>0.47</b>	<b>0.81</b>
$k = 150$			
Wang <i>et al.</i> [WLZT18]	0.77	1.05	1.22
Azencot <i>et al.</i> [ABCCO13]	0.69	1.04	1.43
Ours	<b>0.37</b>	<b>0.41</b>	<b>0.56</b>

**Table 4:** Average accuracy of the three vector field transfer algorithms of Section 5.1 on 20 random pairs of FAUST [BRLB14] for two types of noise, and three noise levels. We use  $k = 30, 70, 150$  eigenvectors for both real and complex Laplacian operators.

can optimize for  $C_{MN}$  in a bijective fashion with:

$$C_{MN} = \arg \min_C \|\Phi_N C - \Pi_{NM} \Phi_M\|_F^2 + \|\Pi_{MN} \Phi_N C - \Phi_M\|_F^2$$

And the same operation can be performed for  $C_{NM}$ .

Afterwards, one can perform a symmetric trick to get  $\Pi_{MN}$  knowing  $C_{MN}$  and  $C_{NM}$ , by additionally noting that in a bijective orthogonal setting,  $C^{-1} = C^T$ :

$$\Pi_{MN} = \arg \min_{\Pi} \|\Pi \Phi_N C_{NM}^T - \Phi_M\|_F^2 + \|\Pi \Phi_N C_{MN} - \Phi_M\|_F^2$$

And the same operation can be performed for  $\Pi_{NM}$ .

In the modified version of this algorithm, the pointwise map directly comes from our complex functional maps, and thus is orientation-aware. Its translation to functional map will also carry that information, making Algorithm 2 more robust than its original version to symmetry errors.

**Complex Discrete Optimization** Secondly, we modify discrete optimisation algorithms [RMWO21] (again, the necessary modifications bear bold line numbers). These algorithms consist in reducing a continuous energy with the same kind of trick used in bijective

ZOOMOUT. It is thus adaptable to all kind of energies. Besides, this method proved more efficient than continuous solvers to reduce natural energies such as conformality or isometry. one simply defines either  $E(\Pi_{MN})$  (respectively  $E(\Pi_{MN}, \Pi_{NM})$  in the case of a bijective energy), rewrites a continuous energy using the pointwise map trick  $C_{MN} = \Phi_N^\dagger \Pi_{NM} \Phi_M$ , and minimizes it for  $\Pi_{MN}$  given  $\mathbb{C}_{NM}$ . Adding in our  $Q$ -step, we get Algorithm 3. This algorithm is both orientation-aware *and* optimizes for desirable maps such as isometries, often resulting in the best map as shown in Table 3 and Figure 7.