Abstract

Efficient and practical representation of geometric data is a ubiquitous problem for several applications in geometry processing. A widely used choice is to encode the 3D objects through their spectral embedding, associating to each surface point the values assumed at that point by a truncated subset of the eigenfunctions of a differential operator (typically the Laplacian). Several attempts to define new, preferable embeddings for different applications have seen the light during the last decade. Still, the standard Laplacian eigenfunctions remain solidly at the top of the available solutions, despite their limitations, such as being limited to near-isometries for shape matching. Recently, a new trend shows advantages in learning substitutes for the Laplacian eigenfunctions. At the same time, many research questions remain unsolved: are the new bases better than the LBO eigenfunctions, and how do they relate to them? How do they act in the functional perspective? And how to exploit these bases in new configurations in conjunction with additional features and descriptors? In this study, we properly pose these questions to improve our understanding of this emerging research direction. We show their applicative relevance in different contexts revealing some of their insights and exciting future directions.

1. Introduction

Finding a convenient representation for deformable 3D objects is a central research question which has attracted attention for decades [7,52,54]. From a historical perspective, the bulk of the work has gone into the construction of shape embeddings into a high-dimensional space, where complex (e.g. non-rigid) transformations are factored out or translated to simpler transformations.

For example, a general procedure is to embed the intrinsic geometry of a 3D shape into a higher-dimensional space where Euclidean distances approximate the intrinsic metric of the object (multidimensional scaling) [52]. The convenience of such an approach is clear, as we are naturally inclined to work with extrinsic distances (lengths of straight lines in Euclidean space) instead of intrinsic ones (lengths of curved paths on a surface).

Similarly, the eigenfunctions of the Laplace-Beltrami Operator (LBO) have played a prominent role in deformable shape representation. Their theoretical properties (i.e., their analogy with the Fourier basis, together with their natural ordering, orthogonality, and smoothness) make them convenient for several applications, including but not limited to surface fairing, function analysis, segmentation, and shape matching [15, 32, 41]. The latter, in particular, is among the most challenging tasks in 3D geometry processing to date. A recent approach to tackle this task is embodied in the framework of functional map [32], which casts the point-to-point correspondence problem as one of recovering a linear map among functional spaces defined over the input surfaces. The estimation of the correspondence via functional maps can be seen as seeking for an alignment between two high-dimensional point clouds [21, 32, 44] given by the underlying shape via the Laplacian eigenfunctions. While this framework does not rely on any particular choice of bases, the LBO eigenbasis constitutes a natural choice due to its optimality for representing smooth functions [3]. Further, while many variants have been proposed [10,23,24,28], the literature is almost unanimous in adopting the LBO eigen-
Our GFM DN

Figure 1. An example of extreme non-isometric matching. The left shape is a statue from [1], the second is a gorilla from [8].

basis as the de facto standard for shape matching. Along this line of thought, [22] proposed to learn an alternative embedding. It offers an introduction to this framework and provides promising results, but tackling only sparse point cloud (where LBO approximation is less stable) and requiring large datasets for training. [22] only scratched the surface of this direction without offering any analysis of the learned basis. This lack is also a consequence of addressing a domain in which no good alternatives are known.

Concerning [22], we instantiate the method to work on meshes providing a more coherent comparison with LBO. For this reason, we adopt a state-of-the-art feature extractor for surfaces [48]. This approach casts our work directly in the domain where LBO eigenfunctions have played the lion’s share for decades. For the first time, we investigate the properties of the learned embedding, motivating this procedure not only for point clouds but also for meshes. Moreover, exploiting the prior given by the surface (connectivity information), we prove that our method can deal with limited training samples. The new data-driven embedding provides better performance and generalization than LBO eigenfunctions and makes it possible to investigate properties that distinguish the two; which ones have contributed to the LBO eigenfunctions success, and which ones need to be relaxed. Subsequently, we show that learned embedding can be not only a competitor of LBO eigenfunctions but also a completion of them. Using ZoomOut refinement [26] we scale the learned embedding to higher frequencies, inheriting the latest advances in correspondence refinement and opening to a novel perspective for mixed embeddings. We think this aspect is exciting, highlighting the relationship between matching and function representation. In this, we also introduce further losses.

To summarize, our work presents the following novelties and contributions:

1. Extending [22] to meshes, overtaking both LBO and [22] on functional representation in several challenging settings; to our best knowledge, this is the first study that presents a basis set that can better represent shape correspondence than LBO eigenfunctions.

2. Combining an axiomatic embedding with a learned one and scaling them through ZoomOut refinement; this provides a new perspective in shape matching, merging the benefits of the two representations.

3. Involving learned embedding in the function representation task, exploring different kinds of losses; such application for learned embedding has never been addressed before and reveals characteristics of the obtained basis set.

4. Offering an extensive analysis of the learned embedding; while LBO has a long tradition, it is an original field for learned functional bases that deserve analyses and investigations.

All the data and code will be made publicly available.

2. Notation and background

3D shapes. We model 3D shapes as 2-dimensional Riemannian manifolds embedded in \( \mathbb{R}^3 \). In the discrete setting, we encode these objects as triangular meshes composed of vertices and faces defined by the oriented triplet of vertices that belong to the same triangular face. Each face is glued to a maximum of three other faces, one for each edge. We denote with \( \mathcal{M} \) and \( \mathcal{N} \) a pair of shapes.
and with $X_M \in \mathbb{R}^{n_M \times 3}$ and $X_N \in \mathbb{R}^{n_N \times 3}$ the list of the 3D coordinates of their $n_M$ and $n_N$ vertices. To each of these shapes, we associated the Laplace-Beltrami Operator (LBO), the second-order partial differential operator extending the standard Laplacian to non-Euclidean domains and denoted respectively as $\Delta_M$ and $\Delta_N$. We adopt the same notation referring to the square matrices, with size $n_M$ and $n_N$ respectively, which encode these operators in the discrete setting and that can be estimated through the cotangent weight formula [30, 36].

The LBO is a symmetric, positive semi-definite operator which admits an eigendecomposition with non-negative real eigenvalues, sorted in non-descending order $0 = \lambda_1 \leq \lambda_2 \leq \ldots$. The eigenfunctions $\phi_1^\Delta \leq \phi_2^\Delta \leq \ldots$ associated with each eigenvalue compose a basis for the space of square-integrable functions defined over the surface, in analogy with the Fourier basis on Euclidean domains. In the discrete setting, each eigenfunction corresponds to a vector with a length equal to the number of vertices. We store the set of the $k$ eigenfunctions associated to the first $k$ eigenvalues with smallest absolute values, as columns of a matrix $\Phi^\Delta = [\phi_1^\Delta, \ldots, \phi_k^\Delta]$. Each row of this matrix is a vector in $\mathbb{R}^k$ and is referred to as the spectral embedding of the corresponding vertex on the shape. The matrix $\Phi^\Delta$ thus encodes the spectral embedding of the shape.

**Functional maps.** Given the two shapes $M$ and $N$, together with their truncated set of LBO eigenfunctions $\Phi^\Delta_M$ and $\Phi^\Delta_N$ respectively, we denote with $\mathcal{F}(M)$ and $\mathcal{F}(N)$ the space of real-valued functions defined on $M$ and $N$. Any point-to-point correspondence $T_{MN} : M \rightarrow N$ induces a functional mapping (with opposite direction) $T_{NM}^\mathcal{F} : \mathcal{F}(N) \rightarrow \mathcal{F}(M)$ via pull-back. Exploiting the Fourier analogy, we can approximate the spaces $\mathcal{F}(M)$ and $\mathcal{F}(N)$ in the given bases $\Phi^\Delta_M$ and $\Phi^\Delta_N$ of size $k$.

Thanks to this approximation, we can compactly encode the mapping $T_{NM}^\mathcal{F}$ in a matrix $C_{NM} \in \mathbb{R}^{k \times k}$ which corresponds to the linear transformation that maps the coefficients of functions approximated by $\Phi^\Delta_N$ to the coefficients of their images through $T_{NM}^\mathcal{F}$ represented by $\Phi^\Delta_M$. In matrix notation, if we encode the point-to-point map $T_{MN}$ in a binary matrix $\Pi_{MN} \in \mathbb{R}^{n_M \times n_N}$, such that its entries $\Pi_{MN}(i, j) = 1$ if and only if the correspondence $T_{MN}$ associates, to the $i$-th vertex of $M$, the vertex of index $j$ on $N$, then we can explicitly compute the functional map $C_{NM} = \Phi^\Delta_M \Pi_{MN} \Phi^\Delta_N$, where we denote with $\dagger$ the Moore–Penrose pseudoinverse. In this framework, due to the analogy with Fourier, the matrices $\Phi^\Delta_M$ and $\Phi^\Delta_N$ play the role of bases for the functional spaces. For this reason, we will refer to them both as spectral embedding and as basis.

**Linearly invariant training** Our work follows the architecture introduced in [22] but modifies it accordingly to work with triangular meshes. Two networks are trained in sequence. As a first step, we train an embedding network $E$ given a set of shape pairs, where each pair has a ground truth correspondence $\Pi_{MN}^{gt}$. This network takes as input the coordinates $X_M$ of a shape $M$ and outputs a high-dimensional embedding $\Phi^E_M$. The network is trained by considering the optimal linear transformation between the two shape embeddings, namely:

$$C_{NM}^{opt} = (\Phi^E_M)^\dagger \Pi_{MN}^{gt} \Phi^E_N,$$

which is converted to a penalty measuring how well the embedded points are aligned; since nearest-neighbor is not dif-
ferentiable, we cast this problem as follows:

\[ D = \text{dist}(\Phi^E_{M_{C_{N_{M}}}}^\text{opt}, \Phi^E_{N_{N}}) \]  
\[ S_{MN} = \text{softmax}(-D), \]  

where \( \text{dist} \) computes the matrix of Euclidean distances in the embedding space, and \( S_{MN} \) acts as a score of similarity between points. Finally, the network loss is formulated as:

\[ \text{Loss}(\Phi^E_{M_{C_{N_{M}}}}^\text{opt}, \Phi^E_{N_{N}}) = \sum \| S_{MN} X_N - \Pi_{MN}^\text{opt} X_N \|^2. \]

### 3. Questions and results

The research questions that drive our research are explicit in the following: after each, we report the related results and share derived insights. If not differently stated, we train our embedding networks to output 40-dimensional embeddings. For the matching applications, the descriptors network is trained to output 80 functions. Since [22] does not describe the relationship between basis and descriptors cardinality, this choice is motivated by an analysis that we report in the Supplementary Material.

#### 3.1. What is theoretically better in a nearly-isometric case?

To analyze properties that make an embedding suitable for matching, we trained our method on two settings: training on SCAPE [4] and testing on FAUST [6] called S+F, and the opposite that we refer to as F+S. During training, we also do data augmentation by applying rotations. For each one, we train 5 embedding networks with a different number of output dimensions (i.e., 5, 10, 20, and 40). On the test dataset, we computed the optimal matching between the 50 pairs using the ground truth correspondence as described above:

\[ C_{N_{M}}^{\text{opt}} = \Phi^M_{N_{N}} \text{softmax} \]  
\[ \Pi_{MN} = NN(\Phi^M_{C_{N_{M}}}, \Phi^E_{N_{N}}). \]

Even using the ground truth \( \Pi_{MN} \) to compute \( C_{N_{M}}^{\text{opt}} \), the bases may present some deformations which are not linearly alignable, and we measure exactly the error generated by this loss of information.

The results are reported in Figure 2. The first observation is that increasing the embedding size has a positive effect, while a saturation occurs for both \( \Phi^E \) and \( \Phi^\Delta \). Secondly, we can compress the LBO representation in significantly smaller space (i.e., with 5 learned dimensions, we have the same quality of 20 LBO basis, with 10 learned the same of 30). But finally, and most importantly, our basis always produces a better matching w.r.t. LBO basis. On the right of Table 2, we also depict the error of the two basis sets. With small bases, the \( \Phi^\Delta \) cannot distinguish between the back and front of the shape, and at higher dimensions, the error peak at the protrusions. Learned embedding distributes the error more homogeneously over the surface.

### What is special with learned bases?

We further studied what are the properties that make our embedding well-suited for shape matching. First, our embedding is always full rank, despite our basis not being orthogonal (unlike the LBO basis, which is orthogonal by construction). A second property worth investigating is smoothness. LBO eigenfunctions form the smoothest orthonormal basis as measured by the Dirichlet energy, which in the discrete case

\[ \text{Loss}(\Phi^E_{M_{C_{N_{M}}}}^\text{opt}, \Phi^E_{N_{N}}) = \sum \| S_{MN} X_N - \Pi_{MN}^\text{opt} X_N \|^2. \]
reads for a basis vector $\Phi_i$:

$$E_D(\Phi_i) = (\Phi_i/\|\Phi_i\|_2)^T \Delta(\Phi_i/\|\Phi_i\|_2).$$  \hspace{1cm} (8)

In Figure 3, we depict the embedding dimensions as functions over the surfaces and report the Dirichlet energy for each. Our embedding is significantly smoother than LBO one; this is possible since we do not impose orthogonality.

**Insights.** The observed saturation on the two basis sets is caused by different reasons. Including more $\Phi^\Delta$ basis functions introduces instability due to high frequencies, that are hardly alignable linearly. Instead, $\Phi^E$, being a learning method, is prone to overfitting if the representation is too rich. Also, to guarantee a good matching, smoothness seems a more desirable property than orthogonality. Intuitively, a linear alignment is much simpler to achieve if there are no outliers or drastic changes. Hence, imposing the linearity of the transformation between the embeddings enforces their smoothness, since the alignment of a few points naturally produces alignment for the entire shape. The need that all vertices match correctly also requires that they are distinguishable, avoiding smooth but trivial solutions (e.g., constant vectors).

### 3.2. Can we refine the learned embeddings?

One of the main advantages of a spectral embedding is its arbitrary dimensionality, which can be selected at test time. However, in [22] the basis dimension is fixed a priori. Inspired by this, we propose to incorporate ZoomOut [26] in our pipeline, considering our 40-dimensional embedding equivalent to the first 40 eigenfunctions of the LBO. Then, we apply ZoomOut by introducing LBO eigenfunctions (starting from the 41st). In the first two rows of Table 1, we initialize with the matching produced by our method the $C \in \mathbb{R}^{40 \times 40}$ ($I_{O_{ur}}$) exploiting our basis (1st row) or the first 40 LBO basis functions (2nd row). Then, we applied ZoomOut to increase the set with higher frequencies.

We observe that our basis can replace the low frequencies of the LBO. In the 3rd and 4th rows, we repeated the experiment initializing with the matching provided by $GFM_{DN}$ ($I_{GFM_{DN}}$), improving a matching optimized for another basis. ZoomOut shows instability for some dimensionality while, remarkably, our embedding seems more stable.

We depict in Figure 4 the different effects of ZoomOut on the linear transformation. Starting with $\Pi_{MN}$, we obtain the $40 \times 40$ transformation using either our embedding ($C_{init}^\Phi$) or the first 40 LBO basis functions ($C_{init}^{\Phi^\Delta}$). Then, we applied ZoomOut obtaining $C_{ZO}^{\Phi^E}$ and $C_{ZO}^{\Phi^\Delta}$ of dimension $60 \times 60$. We observe that $C_{ZO}^{\Phi^E}$ presents a block structure: the top-left block highlights how $\Phi^E$ have been recombined to match higher frequencies of $\Phi^\Delta$; the bottom-left block is almost 0. On the right, we report the variations in the upper left $20 \times 20$; the $\Phi^\Delta$ one is left unchanged, while our basis shows flexibility thanks to its non-orthogonality.

**Insights.** This experiment shows that learned embeddings are a good initialization for refinement techniques and improve existing learning pipelines. The structure of the final $C$ matrix also reveals the reason behind this. In the obtained matrix using only $\Phi^\Delta$, the top right and bottom left rectangles are almost empty, and the only significant interaction appears in the added frequencies. Considering the $C$ produced with $\Phi^E$, this interaction is almost unchanged. However, the top right block shows the interaction between $\Phi^E$ and higher frequencies. This opens to a new perspective in the field of shape matching: instead of seeking competition between representations, incorporating the two seems a promising direction. Note that our merging is naive, and
Table 1. Matching results using ZoomOut as further refinement of an input matching. Replacing the first 40 $\Phi^\Delta$ with $\Phi^E$, we obtain a better matching even for the initialization provided by $GFM_{DN}$.

Figure 4. ZoomOut applied to $\Phi^E$ and $\Phi^\Delta$. On the left, the two transformation matrices have been initialized by the same correspondence $\Pi_{MN}$. Then, in the middle, we increased their size by 20 dimensions, using in both cases the LBO basis from 41 to 60. We show the absolute difference between the initialization and the new upper right part of the matrix on the right.

3.3. How do learned embeddings generalize to challenging settings?

Many previous works that we will discuss in Section 5 validate their performance on the S+F and F+S settings. For the sake of completeness, we also report their results in the Supplementary Material. Here we focus on more exciting challenges: non-isometric shapes, in which the considered objects may have a significant difference in their surface metric; and partiality, where at least the source shape has a missing component. The community has pushed the LBO to tackle both cases, working with ad-hoc refinement techniques, regularization, or new descriptors. We show how flexibility and resilience of the learned embedding approach help in these contexts.

Non-isometric cases We tested our capability of doing shape matching using the learned embedding and features on SMAL dataset [55]. This dataset comprises different animal groups (big cats, canines, ovine, bovine, and hippos), presenting significant non-isometries. The training set is composed of 25 shapes of different species and poses, that is definitely small if compared to the ones generally adopted. The test set is composed of 300 pairs of 25 shapes unseen at training time. LBO eigenfunctions instability to metric variation is well-known, and it is hard to overcome. The results are reported in the table of Figure 5.

Partiality. To test the resilience of our embedding to missing geometry, we used $\Phi^E$ trained on the full shapes of SCAPE (i.e., without any data augmentation with partial shapes). For the 50 FAUST test pairs, we picked a point on the right foot, and then we removed all the surface at a certain geodesic distance from it (at 0.4 and 0.8 respectively, where 0.0 means that we used the full shapes). We argue that this dataset is more interesting from an applicative perspective compared to SHREC16 [11]: the latter has been designed to analyze some specific theoretical properties of LBO basis. Instead, we are not aware of any other work that tackled the problem of missing limbs in human shape matching, also providing a more fair representation of different human beings. Results are reported in Table 2; in the first two rows, we consider the optimal matching, while in the last two, we considered the one with the features obtained by $\mathcal{F}$. We report qualitative examples in the Supplementary Materials. For the sake of completeness, we also report a qualitative case on [11] in Figure 6. The second column is obtained with our method. The third one uses $GFM_{DN}$, while the last one uses LBO basis and our descriptors.

Insights. In the non-isometric case, we interestingly observe that the learned embedding methods for surfaces (Our
3.4. Can the learned embedding be used for function representation?

|                          | 0.0 | 0.4 | 0.8 |
|--------------------------|-----|-----|-----|
| $\Phi^E C_{\text{opt}}$ | 1.58 | 6.8 | 10.5 |
| $\Phi^D C_{\text{opt}}$ | 2.05 | 7.3 | 10.6 |
| Our $GFM_{DN}$           | 3.80 | 8.4 | 12.0 |
| $GFM_{DN}$               | 2.90 | 8.2 | 11.9 |

Figure 5. On the left, a quantitative comparison on non-isometric shape matching on SMAL dataset. On the right, a pair of animals from TOSCA [8] dataset, showing our generalization capability.

and Universal) outperform any other method based on LBO ($GFM_{DN}, GFM_{KP},$ Deep Shells). However, the worst method is [22], which is based on point clouds. The lack of surface information would require a significant amount of data to infer it. In the partial setting, the learned embedding and LBO eigenfunctions performance are similar when used with descriptors. At the same time, we notice that the representation capacity of $\Phi^E C_{\text{opt}}$ still outperforms $\Phi^D C_{\text{opt}}$, showing the robustness of the data-driven embedding; as discussed before, this means that the embedding better preserves the structure (i.e., the smoothness) of the representation. Also, while the embeddings deteriorate at almost the same velocity, the descriptors seem to work with a similar relation w.r.t. the best possible matching, showing (together with the case reported in Figure 6) that the underlying representation is the relevant aspect.

### 3.4. Can the learned embedding be used for function representation?

|                           | F+S | S+F |
|----------------------------|-----|-----|
| $L_{\text{LBO}}$          | 2.86 | 2.26 |
| $L_{\text{LBO+D}}$        | 2.88 | 2.01 |
| $L_{\text{Coord}}$        | 47.58 | 45.90 |
| $L_{\text{Coord}}$        | 2.96 | 3.06 |

Table 3. Results in coordinates representation.

Until now, the functional perspective has been carried only by the functional maps paradigm. How learned basis can represent functions was left unexplored by [22], but we think it is an important aspect.

We focus our attention on coordinates as a triplet of functions defined over the vertices. The interest in how the basis can represent such functions is multiple. Coordinates evolve smoothly across a shape, particularly in the human domain; hence, they are good examples of a continuous global function. Representing the coordinates of each point can be visualized, and it gives an interpretation of the loss of information. Finally, and importantly, reconstructing the coordinates means that the basis can represent the points on the surface. In this case, we considered three different trainings of $\Phi^E$. In the first one, we used the $L_{\text{LBO}}$ defined in Eq. 4. Then, we also considered two setups that include useful information to represent functions. The first one is directly trained to represent the coordinates, with the loss:

$$L_{\text{Coord}}(\Phi^E_M) = \sum \| \Phi^E_M \Phi^E_M X_M - X_M \|^2_2.$$  \hspace{1cm} (9)

Then, we also trained a more general one, imposing two useful properties for representing smooth functions:

$$L_{\text{LBO}}(\Phi^E_M) = \sum \| \Phi^E_M \|^1_1$$  \hspace{1cm} (10)

$$L_{\text{Coord}}(\Phi^E_M) = \sum \| \text{diag}(\Phi^E_M \Phi^E_M \Phi^E_M A \Phi^E_M) \|^2_2$$  \hspace{1cm} (11)

$$L_{\text{LBO+D}} = L_{\text{LBO}} + L_{\text{Coord}}.$$  \hspace{1cm} (12)
training requires \( \sim 3.31 \text{GHz} \) and an RTX 2080 Ti GPU card. Embedding

can be conveniently used for shape matching. The main idea is to realize the geodesic metric of a given 3D object as the Euclidean (also called restricted) metric of some high-dimensional space. Such approaches are variational since, in general, a solution does not exist even for simple cases [19]. More recently, several works have proposed extensions of the Laplacian eigenbasis to deal with difficult settings. In [28], it was proposed to augment the LBO set with a locally supported basis, defined only in a specific region to increase its local representation capability; [24] proposes to add extrinsic information by appending the orthonormalized version of the three coordinates vectors; [18] suggested finding a basis by jointly diagonalizing the Laplacians of the input pair of shapes. Interestingly, all the previous methods start from the standard LBO basis to augment or modify its representation.

Another popular approach in shape matching is finding a set of features (also called descriptors) that are able to identify each point uniquely, such as the Global Point Signature [48], Wave Kernel Signature [5], Heat Kernel Signature [49], AWFT [27], SHOT [51] and it robust version proposed in [29]. However, such features alone are generally insufficient to provide accurate point-to-point matching, while they are widely used in functional maps pipelines.

Functional maps [32, 33] provide the paradigm considered in this paper. They suggest to move the correspondence problem to a functional domain. It has been extended to take into account partiality [43], to provide precise matching within triangle faces [14], and improved with several regularizations [31, 40] or iterative refinement [26, 34, 39].

Learning-based pipelines Data-driven approaches have also been employed to learn for shape correspondence. Several such pipelines only address the rigid case [17, 35, 47], while 3DCoded [16] addresses the non-rigid setting at the cost of an expensive optimization at test time. In the latter setting, an entire line of research has been devoted to learning the descriptors used by functional maps. This has been done using random forests [42], and more recently via deep learning models such as FMNet [20]. The general idea is to take some input feature as input, and to transform it further to obtain better features for matching. Recently, [12] proposed to learn the features directly from the point cloud coordinates while keeping the LBO basis. The main inspiration of our work is [22], which proposes to also learn the basis itself; the method only works on 3D point clouds, since it uses the PointNet architecture [37] without exploiting connectivity that might be given as input, nor investigating the properties and applicability of the learned basis. Among the other contributions, our work also aims to fill this gap.

6. Limitations and Conclusions

Limitations. While our work pushes further our understanding of learned embeddings, several questions remain...
unanswered. Investigating other classes of transformation different from the linear one is still entirely open. Investigating the representation of different functions (e.g., segmentations) may also highlight other properties to impose during the training. Other challenges touched in our work are the presence of clutter and partial-to-partial matching, which may require more sophisticated strategies.

Conclusions. Why should one learn a functional basis, then? Our work shows a series of promising properties, and we wish to motivate the community to foster the research in this direction. Pointing out that smoothness seems a superior property to orthogonality sheds new light on the study of shape embedding for matching, which for many years focused in a contrary direction [28, 38]. Combining learned and axiomatic embeddings opens to mixed methodologies, which could be interesting to analyze already at training time. Generalization to challenging scenarios is appealing and highlights the importance of proper embedding instead of refining existing ones. Finally, we have focused on the functional representation capability of such embeddings that naturally arise from matching training. This discovery opens to look for other properties to impose and a broad set of different applications.

References

[1] Scan the world project. [Online; accessed 16-November-2021]. 2, 16
[2] Abubakar Abid, Ada Poon, and James Zou. Linear regression with shuffled labels. arXiv preprint arXiv:1705.01342, 2017. 14
[3] Y. Aflalo, H. Brezis, and R. Kimmel. On the optimality of shape and data representation in the spectral domain. SIAM Journal on Imaging Sciences, 8(2):1141–1160, 2015. 1
[4] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In ACM SIGGRAPH 2005 Papers, pages 408–416, 2005. 4
[5] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, pages 1626–1633. IEEE, 2011. 8
[6] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Piscataway, NJ, USA, June 2014. IEEE. 4
[7] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. Proceedings of the National Academy of Sciences, 103(5):1168–1172, 2006. 1, 8
[8] Alexander M Bronstein, Michael M Bronstein, and Ron Kimmel. Numerical geometry of non-rigid shapes. Springer Science & Business Media, 2008. 2, 7, 14, 16, 17
[9] Michael M Bronstein, Alexander M Bronstein, Ron Kimmel, and Irad Yavneh. Multigrid multidimensional scaling. Numerical linear algebra with applications, 13(2-3):149–171, 2006. 8
[10] Yoni Choukroun, A. Shtern, A. Bronstein, and R Kimmel. Hamiltonian operator for spectral shape analysis. arXiv:1611.01990, 2016. 1
[11] Luca Cosmo, Emanuele Rodola, Michael M Bronstein, Andrea Torsello, Daniel Cremers, and Y Sahlihiouglo. Shrec’16: Partial matching of deformable shapes. Proc. 3DOR, 2(9):12, 2016. 6, 7
[12] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 4, 7, 8
[13] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 10491–10502. Curran Associates, Inc., 2020. 4, 7
[14] Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. Computer Graphics Forum, 36(5):165–174, 2017. 8
[15] David A Field. Laplacian smoothing and delaunay triangulations. Communications in applied numerical methods, 4(6):709–712, 1988. 1
[16] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 230–246, 2018. 8
[17] Xiaoshui Huang, Guofeng Mei, and Jian Zhang. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. 8
[18] Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In Computer Graphics Forum, volume 32, pages 439–448. Wiley Online Library, 2013. 8
[19] Nathan Linial. Finite metric spaces–combinatorics, geometry and algorithms. arXiv preprint math/0304466, 2003. 8
[20] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In Proceedings of the IEEE International Conference on Computer Vision, pages 5659–5667, 2017. 8
[21] O. Litany, E. Rodol`a, A. M. Bronstein, and M. M. Bronstein. Fully spectral partial shape matching. Computer Graphics Forum, 36(2):247–258, 2017. 1
[22] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, Advances in Neural
Simone Melzi,Sparse representation of step functions on manifolds. *Computers & Graphics*, 82:117–128, 2019. 1

Simone Melzi, Riccardo Marin, Pietro Musoni, Filippo Bardon, Marco Tarini, and Umberto Castellani. Intrinsic/extrinsic embedding for functional remeshing of 3d shapes. *Computers & Graphics*, 88:1 – 12, 2020. 1, 8

S. Melzi, R. Marin, E. Rodolà, U. Castellani, J. Ren, A. Poulenard, P. Wonka, and M. Ovsjanikov. Matching Humans with Different Connectivity. In Silvia Biasotti, Guillaume Lavoüe, and Remco Veltkamp, editors, *Eurographics Workshop on 3D Object Retrieval*. The Eurographics Association, 2019. 16

Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics (TOG)*, 38(6):155, 2019. 2, 5, 8

Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael Bronstein. Shape analysis with anisotropic windowed fourier transform. In *International Conference on 3D Vision (3DV)*, 2016. 8

Simone Melzi, Emanuele Rodolà, Umberto Castellani, and Michael Bronstein. Localized manifold harmonics for spectral shape analysis. *Computer Graphics Forum*, 37(6):20–34, 2018. 1, 8, 9

Simone Melzi, Riccardo Spezialletti, Federico Tombari, Michael M. Bronstein, Luigi Di Stefano, and Emanuele Rodola. Gframes: Gradient-based local reference frame for 3d shape matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4629–4638. IEEE, June 2019. 8

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, New York, NY, 2003. 3

Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. In *Computer Graphics Forum*, volume 36, pages 259–267. Wiley Online Library, 2017. 8

Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Transactions on Graphics (TOG)*, 31(4):30:1–30:11, 2012. 1, 8

Maks Ovsjanikov, Etienne Corman, Michael Bronstein, Emanuele Rodolà, Mirela Ben-Chen, Leonidas Guibas, Frederic Chazal, and Alex Bronstein. Computing and processing correspondences with functional maps. In *SIGGRAPH 2017 Courses*. 2017. 8

G Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *CVPR*, 2021. 8

G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C. Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7193–7203, 2020. 8

Ulrich Pinkall and Konrad Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics*, 2(1):15–36, 1993. 3

Charles R Qi, Hao Su, Kaichen Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 8

Arianna Rampini, Irene Tallini, Maks Ovsjanikov, Alex M Bronstein, and Emanuele Rodolà. Correspondence-free region localization for partial shape similarity via hamiltonian spectrum alignment. In *International Conference on 3D Vision (3DV)*, 2019. 9

Jing Ren, Simone Melzi, Maks Ovsjanikov, and Peter Wonka. Maptree: Recovering multiple solutions in the space of maps. *ACM Trans. Graph.*, 39(6), Nov. 2020. 8

Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (TOG)*, 37(6):1–16, 2018. 8

Martin Reuter, Silvia Biasotti, Daniela Giorgi, Giuseppe Patané, and Michela Spagnuolo. Discrete laplace–beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009. 1

Emanuele Rodolà, Samuel Bulò, Thomas Windheuser, Matthias Vestner, and Daniel Cremers. Dense non-rigid shape correspondence using random forests. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4184, 2014. 8

Emanuele Rodolà, Luca Cosmo, Michael M. Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*, volume 36, pages 222–236. Wiley Online Library, 2017. 8

E. Rodolà, M. Moeller, and D. Cremers. Regularized pointwise map recovery from functional correspondence. *Computer Graphics Forum*, 36(8):700–711, 2017. 1

Raif M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233. Eurographics Association, 2007. 8

Yusuf Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020. 8

Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srinivasan, Simon Lucey, and Howie Choset. Pcmnet: Point cloud registration network using pointnet encoding. *arXiv preprint arXiv:1908.07906*, 2019. 8

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusion is all you need for learning on surfaces. *arXiv preprint arXiv:2012.00888*, 2020. 2, 4

Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28(5):1383–1392, 2009. 8

Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois Goulette, and Leonidas J.
Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[51] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *Proc. ECCV*, pages 356–369. Springer, 2010.

[52] Warren S Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[53] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer graphics forum*, volume 30, pages 1681–1707. Wiley Online Library, 2011.

[54] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[55] Silvia Zuffi, Angjoo Kanazawa, David Jacobs, and Michael J. Black. 3D menagerie: Modeling the 3D shape and pose of animals. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
Abstract

Here we list additional material as support to the main manuscript. We provide more details about the training processes and hyperparameters choice. We also offer an analysis on the relationship between the cardinality of basis and descriptors, quantitative results in isometric cases, and qualitative results both on humans and non-isometric animals. We describe more in detail the partiality setting, and finally, we include some more examples of ZoomOut process.

1. Training Details

In all cases, we composed the Diffusion networks with 12 Diffusion Blocks. In the diffusion process, we used 128 eigenvalues and eigenvectors. We used an ADAM optimizer with an initial learning rate of $10^{-4}$ and a Cosine Annealing Warm Restarts scheduler. Due to computational reasons, we used batch size 2 and accumulated the gradient for 4 batches to simulate a batch size of 8. The maximum number of epochs has been set to 1200. We will release our data and method implementation.

1.1. Loss stability

In Figure 8 we report the behavior of training loss when we train our basis, our descriptor, and the universal embedding in the two settings of S+F and F+S. Both our networks show stability. We attribute the peaks to the Cosine Annealing scheduler restart, which increases the learning rate to escape from local minima. For this reason, we think our solutions spaces are more regular than the one of the universal embedding.
Figure 8. Training losses of our embedding (top left), features (top right) and Universal embedding formulation (bottom).
2. Matching

2.1. On the quality of the embedding

For completeness, in Table 4 we report the numerical series used for the curves of Figure 2 of the main manuscript. For each of the two settings, we mark in bold the best basis set.

| #Basis | S+F | F+S |
|--------|-----|-----|
|        | OUR | LBO | OUR | LBO |
| 5      | 3.79 | 8.40 | 4.02 | 9.46 |
| 10     | 2.51 | 5.61 | 2.79 | 6.30 |
| 20     | 1.79 | 3.42 | 2.26 | 3.63 |
| 30     | 1.40 | 2.71 | 1.68 | 2.73 |
| 40     | 1.58 | 2.05 | **1.52** | 1.97 |

Table 4. For each of the two settings, we compare our bases against LBO ones at a different number of them in the optimal setting (i.e., computing the ground truth transformation $C$, as reported in the main manuscript). Each row is a different training. For each row and each setting, our bases provide better results.

2.2. Basis and Descriptors relation

Given that [22] does not provide any insight on the relation between the basis and descriptors, we investigated this relation. In Table 6, we performed an extensive analysis with different combinations of embedding dimensions and a number of features. Each entry of Table 6 corresponds to a separate training. Concerning the dimensionality of the embedding, we notice that coherently with Figure 2 of the main manuscript, an increasing number of basis functions does not always produce better results. For example, in the F+S case, 40 basis functions tend to be hardly alignable. We believe this is mainly because the SCAPE dataset contains a wide variety of poses. Hence, overfitting FAUST by using a larger representation would make it difficult to generalize at test time.

Our analysis also gives a good rule of thumb for features: a number bounded between $\times 2$ and $\times 3$ seems the one producing the most stable results. We consider this number reasonable to overcome the noise and in line with the literature [2]. This analysis sheds light on the role of the two components of the matching pipeline: the embedding should provide a structure shared across the objects to be easily alignable. The features help to identify such structure. A complex structure is not desirable, while sometimes redundant information in the features can help to identify the correct transformation.

Given such analysis, we decide to keep 40 dimensions for the basis and 80 for the features for all our experiments. This choice provides coherence in the results and good performance in all the settings.

2.3. Near-Isometric matching

Hence, we tested our capability of doing matching near-isometric shapes using the learned embedding and features. In Table 7 we reported a comparison on our three settings. For all the experiments, we kept a 40-dimensional embedding and 80. As can be seen, we perform better F+S settings where generalization is relevant to obtain good results.

In Figure 9, we show texture transfer on S+F and F+S. While all methods produce an excellent matching, we observe a misalignment of high-frequency details.

2.4. Qualitative results

Here we report some other qualitative results:

1. In Figure 10 we report another example of our matching on two statues. Notice that they do not share the same topology structure: despite this, we obtain a smooth matching.

2. In Figure 11 we report an example between two similar animals. Even in the isometric case (which is favorable to the LBO) we show better results.

3. In Figure 12 we have a significant non-isometric case; the error is localized on protrusions (ears, legs).

4. In Figure 13 we show our results on three pairs of animals from the TOSCA dataset [8]. We matched highly
Table 6. Analysis with different number of basis embedding dimensions and features.

| Basis | 5      | 10     | 20     | 30     | 40     |
|-------|--------|--------|--------|--------|--------|
|       | Basis  | S+F    | F+S    |        |        |
|       | 20     | 40     | 80     | 120    | 20     | 40     | 80     | 120    |
|       | Basis  | GT     | Our    | GFM    | DN     | Universal |
| 5     | Basis  | 34.75  | 30.80  | 18.29  | 29.81  | 15.28  | 13.14  | 10.61  | 15.56  |
| 10    | Basis  | 11.01  | 19.88  | 6.85   | 15.03  | 8.83   | 18.92  | 16.30  | 12.90  |
| 20    | Basis  | 32.07  | 9.83   | 4.67   | 10.77  | 40.62  | 12.72  | 7.68   | 12.78  |
| 30    | Basis  | 14.60  | 13.18  | 3.98   | 3.46   | 8.34   | 29.71  | 8.87   | 9.82   |
| 40    | Basis  | 7.61   | 29.66  | 3.77   | 4.63   | 12.68  | 49.85  | 8.76   | 11.49  |

Table 7. Quantitative comparison on shape matching.

| Method  | S+F | F+S |
|---------|-----|-----|
| Our     | 3.8 | 8.8 |
| GFM$_{DN}$ | 2.9 | 10.2|
| Universal | 4.5 | 12.4|

5. In Figures 14, 15, and 16 we show three more qualitative results. We would like to highlight that our methods seem to obtain better matching on protrusions (i.e., legs and arms).
6. in Figure 17 we show an example on two further different datasets. On the left, a pair from SHREC19 [25]; notice that the two shapes are different for their pose but also for their quality (i.e., the \( M \) is a real scan, while the target one is a synthetic model). On the right, a woman is matched with a gorilla from TOSCA dataset [8]. We consider this case particularly extreme since arms and legs have entirely different proportions. Even if there are some evident artifacts on the stomach, we observe an overall coherence in the obtained matching.

3. Partiality
In the partiality setting, we constructed our dataset as follows:

1. We consider the 100 FAUST shapes
2. For each shape, we take a landmark on the right foot
3. We remove all surface that is within a certain geodesic distance to the landmark. We consider two different ranges: 0.4 and 0.8.

Then, we kept the 50 pairs considered in the other experiments, substituting the source shape with the partial one. Hence, we look for a point on the complete shape for each point of the partial shape. Notice that the partial and the complete are in general of different subjects in different poses. We want to remark that partiality has not to be seen at training time. In Figures 18 and 19 we depict two qualitative examples. On the left, the full shape. On the top right, the ground-truth, our and GFM\(DN\) matchings in the 0.4 setting, with texture transfer and error. On the bottom right, the same for the 0.8 setting. The LBO bases are unstable, the results of matching between 0.4 and 0.8 settings vary significantly even in regions far from the missing part (e.g., the right arm).

4. ZoomOut
In Figures 20 and 21 we report two other cases of ZoomOut and their impact on the initialization matrix (the considered pairs are the ones of Figure 14 and 15 respectively). In the first one, we observe that the initial LBO matrix has a less diagonal behaviour than the latter. This is in general due to a non-isometric deformation that in this case, could be reasonably caused by a twist of the torso of Figure 14. The \( C_{\text{init}} \) seems to be almost preserved by the ZoomOut process. In the second case, the initialization is more diagonal, witnessing two almost isometrical shapes. In this case, the ZoomOut process can recombine a few parts of the LBO \( C_{\text{init}} \), while this impacts mainly the last dimensions. However, in both cases, our matrix shows better flexibility. Also, in all experiments, we observe the block division of our \( C_{ZO} \) matrix discussed in the main manuscript.

4.1. Step size ablation study
In the main manuscript, we reported the results by using ZoomOut and step size 5 (i.e., including 5 more basis at each ZoomOut iteration). Here, we report also results using a step of 1 (Table 8) and 2 (Table 9). We show that our basis is always a better choice to improve the matching. We noticed that by reducing the step size, the results at higher \( C \) dimensions become more unstable. We believe this is due to the more iterations required to reach the same \( C \) dimensions, introducing noise in the correspondence.
Figure 11. An isometric pair from the remeshed SMAL dataset. From the left: the source shape, the target shape with color transferred using $\Pi^\text{gt}_{\text{MAX}}$. Then, the color transfer is performed by our, $\text{GFM}_{\text{DN}}$, and the Universal Embedding Baseline. On the right, the geodesic error depicted over the surface.

Figure 12. A non-isometric pair from the remeshed SMAL dataset. From the left: the source shape, the target shape with color transferred using $\Pi^\text{gt}_{\text{MAX}}$. Then, the color transfer performed by our, $\text{GFM}_{\text{DN}}$, and the Universal Baseline. On the right, the geodesic error depicted over the surface.

Figure 13. Three pair of animals from TOSCA [8] dataset, showing our generalization capability.

Figure 14. Texture transfer for matching quality comparison (F+S setting).
Figure 15. Texture transfer for matching quality comparison (F+S setting).

Figure 16. Texture transfer for matching quality comparison (S+F setting).

Table 8. Matching results using ZoomOut with step size 1.

|       | $\Phi^e$ | $\Phi^\Delta$ |
|-------|----------|---------------|
| Init  | ZO10   | ZO20 | ZO40 | ZO60 | Init  | ZO10   | ZO20 | ZO40 | ZO60 |
| $I_{\text{Our}}$ | 3.77 | 3.45 | 3.45 | 3.67 | 3.77 | 7.78 | 7.50 | 7.37 | 7.42 |
| $I_{\text{GFMDN}}$ | 2.85 | 2.88 | 2.93 | 3.00 | 2.85 | 2.92 | 3.28 | 3.70 | 3.95 |

Table 9. Matching results using ZoomOut with step size 2.

|       | $\Phi^e$ | $\Phi^\Delta$ |
|-------|----------|---------------|
| Init  | ZO10   | ZO20 | ZO40 | ZO60 | Init  | ZO10   | ZO20 | ZO40 | ZO60 |
| $I_{\text{Our}}$ | 3.77 | 3.34 | 3.36 | 3.52 | 3.54 | 8.26 | 7.54 | 7.35 | 7.27 | 7.28 |
| $I_{\text{GFMDN}}$ | 2.85 | 2.76 | 2.83 | 3.03 | 3.03 | 10.17 | 9.44 | 9.35 | 9.43 | 9.38 |
| $I_{\text{GFMDN}}$ | 2.85 | 2.97 | 3.25 | 3.51 | 3.56 | 9.65 | 9.18 | 9.22 | 9.34 | 10.19 | 10.48 |
Figure 17. Color transfer for matching quality comparison on two out of distributions couples.

Figure 18. An example of partiality involved in our experiments. On the top, the 0.2 setting. On the bottom, the 0.4 one. Comparison between Our and $GFM_{DN}$. Near to each case we report the average geodesic error over the surface.
Figure 19. An example of partiality involved in our experiments. On the top, the 0.2 setting. On the bottom, the 0.4 one. Comparison between Our and $GFM_{DN}$. Near to each case we report the average geodesic error over the surface.

Figure 20. ZoomOut applied to $\Phi^E$ and $\Phi^\Delta$. On the left, the two transformation matrices have been initialized by the same correspondence $\Pi_{MAP}$. Then, in the middle, we increased their size of 20 dimensions, using in both cases the LBO basis from 41 to 60. On the right, we show the absolute difference between the initialization and the new upper left part of the matrix. The considered example refers to the pair shown in Figure 14.

|       | $S+F$ |       | $F+S$ |
|-------|-------|-------|-------|
|       | Init  | ZO10  | ZO20  | ZO40  | ZO60  | Init  | ZO10  | ZO20  | ZO40  | ZO60  |
| $I_{Our} \Phi^E$ | 3.77  | 3.18  | 3.16  | 3.18  | 3.21  | 8.76  | 7.33  | 7.20  | 7.30  | 7.30  |
| $I_{Our} \Phi^\Delta$ | 3.98  | 4.02  | 4.13  | 4.17  |       | 8.32  | 8.58  | 8.58  | 8.71  |       |
| $I_{GFM_{DN}} \Phi^E$ | 2.85  | 2.55  | 2.54  | 2.57  | 2.62  | 10.17 | 8.99  | 8.97  | 9.03  | 9.11  |
| $I_{GFM_{DN}} \Phi^\Delta$ | 2.96  | 3.04  | 3.12  | 3.18  |       | 9.63  | 9.72  | 10.03 | 10.08 |       |

Table 10. Matching results using ZoomOut with step size 5. This is the same table reported in the main manuscript.
Figure 21. ZoomOut applied to $\Phi^e$ and $\Phi^\Delta$. On the left, the two transformation matrices have been initialized by the same correspondence $\Pi_{MN}$. Then, in the middle, we increased their size of 20 dimensions, using in both cases the LBO basis from 41 to 60. On the right, we show the absolute difference between the initialization and the new upper left part of the matrix. The considered example refers to the pair shown in Figure 15.