Self-Supervised Learning for Multimodal Non-Rigid 3D Shape Matching

Dongliang Cao Florian Bernard
University of Bonn

Abstract

The matching of 3D shapes has been extensively studied for shapes represented as surface meshes, as well as for shapes represented as point clouds. While point clouds are a common representation of raw real-world 3D data (e.g. from laser scanners), meshes encode rich and expressive topological information, but their creation typically requires some form of (often manual) curation. In turn, methods that purely rely on point clouds are unable to meet the matching quality of mesh-based methods that utilise the additional topological structure. In this work we close this gap by introducing a self-supervised multimodal learning strategy that combines mesh-based functional map regularisation with a contrastive loss that couples mesh and point cloud data. Our shape matching approach allows to obtain intramodal correspondences for triangle meshes, complete point clouds, and partially observed point clouds, as well as correspondences across these data modalities. We demonstrate that our method achieves state-of-the-art results on several challenging benchmark datasets even in comparison to recent supervised methods, and that our method reaches previously unseen cross-dataset generalisation ability. Our code is available at `https://github.com/dongliangcao/Self-Supervised-Multimodal-Shape-Matching`.

1. Introduction

Matching 3D shapes, i.e. finding correspondences between their parts, is a fundamental problem in computer vision and computer graphics that has a wide range of applications [11, 16, 31]. Even though it has been studied for decades [56, 57], the non-rigid shape matching problem remains highly challenging. One often faces a large variability in terms of shape deformations, or input data with severe noise and topological changes.

With the recent success of deep learning, many learning-based approaches were proposed for 3D shape matching [17, 19, 28, 33]. While recent approaches demonstrate near-perfect matching accuracy without requiring ground truth annotations [8, 17], they are limited to 3D shapes represented as triangle meshes and strongly rely on clean data.

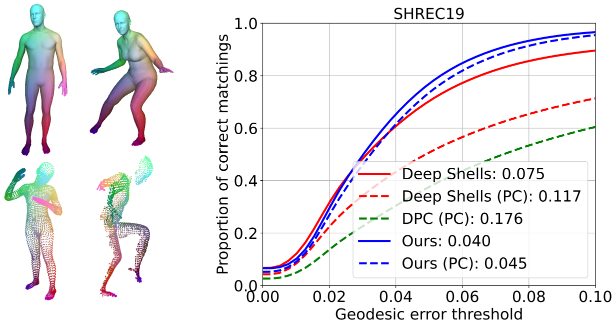

Figure 1. **Left:** Our method obtains accurate correspondences for triangle meshes, point clouds and even partially observed point clouds. **Right:** Proportion of correct keypoints (PCK) curves and mean geodesic errors (scores in the legend) on the SHREC'19 dataset [34] for meshes (solid lines) and point clouds (dashed lines). Existing point cloud matching methods (DPC [26], green line), or mesh-based methods applied to point clouds (Deep Shells [17], red dashed line) are unable to meet the matching performance of mesh-based methods (solid lines). In contrast, our method is multimodal and can process both meshes and point clouds, while enabling accurate shape matching with comparable performance for both modalities (blue lines).

Since point clouds are a common representation for real-world 3D data, many unsupervised learning approaches were specifically designed for point cloud matching [20, 26, 63]. These methods are often based on learning per-point features, so that point-wise correspondences are obtained by comparing feature similarities. The learned features were shown to be robust under large shape deformations and severe noise. However, although point clouds commonly represent samples of a surface, respective topological relations are not explicitly available and thus cannot effectively be used during training. In turn, existing point cloud correspondence methods are unable to meet the matching performance of mesh-based methods, as can be seen in Fig. 1. Moreover, when applying state-of-the-art unsupervised methods designed for meshes (e.g. Deep Shells [17]) to point clouds, one can observe a significant drop in matching performance.

In this work, we propose a self-supervised learning framework to address these shortcomings. Our method uses

Methods	Unsup.	Mesh	Point Cloud	FM-based	Partiality	Robustness	w.o. Refinement	Required train data**
FMNet [28]	✗	✓	✗*	✓	✗	✗	✓	Small
GeomFMaps [13]	✗	✓	✗*	✓	✗	✗	✓	Small
DiffFMaps [32]	✗	✓	✓	✓	✗	✓	✓	Moderate
DPFM [2]	✗	✓	✗*	✓	✓	✗	✓	Small
3D-CODED [19]	✗	✓	✓	✗	✓	✓	✗	Large
IFMatch [55]	✗	✓	✓	✗	✓	✓	✗	Moderate
UnsupFMNet [21]	✓	✓	✗*	✓	✗	✗	✓	Small
SURFMNet [46,51]	✓	✓	✗*	✓	✗	✗	✓	Small
Deep Shells [17]	✓	✓	✗*	✓	✗	✗	✗	Small
ConsistFMaps [8]	✓	✓	✗*	✓	✓	✗	✓	Small
CorrNet3D [63]	✓	✓	✓	✗	✗	✓	✓	Large
DPC [26]	✓	✓	✓	✗	✗	✓	✓	Moderate
Ours	✓	✓	✓	✓	✓	✓	✓	Small

Table 1. **Method comparison.** Our method is the first learning-based approach that combines a unique set of desirable properties.
* Methods are originally designed for meshes, directly applying them to point clouds leads to a large performance drop.
** Categorisation according to the amount of training data: *Small (<1000)*, *Moderate (≈5,000)* and *Large (>10,000)*.

a combination of triangle meshes and point clouds (extracted from the meshes) for training. We first utilise the structural properties of functional maps for triangle meshes as strong unsupervised regularisation. At the same time, we introduce a self-supervised contrastive loss between triangle meshes and corresponding point clouds, enabling the learning of consistent feature representations for both modalities. With that, our method does not require to compute functional maps for point clouds at inference time, but directly predicts correspondences based on feature similarity comparison. Overall, our method is the first learning-based approach that combines a unique set of desirable properties, i.e. it can be trained without ground-truth correspondence annotations, is designed for both triangle meshes and point clouds (throughout this paper we refer to this as *multimodal*), is robust against noise, allows for partial shape matching, and requires only a small amount of training data, see Tab. 1. In summary, our main contributions are:

- For the first time we enable *multimodal* non-rigid 3D shape matching under a simple yet efficient *self-supervised learning* framework.
- Our method achieves accurate matchings for triangle meshes based on *functional map regularisation*, while ensuring matching robustness for less structured point cloud data through *deep feature similarity*.
- Our method outperforms *state-of-the-art* unsupervised and even supervised methods on several challenging 3D shape matching benchmark datasets and shows previously unseen *cross-dataset generalisation ability*.
- We extend the SURREAL dataset [58] by SURREAL-PV that exhibits disconnected components in partial views as they occur in 3D scanning scenarios.

2. Related work

Shape matching is a long-standing problem in computer vision and graphics. In the following, we will focus on reviewing those methods that are most relevant to our work. A more comprehensive overview can be found in [56, 57].

2.1. Shape matching for triangle meshes

Triangle meshes are the most common data representation for 3D shapes in computer graphics, thus a large number of matching methods are specifically designed for them [4, 14, 22, 45, 47, 61]. Notably, the functional map framework [38] is one of the most dominant pipelines in this area and was extended in numerous follow-up works, e.g., in terms of improving the matching accuracy and robustness [35, 42, 59], or by considering non-isometric shape matching [15, 37, 43], multi-shape matching [18, 23, 24], and partial shape matching [29, 44]. Meanwhile, with the success of deep learning, many learning-based methods were introduced with the aim to learn improved features compared to handcrafted feature descriptors, such as HKS [7], WKS [3] or SHOT [49]. FMNet [28] was proposed to learn a vertex-wise non-linear transformation of SHOT descriptors [49], which is trained in a supervised manner. Later works [21, 46] enable the unsupervised training of FMNet, and point-based networks [41, 60] were introduced to improve the matching performance [13, 51]. To enable both local and global information propagation, DiffusionNet [52] introduced a diffusion layer, which was shown to achieve state-of-the-art performance for 3D shape matching [2, 8, 12, 30]. Even though deep functional map methods were shown to lead to state-of-the-art results for shapes represented as meshes, they are not directly applicable for point cloud matching, since the latter only admit an inaccurate estimation of the Laplace-Beltrami operator (LBO) eigenfunctions [32]. To overcome this limitation, our method proposes self-supervised deep feature learning for point clouds without relying on the functional map framework during inference.

2.2. Shape matching for point clouds

Point clouds are a commonly used 3D data representation in various real-world applications, e.g., robotics, autonomous driving, AR/VR, etc. Point cloud matching can be roughly classified into two categories: rigid point cloud registration and non-rigid point cloud matching [56]. In this work, we focus on reviewing the learning-based methods for non-rigid point cloud matching. 3D-CODED [19] was proposed to learn a deformation field from a template shape to the given shape to establish correspondences between them. IFMatch [55] extends vertex-based shape deformation to shape volume deformation to improve the matching robustness. Instead of choosing a template shape beforehand, some works [20, 63] attempted to learn a pairwise deformation field in an unsupervised manner by shape reconstruction and cycle consistency. However, the introduced deformation network requires a large amount of data to train and is category-specific, which limits the generalisation capability [26]. In contrast, our method requires much less training data and shows previously unseen cross-dataset generalisation ability.

To incorporate the functional map framework into point cloud matching, DiffFMaps [32] attempted to learn basis functions with ground truth correspondence supervision to replace the LBO eigenfunctions [40]. More recently, DPC [26] replaced the deformation network by using the shape coordinates themselves to reconstruct shape. Nevertheless, there is still a huge performance gap between unsupervised mesh-based shape matching and point cloud matching [55]. In this work, we propose a multimodal learning approach to bridge the gap.

3. Functional maps in a nutshell

Our approach is based on the functional map framework, which we recap in the following. Unlike finding point-to-point correspondences, which often leads to NP-hard combinatorial optimisation problems [27], the functional map framework encodes the correspondence relationship into a small matrix that can be efficiently solved [38].

Basic pipeline. Given is a pair of 3D shapes $\mathcal{X}$ and $\mathcal{Y}$ with n_x and n_y vertices, respectively. The functional map framework uses truncated basis functions, i.e. the first k LBO eigenfunctions [40] $\Phi_x \in \mathbb{R}^{n_x \times k}, \Phi_y \in \mathbb{R}^{n_y \times k}$, to approximate given features defined on each shape $\mathcal{F}_x \in \mathbb{R}^{n_x \times c}, \mathcal{F}_y \in \mathbb{R}^{n_y \times c}$. To this end, the corresponding coefficients $A = \Phi_x^{\dagger}\mathcal{F}_x \in \mathbb{R}^{k \times c}, B = \Phi_y^{\dagger}\mathcal{F}_y \in \mathbb{R}^{k \times c}$ are computed for each shape, respectively. The functional map $C_{xy} \in \mathbb{R}^{k \times k}$ can be computed by solving the continuous optimisation problem

$$C_{xy} = \mathrm{argmin}_C \; E_{\mathrm{data}}\left(C\right) + \lambda E_{\mathrm{reg}}\left(C\right), \tag{1}$$

where minimising $E_{\mathrm{data}} = \left\|CA - B\right\|^2$ imposes descriptor preservation, whereas minimising the regularisation term E_{reg} imposes certain structural properties [38], see Sec. 4.4. From the optimal C_{xy}, the point map $\Pi_{yx} \in \{0,1\}^{n_y \times n_x}$ can be recovered based on the relationship $\Phi_y C_{xy} \approx \Pi_{yx}\Phi_x$, e.g. either by nearest neighbour search or by other post-processing techniques [35, 39, 59].

4. Non-rigid 3D shape matching

The whole framework of our approach is depicted in Fig. 2. Our approach aims to train a feature extraction network that can be used to extract expressive features for multimodal shape matching. To this end, we pursue a self-supervised training strategy using multimodal data that comprises meshes and point clouds extracted from these meshes.

To be precise, our multimodal training strategy utilises the shapes $\mathcal{X}$ and $\mathcal{Y}$ represented as triangle meshes, together with corresponding point clouds $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ that we obtain by discarding the mesh connectivity information and perturbing the vertex coordinates. The same feature extraction network is used to process both triangle meshes and point clouds, resulting in pointwise features in both cases. Analogous to previous deep functional map methods [8, 46, 51], a non-trainable FM solver is used to compute bidirectional functional maps C_{xy}, C_{yx} based on the features extracted from the triangle meshes. At the same time, the features extracted from the point clouds are used to construct a soft correspondence matrix $\hat{\Pi}_{xy}$ via feature similarity measurement. To enable the self-supervised training of our feature extractor, we use functional map regularisation. In addition, by using a contrastive loss we enforce that the features from the triangle meshes and the point clouds are similar. At inference time, the functional map framework (see Sec. 3) is used for finding correspondences for 3D shapes represented as triangle meshes, while the correspondences for point clouds (or between triangle meshes and point clouds) are computed based on deep feature similarity, thereby avoiding the problem of point clouds only admitting an inaccurate estimation of the LBO eigenfunctions [6, 9, 53]. In the following, we explain the individual components of our method in detail.

4.1. Feature extractor

The feature extractor aims to extract features of both triangle meshes and point clouds that are robust to shape deformations and to the sampling. To this end, we use the DiffusionNet architecture [52] throughout our work, similar to other recent methods [2, 8]. DiffusionNet is based on

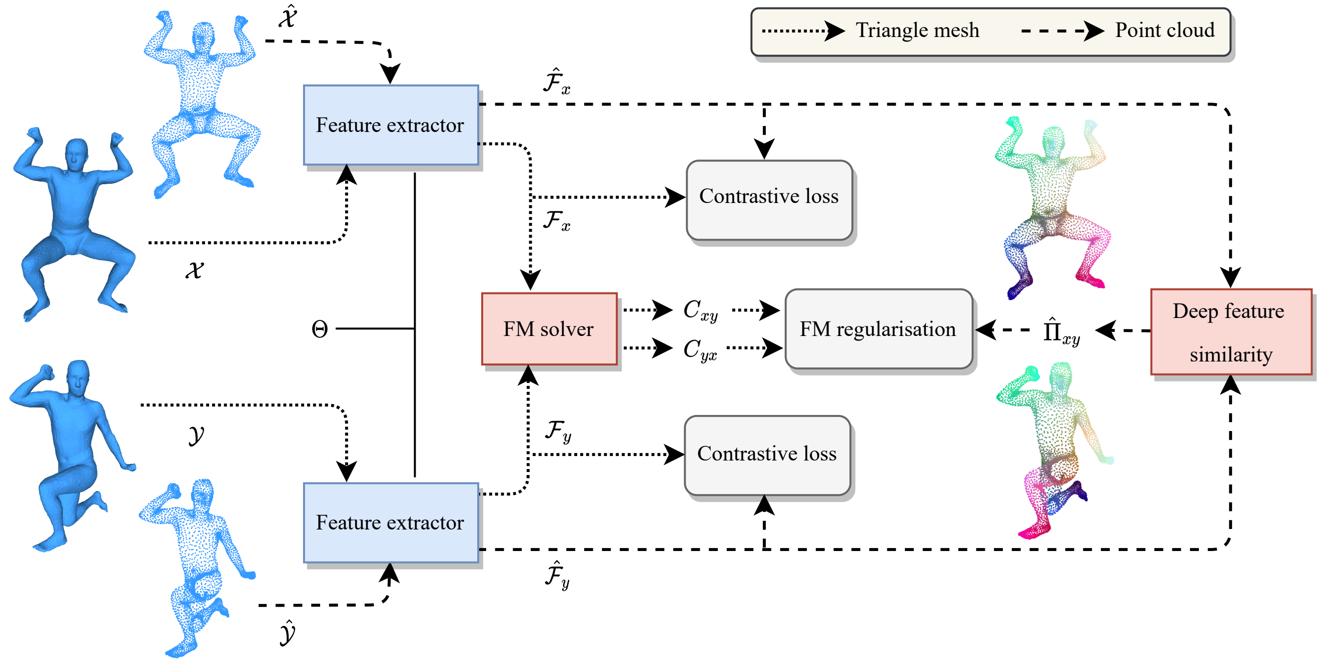

Figure 2. **Method overview.** During training a Siamese feature extraction network with shared weights Θ learns to extract mesh features $\mathcal{F}_x, \mathcal{F}_y$ for input meshes $\mathcal{X}, \mathcal{Y}$, as well as point cloud features $\hat{\mathcal{F}}_x, \hat{\mathcal{F}}_y$ for corresponding point clouds $\hat{\mathcal{X}}, \hat{\mathcal{Y}}$. The mesh features $\mathcal{F}_x, \mathcal{F}_y$ are then used to compute bidirectional functional maps C_{xy}, C_{yx} using the parameter-free FM solver (red). In contrast, the features from point clouds $\hat{\mathcal{F}}_x, \hat{\mathcal{F}}_y$ are used to construct a soft correspondence matrix $\hat{\Pi}_{xy}$ based on the feature similarity (red). The FM regularisation and contrastive loss together form our overall loss function (gray). The feature extractor (blue) is the only trainable part in our method.

an intrinsic surface diffusion process [52] and leads to the state-of-the-art performance in the context of shape matching [2, 8, 12, 30]. Moreover, DiffusionNet allows to extract features from both meshes and point clouds.

Following [8], our feature extractor is used in a Siamese way, i.e. the same network with shared wights Θ is applied for both source shapes $\mathcal{X}, \hat{\mathcal{X}}$ and target shapes $\mathcal{Y}, \hat{\mathcal{Y}}$.

4.2. Functional map solver

The goal of the functional map solver (FM solver) is to compute the bidirectional functional maps C_{xy}, C_{yx} based on the extracted features $\mathcal{F}_x, \mathcal{F}_y$. The basic pipeline is explained in Sec. 3. Analogous to previous methods [2, 8], we use a regularised functional map solver [42] to improve the robustness when computing the functional map. To this end, the regularisation term E_{reg} in Eq. (1) can be expressed in the form

$$E_{\mathrm{reg}} = \sum_{ij} C_{ij}^2 M_{ij}, \tag{2}$$

where M_{ij} is the resolvent mask that can be viewed as an extension of Laplacian commutativity, see [42] for details.

4.3. Deep feature similarity

The goal of the deep feature similarity module is to predict a correspondence matrix $\hat{\Pi}_{xy}$ to explicitly indicate the correspondences between given input point clouds $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ with n_x and n_y points, respectively. Theoretically, $\hat{\Pi}_{xy}$ should be a (partial) permutation matrix, i.e.

$$\left\{\Pi \in \{0,1\}^{n_x \times n_y} : \Pi \mathbf{1}_{n_y} \leq \mathbf{1}_{n_x}, \mathbf{1}_{n_x}^\top \Pi \leq \mathbf{1}_{n_y}^\top \right\}, \tag{3}$$

where the element at position (i, j) of $\hat{\Pi}_{xy}$ indicates whether the i-th point in $\hat{\mathcal{X}}$ corresponds to the j-th point in $\hat{\mathcal{Y}}$. However, the construction of such a binary matrix is non-differentiable. To this end, a soft correspondence matrix is used to approximate the binary constraints (3) in practice [8, 16, 48, 63]. The key idea to construct the soft correspondence matrix $\hat{\Pi}_{xy}$ is based on the similarity measurement between features $\mathcal{F}_x$ and $\mathcal{F}_y$ defined on each shape. The construction process can be expressed in the form

$$\hat{\Pi}_{xy} = \mathrm{Corr}\left(\langle \mathcal{F}_x, \mathcal{F}_y \rangle\right), \tag{4}$$

where $\langle \cdot, \cdot \rangle$ is the $(n_x \times n_y)$-dimensional matrix of the dot products between pairs of feature vectors and $\mathrm{Corr}(\cdot)$ is an operator to construct a soft correspondence matrix based on the similarity matrix [25, 36].

In this work, we use Sinkhorn normalisation [36, 54] to construct the soft correspondence matrix. Sinkhorn normalisation iteratively normalises rows and columns of a matrix using the softmax operator. During inference, we quantise $\hat{\Pi}_{xy}$ to a binary matrix.

4.4. Self-supervised loss

To train our feature extractor in a self-supervised manner, we combine unsupervised functional map regularisation [8, 46, 51] with self-supervised contrastive learning [62]. Our unsupervised functional map regularisation can be divided into two parts.

The first part regularises the structural properties of the predicted functional maps C_{xy}, C_{yx}. Following [46], the functional map regularisation can be expressed in the form

$$E_{\text{fmap}} = \lambda_{\text{bij}} E_{\text{bij}} + \lambda_{\text{orth}} E_{\text{orth}}. \tag{5}$$

In Eq. (5), E_{bij} is the bijectivity constraint to ensure the map from $\mathcal{X}$ through $\mathcal{Y}$ back to $\mathcal{X}$ is the identity map, E_{orth} represents the orthogonality constraint to prompt a locally area-preserving matching, see [8] for more details.

The second part regularises the predicted soft correspondence matrix $\hat{\Pi}_{xy}$ based on the relationship $\Phi_x C_{yx} \approx \hat{\Pi}_{xy} \Phi_y$. Following [8], our unsupervised loss can be expressed in the form

$$E_{\text{align}} = \|\Phi_x C_{yx} - \hat{\Pi}_{xy} \Phi_y\|_F^2. \tag{6}$$

It is important to note that our correspondence matrix $\hat{\Pi}_{xy}$ is directly predicted based on the deep feature similarity between point clouds. This is in contrast to [8], where a universe classifier is required to predict shape-to-universe point maps. In turn, our framework is more efficient and flexible, since we do not rely on the universe classifier and the knowledge of the number of universe vertices.

In addition to functional map regularisation, we further utilise a self-supervised contrastive loss to encourage similar features for corresponding points from the input mesh and the corresponding point cloud. To this end, we use the PointInfoNCE loss [62], which maximises the feature similarity between corresponding points in a given triangle mesh $\mathcal{X}$ and a given point cloud $\hat{\mathcal{X}}$, while at the same time minimising the feature similarity between other points. The loss term can be expressed in the form

$$E_{\text{nce}} = -\sum_{i=1}^{n_x} \log \frac{\exp\left(\langle \mathcal{F}_x^i, \hat{\mathcal{F}}_x^i \rangle / \tau\right)}{\sum_{j=1}^{n_x} \exp\left(\langle \mathcal{F}_x^i, \hat{\mathcal{F}}_x^j \rangle / \tau\right)}, \tag{7}$$

where τ is a scaling factor. Similarly, E_{nce} is also applied to both shape $\mathcal{Y}$ and $\hat{\mathcal{Y}}$. Finally, the overall loss for training is a weighted sum of the individual losses above, i.e.

$$E_{\text{total}} = E_{\text{fmap}} + \lambda_{\text{align}} E_{\text{align}} + \lambda_{\text{nce}} E_{\text{nce}}. \tag{8}$$

4.5. Implementation details

We implement our framework in PyTorch. We use DiffusionNet [52] with default settings for our feature extractor. In the context of the FM solver, we set $\lambda = 100$ in Eq. (1). For training, we empirically set $\lambda_{\text{bij}} = 1.0, \lambda_{\text{orth}} = 1.0$ in Eq. (5), $\tau = 0.07$ in Eq. (7), and $\lambda_{\text{align}} = 10^{-3}, \lambda_{\text{nce}} = 10$ in Eq. (8). See the supplementary document for more details.

5. Experimental results

In this section we demonstrate the advantages of our method for multimodal non-rigid 3D shape matching under different challenging scenarios.

5.1. Complete shape matching

Datasets. We evaluate our method on several standard benchmark datasets, namely FAUST [5], SCAPE [1] and SHREC'19 [34] dataset. Instead of the original datasets, we choose the more challenging remeshed versions from [13, 43]. The FAUST dataset consists of 100 shapes (10 people in 10 poses), where the evaluation is performed on the last 20 shapes. The SCAPE dataset contains 71 different poses of the same person, where the last 20 shapes are used for evaluation. The SHREC'19 dataset is a more challenging benchmark dataset due to significant variations in mesh connectivity. It comprises 44 shapes and a total of 430 evaluation pairs.

Results. The mean geodesic error is used for method evaluation. We compare our method with state-of-the-art axiomatic, supervised and unsupervised methods. To further evaluate the method robustness, we randomly add Gaussian noise to input point clouds. Note that for a fair comparison, we use DiffusionNet [52] as the feature extractor for all deep functional map methods, as it can significantly improve shape matching accuracy [52]. The results are summarised in Tab. 2. We note that directly using DiffusionNet as feature extractor in previous deep functional map methods does not lead to accurate point cloud matching, see the last column in Tab. 2. In contrast, the matching performance of triangle meshes and point clouds remains almost the same for our method. As a result, our method outperforms the previous state-of-the-art in most settings, even in comparison to supervised methods, which is particularly prominent in the case of point cloud matching. Fig. 3 shows a visual comparison of different methods in terms of point cloud matching.

5.2. Cross-dataset generalisation

Datasets. We evaluate the cross-dataset generalisation ability by training on the synthetic SURREAL [58] dataset and evaluating on FAUST, SCAPE and SHREC'19 datasets. Following [13], we use all (230k) shapes of the SURREAL dataset for point cloud matching methods, while only the first 5k shapes for deep functional map methods, since functional map regularisation is a strong regularisation requiring only a small amount of data to train [13].

Geo. error (×100)	**FAUST**			**SCAPE**			**SHREC'19**			**FM-based**
	Mesh	PC	Noisy PC	Mesh	PC	Noisy PC	Mesh	PC	Noisy PC	
				Axiomatic Methods						
BCICP [43]	6.4	-	-	11.0	-	-	8.0	-	-	✓
ZOOMOUT [35]	6.1	-	-	7.5	-	-	7.8	-	-	✓
Smooth Shells [15]	2.5	-	-	4.7	-	-	7.6	-	-	✓
				Supervised Methods						
FMNet [28]	3.1	8.5	14.0	9.1	15.0	21.3	10.4	14.3	19.1	✓
3D-CODED [19]	2.5	2.5	2.8	9.8	9.8	10.0	7.7	7.7	7.9	✗
IFMatch [55]	2.6	2.6	2.7	11.0	11.0	11.2	6.5	6.5	6.6	✗
DiffFMaps [32]	10.5	10.5	11.7	23.1	23.1	22.7	18.2	18.2	19.4	✓
GeomFMaps [13]	2.6	6.1	10.2	3.0	7.7	13.3	4.1	10.6	14.6	✓
				Unsupervised Methods						
SURFMNet [46, 51]	2.4	6.0	13.5	6.0	11.3	20.1	4.8	13.9	19.1	✓
UnsupFMNet [21]	4.8	9.6	17.8	9.6	11.3	15.5	11.1	17.3	23.8	✓
Deep Shells [17]	**1.7**	6.0	11.2	5.3	7.8	11.1	7.5	11.7	14.4	✓
ConsistFMaps [8]	2.4	11.2	16.9	5.1	12.3	16.4	4.2	13.7	17.2	✓
CorrNet3D [63]	26.5	26.5	27.0	37.3	37.3	36.8	33.7	33.7	34.0	✗
DPC [26]	11.6	11.6	14.6	16.0	16.0	18.6	17.6	17.6	19.4	✗
Ours	2.0	**2.4**	**4.4**	**3.1**	**4.1**	**6.6**	**4.0**	**4.5**	**5.8**	✓

Table 2. Quantitative results on the FAUST, SCAPE and SHREC'19 datasets in terms of mean geodesic errors. We evaluate all methods on individual dataset for shapes represented as triangle meshes and point clouds. The **best** results from the unsupervised methods in each column are highlighted. The last column indicates whether the method is based on the functional map framework. Our method outperforms previous unsupervised methods and bridges the matching performance gap between meshes and point clouds.

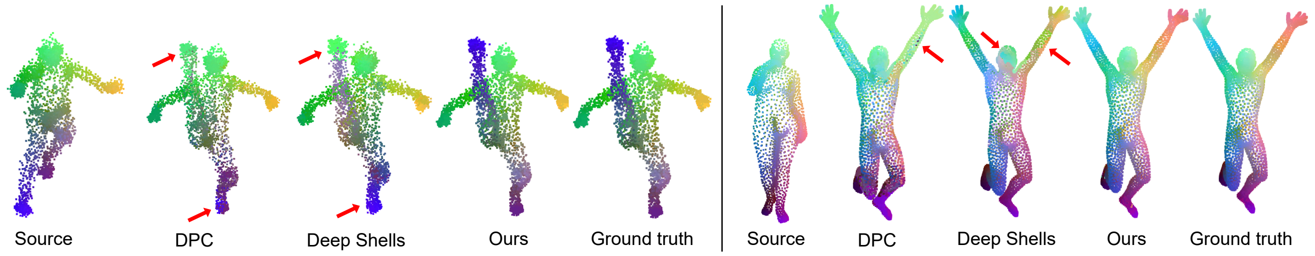

Figure 3. Qualitative results of different methods applied to point clouds from the SHREC'19 dataset. Errors are indicated by red arrows.

Geo. (×100)	**F (PC)**	**S (PC)**	**S19 (PC)**	**|Data|**
		Supervised Methods		
FMNet [28]	3.8 (12.2)	10.2 (15.3)	13.8 (22.7)	5k
DiffFMaps [32]	26.5 (26.5)	34.8 (34.8)	42.2 (42.2)	230k
GeomFMaps [13]	2.7 (10.4)	3.3 (8.7)	4.7 (14.1)	5k
		Unsupervised Methods		
SURFMNet [46, 51]	2.3 (16.0)	3.3 (14.7)	8.3 (27.8)	5k
Deep Shells [17]	8.1 (12.5)	12.2 (14.1)	12.1 (15.9)	5k
ConsistFMaps [8]	3.2 (19.3)	6.7 (17.3)	13.7 (24.2)	5k
CorrNet3D [63]	18.1 (18.1)	18.3 (18.3)	18.8 (18.8)	230k
DPC [26]	13.4 (13.4)	15.8 (15.8)	17.4 (17.4)	230k
Ours	**2.0** (**3.5**)	**3.2** (**3.8**)	**4.4** (**6.6**)	5k

Table 3. Cross-dataset generalisation evaluated on the **F**AUST, **S**CAPE and **S**HREC'**19** datasets and trained on the SURREAL dataset. The **best** results in each column are highlighted. The last column indicates the amount of data used for training. Our method outperforms previous supervised and unsupervised methods.

Results. As shown in Tab. 3, our method achieves a better cross-dataset generalisation ability and outperforms both state-of-the-art supervised and unsupervised methods. We note that deformation-based methods (e.g. CorrNet3D [63]) require a large amount of data to train, since it achieves better results when it is trained on large SURREAL dataset compared to trained on the same small dataset (see comparison between Tab. 2 and Tab. 3). In contrast, our method requires only a small amount of training data and achieves similar performance compared to intra-dataset training. Fig. 4 shows some qualitative results of our method.

5.3. Partial shape matching

Our framework can be easily adapted for unsupervised *partial* shape matching, see the supplementary document for details.

Datasets. We evaluate our method in the context of partial shape matching on the challenging SHREC'16 [10] dataset.

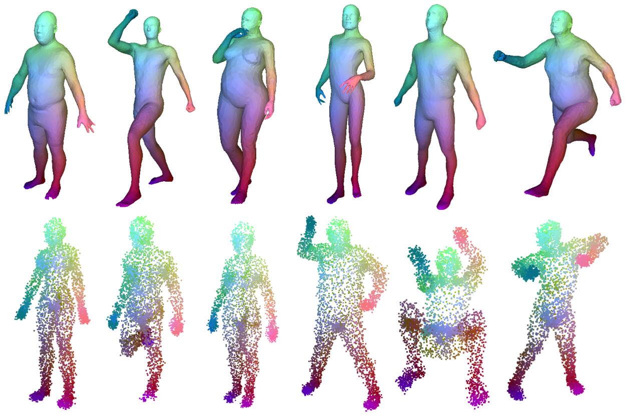

Figure 4. Qualitative results of our method on the SHREC'19 dataset (trained on SURREAL dataset) for both mesh and noisy point cloud matching. The top-left shape is the source shape. Our method demonstrates previously unseen generalisation ability.

This dataset consists of 200 training shapes, categorised into 8 classes (humans and animals). Each class has a complete shape to be matched by the other partial shapes. The dataset is divided into two subsets, namely CUTS (missing a large part) with 120 pairs, and HOLES (missing many small parts) with 80 pairs. Following [2], we train our method for each subset individually and evaluate it on the corresponding unseen test set (200 shapes for each subset).

Geo. error (×100)	CUTS (PC)	HOLES (PC)
Axiomatic Methods		
PFM [44]	9.7 (-)	23.2 (-)
FSP [29]	16.1 (-)	33.7 (-)
Supervised Methods		
GeomFMaps [13]	8.0 (18.5)	12.9 (18.9)
DPFM sup [2]	3.2 (10.4)	11.8 (17.0)
Unsupervised Methods		
ConsistFMaps [8]	8.4 (26.6)	17.9 (27.0)
DPFM unsup [2]	9.0 (20.9)	20.5 (22.8)
Ours	**7.6 (12.2)**	**15.9 (16.7)**

Table 4. Quantitative results on the CUTS and HOLES subsets of the SHREC'16 dataset. The **best** results from the unsupervised methods in each column are highlighted. Our method outperforms previous axiomatic and unsupervised methods.

Results. Tab. 4 summarises the results. Our method outperforms previous axiomatic and unsupervised methods. Compared to the *supervised* DPFM [2], our *self-supervised* method achieves comparable results for point cloud matching. Fig. 5 shows qualitative results of different methods for point cloud matching on the challenging HOLES subset.

5.4. Partial view matching

In this proof-of-concept experiment we consider the matching of a partially observed point cloud to a complete shape, which is a common scenario for data acquired from 3D scanning devices. To this end, we evaluate our method in terms of partial view matching, in which a partially observed point cloud is matched to a complete template shape.

Datasets. We create SURREAL-PV, a new partial view matching dataset based on the SURREAL [58] dataset. Given a complete shape represented as triangle mesh, we use raycasting to obtain a partial shape (both triangle mesh and point cloud) from a randomly sampled viewpoint. In total, we create 5k shape pairs and divide them into 80% training set and 20% test set. Compared to Sec. 5.3, the challenge of partial view matching is that there exists many disconnected components in the partial shapes and the sampling for partial point clouds is different compared to the complete shapes.

Geo. error (×100)	SURREAL-PV
DPFM sup [2]	7.8
DPFM unsup [2]	12.0
Ours	**6.3**

Table 5. Quantitative results on the SURREAL partial view dataset. The **best** results is highlighted. Our method outperforms both supervised and unsupervised DPFM.

Results. The results are shown in Tab. 5. We compare our method with DPFM [2], which is the state-of-the-art partial matching method (see Tab. 4). Our method even outperforms the supervised version of DPFM. Fig. 6 shows some qualitative results of our method.

5.5. Multimodal medical shape data

To demonstrate the potential for real-world applications, we conduct an experiment for multimodal matching in the context of medical image analysis. To this end, we use the real-world LUNA16 dataset [50], which provides chest CT-scans with corresponding lung segmentation mask. Based on the provided segmentation masks, we reconstruct 3D lung shapes represented as triangle meshes and simulate partial point clouds using a subset of slices of the volumetric images. Since there are no ground-truth correspondences available, we restrict ourselves to qualitative results, which are shown in Fig. 7, where it can be seen that reliable correspondences between different lung shapes can be obtained from our method.

6. Ablation study

We evaluate the importance of our introduced loss terms E_{align} in Eq. (6) and E_{nce} in Eq. (7) by discarding them individually. For this experiment, we consider the same experimental setting as in Sec. 5.1. More ablative experiments are provided in the supplementary document.

Results. Tab. 6 summarises the quantitative results. By comparing the first row with the second row, we can con-

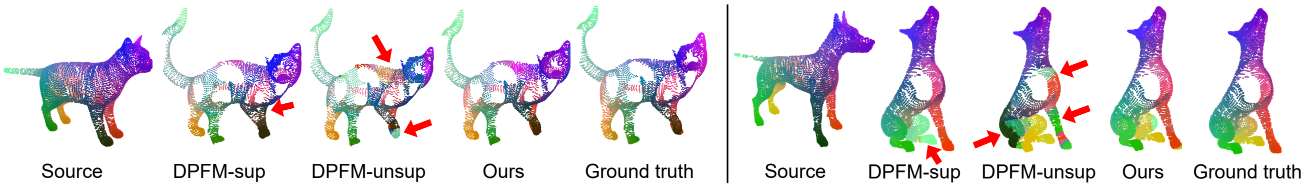

Figure 5. Qualitative results of different methods applied to point clouds from the HOLES subset. Errors are indicated by red arrows.

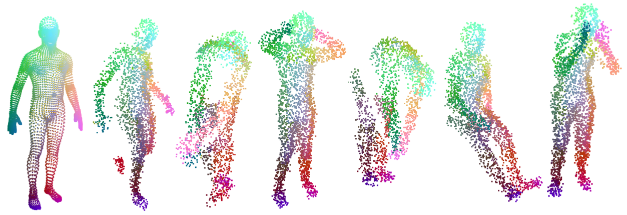

Figure 6. Qualitative results of our method on the SURREAL-PV dataset. The leftmost shape is the source shape. Our method obtains accurate correspondences even for partially-observed point clouds with different sampling and disconnected components.

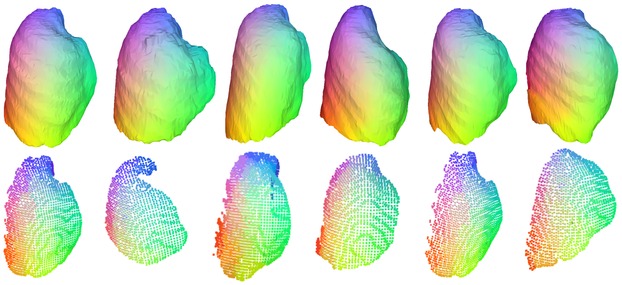

Figure 7. We obtain reliable correspondences for the matching of meshes and partial point clouds of 3D lung shapes.

Geo. error ($\times$**100**)	**F (PC)**	**S (PC)**	**S19 (PC)**
w.o. E_{align}, E_{nce}	2.3 (5.2)	4.8 (9.3)	4.3 (11.2)
w.o. E_{nce}	2.1 (2.7)	4.2 (5.6)	4.1 (5.1)
w.o. E_{align}	2.0 (22.7)	3.2 (20.9)	4.0 (30.8)
Ours	**2.0** (**2.4**)	**3.1** (**4.1**)	**4.0** (**4.5**)

Table 6. Ablation study on the **F**AUST, **S**CAPE and **S**HREC'**19** datasets. The **best** results in each column are highlighted.

clude that E_{align} plays the key role for accurate point cloud matching. By comparing the first row with the third row, we notice that E_{nce} can boost the matching performance for triangle meshes, while it hampers the matching performance for point clouds. Together with both loss terms, our method achieves accurate multimodal matchings.

7. Limitations and discussion

Our work is the first self-supervised approach for multimodal 3D non-rigid shape matching and achieves state-of-the-art performance on a diverse set of relevant tasks. Yet, there are also some limitations that give rise to interesting future research questions.

Unlike previous deep functional map methods that only work for noise-free meshes, our method can handle both meshes and point clouds, even under noise and partiality. However, our method struggles with severe outliers, since our method does not have an explicit outlier rejection mechanism and assumes that vertices on the shape with fewer vertices always have a correspondence on the other.

Analogous to many learning-based shape matching approaches, our method takes 3D vertex positions as input, and is thus not rotation-invariant. However, unlike deformation-based methods [19, 55, 63], which predict coordinate-dependent deformation fields and thus require rigidly-aligned shapes, our method allows for data augmentation by randomly rotating shapes during training (similar to [2, 13]), so that it is more robust to the initial pose, see the supplementary document for an experimental evaluation.

8. Conclusion

In this work we propose the first self-supervised learning framework for multimodal non-rigid shape matching. Our method allows to compute intramodal correspondences for meshes, complete point clouds, and partial point clouds, as well as correspondences across these modalities. This is achieved by introducing a novel multimodal training strategy that combines mesh-based functional map regularisation with self-supervised contrastive learning coupling mesh and point cloud data. We experimentally demonstrate that our method achieves state-of-the-art performance on numerous benchmarks in diverse settings, including complete shape matching, cross-dataset generalisation, partial shape and partial view matching, as well as multimodal medical shape matching. Overall, we believe that our method will be a valuable contribution towards bridging the gap between the theoretical advances of shape analysis and its practical application in real-world settings, in which partially observed and multimodal data plays an important role.

Acknowledgements

We thank Zorah Lähner for helpful feedback. This project was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 458610525.

References

[1] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM SIGGRAPH*. 2005. 5

[2] Souhaib Attaiki, Gautam Pai, and Maks Ovsjanikov. Dpfm: Deep partial functional maps. In *International Conference on 3D Vision (3DV)*, 2021. 2, 3, 4, 7, 8

[3] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *ICCV*, 2011. 2

[4] Florian Bernard, Zeeshan Khan Suri, and Christian Theobalt. Mina: Convex mixed-integer programming for non-rigid shape alignment. In *CVPR*, 2020. 2

[5] Federica Bogo, Javier Romero, Matthew Loper, and Michael J Black. Faust: Dataset and evaluation for 3d mesh registration. In *CVPR*, 2014. 5

[6] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Michael M Bronstein, and Daniel Cremers. Anisotropic diffusion descriptors. In *Computer Graphics Forum*, volume 35, pages 431–441. Wiley Online Library, 2016. 3

[7] Michael M Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *CVPR*, 2010. 2

[8] Dongliang Cao and Florian Bernard. Unsupervised deep multi-shape matching. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 7

[9] Ulrich Clarenz, Martin Rumpf, and Alexandru C Telea. Finite elements on point based surfaces. In *PBG*, pages 201–211, 2004. 3

[10] Luca Cosmo, Emanuele Rodola, Michael M Bronstein, Andrea Torsello, Daniel Cremers, and Y Sahillioglu. Shrec'16: Partial matching of deformable shapes. *Proc. 3DOR*, 2(9):12, 2016. 6

[11] Huong Quynh Dinh, Anthony Yezzi, and Greg Turk. Texture transfer during shape transformation. *ACM Transactions on Graphics (ToG)*, 24(2):289–310, 2005. 1

[12] Nicolas Donati, Etienne Corman, and Maks Ovsjanikov. Deep orientation-aware functional maps: Tackling symmetry issues in shape matching. In *CVPR*, 2022. 2, 4

[13] Nicolas Donati, Abhishek Sharma, and Maks Ovsjanikov. Deep geometric functional maps: Robust feature learning for shape correspondence. In *CVPR*, 2020. 2, 5, 6, 7, 8

[14] Marvin Eisenberger, Zorah Lähner, and Daniel Cremers. Divergence-free shape correspondence by deformation. In *Computer Graphics Forum*, volume 38, pages 1–12. Wiley Online Library, 2019. 2

[15] Marvin Eisenberger, Zorah Lahner, and Daniel Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *CVPR*, 2020. 2, 6

[16] Marvin Eisenberger, David Novotny, Gael Kerchenbaum, Patrick Labatut, Natalia Neverova, Daniel Cremers, and Andrea Vedaldi. Neuromorph: Unsupervised shape interpolation and correspondence in one go. In *CVPR*, 2021. 1, 4

[17] Marvin Eisenberger, Aysim Toker, Laura Leal-Taixé, and Daniel Cremers. Deep shells: Unsupervised shape correspondence with optimal transport. *NIPS*, 2020. 1, 2, 6

[18] Maolin Gao, Zorah Lahner, Johan Thunberg, Daniel Cremers, and Florian Bernard. Isometric multi-shape matching. In *CVPR*, 2021. 2

[19] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. 3d-coded: 3d correspondences by deep deformation. In *ECCV*, 2018. 1, 2, 3, 6, 8

[20] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Unsupervised cycle-consistent deformation for shape matching. In *Computer Graphics Forum*. Wiley Online Library, 2019. 1, 3

[21] Oshri Halimi, Or Litany, Emanuele Rodola, Alex M Bronstein, and Ron Kimmel. Unsupervised learning of dense shape correspondence. In *CVPR*, 2019. 2, 6

[22] Benjamin Holzschuh, Zorah Lähner, and Daniel Cremers. Simulated annealing for 3d shape correspondence. In *2020 International Conference on 3D Vision (3DV)*, 2020. 2

[23] Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics (ToG)*, 33(4):1–11, 2014. 2

[24] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. In *Computer Graphics Forum*. Wiley Online Library, 2020. 2

[25] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *ICLR*, 2017. 4

[26] Itai Lang, Dvir Ginzburg, Shai Avidan, and Dan Raviv. Dpc: Unsupervised deep point correspondence via cross and self construction. In *International Conference on 3D Vision (3DV)*, 2021. 1, 2, 3, 6

[27] Eugene L Lawler. The quadratic assignment problem. *Management science*, 9(4):586–599, 1963. 3

[28] Or Litany, Tal Remez, Emanuele Rodola, Alex Bronstein, and Michael Bronstein. Deep functional maps: Structured prediction for dense shape correspondence. In *ICCV*, 2017. 1, 2, 6

[29] Or Litany, Emanuele Rodolà, Alexander M Bronstein, and Michael M Bronstein. Fully spectral partial shape matching. In *Computer Graphics Forum*. Wiley Online Library, 2017. 2, 7

[30] Shengjun Liu, Haojun Xu, Dong-Ming Yan, Ling Hu, Xinru Liu, and Qinsong Li. WTFM Layer: An Effective Map Extractor for Unsupervised Shape Correspondence. *Computer Graphics Forum*, 2022. 2, 4

[31] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. Smpl: A skinned multi-person linear model. *ACM Transactions on Graphics (ToG)*, 34(6):1–16, 2015. 1

[32] Riccardo Marin, Marie-Julie Rakotosaona, Simone Melzi, and Maks Ovsjanikov. Correspondence learning via linearly-invariant embedding. *NIPS*, 2020. 2, 3, 6

[33] Jonathan Masci, Davide Boscaini, Michael Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *Proceedings of the IEEE international conference on computer vision workshops*, 2015. 1

[34] Simone Melzi, Riccardo Marin, Emanuele Rodolà, Umberto Castellani, Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Shrec 2019: Matching humans with different connectivity. In *Eurographics Workshop on 3D Object Retrieval*, 2019. 1, 5

[35] Simone Melzi, Jing Ren, Emanuele Rodola, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *arXiv preprint arXiv:1904.07865*, 2019. 2, 3, 6

[36] Gonzalo Mena, David Belanger, Scott Linderman, and Jasper Snoek. Learning latent permutations with gumbel-sinkhorn networks. In *ICLR*, 2018. 4

[37] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. In *Computer Graphics Forum*. Wiley Online Library, 2017. 2

[38] 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):1–11, 2012. 2, 3

[39] 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. 3

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

[41] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NIPS*, 2017. 2

[42] Jing Ren, Mikhail Panine, Peter Wonka, and Maks Ovsjanikov. Structured regularization of functional map computations. In *Computer Graphics Forum*. Wiley Online Library, 2019. 2, 4

[43] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics (ToG)*, 37:1–16, 2018. 2, 5, 6

[44] Emanuele Rodolà, Luca Cosmo, Michael M Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. In *Computer Graphics Forum*. Wiley Online Library, 2017. 2, 7

[45] Paul Roetzer, Paul Swoboda, Daniel Cremers, and Florian Bernard. A scalable combinatorial solver for elastic geometrically consistent 3d shape matching. In *CVPR*, 2022. 2

[46] Jean-Michel Roufosse, Abhishek Sharma, and Maks Ovsjanikov. Unsupervised deep learning for structured shape matching. In *ICCV*, 2019. 2, 3, 5, 6

[47] Yusuf Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer*, 36(8):1705–1721, 2020. 2

[48] Mahdi Saleh, Shun-Cheng Wu, Luca Cosmo, Nassir Navab, Benjamin Busam, and Federico Tombari. Bending graphs: Hierarchical shape matching using gated optimal transport. In *CVPR*, 2022. 4

[49] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 2

[50] Arnaud Arindra Adiyoso Setio, Alberto Traverso, Thomas De Bel, Moira SN Berens, Cas Van Den Bogaard, Piergiorgio Cerello, Hao Chen, Qi Dou, Maria Evelina Fantacci, Bram Geurts, et al. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge. *Medical image analysis*, 42:1–13, 2017. 7

[51] Abhishek Sharma and Maks Ovsjanikov. Weakly supervised deep functional maps for shape matching. *NIPS*, 2020. 2, 3, 5, 6

[52] Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *arXiv preprint arXiv:2012.00888*, 2020. 2, 3, 4, 5

[53] Nicholas Sharp and Keenan Crane. A laplacian for nonmanifold triangle meshes. In *Computer Graphics Forum*, volume 39, pages 69–80. Wiley Online Library, 2020. 3

[54] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 4

[55] Ramana Sundararaman, Gautam Pai, and Maks Ovsjanikov. Implicit field supervision for robust non-rigid shape matching. In *ECCV*, 2022. 2, 3, 6, 8

[56] Gary KL Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C Langbein, Yonghuai Liu, David Marshall, Ralph R Martin, Xian-Fang Sun, and Paul L Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE transactions on visualization and computer graphics*, 19(7):1199–1217, 2012. 1, 2, 3

[57] Oliver Van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A survey on shape correspondence. In *Computer Graphics Forum*, 2011. 1, 2

[58] Gul Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 2, 5, 7

[59] Matthias Vestner, Roee Litman, Emanuele Rodola, Alex Bronstein, and Daniel Cremers. Product manifold filter: Non-rigid shape correspondence via kernel density estimation in the product space. In *CVPR*, 2017. 2, 3

[60] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (ToG)*, 38(5):1–12, 2019. 2

[61] Thomas Windheuser, Ulrich Schlickewei, Frank R Schmidt, and Daniel Cremers. Geometrically consistent elastic matching of 3d shapes: A linear programming solution. In *ICCV*, 2011. 2

[62] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pretraining for 3d point cloud understanding. In *ECCV*, 2020. 5

[63] Yiming Zeng, Yue Qian, Zhiyu Zhu, Junhui Hou, Hui Yuan, and Ying He. Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. In *CVPR*, 2021. 1, 2, 3, 4, 6, 8