A. Choice of the Laplacian

The original functional maps paper introduced the use of the LBO eigenbasis because it satisfies the desired properties of compactness of representation and stability under near-isometric deformations [87, §5.1]. Furthermore, later works from Aflalo et al. [3, 4] have shown that, in a general setting, and especially when working with smooth signals, the LBO eigenbasis of a manifold $\mathcal{M}$ (of arbitrary dimension) is the optimal orthonormal basis for representing a finite-dimensional subspace of $L^2(\mathcal{M})$. In this setting, it is reasonable to extend the choice of the LBO to the volumetric domain, especially considering that volumetric deformations do not induce a larger geodesic distortion with respect to near-isometric surface deformations (see Sec. C for an empirical study in this regard).

On the other hand, it should be noted that discretizing the LBO operator in a way that fulfills all the algebraic properties of the same operator in the smooth setting is often difficult [119]. In literature there exist multiple alternative discretizations [8, 45, 82], each one with its own pros and cons. Among the various alternatives, we chose the cotangent formula due to its popularity, ease of implementation, and efficiency. This choice is not restrictive. Any alternative discretization can be used equivalently. Nonetheless, since most applications that we address involve analyzing information at the surface (which is the boundary of a tetrahedral mesh), we are forced to a discretization that imposes Neumann's boundary conditions to avoid zero-valued eigenfunctions on the surface.

Volumetric Cotangent Formula We discretize the LBO of a tetrahedral mesh $\mathcal{M} = (V_{\mathcal{M}}, T_{\mathcal{M}})$ with the following per vertex relation

$$\Delta(v_i) = \sum_{j \in N(v_i)} w_{ij}(v_i - v_j)\,, \tag{10}$$

using the n-dimensional cotangent formula [20, 57] to compute per edge weights w_{ij}. These weights consider all tetrahedra $ijkl$ incident to edge ij, according to the following formula

$$w_{ij} = \frac{1}{6} \sum_{ijkl} |v_k - v_l| \cot \theta_{kl}\,, \tag{11}$$

where angle θ_{kl} denotes the dihedral angle at the edge kl opposite to edge ij w.r.t. tetrahedron $ijkl$.

B. Spectral Comparison

LBO Eigenbasis. On continuous Riemannian manifolds, the LBO and its eigenbasis are invariant to isometric deformations. However, in the discrete setting, the position and connectivity of the vertices have a significant impact on the equivalence of the operator across different meshes, and changing the triangulation of a surface or its vertex density can strongly affect the spectral decomposition [53].

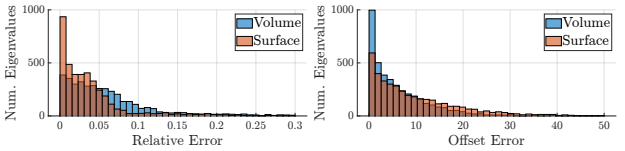

Figure 9. Eigenvalue errors distribution for the volumetric LBO (blue) and the surface LBO (orange).

Moving from discrete surfaces to discrete volumes makes this problem even more evident. In a regular triangular mesh, each vertex has on average 6 neighboring vertices and 6 neighboring triangles, while in regular tetrahedral meshes, the average size of a vertex neighborhood is 12 vertices and 20 tetrahedra.

To ensure that the LBO has a consistent behavior across isometric or similar volumetric shapes, we compute the spectral decomposition of the LBO on 40 pairs of shapes with identical connectivity released by Su et al. [113] and compare both the eigenvalues and the eigenfunctions of the LBO. Our results are summarized in Fig. 9. The first two rows of the figure show that two humanoid shapes share the same LBO eigenfunctions (up to sign flips), even if they are not isometric and do not represent the same subject; the overall similarity is sufficient to generate the same low-frequency spectrum. To compare the eigenvalues, we first compute the vectors $\boldsymbol{\lambda}_{\mathcal{M}}, \boldsymbol{\lambda}_{\mathcal{N}}$ of the first 100 LBO eigenvalues on two tetrahedral meshes. Since the absolute difference between LBO eigenvalues is notoriously not informative, we rely on both the relative difference and the offset difference proposed by Moschella et al. [81] (see below).

For each pair of tetrahedral meshes, we also extract the triangular meshes representing their boundary surface and compute the same quantities. For a more meaningful comparison, all the tetrahedral meshes are rescaled to have unitary volume, and all the surface meshes are rescaled to have unitary surface area. The error distributions are shown at the bottom of Fig. 9, where we can see that the spectral similarity between pairs of tetrahedral and surface meshes is comparable.

Offset Error. The eigenvalues of the Laplace-Beltrami operator form a non-decreasing sequence that grows with a rate $\lambda_k \in \mathcal{O}(\sqrt[d]{k^2})$, where d is the dimensionality of the manifold [121]. A direct comparison of the eigenvalues is thus misleading, as the difference will be dominated by the eigenvalues with larger index. One possibility for handling this issue is to introduce relative differences in the comparison. Alternatively, Moschella et al. [81] introduced the offset representation for comparing and handling Laplacian spectra. Given two manifolds $\mathcal{M}$ and $\mathcal{N}$, the offset

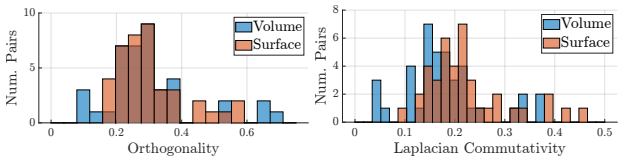

Figure 10. Distribution of the orthogonality (left) and Laplacian commutativity (right) metrics for pairs of volumetric shapes (blue) and surface meshes (orange).

difference can be defined as follows

$$\text{offset}(\boldsymbol{\lambda}_{\mathcal{M}}, \boldsymbol{\lambda}_{\mathcal{N}}) = |\text{off}(\boldsymbol{\lambda}_{\mathcal{M}}) - \text{off}(\boldsymbol{\lambda}_{\mathcal{N}})|\,, \tag{12}$$

$$\text{off}(\boldsymbol{\lambda}_{\mathcal{M}}) = (\text{off}(\lambda_{\mathcal{M},1}), \text{off}(\lambda_{\mathcal{M},2}), \cdots)\,, \tag{13}$$

$$\text{off}(\lambda_{\mathcal{M},i}) = \lambda_{\mathcal{M},i} - \lambda_{\mathcal{M},i-1}\,. \tag{14}$$

This method for comparing eigenvalues is designed to compensate for the Weyl's law on 2-dimensional manifolds: since $\lambda_k \in \mathcal{O}(k)$, comparing consecutive differences (which asymptotically behave like a constant function) is more informative than comparing absolute values. However, when dealing with 3-dimensional manifolds, the Weyl's law changes to $\lambda_k \in \mathcal{O}(\sqrt[3]{k^2})$ and such consecutive differences are not asymptotically constant anymore. This change to the asymptotic behavior also affects the relative difference: if we consider that the k-th eigenvalue can be written as the Weyl's asymptotic relation plus a correction term, namely $\lambda_k = c\sqrt[d]{k^2} + r$, we have that

$$\frac{c_1\, k + r_1 - c_2\, k - r_2}{c_1\, k + r_1} \not\approx \frac{c_1 \sqrt[3]{k^2} + r_1 - c_2 \sqrt[3]{k^2} - r_2}{c_1 \sqrt[3]{k^2} + r_1}\,. \tag{15}$$

As a consequence neither the relative difference or the offset difference have the same scale for 2-dimensional and 3-dimensional manifolds.

However, we can use the Weyl's law to compensate the non-linear growth of the eigenvalues. Since $\lambda_k \in \mathcal{O}\sqrt[3]{k^2}$, before computing the offsets we can transform the eigenvalues as $\lambda'_k = \sqrt{\lambda_k^3} \in \mathcal{O}(k)$. With this transformation, the error measures has the same scale for both the surfaces and the volumes. Clearly, this holds under the assumption that all the 2-dimensional manifolds have unitary area and all the 3-dimensional manifolds have unitary volume, which is easy to enforce as a preprocessing step.

Volumetric Functional Maps. Given a definition for the discrete LBO, the definition of the functional maps framework in tetrahedral meshes follows directly from its 2-dimensional counterpart (see Sec. 3). Nonetheless, in order to ensure a seamless integration of the existing approaches in the volumetric setting, we study how the usual properties of a functional map behave on tetrahedral meshes.

Given two meshes $\mathcal{M}$ and $\mathcal{N}$ and a functional map $\mathbf{C} : \mathcal{F}(\mathcal{M}) \to \mathcal{F}(\mathcal{N})$, Lescoat et al. [53] propose two metrics for evaluating the ideal qualities of $\mathbf{C}$: the orthogonality $\|\mathbf{C}\|_O$ and the Laplacian commutativity $\|\mathbf{C}\|_L$

$$\|\mathbf{C}\|_O = \frac{\|\mathbf{C}^\top\mathbf{C} - \mathbf{I}\|}{\|\mathbf{I}\|}\,, \qquad \|\mathbf{C}\|_L = \frac{\|\mathbf{C}\Lambda_{\mathcal{M}} - \Lambda_{\mathcal{N}}\mathbf{C}\|}{\|\mathbf{C}\Lambda_{\mathcal{M}}\|}\,, \tag{16}$$

where $\mathbf{I}$ is the identity matrix, $\|\cdot\|$ is the Frobenius norm, and $\Lambda_{\mathcal{M}}, \Lambda_{\mathcal{N}}$ are the diagonal matrices containing the LBO eigenvalues of, respectively, $\mathcal{M}$ and $\mathcal{N}$.

Again, we compute the same quantities for tetrahedral and triangular meshes across the same 40 pairs of shapes, summarizing the results in Fig. 10. As shown in the figure, the quality of volumetric functional maps is comparable to the surface cases.

C. Volumetric Isometries

The functional maps framework relies on the assumption that the two shapes are isometric or quasi-isometric. In practice, the framework proposed by Ovsjanikov et al. [87] works well even under mild non-isometric deformations and meshes at a different scale. However, when dealing with strong non-isometric distortion, more sophisticated approaches are needed, which cannot be easily lifted to manifolds of higher dimensions [90].

In order to verify that the functional maps framework can be meaningfully transported to a volumetric setting, we need to verify that the level of isometric distortion does not change when a 3D shape is represented as a 2-dimensional surface or a 3-dimensional volume. The dataset introduced by Su et al. [113] contains 40 pairs of tetrahedral meshes in bijective correspondence, and thus it is easy to compute the distortion of geodesic distances induced by the correspondence. For each pair of shapes $\mathcal{M}$ and $\mathcal{N}$, we consider 100k random pairs of points $x, y \in \partial\mathcal{M}$. We then compute the geodesic distortion induced by the ground truth correspondence $\pi : \mathcal{M} \to \mathcal{N}$ for both the surface and the volumetric setting, respectively as

$$\text{SurfErr}(x, y) = \frac{|d_{\partial\mathcal{M}}(x, y) - d_{\partial\mathcal{N}}(\pi(x), \pi(y)|}{d_{\partial\mathcal{M}}(x, y)}\,, \tag{17}$$

$$\text{VolErr}(x, y) = \frac{|d_{\mathcal{M}}(x, y) - d_{\mathcal{N}}(\pi(x), \pi(y)|}{d_{\mathcal{M}}(x, y)}\,, \tag{18}$$

where $d_{\mathcal{S}}$ denotes the approximation of the geodesic distance in the volume $\mathcal{S}$ provided by the Dijkstra distance, and $d_{\partial\mathcal{S}}$ denotes the analogous approximation over the boundary surface $\partial\mathcal{S}$.

The results are summarized in Fig. 11, where we report for each pair of shapes the average distortion over the surface (in blue) and the volume (in orange), together with the standard deviation. We can see that, in all cases, the distortions are comparable, with the tetrahedral meshes generally achieving a slightly better preservation of the geodesic distances. As a consequence, we guess that, for most practical

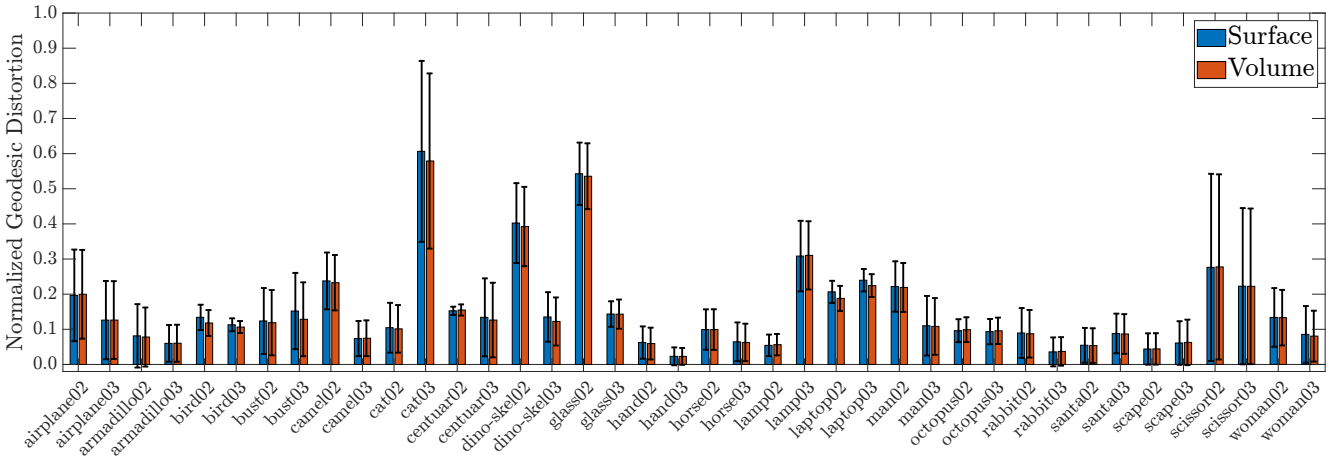

Figure 11. For each subject, the relative geodesic distortion between pairs of volumes (orange) and pairs of surfaces (blue). The plot provides both the average distortion and the standard deviation.

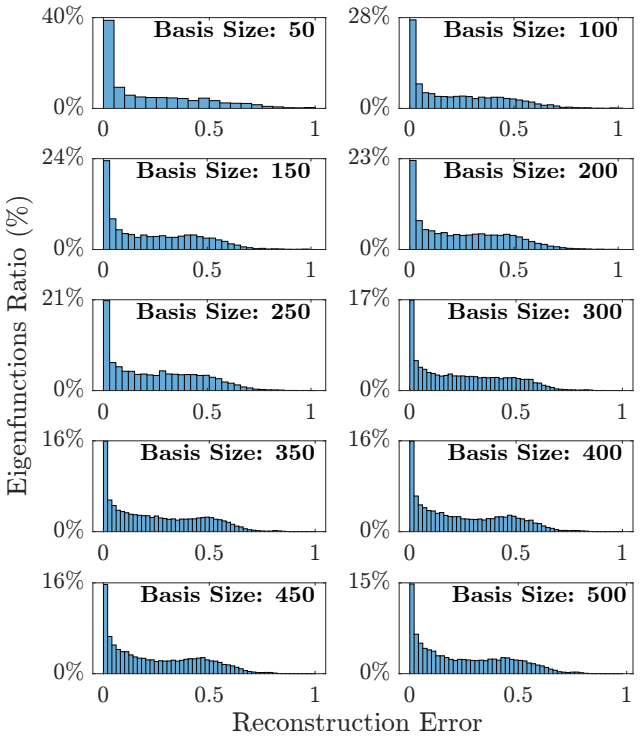

Figure 12. For a varying basis size, we try to reconstruct the LBO eigenfunctions of the surface using the restriction of the volumetric LBO eigenfunctions at the boundary. The reconstruction error lies in the range $[0, 1]$.

applications, the property of a pair of shapes of being isometric is preserved between a surface and volumetric representation, thus allowing us to safely extend the functional maps framework from surfaces to volumes.

D. Boundary Trace of Eigenfunctions

As mentioned in Sec. 4.1, given the volumetric LBO eigenfunctions $\Phi_{\mathcal{M}}$ of a 3-dimensional manifold $\mathcal{M}$, the restriction $\Phi_{\partial\mathcal{M}}$ of such basis at the boundary surface $\partial\mathcal{M}$ (often also called *boundary trace*) forms a set of linearly independent functions over $\partial\mathcal{M}$ [51, 117].

However, $\Phi_{\partial\mathcal{M}}$ is not necessarily equivalent to the LBO eigenbasis $\mathbf{\Psi}_{\partial\mathcal{M}}$ of the surface, meaning that it could be a sub-optimal basis [3, 4]. In order to evaluate the expressive power of the boundary trace of the volumetric eigenfunctions, we try to reconstruct the basis $\mathbf{\Psi}_{\partial\mathcal{M}}$ using the basis $\Phi_{\partial\mathcal{M}}$. For our experiment, we use the 80 shapes from the dataset presented by Su et al. [113] and we try to reconstruct the first k eigenfunctions of the surface with the first k boundary traces. In particular, we first compute the reconstruction $\mathbf{\Psi}' = \Phi_{\partial\mathcal{M}}\Phi^{\dagger}_{\partial\mathcal{M}}\mathbf{\Psi}_{\partial\mathcal{M}}$. Then, for each surface eigenfunction ψ_i and its reconstruction ψ'_i, we compute the reconstruction error as

$$\mathrm{err}(\psi_i) = \int_{\partial\mathcal{M}} (\psi_i(x) - \psi'_i(x))^2 \,\mathrm{d}x\,. \tag{19}$$

This error measure lies in the range $[0, 1]$.

In Fig. 12 we show the distributions of the reconstruction errors across the entire dataset for different values of k. As shown in the figure, very few of the surface eigenfunctions are close to be orthogonal to the boundary traces. However, the boundary restriction of the volumetric eigenfunctions is able to faithfully represent a significant portion of the surface eigenbasis, and provides a reasonable approximation for most of them. Therefore, while not being the optimal basis, the boundary trace of the volumetric LBO eigenfunctions still provides a good candidate for representing a k-dimensional function space over the surface.

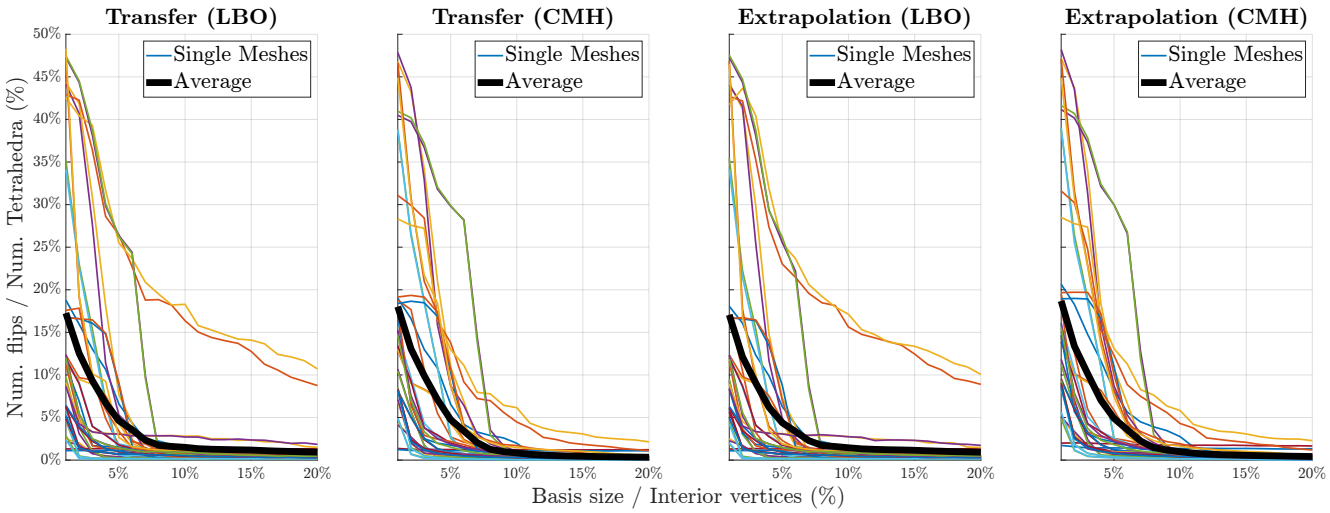

Figure 13. Number of flipped tetrahedra obtained by transferring the mesh connectivity considering a growing number of basis functions. From left to right: coordinate transfer using the LBO basis; coordinate transfer using the CMH basis; coordinate extrapolation using the LBO basis; coordinate extrapolation using the CMH basis. The smaller colored lines show the trend for each pair of shapes in our reference dataset [113]. The bold black line is the average trend across the entire dataset. The two pairs that take longer to converge with the LBO basis functions (yellow and orange lines) are the ones where the source and target differ the most and are strongly non-isometric. This explains why using only the intrinsic information from the LBO eigenbasis makes the method converge slowly and adding extrinsic information from CMH aligns the pairs with the other results.

Method	Continuity	Coverage	Dirichlet	Flips
FMaps	1.32	91.48%	3.82%	10.03%
ZoomOut	1.26	92.63%	3.80%	9.72%
ZoomOut Fast	1.32	92.88%	3.79%	10.80%
Orthoprods	1.29	87.78%	3.81%	5.80%

Table 3. Additional metrics for the volumetric shape matching experiments from Sec. 5.1 on the dataset from Su et al. [113].

E. Volumetric Functional Maps Metrics

Finally, we consider additional metrics to better evaluate the quality of volumetric shape matching discussed in Sec. 5.1. In particular, we report in Tab. 3 the continuity and coverage of the correspondence, as well as its Dirichlet energy, which provide a further characterization of maps in general [98, §5]. We also report the average percentage of flipped tetrahedra, which is an intrinsically volumetric measure. All the values are averaged across the entire dataset from Su et al. [113]. We stress that all reported values (including those in Sec. 5.1) refer to volumetric estimates of the metrics.

F. Connectivity Transfer and Basis Size

While the average behavior of our connectivity transfer approaches is clear from the results shown in Tab. 1, there are additional information that can be better appreciated by examining the behavior of the methods on the single pairs. In particular, we can see from Fig. 13 that some pairs are particularly challenging and results in a number of flips that is significantly higher than the average, when using the LBO eigenbasis. However, CMH seems to be a strong regularizer: the basis extended with the vertex coordinates does not seem to have a significant impact on the other curves, but it is definitely able to align the behavior of these outliers to the average.

On the other hand, it seems that the introduction of the coordinate harmonics negatively affects smaller bases. Indeed, on the left end of the curves, the use of CMH as a basis seems to slow down the decrease rate of the curve.

In summary, by inspecting these results we educatedly guess that a good rule of thumb could be to use CMH when we have access to a larger basis, and stick to the LBO eigenbasis if we can only afford to compute a few basis functions.

We stress that using 20% of the entire basis, in general, is computationally expensive, and we report the values only for analysis purposes. Indeed, for our experiments on the datasets from Nigolian et al. [83] (see Sec. 5.2), we consider a significantly smaller basis with 500 eigenfunctions, covering from 0.5% to 5% of the total basis, depending on the resolution of the tetrahedral mesh.

G. Volumetric Shape Matching Dataset

As mentioned in Sec. 5.4, well-established datasets for shape matching only provide surfaces. On top of that, most of the available recent datasets try to highlight some challenges in the problem of finding a correspondence; this includes partialities, holes, heavy topological noise, self-

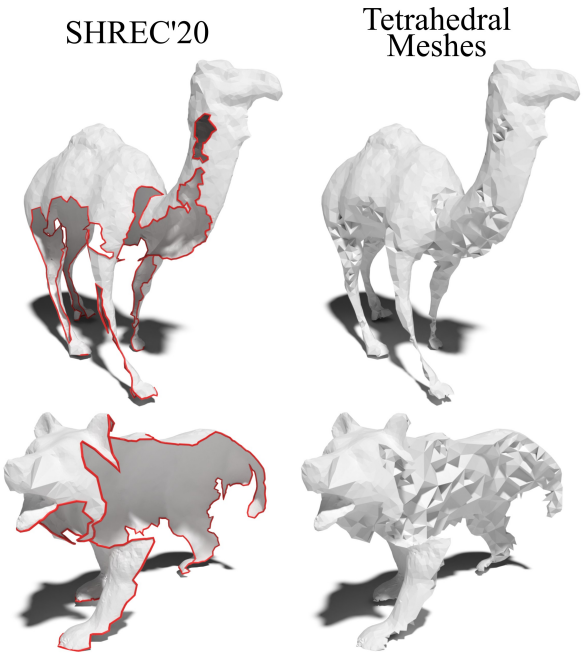

Figure 14. The two partial surfaces from the SHREC'20 dataset that we discarded from our experiments because once turned into tetrahedral meshes they do not correctly convey the intended shape, introducing an unwanted bias in the comparative analysis with surface-based methods.

intersections, clutter, and so on.

Dropping the assumptions of manifoldness, watertightness, and absence of self-intersections for the surfaces, the most robust tool for producing tetrahedral meshes results to be fTetWild [43]. Nonetheless, for some of the surfaces, the problem of finding a tetrahedralization is so ill-posed that any resulting volume would not provide meaningful information about the 3D shape.

Indeed, for SHREC'20, we discard two partial meshes (see Fig. 14) because the inside/outside system of fTetWild was not able to correctly reproduce a volumetric mesh that allows a meaningful comparison with surface methods. For the same reason, we are not able to consider alternative datasets such as SMAL [123], which contains numerous self-intersections that yield topological changes during tetrahedralization. Attempts to sanitize these models with various combinations of geometry processing tools [23, 24, 39, 60, 93] ended with no luck. Indeed, our difficulties in creating a dataset that allows to meaningfully compare surface-based and volume-based shape matching methods raises an open research challenge that will hopefully be addressed by the community in the near future (Sec. 6).

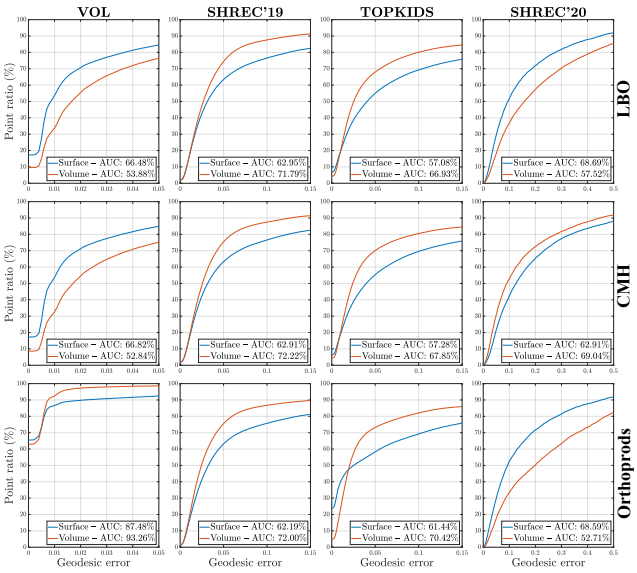

Figure 15. Cumulative plot of the geodesic error obtained with the standard functional maps framework (blue) on surfaces and our volumetric functional maps approach (orange) on four different datasets (columns) and using three different bases (rows). The results are averaged across each dataset.

Figure 16. The tool we use to tetrahedralize surface matching datasets [43] is robust, but occasionally introduces topological noise, closing unwanted handles that alter the mapping task (closeup). The LBO eigenbasis is not robust against such defects. Conversely, the CMH basis is robuster and greatly improves the mapping quality. Both the camel and the cow are from the SHREC'20 dataset.

H. Choice of the Basis for Shape Matching

More detailed results about the comparison can be found in Fig. 15, where we show the cumulative geodesic error curves averaged across each dataset and their corresponding area under the curve (AUC). By comparing these curves against the results shown in Tab. 2, we can appreciate how the results summarized by the average geodesic error reflect on the cumulative curves, highlighting the consistently better behavior of our approach, as well as the importance in the choice of the basis.

In this regard, the use of the CMH basis seems to be particularly effective on SHREC'20, where the surface-based approach achieves better accuracy otherwise. Based

on our understanding, this depends on the basis alignment algorithm we use [85], which is known to not be robust under strong non-isometric deformations and topological changes, which occasionally occurred when tetrahedralizing our surface-based reference datasets. This behavior can be partially compensated by using more informative bases such as CMH, which provide a better tool for representing extrinsic information. This can be better appreciated in the example from Fig. 16, where we show a surface coordinate transfer in a pair from the SHREC'20 dataset with our volumetric approach using the standard LBO eigenbasis and the CMH basis. However, integrating more robust basis alignment methods is certainly worth of future investigation.

As a final note, for all our experiments we use the values suggested in the original papers when selecting the size of the bases. In particular, for the LBO eigenbasis we compute a 20×20 functional map using descriptors derived from 200 eigenfunctions for each shape. The map is extended up to the size 120×120 using ZoomOut, with a step size of 5 basis functions per iteration. In the case of CMH, the last 3 functions are replaced with the orthogonalized coordinates. For the Orthoprods basis, we use the suggested values of 40 eigenfunctions and second order polynomials. However, instead of the numerically unstable derivation of the map proposed in the original paper [68], we extend the functional map with ZoomOut.

I. Algorithmic Details

A sample MATLAB implementation of our volumetric functional maps framework is available at https://github.com/filthynobleman/vol-fmaps. Here we provide the pseudocode for both our connectivity transfer approaches (Sec. 4.1) and our volume-aware surface correspondence method (Sec. 4.2).

Algorithm 1 summarizes our solution for transfering connectivity through functional mapping. Firstly, we compute a tetrahedralization $\mathcal{N}$ for the interior of the target surface $\partial\mathcal{N}$ (Line 2). The volumetric meshes $\mathcal{M}$ and $\mathcal{N}$ are then put in functional correspondence via the map $\mathbf{C}$, which is approximated from the known alignment of the eigenfunctions at the surface (Lines 3-5). The source volume $\mathcal{M}$ is used to initialize the final mesh $\mathcal{R}$ (Line 6), whose coordinate are replaced from the functional transfer of the coordinates of $\mathcal{N}$.

The approach for extrapolating the interior volume from the surface is summarized in Algorithm 1. In this case, we only need the LBO eigenfunctions of $\mathcal{M}$ (Line 2). Again, we initialize a final mesh $\mathcal{R}$ from the source volume $\mathcal{M}$ (Line 3). However, with this approach the interior volume is obtained by projecting the coordinates at the surface in the functional space and reconstructing them in the whole volumetric mesh. The projection is performed using the boundary traces of the eigenfunctions, and the reconstruction is done with the full basis.

Finally, we summarize in Algorithm 3 our procedure to compute the surface correspondence by exploiting the volumetric information. We first compute the interior volumes $\mathcal{M}$ and $\mathcal{N}$ for the surfaces $\partial\mathcal{M}$ and $\partial\mathcal{N}$, respectively (Line 2-3). The LBO eigenbases are then computed on the entire volume (Lines 4-5) and put in functional correspondence through the optimization of the map $\mathbf{C}$ (Line 6). Finally, we compute the correspondence π via nearest neighbor, limiting the search to the boundary traces of the eigenfunctions.

Algorithm 1 Functional connectivity transfer algorithm.

1: **procedure** TRANSFER($\mathcal{M}$, $\partial\mathcal{N}$, $\pi : \partial\mathcal{M} \to \partial\mathcal{M}$)
2: $\quad\mathcal{N} \leftarrow$ computed interior of $\partial\mathcal{N}$
3: $\quad\Phi_{\mathcal{M}} \leftarrow$ LBODECOMPOSITION($\mathcal{M}$)
4: $\quad\Phi_{\mathcal{N}} \leftarrow$ LBODECOMPOSITION($\mathcal{N}$)
5: $\quad\mathbf{C} \leftarrow \Phi^{\dagger}_{\partial\mathcal{M}}\, T_{\pi}(\Phi_{\partial\mathcal{N}})$
6: $\quad\mathcal{R} \leftarrow \mathcal{M}$
7: $\quad\boldsymbol{x}_{\mathcal{R}} \leftarrow \Phi_{\mathcal{M}}\, \mathbf{C}\, \Phi^{\dagger}_{\mathcal{N}}\, \boldsymbol{x}_{\mathcal{N}}$
8: $\quad\boldsymbol{y}_{\mathcal{R}} \leftarrow \Phi_{\mathcal{M}}\, \mathbf{C}\, \Phi^{\dagger}_{\mathcal{N}}\, \boldsymbol{y}_{\mathcal{N}}$
9: $\quad\boldsymbol{z}_{\mathcal{R}} \leftarrow \Phi_{\mathcal{M}}\, \mathbf{C}\, \Phi^{\dagger}_{\mathcal{N}}\, \boldsymbol{z}_{\mathcal{N}}$
10: $\quad$**return** $\mathcal{R}$
11: **end procedure**

Algorithm 2 Spectral coordinate extrapolation algorithm.

1: **procedure** EXTRAPOLATE($\mathcal{M}$, $\partial\mathcal{N}$, $\pi : \partial\mathcal{M} \to \partial\mathcal{M}$)
2: $\quad\Phi_{\mathcal{M}} \leftarrow$ LBODECOMPOSITION($\mathcal{M}$)
3: $\quad\mathcal{R} \leftarrow \mathcal{M}$
4: $\quad\boldsymbol{x}_{\mathcal{R}} \leftarrow \Phi_{\mathcal{M}}\, \Phi^{\dagger}_{\partial\mathcal{M}}\, T_{\pi}(\boldsymbol{x}_{\partial\mathcal{N}})$
5: $\quad\boldsymbol{y}_{\mathcal{R}} \leftarrow \Phi_{\mathcal{M}}\, \Phi^{\dagger}_{\partial\mathcal{M}}\, T_{\pi}(\boldsymbol{y}_{\partial\mathcal{N}})$
6: $\quad\boldsymbol{z}_{\mathcal{R}} \leftarrow \Phi_{\mathcal{M}}\, \Phi^{\dagger}_{\partial\mathcal{M}}\, T_{\pi}(\boldsymbol{z}_{\partial\mathcal{N}})$
7: $\quad$**return** $\mathcal{R}$
8: **end procedure**

Algorithm 3 Surface restricted volume matching algorithm.

1: **procedure** VOL2SURF($\partial\mathcal{M}$, $\partial\mathcal{N}$)
2: $\quad\mathcal{M} \leftarrow$ computed interior of $\partial\mathcal{M}$
3: $\quad\mathcal{N} \leftarrow$ computed interior of $\partial\mathcal{N}$
4: $\quad\Phi_{\mathcal{M}} \leftarrow$ LBODECOMPOSITION($\mathcal{M}$)
5: $\quad\Phi_{\mathcal{N}} \leftarrow$ LBODECOMPOSITION($\mathcal{N}$)
6: $\quad\mathbf{C} \leftarrow$ OPTIMIZEVOLFMAP($\Phi_{\mathcal{M}}, \Phi_{\mathcal{N}}$)
7: $\quad\pi \leftarrow$ NNSEARCH($\Phi_{\partial\mathcal{M}}\mathbf{C}$, $\Phi_{\partial\mathcal{N}}$)
8: $\quad$**return** π
9: **end procedure**