

# Continuous Matching via Vector Field Flow

Etienne Corman<sup>1,2</sup> Maks Ovsjanikov<sup>1</sup> Antonin Chambolle<sup>2</sup>

<sup>1</sup>LIX, École Polytechnique, CNRS, France  
<sup>2</sup>CMAP, École Polytechnique, CNRS, France

---

## Abstract

*We present a new method for non-rigid shape matching designed to enforce continuity of the resulting correspondence. Our method is based on the recently proposed functional map representation, which allows efficient manipulation and inference but often fails to provide a continuous point-to-point mapping. We address this problem by exploiting the connection between the operator representation of mappings and flows of vector fields. In particular, starting from an arbitrary continuous map between two surfaces we find an optimal flow that makes the final correspondence operator as close as possible to the initial functional map. Our method also helps to address the symmetric ambiguity problem inherent in many intrinsic correspondence methods when matching symmetric shapes. We provide practical and theoretical results showing that our method can be used to obtain an orientation preserving or reversing map starting from a functional map that represents the mixture of the two. We also show how this method can be used to improve the quality of maps produced by existing shape matching methods, and compare the resulting map's continuity with results obtained by other operator-based techniques.*

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: —Shape Analysis

---

## 1. Introduction

Computing correspondences or mappings between 3D shapes is one of the key building blocks in many areas of digital geometry processing, including deformation transfer [SZGP05], shape interpolation (morphing) [KMP07] and statistical shape analysis [HSS\*09] among many others. This problem is particularly challenging in the case of shapes undergoing non-rigid deformations, where the notion of the optimal map may be difficult to define and optimize for. Thus, although a number of robust techniques have been proposed to address the rigid alignment problem [MAM14], non-rigid shape matching remains challenging [TCL\*13].

Most of the successful global methods proposed to find correspondences between pairs of non-rigid shapes in the recent years have relied on a variant of the conformal [WWJ\*07, LF09, KLF11] or fully isometric [BBK06, TBW\*09, SY11, OBCS\*12] deformation models, which assume that either the angles or the geodesic distances between pairs of points are approximately preserved by the mapping. Although such models have very appealing theoretical properties, using them directly can often lead to difficult non-linear, non-convex

optimization problems [BBK06]. Therefore, most recent work in this direction have concentrated on finding a low-dimensional parameterization of the space of mappings, that allows for efficient optimization techniques (e.g. [LF09, OMMG10, BWW\*14]).

Among such low-dimensional representations of the space of correspondences, one particularly appealing approach is based on the framework of functional maps [OBCS\*12], which consider mappings as linear operators between the corresponding function spaces. This representation has the advantage of being computationally efficient and easy to manipulate, since typically it allows to encode a correspondence with a small-sized matrix using a multi-scale functional basis. Moreover, finding the optimal functional map, can often be formulated using relatively simple optimization problems [OBCS\*12, PBB\*13]. As a result, methods based on this representation, have recently been used to achieve state-of-the-art results for near-isometric shape matching problems [PBB\*13] and co-segmentation of shape collections [HWG14].

One of the weaknesses of the functional map representation, however, is that by representing mappings

as correspondences between functions, it requires an additional post-processing step to obtain a point-to-point map after computing the optimal functional map. The basic approach for this conversion step, proposed in [OBCS\*12] and used in most follow-up works, assigns points by considering the mapping between the corresponding Dirac delta-functions. Since each delta-function is mapped independently, however, this approach can (and most often does) introduce significant artifacts and discontinuities into the final point-to-point mapping (see the first two columns of Figure 6). This makes the resulting correspondences unusable in settings that require continuity of the mapping, such as texture transfer. Additional pair-wise terms can potentially be introduced in the conversion procedure, but this would require creating variables for points with potentially very expensive consistency constraints, which very quickly loses the appeal of the functional map framework, and reduces to direct optimization.

In this context, we propose a novel method for converting a functional map to a point-to-point map, which guarantees continuity and does not rely on any pairwise consistency constraints, making it computationally efficient. Our main idea is to represent the target point-to-point map as a composition of an arbitrary continuous map between the two surfaces and a flow associated with an unknown vector field on one of them. By relying on the recently proposed operator representation of vector fields [ABCCO13], we show that the optimal vector field can be computed efficiently entirely within the functional map framework, and the computation of the final map requires a single discretization of vector field advection. We also employ a recently proposed supervised learning technique [COC14] that not only helps to obtain better functional maps but also helps to identify functional subspaces where the map is reliable, which significantly helps to improve the final point-to-point map.

Our method also helps to address the symmetric ambiguity problem inherent in many intrinsic correspondence methods when matching symmetric shapes. We provide practical and theoretical results showing that our method can be used to obtain an orientation preserving or reversing map starting from a functional map that represents the mixture of the two. Finally, we test our method on a shape collection and show that we can produce maps that are both continuous and have smaller geodesic distortion compared to the results obtained by existing techniques.

## 2. Related Work

Non-rigid shape matching is an extremely very well-developed area and we refer the interested reader to

recent surveys (e.g., [BBK08, VKZHC011, TCL\*13]) for an in-depth review of all of the related work. Below we concentrate on the recent works that are directly related to ours, consisting of methods for global near-isometric shape matching with special emphasis on approaches that guarantee the continuity of the resulting maps.

As mentioned in the introduction, most of the existing techniques for non-rigid shape matching use a deformation model for finding correspondences between 3D shapes. The two most common models in this setting include approximate intrinsic isometries and conformal mappings. The former model, which was originally introduced by Bronstein et al. [BBK06] and Mémoli [Mém07] assumes that pairwise geodesic distances are approximately preserved by the deformation. The first works that use this assumption lead to continuous maps by design, but result in very challenging optimization problems that are difficult to solve with more than a small number of points [BBK06]. As a result, many follow-up techniques have used a relaxed version of the isometric mapping assumption, which result in more manageable optimization problems, but can often fail to guarantee a low distortion continuous mapping (e.g., [HAWG08, TBW\*09, OMMG10, SY11, BWW\*14]). Furthermore, an additional challenge in using the isometric model assumption is that exact intrinsic isometries are extremely rare, both in theory [Glu75] and in practice, since most deformable shapes induce some amount of distortion.

Another set of successful techniques, which are more widely applicable than those based on the isometric mapping assumption are those that assume that the mapping is conformal, and thus only preserves angles (e.g., [HAT\*00, WWJ\*07, HS09, LF09, KLF11]). These techniques are appealing because a conformal mapping is known to exist between any pair of shapes with the same topology, but also because the set of such mappings can be parameterized relatively easily by using a canonical domain, such as a sphere for genus zero surfaces. Moreover, the resulting maps obtained by these approaches are typically continuous. At the same time, conformal mappings can often induce large area distortion, which can result in unrealistic correspondences between non-rigid shapes, which limits their use significantly.

A recent set of approaches that overcome the above-mentioned challenges to some extent is based on the functional map representation, introduced in [OBCS\*12]. This framework is based on representing maps as linear operators acting on real-valued functions, and which can be encoded compactly by small-sized matrices in the discrete setting by using a multi-scale basis. Although the original approach and

the follow-up works, including [KBB\*13, PBB\*13], all implicitly use the isometric deformation assumption, they have been shown to be very robust to small non-isometric distortions, by extensive use of strong geometric and linear-algebraic regularization techniques. Moreover, several recent works have shown how this framework can be used in the supervised learning setting, where functional maps between unseen shapes can be obtained by exploiting information present in a small set of example maps [RBW\*14, COC14].

Despite its practical appeal, one of the limitations of the functional map framework, however, is that a post-processing step is necessary to convert a functional map to a point-to-point one. The method used in [OBCS\*12] is based on mapping Dirac delta functions. However as the points are considered independently the continuity of the resulting map is not ensured. This problem can be particularly prominent in shapes that contain intrinsic symmetries, which contain at least two equally good solutions for the optimal functional map, and the computed one is at best a linear blending of the two.

Note that, closely related to our technique, especially in the use of flows for computing continuous maps (diffeomorphisms) is the LDDMM framework [BMTY05, MTY06], widely used in the medical imaging community. Unlike these methods, however, our approach is purely intrinsic and operates directly on the surface of the target model, rather than deforming a template in space.

**Contributions** In this paper we propose a novel method for converting a functional map into a point-to-point one, which combines the strengths of the functional map framework that allows to compute low-distortion functional maps, with those of the conformal mapping approaches, which produce continuous correspondences. Namely, starting from a map computed using the state-of-the-art conformal-based Blended Intrinsic Map approach [KLF11], we modify it by computing the optimal vector field, whose flow, composed with the original map, would result in a functional map as close as possible to the given one. By using the recently proposed operator representation of vector fields [ABCCO13] and the connection between advection and matrix exponentiation, we propose an efficient optimization approach for computing the optimal vector field entirely within the functional map framework. Moreover, we show theoretically that this approach is guaranteed to produce the correct continuous map when the input functional map represents a blending of the orientation preserving and reversing maps under certain assumptions, and demonstrate this projection step in practice.

### 3. Functional maps

The functional map representation introduced in [OBCS\*12] provides a flexible framework for representing and manipulating maps between shapes. Given two surfaces  $M$  and  $N$ , a point-to-point map  $T : N \rightarrow M$  induces a map between function spaces  $C_T : L^2(M) \rightarrow L^2(N)$ , where  $L^2(M)$  is the set of square-integrable functions defined on the surface  $M$ . The functional map  $C_T$  is defined by composition with  $T$  as  $C_T(f) = f \circ T$ . The operator  $C_T$  is a linear transformation and given a basis it can be represented as a matrix in the discrete setting. This matrix can be easily computed if the map  $T$  is known.

Following the pipeline proposed in [OBCS\*12], we use a two stage algorithm to tackle the shape matching problem.

#### 3.1. Functional maps pipeline

If  $T$  is unknown the first step is to approximate  $C_T$  by formulating functional constraints of the type  $C_T f = g$ , where  $f$  and  $g$  are functions of  $N$  and  $M$  respectively. The functional constraints used in the original work [OBCS\*12] come from local shape descriptors that are stable under nearly isometric deformations. Common robust descriptors include the Heat Kernel Signature (HKS) [SOG09] and the Wave Kernel Signature (WKS) [ASC11], as well as descriptors coming from segment correspondence constraints. We will let  $F$  and  $G$  denote the matrices whose each column contains corresponding functions on  $N$  and  $M$ , which implies that  $C_T F \approx G$ . Additionally a regularization is added using the assumption that the deformation is nearly isometric, which it is equivalent to  $C_T \Delta_N \approx \Delta_M C_T$ , where  $\Delta_N$  and  $\Delta_M$  are the Laplace-Beltrami operators on  $N$  and  $M$  respectively. This leads to the least square problem:

$$C = \arg \min_X \|XF - G\|_F^2 + \alpha \|X\Delta_N - \Delta_M X\|_F^2, \quad (1)$$

where  $\|\cdot\|_F$  denotes the Frobenius norm. It has been shown that solving Eq. (1) can lead to good approximation  $C$  of the functional map  $C_T$ . In this paper we will use several modifications of this model introduced in [COC14] that weighted the functional correspondence by a diagonal matrix  $D$ :  $\|(XF - G)D\|^2$ . This weight is automatically set using a learning procedure, which leads to functional maps of significantly better quality. Namely, from a set of shapes  $N_i$  with known ground-truth functional maps  $C_i^* : L^2(M) \rightarrow L^2(N_i)$  we find the set of weights  $D$  that minimizes the difference between the approximation and the actual map:

$$\min_D \|C_i(D) - C_i^*\|, \text{ where } C_i(D) = \arg \min_X E(X) \\ E(X) = \|(XF - G)D\|_F^2 + \alpha \|X\Delta_{N_i} - \Delta_M X\|_F^2 \quad (2)$$

Moreover this framework provides a way to learn a basis  $Y_p$  (where  $p$  is the basis size) of the functions whose transfer is the most accurate, by minimizing  $\sum_i \|(C_i(D) - C_i^*)Y_p\|$ , given a fixed weight matrix  $D$ .

The basis  $Y_p$  is particularly useful in factoring out badly matched functions, which typically represent the parts of shapes, for which the descriptor constraints fail to provide reliable information.

Once the functional map  $C$  is computed, the goal of the second step is to convert it to a point-to-point map. The method proposed in [OBCS\*12] and reused in most of the follow-up work consists in finding the nearest neighbors of the images of Dirac-delta functions on  $M$  by  $C$  among the Dirac functions on  $N$ . Namely, for each point  $x \in M$ , the map:  $T(x)$  is computed as via  $T(x) = \arg \min_y \|\delta_y - C\delta_x\|$ , where  $\delta_x$  is an indicator function on vertex  $x$ , written in the appropriate basis.

### 3.2. Main challenges

While both steps described above are very efficient in practice, the second stage has a very serious limitation, in that it processes each point independently, meaning that the final map  $T$  may not be (and often is not) continuous. The first two columns of Figure 6 provide examples of discontinuous maps resulting of this conversion.

To illustrate this phenomenon, let us assume that the target shape  $N$  has an orientation-reversing (reflectional) intrinsic symmetry  $S : N \rightarrow N$ . In this case, there exist at least two equally good potential solutions for Eq. (1) and similarly, each point  $x$  may have several candidate correspondences.

In practice the functional constraints are not sufficient to resolve symmetric ambiguities, in large part because most robust descriptors are invariant under intrinsic isometries. The best we can hope for when approximating  $C_T$  is an exact functional map for symmetric functions (i.e.  $f$ , s.t.  $f \circ S = f$ ) and a noisy or zero functional map for antisymmetric functions (i.e.  $f \circ S = -f$ ). Since our approximations are obtained by solving a linear system, most likely a solution of the least squares problem will be a linear blending between the orientation preserving and reversing functional map:

$$C_T^\alpha = (1 - \alpha)C_T + \alpha C_{T \circ S} \tag{3}$$

Note that  $\alpha = 0.5$  implies that all antisymmetric functions are mapped to zero.

The conversion of  $C$  to a point-to-point map in itself gives no guaranty of continuity in the resulting map. Since each Dirac function of a point  $x$  is treated

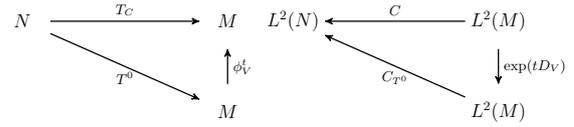


Figure 1: Left: the unknown continuous map  $T_C$  is a composition of the input  $T^0$  and the flow  $\phi_V^t$  of a vector field  $V$ . Right: dual representation as functional maps.

independently it can be mapped indifferently to its image  $T^{-1}(x)$  or to its symmetric alternative  $S(T^{-1}(x))$ . Moreover this process is not designed to be stable under the blending noise  $\alpha$ , as in (Eq. 3).

In this context, the key idea developed in this paper is to construct a point-to-point map from the functional map  $C$  by following a procedure that *guarantees* continuity, while being robust to blending noise. In particular, starting from an arbitrary continuous map between  $M$  and  $N$ , we find an optimal vector field, whose flow makes the final correspondence operator as close as possible to the initial functional map. Since the flow of a vector field provides a continuous, and orientation-preserving map, the final correspondence is both continuous and has the orientation of the initial map. As we show below, this can significantly improve the quality of the resulting point-to-point map, while remaining computationally tractable and avoiding expensive second-order pairwise constraints.

### 3.3. Algorithm overview

The algorithm proposed in this paper takes as input a functional map  $C : L^2(M) \rightarrow L^2(N)$  and an arbitrary continuous map  $T^0 : N \rightarrow M$ . It then outputs a continuous point-to-point map  $T_C : N \rightarrow M$ .

As mentioned above, the main idea of our algorithm is to construct the map  $T_C$  by composing  $T^0$  with the flow  $\phi_V^t$  of a well-chosen vector field  $V$  (see Figure 1). We will choose the vector field  $V$  such that  $\phi_V^t \circ T^0$  represented as a functional map is as close as possible to the input  $C$ . This can be done efficiently by representing  $\phi_V^t$  as an operator (Section 4) and then solving a small-scale optimization problem as explained in Section 5. To find the map  $T_C$  we solve a system of ODEs with a simple solver (Section 6).

The main steps of the proposed algorithm are described in Algorithm 1.

## 4. Family of diffeomorphisms

In this section we construct a family of diffeomorphisms which map  $N$  onto  $M$  and derive their repre-

---

**Algorithm 1:** FUNCTIONAL MAP CONVERSION

---

- Input** :  $C : L^2(M) \rightarrow L^2(N)$  functional map  
 $T^0 : N \rightarrow M$  initial continuous map  
**Output:**  $T_C : C$  converted into a continuous map
- 1 Find Optimal Vector field (Section 5);
  - 2 Convert  $T^0$  to a functional map  $C_{T_0}$ ;
  - 3 Solve:  $a^* \in \arg \min_{a \in \mathbb{R}^n} \|C_{T_0} \exp(\sum_{i=1}^n a_i D_{V_i}) - C\|_\phi$ ;
  - 4 Set:  $V := \sum_{i=1}^n a_i^* D_{V_i}$ ;
  - 5 Compute  $T_C$  (Section 6);
  - 6 Solve:  $\frac{d}{dt} \phi_V^t(p) = V(\phi_V^t(p))$ ,  $\phi_V^0(p) = p \in N$ ;
  - 7 **return**  $T_C := \phi_V^1 \circ T^0$ ;
- 

sentation as functional maps. The point-to-point map which converts the given functional map  $C$  will be chosen among this family.

**Vector field flow** Given a family of tangent vector fields  $\{V_i\}_{1 \leq i \leq n}$  on  $M$ , we let  $\mathcal{V}$  be the space spanned by the linear combinations of the  $V_i$ . Any vector field  $V \in \mathcal{V}$ , defines a one-parameter family of maps  $\phi_V^t : M \rightarrow M$  called the flow of  $V$ . The flow is formally defined as the unique solution to the differential equation:

$$\frac{d}{dt} \phi_V^t(p) = V(\phi_V^t(p)), \quad \phi_V^0(p) = p \in N. \quad (4)$$

Given an arbitrary diffeomorphism  $T : N \rightarrow M$  we construct a family of diffeomorphisms  $\mathcal{T}$  parametrized by  $t \in \mathbb{R}$  and  $a \in \mathbb{R}^n$ :

$$T_a^t(p) = \phi_{V_a}^t \circ T(p), \quad V_a = \sum_{i=1}^n a_i V_i \quad (5)$$

Remark that the orientation of a map  $T_a^t \in \mathcal{T}$  is given by the orientation of  $T$  since the flow of a vector field is orientation preserving.

**Functional Representation of the family** The family of mappings  $\mathcal{T}$  has an easy representation in the functional map framework as explained in [ABCCO13]. This is because, a vector field  $V$  on a smooth manifold can be represented as an operator  $D_V$  acting on a function  $f$ :

$$D_V(f)(p) = \langle V(p), \nabla f(p) \rangle_p. \quad (6)$$

Since the action of  $D_V$  is linear, the operator is conveniently represented as a matrix in the discrete setting.

It is well known that  $g_t = f \circ \phi_V^t$  is the unique solution of the PDE:

$$\frac{\partial g}{\partial t}(t, p) = D_V(g)(t, p), \quad g(0, p) = f(p).$$

A key property of the operator representation of vector fields, introduced in [ABCCO13] is that for analytic functions the functional map  $C_{\phi_V^t}$  is represented by the exponential of the operator  $D_V$  since one has:

$$C_{\phi_V^t} f := f \circ \phi_V^t = \exp(t D_V)(f)$$

Since map composition is achieved via matrix multiplication in the functional representation, this yields a simple way of describing our family of diffeomorphisms  $\mathcal{T}$ . Let  $T_a^t \in \mathcal{T}$  then

$$C_{T_a^t} = C_T \exp\left(t \sum_{i=1}^n a_i D_{V_i}\right). \quad (7)$$

## 5. Optimal vector field

### 5.1. Optimization Problem

Our main idea, developed in the section, is to project the input functional map  $C$  onto the appropriate set of diffeomorphisms  $\mathcal{T}$ . Namely our goal is to find a vector field  $V \in \mathcal{V}$  such that the operator representation (7) of  $T_a^t$  is as-close-as possible to  $C$ . This projection is easily written thanks to the operator representation, and computationally it reduces to solving the optimization problem:

$$\min_{a \in \mathbb{R}^n} \|C_T \exp\left(\sum_{i=1}^n a_i D_{V_i}\right) - C\|_\phi, \quad (8)$$

for an appropriate choice norm  $\|\cdot\|_\phi$ . Here we note briefly that the norm is chosen to be differentiable.

In practice, the problem (8) can be solved using a first order method such as the L-BFGS algorithm. The main difficulty lies in finding the gradient of the objective function is the computation of derivative of  $\exp(\sum_{i=1}^n a_i D_{V_i})$  in the direction  $V_j$ . While there exists a vast literature on approximating the exponential of a matrix (for a survey see [MVL03]), to the best of our knowledge few methods address the problem of computing the directional derivative of the matrix exponential, which is conceptually non-trivial. As we show in the Appendix, however, the directional derivative can be obtained as a block of the matrix exponential of a bigger operator.

### 5.2. Properties

One of the advantages of the formulation of the problem of finding the optimal point-to-point map from a functional map via Eq. (8) is that it makes no assumptions on the input map  $C$ . This is particularly important since, as mentioned above, in the presence of intrinsic symmetries the functional map  $C$  can, even in the best case, be a linear blending of the functional representation of an orientation-preserving

and orientation-reversing map. However, one potential problem is that the presence of the “noisy” part in the functional map can adversely affect the final output map  $T$  obtained by optimizing Eq. (8).

Fortunately, both in theory and in practice this is not the case. Namely, under some suitable assumptions, the orientation-preserving ground-truth functional map, must be a local minimum of the problem (8) even when the functional map  $C$  is given by the symmetry blending defined at (3). In particular, as we show in the Appendix if the norm  $\|\cdot\|_\phi = \|\cdot\|_F^2$  is the squared Frobenius norm, the set of vector fields considered  $\mathcal{V}$  is divergence-free and the initial transformation  $T$  approximately isometric, then  $C_T$  must be a local minimum of Eq. (8).

### 5.3. Practical choice of the norm

As stated before  $C$  is not reliable for antisymmetric functions. Therefore there is some function subspace on which  $C$  and  $C_T \exp(t \sum_{i=1}^n a_i D_{V_i})$  cannot agree. The choice of the norm  $\|\cdot\|_\phi$  in the problem (8) is of critical importance. The naive choice of the squared Frobenius norm is not well-suited for this problem since it is the sum of the squared singular values. As such, it will give a large weight on badly matched function subspace and a small weight on well matched function subspace. However, since typically we have almost no information about antisymmetric functions so the optimization problem based on this problem will put a lot of effort matching functions that we cannot hope to match and few matching interesting subspace. A better choice for  $\|\cdot\|_\phi$  is a regularization of the nuclear norm. We choose

$$\|A\|_\phi = \|AY_p\|_{\epsilon,*} \tag{9}$$

where  $Y_p$  is a basis of  $p$  functions that we want to focus on obtained using the approach described in Section 3.1 and based on [COC14]. In the unsuperised setting we chose  $Y_p$  to be the identity. The norm  $\|\cdot\|_{\epsilon,*}$  is a defined by  $\sum_{i=1}^N \sqrt{\sigma_i^2 + \epsilon}$  the  $\sigma_i$  are the singular values of the matrix. With this norm, we give smaller weight to the subspaces that are difficult to align and focus on task we are able to complete.

The parameter  $\epsilon$  makes the function  $\|\cdot\|_{\epsilon,*}$  differentiable and should be small and is taken at  $10^{-3}$ . Note that the Jacobian matrix of the singular values is easily computable as explained in [PL00].

### 6. Vector field flow on manifold

Once the optimal vector field  $V$  is found using the procedure described above, we obtain the final point-to-point map by composing the initial map  $T$  with the flow of  $V$ . To compute this flow, we need to solve the

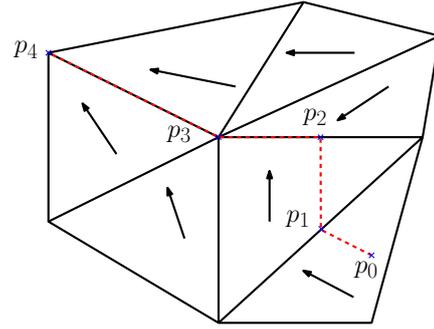


Figure 2: Example of a path trace starting point at  $p_0$ .

system of equations (4) on the given triangle mesh. In principle any advection solver will work with our method. However since computing the flow is known to be potentially difficult, we implemented our own solution. The implementation we use gives a coarse approximation of the flow and might not be accurate for very large deformations. For more accurate solution of this problem we refer to [RS14,MPZ14] which provide more guaranties of continuity of the flow and faster convergence.

In all of our applications we assume that the shape is a triangulated mesh and the vector field is given as a single vector per face. Given this representation, we assume that that the vector field is constant per face and is interpolated at the edges.

Three main situations can occur: the current point could be at inside face, an edge or a vertex.

**At a Face** Since inside a face the vector field is assumed to be constant, we follow it until we reach an edge or a vertex (Figure 2 from  $p_0$  to  $p_1$ ).

**At an Edge** When a point is at an edge, we try to cross the face we did not come from (Figure 2 from  $p_1$  to  $p_2$ ). If the point did not move we follow the edge by interpolating the vector field from the two neighboring faces and end up at a vertex (Figure 2 from  $p_2$  to  $p_3$ ).

**At a Vertex** When the point is at a vertex, we try to follow the vector of each of the neighboring faces and choose the one that goes the furthest. If the point cannot move, we try to follow the neighboring edges, using interpolated directions and to potentially end up at another vertex (Figure 2 from  $p_3$  to  $p_4$ ).

### 7. Results

For all the experiments we express all functions in the basis given by the first 150 eigenfunctions of the Laplace-Beltrami operator. We choose a family of 50

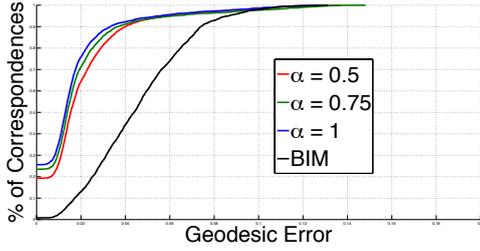


Figure 3: Impact of the noisy functional map  $C_T^\alpha$  on the point-to-point correspondences for various values of  $\alpha$ .

tangent vector fields for the  $V_i$  given by the first eigenfunctions of the 1-form Laplace-de Rham operator, constructed following the procedure described in [FSDH07].

We have evaluated our method for computing point-to-point correspondences on the shapes on the benchmarks of Angelov and al. [ASK\*05] and of Bronstein and al. [BBK08]. In all of the cases, the input continuous map  $T^0$  is the result of the BIM algorithm [KLF11]. This map is most of the time continuous but can be very distorted in some areas. We will show that our method is able to detect the distorted areas and correct them.

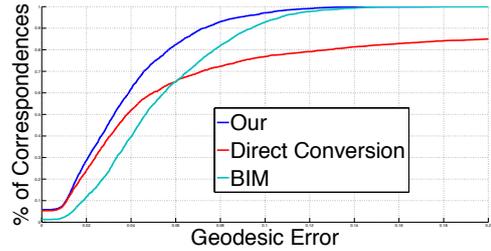
### 7.1. Symmetry blending

As stated above a plausible perturbation for the input functional map  $C$  is given by equation (3). We test our method when  $C$  is the linear blending of the ground-truth functional map and the ground-truth orientation reversing functional map for various values of  $\alpha$ . In this experiment  $Y_p$  is the identity matrix. For this experiment we choose a pair of shapes from the SCAPE dataset.

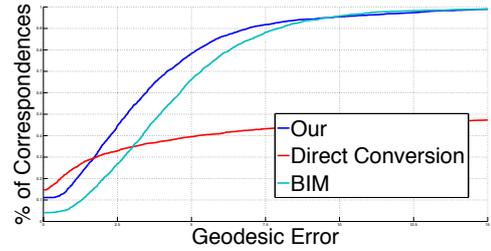
The graph shown in Figure 3 shows the percentage of correspondences smaller than a threshold. Of course the closer  $C_T^\alpha$  is to the ground-truth map the better are the correspondences. However our results are robust even when the target functional map is an exact blending of the direct and symmetric map and are always better than the map coming from BIM. Thus, even when the assumptions of our theoretical observation are not fulfilled, our method can successfully retrieve meaningful information from noisy data.

### 7.2. Error using a computed functional map

In a more realistic scenario, rather than using a ground-truth functional map, we compute it via the inference pipeline described in Section 3.1. In this section the experiments are conducted on several pairs of



(a) SCAPE: average from 7 pairs



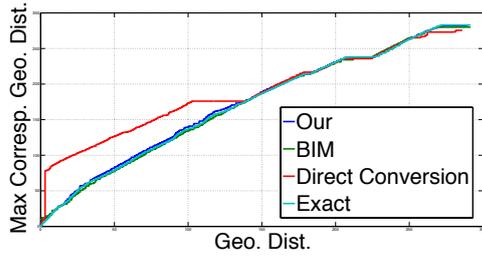
(b) TOSCA: average from 5 pairs

Figure 4: Improvement of the BIM map using our method.

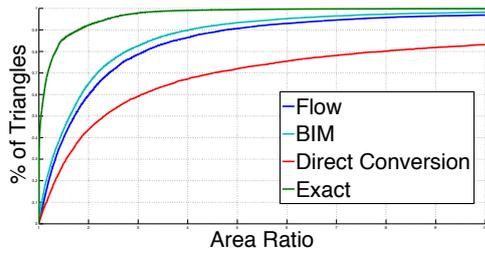
shapes: 7 human pairs (SCAPE) and 5 animal pairs (TOSCA). The functional map  $C$  is computed using the least squares problem (1), where each functional constraints is weighted. The weights are learned solving problem (2) using the algorithm described in [COC14] which also outputs a matrix  $Y_p$  corresponding to the  $p$  best mapped functions, where we let  $p$  equal to 70. The training set is composed of 8 randomly chosen meshes for the SCAPE example and 4 meshes for the TOSCA centaur example. We compute 310 functional constraints equally distributed among these categories:

- Heat Kernel Signature [SOG09],
- Wave Kernel Signature [ASC11] at three different variances,
- Gaussian and Mean Curvature,
- Logarithm of the absolute value of Gaussian and Mean Curvature,
- Mesh Saliency [LVJ05].

We compare our approach with BIM, that serves as  $T^0$ , and with the functional map  $C$  converted to point-to-point map using the method proposed in [OBCS\*12]. The graph in Figure 4 shows the percent of correspondences which have geodesic error smaller than a threshold in average for SCAPE and TOSCA. In this case, we only accept direct correspondences as correct, and consider symmetric points as wrong. Note that our method shows quality improvement over Blended Intrinsic Maps. The direct conversion of  $C$



(a) Maximal distortion computed with Eq. (10)



(b) Area ratio

Figure 5: Comparison of the distortion induced by various for a pair of centaur (TOSCA).

have some point with very large geodesic error due to points mapped to their symmetric counterparts.

We evaluate the continuity of our map with two measures of distortion. First the maximum radius corresponding to a geodesic ball of given size. For a map  $T$  this is formally given by the function:

$$r(t) \mapsto \max_{d_N(x,y) \leq t} d_M(T(x), T(y)), \quad (10)$$

where  $d_N$  is the geodesic distance on  $N$ . If the map is nearly isometric  $r$  should be close to identity. We compare this measure for different mapping in Figure 5a for one example from TOSCA. Our method is comparable to BIM and to the ground truth in terms of continuity while the direct conversion of  $C$  show some very large distortions. Second we compare the ratio between the triangle’s area before and after deformation. Since the deformations in our examples are almost isometric this ratio should be close to one. The graph in Figure 5b shows the percent of triangles which have an area ration smaller than a threshold. We show only the ratio greater than one since most of the discontinuous behavior is due to large jumps. The area ratio of the exact mapping are concentrated around one which is consistent with the fact that the deformation is nearly isometric. Again the direct conversion of  $C$  show some very large area distortions compare to BIM and our method.

This lack of continuity is confirmed by Figure 6 which provides on two examples a visualization of the

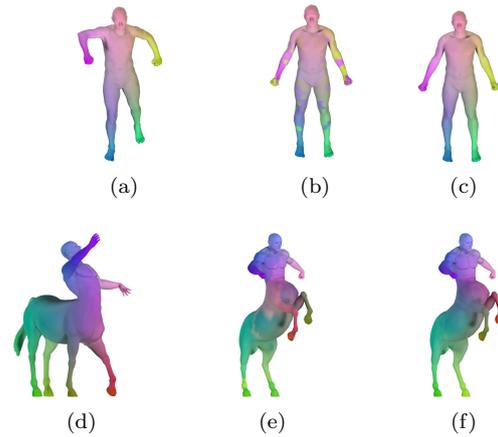


Figure 6: Visualization of the point-to-point mapping through color correspondence. The texture of the first column (6a, 6d) are transferred to the second using the direct conversion of a functional map (6b, 6e) and to the third using our method (6c, 6f).

point-to-point mapping using color correspondence. The direct conversion of the functional map shows some artifacts due to the blending between orientation preserving and orientation reversing maps.

Our method successfully repairs the areas distorted by BIM as shown on Figure 7 for two different matching problems. In this example the BIM maps transfer poorly functions from the source meshes to the target meshes while our method corrects these incorrect matches by providing a more accurate transfer. A visualization of the optimal vector field is provided on Figure 8 for the human example. The vector field on Figure 8b corresponds to the displacement needed to repair the BIM map, the action of this correction can be seen on the upper row of Figure 7.

### 7.3. Parameters dependence

In theory the quality of the initial map  $T^0$  has no impact on the final results. In practice we consider only a small family of vector fields based on the first eigenfunctions of 1-form Laplace-de Rham operator. Therefore in this setting our method will be more efficient in repairing low frequency distortion rather than recovering a high frequency deformation that cannot be represented by the flow of low frequency vector field. Of course the bigger is the vector field basis the better will be the repairs, and the slower will be the method. In the experiments we presented the dimension of the vector field family can be reduced to 40 without influencing too much the point-to-point map.

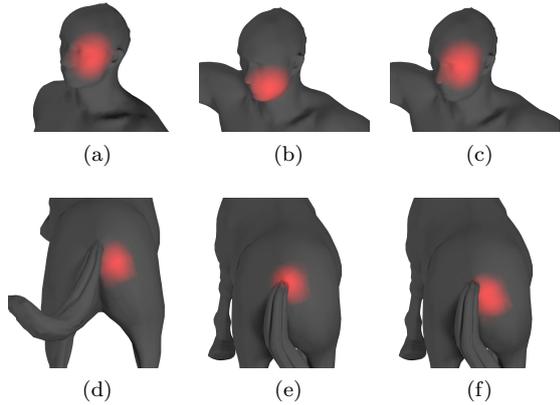


Figure 7: Transfer of a function on the source meshes 7a and 7d to the target meshes using BIM 7b and 7e compared to our method 7c and 7f.

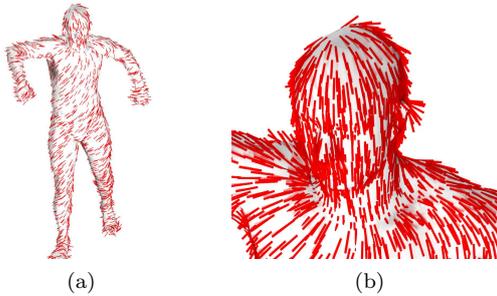


Figure 8: Visualization of the direction of the optimal vector field corresponding to the experiment 7c: complete shape 8a and close-up on the face 8b.

Another critical parameter is the number of eigenfunctions we choose to represent the functional map  $C$  and  $C_{T^0}$ . If the deformation is nearly isometric a small number is sufficient as the functional map  $C$  is almost diagonal. These considerations also apply to the initial map  $T^0$ : a very distorted map is badly approximated by a small number of eigenfunctions and can severely influence our method. We found that lowering the size of the function basis under 150 degrades rapidly the quality of the results.

In principle our method should work for non-isometric deformations provided we are given high-quality functional map as input. To obtain such a map, the choice of the functional basis would have to be modified in order to successfully encode the functional map in a reduced basis. This direction is left as an interesting future work.

Mesh	Vertices	Optimization	Flow
Horse	19248	369s	29.4s
Dog	25290	300s	20.4s
Centaur	15768	381s	39.0s
SCAPE	12500	231.8s	30.8s

Table 1: Average CPU time of each step for different mesh size.

#### 7.4. Performance

For performance evaluation the computation times are given in the Table 1 in various cases. All the experiments have been performed on laptop with a 1.4 GHz processor and 4Go memory without parallelization. The timings are given for the two steps of the method: solving the problem (8) and tracing the flow lines. The time spent solving the optimization problem is almost independent of the number of vertices. The size of this problem depends only on the number of computed eigenfunctions of the Laplace-Beltrami operator and on the dimension of the vector field family, which are constant in all experiments. Note that the computation of the flow does not scale linearly with the number of the vertices. This is explained by the fact we compute a composition with the BIM map which may map many vertices to a single point.

#### 8. Conclusion, Limitations and Future Work

In this paper we presented a method for non-rigid shape matching that is designed to output continuous maps. Our approach combines the strengths of conformal-based approaches, which often guarantee continuity with the functional map framework, which can enable low-distortion maps on the space of functions. Key to our method is enforcing continuity via the flow of a vector field, which allows our method to remain efficient by avoiding expensive pairwise vertex constraints. One of the limitations of our method is that we only approximate the flow of a single vector field, whereas in practice, for complex motions, a combination of flows may be necessary. Extending our method to such cases is possible, while taking care of the robustness and non-accumulation of numerical errors. We are also planning to wider arrays of initial maps and ways to incorporate the continuity directly in the optimization of the functional maps.

#### Acknowledgements

The authors would like to acknowledge the support of the French Direction Générale de l'Armement (DGA), a Google Faculty Research Award, the Marie Curie grant CIG-334283-HRGP, a CNRS chaire d'excellence, and a chaire Jean Marjoulet.

## References

- [ABCCO13] AZENCOT O., BEN-CHEN M., CHAZAL F., OVSJANIKOV M.: An operator approach to tangent vector field processing. *Computer Graphics Forum* 32, 5 (2013), 73–82. 2, 3, 5, 11
- [ASC11] AUBRY M., SCHLICKWEI U., CREMERS D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV Workshops* (2011), pp. 1626–1633. 3, 7
- [ASK\*05] ANGUELOV D., SRINIVASAN P., KOLLER D., THRUN S., RODGERS J., DAVIS J.: Scape: shape completion and animation of people. In *ACM TOG (Proc. SIGGRAPH)* (2005), vol. 24, pp. 408–416. 7
- [BBK06] BRONSTEIN A. M., BRONSTEIN M. M., KIMMEL R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *PNAS* 103, 5 (2006). 1, 2
- [BBK08] BRONSTEIN A., BRONSTEIN M., KIMMEL R.: *Numerical geometry of non-rigid shapes*. Springer, 2008. 2, 7
- [BMTY05] BEG M. F., MILLER M. I., TROUVÉ A., YOUNES L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. *International journal of computer vision* 61, 2 (2005), 139–157. 3
- [BWW\*14] BRUNTON A., WAND M., WUHRER S., SEIDEL H.-P., WEINKAUF T.: A low-dimensional representation for robust partial isometric correspondences computation. *Graphical Models* 76, 2 (2014), 70–85. 1, 2
- [COC14] CORMAN É., OVSJANIKOV M., CHAMBOLLE A.: Supervised descriptor learning for non-rigid shape matching. In *ECCV 2014 Workshops, Part IV* (2014), Springer International Publishing. 2, 3, 6, 7
- [FSDH07] FISHER M., SCHRÖDER P., DESBRUN M., HOPPE H.: Design of tangent vector fields. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, p. 56. 7
- [Glu75] GLUCK H.: Almost all simply connected closed surfaces are rigid. In *Geometric topology*. Springer, 1975, pp. 225–239. 2
- [HAT\*00] HAKER S., ANGENENT S., TANNENBAUM A., KIKINIS R., SAPIRO G., HALLE M.: Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer Graphics* 6, 2 (2000), 181–189. 2
- [HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. *CGF (Proc. SGP)* (2008), 1449–1457. 2
- [HS09] HURDAL M. K., STEPHENSON K.: Discrete conformal methods for cortical brain flattening. *Neuroimage* 45, 1 (2009), S86–S98. 2
- [HSS\*09] HASLER N., STOLL C., SUNKEL M., ROSENHAHN B., SEIDEL H.-P.: A statistical model of human pose and body shape. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 337–346. 1
- [HWG14] HUANG Q., WANG F., GUIBAS L.: Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 36. 1
- [KBB\*13] KOVNATSKY A., BRONSTEIN M. M., BRONSTEIN A. M., GLASHOFF K., KIMMEL R.: Coupled quasi-harmonic bases. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 439–448. 3
- [KLF11] KIM V. G., LIPMAN Y., FUNKHOUSER T.: Blended intrinsic maps. *ACM TOG (Proc. SIGGRAPH)* 30, 4 (2011). 1, 2, 3, 7
- [KMP07] KILIAN M., MITRA N. J., POTTMANN H.: Geometric modeling in shape space. In *ACM Transactions on Graphics (TOG)* (2007), vol. 26, ACM, p. 64. 1
- [LF09] LIPMAN Y., FUNKHOUSER T.: Mobius voting for surface correspondence. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 28, 3 (Aug. 2009). 1, 2
- [LVJ05] LEE C. H., VARSHNEY A., JACOBS D. W.: Mesh saliency. In *ACM Transactions on Graphics (TOG)* (2005), vol. 24, ACM, pp. 659–666. 7
- [MAM14] MELLADO N., AIGER D., MITRA N. J.: Super 4pcs fast global pointcloud registration via smart indexing. *CGF (Proc. SGP)* 33, 5 (2014), 205–215. 1
- [Mém07] MÉMOLI F.: On the use of Gromov-Hausdorff Distances for Shape Comparison. In *Symposium on Point Based Graphics* (2007), pp. 81–90. 2
- [MPZ14] MYLES A., PIETRONI N., ZORIN D.: Robust field-aligned global parametrization. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 135. 6
- [MTY06] MILLER M. I., TROUVÉ A., YOUNES L.: Geodesic shooting for computational anatomy. *Journal of mathematical imaging and vision* 24, 2 (2006), 209–228. 3
- [MVL03] MOLER C., VAN LOAN C.: Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM review* 45, 1 (2003), 3–49. 5
- [OBCS\*12] OVSJANIKOV M., BEN-CHEN M., SOLOMON J., BUTSCHER A., GUIBAS L.: Functional maps: a flexible representation of maps between shapes. *ACM TOG (Proc. SIGGRAPH)* 31, 4 (2012). 1, 2, 3, 4, 7, 11
- [OMMG10] OVSJANIKOV M., MÉRIGOT Q., MÉMOLI F., GUIBAS L.: One point isometric matching with the heat kernel. *CGF* 29, 5 (2010), 1555–1564. 1, 2
- [PBB\*13] POKRASS J., BRONSTEIN A. M., BRONSTEIN M. M., SPRECHMANN P., SAPIRO G.: Sparse modeling of intrinsic correspondences. In *Computer Graphics Forum* (2013), vol. 32, pp. 459–468. 1, 3
- [PL00] PAPADOPOULOU T., LOURAKIS M. I.: Estimating the jacobian of the singular value decomposition: Theory and applications. In *Computer Vision-ECCV 2000*. Springer, 2000, pp. 554–570. 6
- [RBW\*14] RODOLÀ E., BULÒ S. R., WINDHEUSER T., VESTNER M., CREMERS D.: Dense non-rigid shape correspondence using random forests. In *Proc. CVPR* (2014), pp. 4177–4184. 3
- [RS14] RAY N., SOKOLOV D.: Robust polylines tracing for n-symmetry direction field on triangulated surfaces. *ACM Transactions on Graphics (TOG)* 33, 3 (2014), 30. 6
- [SOG09] SUN J., OVSJANIKOV M., GUIBAS L.: A concise and provably informative multi-scale signature based on heat diffusion. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1383–1392. 3, 7
- [SY11] SAHILLIOĞLU Y., YEMEZ Y.: Coarse-to-fine combinatorial matching for dense isometric shape correspondence. *Computer Graphics Forum* 30, 5 (2011), 1461–1470. 1, 2

- [SZGP05] SUMNER R. W., ZWICKER M., GOTSMAN C., POPOVIĆ J.: Mesh-based inverse kinematics. In *ACM SIGGRAPH* (2005), pp. 488–495. 1
- [TBW\*09] TEVS A., BOKELOH M., WAND M., SCHILLING A., SEIDEL H.-P.: Isometric registration of ambiguous and partial data. In *Proc. CVPR* (2009), pp. 1185–1192. 1, 2
- [TCL\*13] TAM G. K., CHENG Z.-Q., LAI Y.-K., LANGBEIN F. C., LIU Y., MARSHALL D., MARTIN R. R., SUN X.-F., ROSIN P. L.: Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *Visualization and Computer Graphics, IEEE Transactions on* 19, 7 (2013), 1199–1217. 1, 2
- [VKZHCO11] VAN KAICK O., ZHANG H., HAMARNEH G., COHEN-OR D.: A survey on shape correspondence. *Computer Graphics Forum* 30, 6 (2011), 1681–1707. 2
- [WWJ\*07] WANG S., WANG Y., JIN M., GU X. D., SAMARAS D.: Conformal geometry and its applications on 3d shape matching, recognition, and stitching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 29, 7 (2007), 1209–1220. 1, 2

## 9. Appendix

**Directional derivative of matrix exponential** Let  $x_0$  be an arbitrary vector and  $x(t) = \exp(t \sum_{i=1}^n a_i D_{V_i}) x_0$ , it is well known that  $x(t)$  satisfies the ODE:

$$x'(t) = \sum_{i=1}^n a_i D_{V_i} x(t), \quad x(0) = x_0.$$

Moreover  $x_h(t) = \exp(t(\sum_{i=1}^n a_i D_{V_i} + hV_j)) x_0$  is solution of

$$x'(t) = (\sum_{i=1}^n a_i D_{V_i} + hV_j)x(t), \quad x(0) = x_0.$$

We denote  $y(t)$  the directional derivative in the direction  $V_j$ :

$$y(t) = \lim_{h \rightarrow 0} \frac{1}{h} \left( \exp \left( t \left( \sum_{i=1}^n a_i D_{V_i} + hV_j \right) \right) - \exp \left( t \sum_{i=1}^n a_i D_{V_i} \right) \right) x_0$$

Therefore the  $y(t)$  is the unique solution of

$$\begin{cases} x'(t) = \sum_{i=1}^n a_i D_{V_i} x(t), & x(0) = x_0 \\ y'(t) = \sum_{i=1}^n a_i D_{V_i} y(t) + V_j x(t), & y(0) = 0 \end{cases}$$

Finally, the directional derivative  $d_j E$  is a block of the matrix exponential of a bigger operator:

$$\begin{pmatrix} E & A \\ d_j E & B \end{pmatrix} = \exp \begin{pmatrix} t \sum_{i=1}^n a_i D_{V_i} & 0 \\ tV_j & t \sum_{i=1}^n a_i D_{V_i} \end{pmatrix}$$

Note that if there are  $n$  vectors in the family of vector fields we have to compute  $n$  matrix exponentials.

**Optimality under blending noise** A necessary condition for  $C_T$  to be a local minimum is:

$$\forall X \in \mathcal{T}(C_T), \quad \langle X, C_T - C_T^\alpha \rangle_F = 0$$

where  $\mathcal{T}(C_T)$  is the tangent space of the set of all functional map at the point  $C_T$ . This tangent space has a simple expression. Consider a small perturbation of  $C_T$  by the flow  $\phi_V^t$  of vector field  $V \in \mathcal{V}$  applied to an arbitrary function  $f$ :

$$\begin{aligned} \lim_{t \rightarrow 0} \frac{1}{t} (C_{\phi_V^t \circ T} f - C_T f) &= \lim_{t \rightarrow 0} \frac{1}{t} (f \circ \phi_V^t \circ T - f \circ T) \\ &= \frac{d}{dt} (f \circ \phi_V^t \circ T) |_{t=0} \\ &= \langle V, \nabla f \rangle \circ T \\ &= C_T(D_V(f)) \end{aligned}$$

The necessary condition becomes:

$$\forall V \in \mathcal{V}, \quad (1 - \alpha) \langle C_T D_V, C_T - C_S C_T \rangle_F = 0$$

If the deformation is nearly isometric isometric  $C_T^\perp \approx C_{T^{-1}}$  see [OBCS\*12], moreover  $C_T^\perp C_S C_T$  is an approximation of the internal symmetry on  $M$ .

Suppose that the functional basis is composed only by even and odd functions with respect to the symmetry  $S$ . Therefore the functional map associated to an internal symmetry is a diagonal matrix with 1 and  $-1$  on the diagonal correspond to the symmetric and antisymmetric eigenfunctions.

The necessary condition becomes:

$$\forall V, \quad (1 - \alpha) \langle D_V, I - C_T^\perp C_S C_T \rangle_F = 0$$

If  $D_V$  represents a divergence free vector field then it is a skew symmetric operator as explained in ([ABCCO13]). Since  $I - C_T^\perp C_S C_T$  is a diagonal matrix the scalar product is always zero. Therefore  $C_T$  is a critical point of (8).