Differentiable Deformation Graph based Neural Non-rigid Registration

Wanquan Feng[1], Hongrui Cai[2], Junhui Hou[3], Bailin Deng[4] and Juyong Zhang[1*]

[1]School of Mathematical Sciences, University of Science and Technology of China, Hefei, China.
[2]School of Data Science, University of Science and Technology of China, Hefei, China.
[3]Department of Computer Science, City University of Hong Kong, Hong Kong, China.
[4]School of Computer Science and Informatics, Cardiff University, Cardiff, United Kingdom.

*Corresponding author(s). E-mail(s): juyong@ustc.edu.cn;
Contributing authors: lcfwq@mail.ustc.edu.cn;
hrcai@mail.ustc.edu.cn; jh.hou@cityu.edu.hk;
DengB3@cardiff.ac.uk;

Abstract

The traditional pipeline for non-rigid registration is to iteratively update the correspondence and alignment such that the transformed source surface aligns well with the target surface. Among the pipeline, the correspondence construction and iterative manner are key to the results, while existing strategies might result in local optima. In this paper, we adopt the widely used deformation graph based representation, while replacing some key modules with neural learning based strategies. Specifically, we design a neural network to predict the correspondence and its reliability confidence rather than the strategies like nearest neighbor search and pair rejection. Besides, we adopt the GRU-based recurrent network for iterative refinement, which is more robust than the traditional strategy. The model is trained in a self-supervised manner, and thus can be used for arbitrary datasets without ground-truth. Extensive experiments demonstrate that our proposed method outperforms the state-of-the-art methods by a large margin.

Keywords: differentiable deformation graph, non-rigid registration

MSC Classification: 65D19 , 68U05

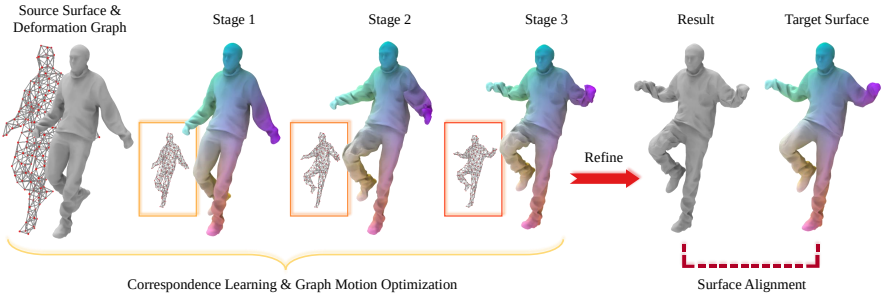

Fig. 1 We propose a deformation graph based neural learning method for non-rigid registration. The pipeline is in a coarse-to-fine fashion, utilizing the deformation graph representation for structure level registration and then vertex level deformation for geometry details. For the structure level registration, compared with the traditional optimization based approaches, we replace the correspondence construction, pair rejection and iterative strategy with the neural learning modules. For the vertex level registration, we apply a point-wise refinement module in the final to further improve the registration result.

1 Introduction

Registration, which is widely used to fuse different observations to reconstruct a global and complete shape, is a quite fundamental problem in computer vision, robotics and graphics. For example, it plays a key role in tracking [1, 2], reconstruction [3, 4], and many other tasks. Although it has been studied for many years, how to effectively and efficiently do registration still remains a challenging problem as it is essentially a combinatorial optimization problem. Compared with rigid registration, non-rigid registration [5–8] is even more challenging due to its large degree of freedom, and it gets even worse for data with partial overlap and missing areas.

A widely adopted pipeline [5, 7] for surface registration iterates between a correspondence step and an alignment step. The correspondence step constructs the point-wise correspondence between source and target surfaces. With the constructed correspondence, the alignment step estimates a spatial transformation by solving an optimization problem, which deforms the source surface to be closer to the target under some metrics. These two steps are applied alternately until convergence. The final registration result heavily relies on the quality of constructed correspondence, which is not trivial to obtain [9–14]. In traditional optimization based methods, the heuristic correspondence construction like nearest neighbor search and the simple iterative manner like

refining results from the last round might result in local optima, causing the lack of robustness.

Recently, learning based methods have made great progress and demonstrated better robustness [6, 8, 15, 16]. However, these methods are mainly developed for rigid registration. Compared with the rigid case, the learning based non-rigid surface registration is less developed. One challenge of non-rigid registration is how to represent the non-rigid deformation due to its high degree of freedom. The point-wise displacement has been applied in the cases where the non-rigid degree is small (e.g. the spline-based disturb [6], the scene flow [8]). A recent method [17] proposes to represent the non-rigid deformation in a recurrent scheme of several rigid transformations, which adopts a new learning based strategy. On the other hand, deformation graph [18] based representation is commonly used for non-rigid registration, due to its advantages of decreasing the representation complexity and resulting in reasonable deformed shapes by utilizing the shape's geometric shape structures. However, as the node number, edge connection and topology might differ in different shapes, it is not easy to directly utilize the deformation graph as input for network training.

In this paper, we propose a differentiable deformation graph based neural learning method for non-rigid registration by replacing some components with neural based strategies to fully take advantage of the shape priors and domain knowledge embedded in trained neural networks. The first replacement is the correspondence prediction module. We design a neural network to predict the correspondence rather than using the hand-craft features (e.g. the nearest neighbors). We also predict the point-wise reliability confidence for robust learning by giving less weight to bad pairs. The second key module is the iterative learning strategy. We employ a GRU [19] based recurrent structure, which keeps more historical memories of the iteration stages to replace the traditional iterative manner. Accordingly, we apply a hierarchical regularization strategy to control the freedom of each iteration, reducing the difficulty of training this recurrent network. Finally, we add a refinement module to refine the deformed shape in the point-wise level, which eliminates the errors caused by the limited representation ability of deformation graph. The whole framework is trained end-to-end in a self-supervised manner. Extensive experiments demonstrate that our proposed method dramatically outperforms the state-of-the-art methods on both synthetic and real-scanned data.

2 Related Works

Non-rigid deformation. Traditional non-rigid deformation estimation methods are based on optimization and can be categorized according to the representation. N-ICP [20] solves point-wise rigid transformation, while some other methods are based on thin plate spline functions [21–24]. The deformation graph can model the deformation as a series of local affine transformations, inspiring the embedded deformation-based methods [18, 25, 26]. Some other

methods [5, 27–30] are based on the Gaussian mixture model (GMM), encoding the points probabilistically and estimating the deformation by the Expectation-Maximization (EM) algorithm. Among them, Coherent point drift (CPD) [5] is the most classic and Bayes Coherent point drift (BCPD) [30] is a recently proposed Bayesian version of CPD that performs better in convergence and effectiveness.

There are also learning based methods proposed in recent years. CPD-Net [6] predicts the point-wise displacement by a PointNet backbone. Similarly, the scene flow estimation works [8, 31, 32] also estimate the point-wise displacement. DispVoxNets [33] regresses 3D displacement fields on regularly sampled proxy 3D voxel grids. PR-Net [34] models the deformation as some control points of thin plate spine and adopts a voxel based strategy to extract shape correlation tensor and predict the control points, supervised by a GMM loss. RMA-Net [17] proposes an iterative parameterized representation that fits the recurrent network structure to reduce the training difficulty, improving the effect for large-scale deformations. Different from the above methods, we employ the deformation graph representation in our network.

Furthermore, optical flow estimation [35, 36] can be treated as a 2D version of non-rigid registration, which also includes optimization based [35, 36] and learning based [37–39] algorithms. Recently, [40] utilizes the depth information to assist optical flow estimation with the help of deformation graph based representation. To deal with the large deformation on image plane, popular optical flow methods are also based on iterative strategies. The RAFT [37] employs a GRU based network to achieve the state-of-the-art performance.

Surface Correspondence. In traditional methods, shape correspondence is solved from a minimization problem through the similarity measures, such as point-wise [9–11, 41, 42] or pair-wise [43, 44] descriptors. Some works [45, 46] aim to obtain a sparse set of point correspondences and extend them to dense mappings. Then, some methods concentrated on measuring and optimizing consistency of sets of maps [47–49]. The functional maps [49] have the most impact among these works, converting the point-level correspondence to the function-level correspondence and reducing dimensionality of the problem drastically. Compared to the highly-complex, non-convex and non-linear optimization methods, the functional maps based methods only need to solve in a low-dimension space composed of the Laplace-Beltrami basis. The point-to-point correspondence can be recovered by some kinds of post-processing [49, 50].

There are also some deep learning methods [12, 13, 13] based on the functional maps in recent years. FMNet [13] proposed a framework that takes the point-wise descriptor SHOT [51] as input and returns the function-level correspondence. A later unsupervised method [12] keeps a similar structure with FMNet and uses geodesic distance matrices for supervision instead of the ground truth correspondence. Then another work [13] employed the heat kernels instead of geodesic distance as supervision, and achieved a similar effect with less training time. Rather than the functional maps, we directly predict

the dense correspondence and employ the point-wise confidence, the graph based regularization as well as a refinement module to increase the tolerance to the incorrect correspondence.

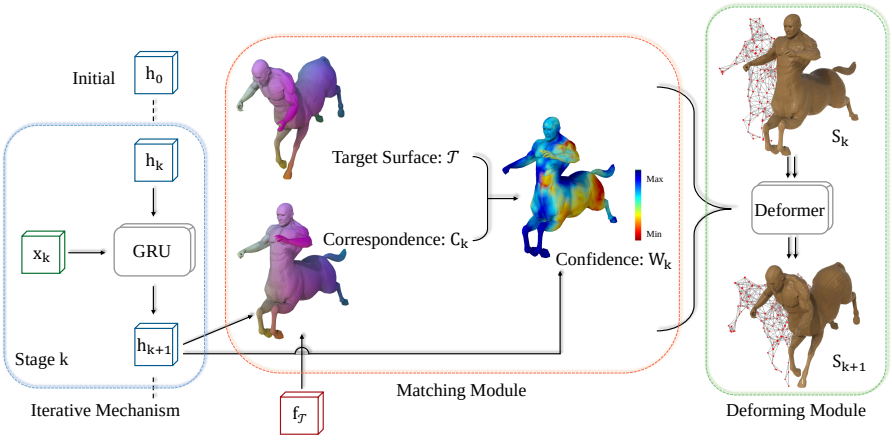

Fig. 2 Our network structure. We employ a GRU based recurrent framework and operates the matching and deforming modules in each iteration stage. In the matching module, the network predicted a soft correspondence with point-wise confidence. In the deforming module, the differentiable deformer optimizes the deformation such that the deformed shape gets closer to the target. Specifically, in the k-th stage, the GRU takes in a combination of the features x_k as input and updates the hidden state from h_{k-1} to h_k, based on an embedded feature of the target $f_{\mathcal{T}}$. For the computation of x_k and the initialization of hidden state h_0, we keep consistent with the previous work [17, 38].

3 Algorithm

Given the source and target surfaces represented as point sets $\mathcal{S} \in \mathbb{R}^{M\times 3}$ and $\mathcal{T} \in \mathbb{R}^{N\times 3}$, we aim to estimate a non-rigid deformed shape $\tilde{\mathcal{S}} \in \mathbb{R}^{M\times 3}$, such that $\tilde{\mathcal{S}}$ aligns well with the target surface $\mathcal{T}$. In this section, we will first introduce our pipeline, then give our loss function design.

3.1 Pipeline

To solve the registration problem, a popular pipeline is to alternately predict the correspondence and solve the deformation in an iterative manner, which is widely used in the optimization based registration methods [25, 26, 52, 53]. Let the subscript k denote the k-th iteration stage, and denoting $\mathcal{S}_0 = \mathcal{S}$, then the iterative process can be summarized as:

$$\begin{aligned}(\mathcal{S}_k, \mathcal{T}) &\xrightarrow{Match} \mathbf{C_k},\\ (\mathcal{S}_k, \mathcal{T}, \mathbf{C_k}) &\xrightarrow{Deform} \mathcal{S}_{k+1},\end{aligned} \tag{1}$$

where the matching process constructs the correspondence $\mathbf{C}$, and the deforming process estimates the deformation, usually by solving an optimization problem. For the deformation representation, the deformation graph [18] is a popular choice, benefiting from the low freedom and the good structure level shape prior.

In this paper, we refer to the above classic pipeline and improve some modules to form a more robust framework. We also utilize the deformation graph representation to achieve a satisfactory structure level registration result. However, some geometry details may be lost because of the small freedom of the deformation graph, which motivates us to employ a point level refinement module in the final, and thus forms the overall coarse-to-fine framework. For the structure level registration, the correspondence construction and iterative manner are re-designed such that a better solution can be achieved compared with currently used heuristic strategies like nearest neighbor search in correspondence construction. An overview of our pipeline is shown in Fig. 1

The overall framework is organized by a GRU [19] based recurrent network, where each stage includes a matching module and a deforming module, illustrated in Fig. 2.

Matching Module. In each iteration, we predict the dense correspondence between the source and target surfaces, represented as a soft matching matrix: $\mathbf{C} \in \mathbb{R}^{M \times N}$ with non-negative elements. Each row of $\mathbf{C}$ sums to 1, with the elements representing the probability for a point in the source surface to correspond to each point on the target surface. To obtain the prediction, we extract the embedded feature of target (denoted as $f_{\mathcal{T}}$) with DGCNN [54] and apply it to each iteration. Compared with searching correspondences in 3D coordinate space, searching in high-dimensional feature space can make the matching module more robust to data noises. In the k-th iteration, the correspondence map $\mathbf{C}_k$ is computed as:

$$\mathbf{C}_k = Softmax(\langle h_k \cdot f_{\mathcal{T}} \rangle), \tag{2}$$

where the h_k denotes the hidden state in the recurrent network, who can remember the history features of each stage. We first compute the inner product of h_k and $f_{\mathcal{T}}$, and then use the softmax normalization on each row. With the soft correspondence, we can form a soft projection from the source surface into the target surface: $\mathcal{S} \rightarrow \mathbf{C}_k\mathcal{T}$, where each point in $\mathcal{S}$ is softly related to the expectation to the target points. According to the $\mathcal{S}$, $\mathbf{C}_k\mathcal{T}$ and h_k, the network then predicts the point-wise confidence $\mathbf{W}_k \in (0, 1)^M$ to the correspondence, representing how reliable the predicted correspondence is.

Deforming Module. Once we obtain the predicted matching matrix $\mathbf{C}_k$ and the confidence $\mathbf{W}_k$, we then use them to solve the deformation. Here the deformation is defined on the deformation graph of $\mathcal{S}$, denoted as $\mathcal{G} =$

$(\mathcal{V}, \mathcal{E}, \mathcal{M})$, where

$$\begin{aligned} \mathcal{V} &= \{v_i \in \mathbb{R}^3, i = 1, \ldots, m\}, \\ \mathcal{E} &= \{(v_i, v_j), v_i \text{ is adjacent to } v_j\}, \\ \mathcal{M} &= \{(R_i, t_i) \in SE(3), i = 1, \ldots, m\}, \end{aligned} \tag{3}$$

denote the set of nodes, the set of edges, and the set of node-wise rigid motions, respectively. With the motion $\mathcal{M}$ defined on $\mathcal{G}$, the source surface $\mathcal{S}$ can be deformed into the target surface $\mathcal{M} \circ \mathcal{S}$, where we use the symbol $\circ$ to denote the deforming operation. (Specific deforming formulation can be seen in [18] or in our supplementary materials.) For the given $\mathcal{S}$ and $\mathcal{V}$, the deformed result $\mathcal{M} \circ \mathcal{S}$ is totally decided by the motion $\mathcal{M}$.

The motion used for deforming is obtained by solving an optimization problem. In the first stage, $\mathcal{M}_1$ is initialized as the identity transformation. When k>1, $\mathcal{M}_k$ is initialized as $\mathcal{M}_{k-1}$. For the optimization, we employ two energy terms. Firstly, the deformed result should be consistent with the predicted correspondence. Specifically, we encourage the deformed result to be close to $\mathbf{C}_k\mathcal{T}$, which has been computed in the matching module, and the relating energy term can be formed as:

$$\mathbb{E}_{deform}(\mathcal{M}_k) = \|\mathbf{W}_k \odot (\mathcal{M}_k \circ \mathcal{S} - \mathbf{C}_k\mathcal{T})\|_F^2. \tag{4}$$

where the $\odot$ means the point-wise multiplication. Moreover, we also use a regularization energy term while solving $\mathcal{M}_k$ on $\mathcal{G}$. Like in [18], for each pair of neighbors, we sum the squared distances between the transformation applied to the neighbors and the actually transformed neighbor positions, encouraging the transformations on neighbor nodes are consistent:

$$\mathbb{E}_{reg}(\mathcal{M}_k) = \sum_{(v_i, v_j) \in \mathcal{E}} \|R_i(v_j - v_i) + (v_i + t_i) - (v_j + t_j)\|_2^2, \tag{5}$$

where the R_i, t_i and t_j denote the transformations defined on the neighbouring nodes v_i and v_j of $\mathcal{M}_k$. Thus, we can solve the optimal motion $\tilde{\mathcal{M}}$ by:

$$\tilde{\mathcal{M}}_k = \arg\min_{\mathcal{M}_k} (\mathbb{E}_{deform}(\mathcal{M}_k) + \lambda_k \cdot \mathbb{E}_{reg}(\mathcal{M}_k)) \tag{6}$$

To solve the optimization problem, we refer to [40] and use the Gauss-Newton method, keeping the deforming module differentiable. Overall, the formulation of the deforming module can be written as:

$$\mathcal{S}_k = \tilde{\mathcal{M}}_k \circ \mathcal{S}. \tag{7}$$

Here, we notice that the λ_k in Eq. (6) is a natural regularization for the final deformed surface $\mathcal{S}_k$. When $\lambda_k \to \infty$, the deformed $\mathcal{S}_k$ should be close to a

rigid transformation of $\mathcal{S}$. Based on this observation, we employ a hierarchical regularization strategy:

$$\lambda_1 > \lambda_2 > \cdots > \lambda_K \approx 0, \tag{8}$$

which would make the network to learn the deformation step by step, reducing the one-step difficulty and promoting the registration effect.

Iterative Mechanism. Instead of the simple iteration scheme (directly use the current result as the input of the next iteration, as in Eq. (1)), we adopt the GRU-based recurrent network for iteration. With a fixed updating format [17, 38, 55], in the k-th stage, the GRU takes in a combination of the features x_k in the current as input, and updates the hidden state from h_{k-1} to h_k. For the computation of x_k and the initialization of hidden state h_0, we keep consistent with the previous work [17, 38]. The computation of the following process is based on the updated hidden state h_k and an embedded feature of the target $f_{\mathcal{T}}$. During the alternate iterations, we pass the h_k and $f_{\mathcal{T}}$ into the matching and deforming modules in the following stage.

After the last iteration, the network still needs to fix the gap between the structure level deformation and the geometry details on the target. Specifically, we employ a refinement module, which predicts the point-wise displacement from h_K and $f_{\mathcal{T}}$ to achieve a vertex level registration, denoted as $\mathcal{S}^{'}$. This module is also easy to train since the distance between the deformed surface and the target surface has been significantly reduced in the earlier iterations.

3.2 Loss

We train our network in an unsupervised manner to suit the situation that the well labeled data of real-scanned deforming objects is difficult to obtain. In this section, we introduce the loss terms which are used for the network training. For the convenience of description, we firstly consider a single iteration, where the predicted matching matrix and deformed surface are denoted as $\mathcal{C}$ and $\tilde{\mathcal{S}}$.

Chamfer Loss. To deform the source surface such that the deformed surface $\tilde{\mathcal{S}}$ is overall aligned to the target $\mathcal{T}$, we adopt the chamfer distance to meausre their distance:

$$\mathcal{L}_{cd}(\tilde{\mathcal{S}}, \mathcal{T}) = \frac{1}{2\mathcal{S}} \sum_{\tilde{s} \in \tilde{\mathcal{S}}} \min_{t \in \mathcal{T}} \|\tilde{s} - t\|_2^2 + \frac{1}{2\mathcal{T}} \sum_{t \in \mathcal{T}} \min_{\tilde{s} \in \tilde{\mathcal{S}}} \|\tilde{s} - t\|_2^2. \tag{9}$$

ARAP Loss. To encourage the predicted correspondence to be smooth, we use the as-rigid-as-possible (ARAP) loss here. Specially, we encourage the neighbor distance to stay close between the original point set and the expectation of the target point $\mathbf{C}\mathcal{T}$. Let the subscripts (i), (j) to denote the row-selecting operation, the ARAP loss is defined as:

$$\mathcal{L}_{arap}(\mathbf{C}) = \sum_{(i,j) \in \mathcal{E}_{\mathcal{S}}} (\|\mathbf{C}_{(i)}\mathcal{T} - \mathbf{C}_{(j)}\mathcal{T}\|_2 - \|\mathcal{S}_{(i)} - \mathcal{S}_{(j)}\|_2)^2, \tag{10}$$

where the $\mathcal{E}_{\mathcal{S}}$ and $\mathcal{E}_{\mathcal{T}}$ means the edge sets of $\mathcal{S}$ and $\mathcal{T}$.

Confidence Loss. In order to prevent the confidence $\mathbf{W}$ from degrading to 0, we encourage the value of the $\mathbf{W}$ not to be too small via the following energy term:

$$\mathcal{L}_{conf}(\mathbf{W}) = -\|\mathbf{W}\|_2^2. \tag{11}$$

Suppose we use K iterations for training, the loss in the k-th iteration can be formulated as:

$$\mathcal{L}_k = \mathcal{L}_{cd}(\mathcal{S}_k, \mathcal{T}) + \beta_1 \mathcal{L}_{arap}(\mathbf{C}_k) + \beta_2 \mathcal{L}_{conf}(\mathbf{W}_k). \tag{12}$$

Refinement Loss. Although the deformation graph based representation performs quite well especially on preserving the shape structure, it only has small freedoms and thus can not deform the geometry details quite well. Therefore, we add a refinement module in the final and encourage refined surface $\mathcal{S}^{'}$ to be close to the target. Moreover, we limit the displacement to be not too large. Considering these aspects, the loss function for the refined surface $\mathcal{S}^{'}$ is defined as:

$$\mathcal{L}_{refine}(\mathcal{S}^{'}) = \mathcal{L}_{cd}(\mathcal{S}^{'}, \mathcal{T}) + \epsilon \|\mathcal{S}^{'} - \mathcal{S}_K\|_2^2. \tag{13}$$

The overall loss can be written as:

$$\mathcal{L}_{sum} = \sum_{k=1}^{K} \gamma^{K-k} \mathcal{L}_k + \mathcal{L}_{refine}. \tag{14}$$

4 Experiments

In this section, we conduct extensive experiments to analyze the algorithm components and show the results and comparisons on real scanned non-rigid shapes, demonstrating the superiority of our method.

4.1 Implementation Details

Dataset. We train and test our model on both synthetic and real-scanned data, including 6 categories of deformable objects: human body, cat, dog, wolf, centaur and horse. For the synthetic data, we apply our model in the TOSCA [56] dataset who has several categories of synthetic animal models. We also train and test our model in the real-scanned dataset, the Dynamic FAUST[57], which contains 10 raw scanned human body surface sequences. From the TOSCA dataset, we choose 16,551 training pairs and 513 testing pairs. From the human sequences, we select a total of 7,802 training pairs and 391 testing pairs. The real-scanned data are lack of the ground truth correspondence between the source and target in each pair, and the model contains random noise as well as incompleteness due to the real scanning process, which increases the difficulty of registration. Moreover, we also test our trained model on some more deficiency data obtained by *Kinect Azure DK* to show our robustness.

Network Training. For all the training and testing surfaces, we sample 4096 points to form the source and target point clouds. For all models, we construct

the deformation graph with node number in range $150-200$. During the Gauss-Newton optimization, we use 5 iterations with the step of the iterations to be 1.0, 0.8, 0.7, 0.6, 0.5. The network contains 3 iterations, with $\{\lambda_k\}_{k=1}^3$ to be 10^3, 10^2, 10 respectively. The β_1 and β_2 in Eq. (12) are set as 5 and 10^{-4}, the ϵ in Eq. (13) is set as 0.03, and the γ in Eq. (14) is set as 0.9. We train the network with the OneCycleLR [58] strategy, with the cycle length as 1000, the maximum learning rate as 10^{-4} and the batch size as 4. The network is totally trained for 500K forward-backward iterations with the Adam [59] optimizer. All the experiments are conducted on a workstation with 40 Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz, 128GB of RAM, and four 24G GeForce RTX 3090Ti GPUs.

Metrics For the results of both the synthetic and real-scanned data, we evaluate the registration performance with CD and EMD, as the same in [17].

4.2 Ablation Study

We analyze the choices of the key settings of our network by experiments, including the iteration number, the refinement module, the point-wise confidence and the deformation node number.

#Iter	$\{\lambda_k\}$	Refine	Confidence	CD ↓	EMD ↓
1	$\{10^3\}$	×	×	22.34	9.32
2	$\{10^3, 10^2\}$	×	×	13.36	8.02
3	$\{10^3, 10^2, 10\}$	×	×	6.33	5.20
4	$\{10^3, 10^2, 10, 1\}$	×	×	7.23	5.78
3	$\{10^3, 10^2, 10\}$	√	×	5.10	4.23
3	$\{10^3, 10^2, 10\}$	√	√	**4.84**	**4.09**

Table 1 Results of the component analysis for the network settings, evaluated by CD($\times 10^{-5}$) and EMD($\times 10^{-3}$).

Network Settings. To explore the best network settings, we study on the iteration times, the refinement module and the correspondence confidence. Using the Dynamic FAUST[57] dataset as benchmark, we apply different strategies to train and test on the same data and with the same training settings, which is listed in Tab. 1.

At first, we use only 1 iteration without the refinement module or the correspondence confidence (shown in the 1-st row). Then we add iteration numbers step by step. From the $1-4$ rows, the best choice about iteration number is 3, with the regularization weights to be 10^3, 10^2, and 10. The result of the 4-th row is not as good as the 3-rd row because the too small regularization weight increases the freedom and causes the training difficulty. Therefore, we use 3 iterations in all our experiments. In the 5-th module, we add the refinement module after the last iteration, and find that the error is reduced again, showing the effect of the refining process. At last, we add the correspondence confidence in the 6-th row, which also promotes the registration effect. This is

also in line with our expectations, because confidence provides a natural soft pair-rejection strategy, which is also important in the optimization methods. In our main experiments on both synthetic and real-scanned datasets, we keep the network settings always the same as the 6-th row.

Number of Deformation Nodes. When constructing the deformation graph, the number of nodes should be adjusted manually. We apply experiments to search for the number of nodes that are most conducive to network training. We collect 30 pairs of shapes from the TOSCA [56] dataset and created 10 versions of the deformation graphs for them, in which the number of nodes gradually increased.

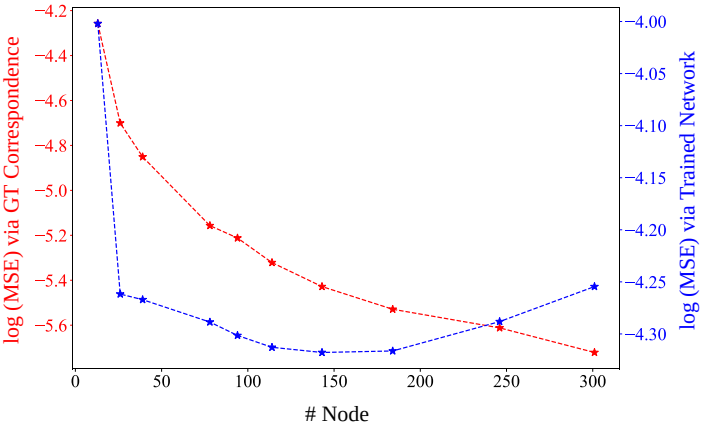

Fig. 3 The analysis on the node number of the graph. Red line represents the error of the optimization with ground truth. Blue line represents the error of the unsupervised training.

Firstly, we directly use the ground truth correspondence to solve the Gauss-Newton optimization ($\lambda = 10$). The MSE between the deformed shape and the target shape is shown as the red line in Fig. 3, where the horizontal axis represents the number of nodes, and the vertical axis (left side, color in red) represents the MSE. Then we apply our network (1 iteration, unsupervised) to train on this collection for enough forward-backward iterations (we set 1000 epochs to make sure the convergence), and shown the fitting error as the blue line in Fig. 3, where the horizontal axis represents the number of nodes, and the vertical axis (right side, color in blue) represents the error of the network result.

Although a larger number of nodes can drive the smaller fitting error by optimization with the ground truth, the best choice for our unsupervised training is a relatively moderate number of nodes. The graphs with too few nodes are lack of expression ability, and the graphs with too many nodes would have too large freedom, which is not conducive to our unsupervised network training.

4.3 Results and Comparisons

In this section, we show the qualitative and quantitative results and comparisons on the synthetic dataset TOSCA and real-scanned dataset Dynamic FAUST with recent state-of-the-art methods, showing the superiority of our method. While CPD [5] is the most classic and well-known optimization method for non-rigid registration, BCPD [30] is a recent Bayesian version of CPD, which performs better on the convergence and achieves the state-of-the-art effect among the optimization methods. Another recent work, RMA-Net [17], is a learning based framework that parameterizes the non-rigid deformation as the combination of a series of rigid transformations, which is also trained in the unsupervised manner and achieves the state-of-the-art effect. Some previous works [60, 61] consider the registration between deformable shapes and a template shape, but we do not compare with them because we do not assume there is a template shape in our task. Moreover, we also compare with a framework that focuses on estimating the surface correspondence. Specifically, we compare with FMNet [62], a supervised network based on the functional maps. The predicted dense correspondence can pull each source point to its correspondence position, with can be viewed as the natural deformation defined by the correspondence.

4.3.1 Registration for Synthetic Data

We firstly train and test our model on the synthetic dataset, and compare it with the CPD, BCPD, RMA-Net and FMNet. From the TOSCA dataset, we choose 5 categories of animals to compose the training and testing set, including cat, dog, horse, centaur and wolf. The Tab. 2 shows the performance comparison of each method on the TOSCA dataset. We can see that our method obtains the best performance.

Metric	Input	CPD	BCPD	FMNet	RMA-Net	Ours
CD↓	47.50	27.29	16.31	59.9	3.12	**2.54**
EMD↓	27.76	4.43	3.05	16.25	0.45	**0.40**

Table 2 Results and comparison on the TOSCA dataset, with metrics CD($\times 10^{-5}$) and EMD($\times 10^{-3}$).

The optimization based methods CPD and BCPD are sensitive to the specific testing sample and may not be able to converge to the right solution, causing the result not good enough. The FMNet predicts the dense correspondence, which is usually evaluated by the accuracy under a geodesic error threshold. In our registration problem that means to estimate the deformation, we try the simple way that directly considers the predicted correspondence of FMNet as the deformation. The low quality also shows that there is still some

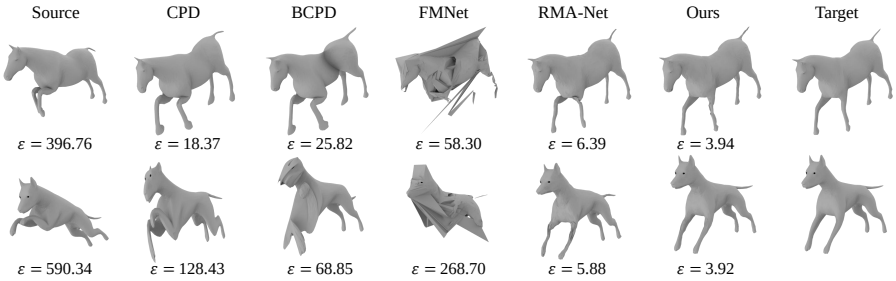

Fig. 4 Comparisons with optimization based methods CPD [5], BCPD [30] and learning based methods FMNet [62], RMA-Net [17] on the synthetic dataset TOSCA [56]. All values are metrics CD($\times 10^{-5}$) of specific examples.

gap from the dense correspondence predicting task directly to a good registration quality, which is actually escaped in our pipeline by some specifically designed modules and the end-to-end trainable framework.

RMA-Net can basically pull the limbs to the correct position (e.g. the legs of the dog in Fig. 4), but there may exist local collapses in some details, which does not appear in our method, thanks to the regularization of the deformation graph. Moreover, RMA-Net may cause misidentification of the topology (e.g. the horse in Fig. 4, where some points of the left thigh are pulled onto the body), which does not occur in our results, either. The reason should be that the deformation graph implies enough shape prior to significantly decrease the probability of topology misidentification.

From both Fig. 4 and Tab. 2, we can conclude that our method works well on the non-rigid surface registration task. Taking good use of the shape prior and registration effect of the deformation graph, our method outperforms the previous methods.

4.3.2 Registration for Real-scanned Data

To prove our feasibility and robustness on the real-scanned data, we train and test our model on the Dynamic FAUST dataset, where the real-scanned data suffers from noise, outliers and incompleteness. Our comparison with CPD, BCPD and RMA-Net is shown in Fig. 5 and Tab. 3. The comparison conclusion is consistent with the synthetic scenarios. Our method achieves the best performance in both qualitative and quantitative experiments.

Metric	Input	CPD	BCPD	RMA-Net	Ours
CD↓	87.68	21.02	11.19	5.03	**4.84**
EMD↓	13.09	8.23	6.78	4.17	**4.09**

Table 3 Results and cmparisons on the Dynamic FAUST dataset, with metrics CD($\times 10^{-5}$) and EMD($\times 10^{-3}$).

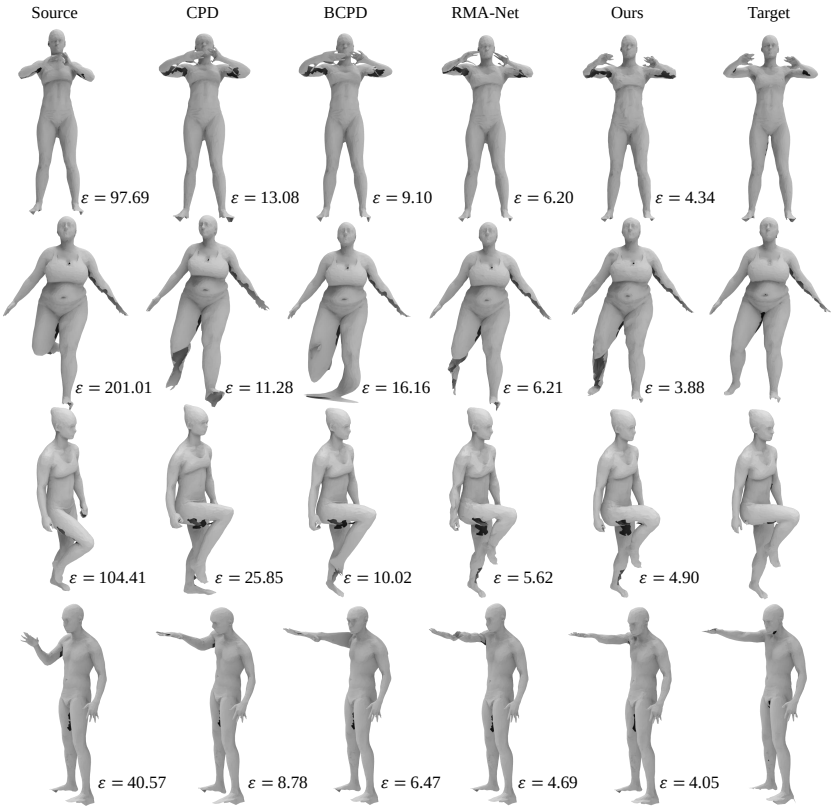

Fig. 5 Comparisons with CPD [5], BCPD [30] and RMA-Net [17] on the real-scanned dataset Dynamic FAUST [57]. All values are metrics CD($\times 10^{-5}$) of specific examples.

In the first row of Fig. 5, our result is the only one that pulls the hands apart to the correct target position, while none of others pulls them apart. In the second row, CPD, RMA-Net and our results pull the leg down successfully, but only our result keeps the original shape of the calf and foot. The third and fourth examples raise the leg and stretch the arm respectively, where our method estimates the correct body motion and keeps the overall surface in a reasonable shape, escaping from elongating the calf (CPD, BCPD in the third row) or bending the arm (CPD, RMA-Net in the fourth row). From these examples, we can see that our method still works quite well for real-scanned data that suffers from low quality.

4.3.3 Registration for More Deficiency Data

To further show the robustness of the framework, We further test on some depth images acquired by *Kinect Azure DK*. We use a dataset for 4D dynamic reconstruction problem [63] which contains 14 depth sequences of different people (we construct 4680 and 121 pairs for training and testing) and tried

to register global human models to the depth images. We also apply RMA-Net on this dataset for comparison. Results are shown in Fig. 6. We can see that our result can keep the basic shape of the source model when changing the pose, while the result of RMA-Net can not keep the basic shape of the source. This shows the robustness of our framework. Although our method improves the quality of non-rigid registration, there are still some failure cases. Like embedded deformation-based methods [25, 26], our method cannot handle topology change, which is still an open problem in non-rigid registration.

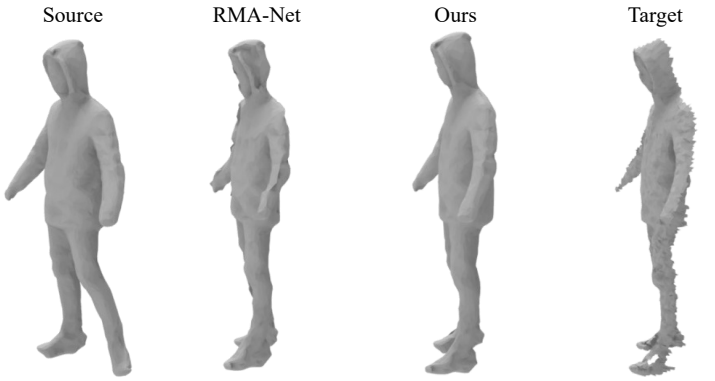

Fig. 6 Comparisons with RMA-Net [17] on the depth images collected by *Kinect Azure DK*.

5 Conclusion

In this work, we propose a deformation graph based neural learning method for non-rigid registration in a coarse-to-fine fashion, applying the structure level and vertex level registration in turns. For the coarse structure level, we follow the traditional deformation graph based pipeline and improve some key modules by learning based strategies, including the correspondence construction and the iterative mechanism. For the vertex level, we utilize a point-wise refinement module to achieve better geometry details. The network is trained end-to-end in the unsupervised manner, and outperforms the previous state-of-the-art methods on both synthetic and real-scanned data by a large margin.

Acknowledgement

This work was supported by National Natural Science Foundation of China (No. 62122071), the Youth Innovation Promotion Association CAS (No. 2018495), "the Fundamental Research Funds for the Central Universities" (No. WK3470000021)

References

[1] Qian, C., Sun, X., Wei, Y., Tang, X., Sun, J.: Realtime and robust hand tracking from depth. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1106–1113 (2014)

[2] Vedula, S., Baker, S., Rander, P., Collins, R.T., Kanade, T.: Three-dimensional scene flow. IEEE Trans. Pattern Anal. Mach. Intell. **27**(3), 475–480 (2005)

[3] Newcombe, R.A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A.J., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.W.: Kinectfusion: Real-time dense surface mapping and tracking. In: IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 127–136 (2011)

[4] Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 343–352 (2015)

[5] Myronenko, A., Song, X.B.: Point set registration: Coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. **32**(12), 2262–2275 (2010)

[6] Wang, L., Fang, Y.: Coherent point drift networks: Unsupervised learning of non-rigid point set registration. CoRR **abs/1906.03039** (2019)

[7] Besl, P.J., McKay, N.D.: A method for registration of 3-d shapes. IEEE Trans. Pattern Anal. Mach. Intell. **14**(2), 239–256 (1992)

[8] Liu, X., Qi, C.R., Guibas, L.J.: Flownet3d: Learning scene flow in 3d point clouds. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 529–537 (2019)

[9] Rustamov, R.M.: Laplace-beltrami eigenfunctions for deformation invariant shape representation. In: Eurographics Symposium on Geometry Processing, pp. 225–233 (2007)

[10] Sun, J., Ovsjanikov, M., Guibas, L.J.: A concise and provably informative multi-scale signature based on heat diffusion. Comput. Graph. Forum, 1383–1392 (2009)

[11] Tombari, F., Salti, S., di Stefano, L.: Unique signatures of histograms for local surface description. In: European Conference on Computer Vision (ECCV), pp. 356–369 (2010)

[12] Halimi, O., Litany, O., Rodolà, E., Bronstein, A.M., Kimmel, R.: Unsupervised learning of dense shape correspondence. In: IEEE/CVF Conference

on Computer Vision and Pattern Recognition (CVPR), pp. 4370–4379 (2019)

[13] Aygün, M., Lähner, Z., Cremers, D.: Unsupervised dense shape correspondence using heat kernels. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 573–582 (2020)

[14] Zeng, Y., Qian, Y., Zhu, Z., Hou, J., Yuan, H., He, Y.: Corrnet3d: Unsupervised end-to-end learning of dense correspondence for 3d point clouds. CoRR **abs/2012.15638** (2020)

[15] Aoki, Y., Goforth, H., Srivatsan, R.A., Lucey, S.: Pointnetlk: Robust & efficient point cloud registration using pointnet. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7163–7172 (2019)

[16] Wang, Y., Solomon, J.: Deep closest point: Learning representations for point cloud registration. In: IEEE International Conference on Computer Vision (ICCV), pp. 3522–3531 (2019)

[17] Feng, W., Zhang, J., Cai, H., Xu, H., Hou, J., Bao, H.: Recurrent multi-view alignment network for unsupervised surface registration. CoRR **abs/2011.12104** (2020)

[18] Sumner, R.W., Schmid, J., Pauly, M.: Embedded deformation for shape manipulation. ACM Trans. Graph. **26**(3), 80 (2007)

[19] Cho, K., van Merrienboer, B., Bahdanau, D., Bengio, Y.: On the properties of neural machine translation: Encoder-decoder approaches. In: Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pp. 103–111 (2014)

[20] Amberg, B., Romdhani, S., Vetter, T.: Optimal step nonrigid ICP algorithms for surface registration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2007)

[21] Chui, H., Rangarajan, A.: A new point matching algorithm for non-rigid registration. Comput. Vis. Image Underst. **89**(2-3), 114–141 (2003)

[22] Jian, B., Vemuri, B.C.: Robust point set registration using gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1633–1645 (2011)

[23] Chen, J., Ma, J., Yang, C., Ma, L., Zheng, S.: Non-rigid point set registration via coherent spatial mapping. Signal Process. **106**, 62–72 (2015)

[24] Yang, Y., Ong, S.H., Foong, K.W.C.: A robust global and local mixture distance based non-rigid point set registration. Pattern Recognit. **48**(1), 156–173 (2015)

[25] Li, H., Sumner, R.W., Pauly, M.: Global correspondence optimization for non-rigid registration of depth scans. Comput. Graph. Forum **27**(5), 1421–1430 (2008)

[26] Yao, Y., Deng, B., Xu, W., Zhang, J.: Quasi-newton solver for robust non-rigid registration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7597–7606 (2020)

[27] Kolesov, I., Lee, J., Sharp, G., Vela, P.A., Tannenbaum, A.R.: A stochastic approach to diffeomorphic point set registration with landmark constraints. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 238–251 (2016)

[28] Ma, J., Zhao, J., Yuille, A.L.: Non-rigid point set registration by preserving global and local structures. IEEE Trans. Image Process. **25**(1), 53–64 (2016)

[29] Golyanik, V., Taetz, B., Reis, G., Stricker, D.: Extended coherent point drift algorithm with correspondence priors and optimal subsampling. In: Winter Conference on Applications of Computer Vision (WACV), pp. 1–9 (2016)

[30] Hirose, O.: A bayesian formulation of coherent point drift. IEEE Trans. Pattern Anal. Mach. Intell. **PP**(99), 1–1 (2020)

[31] Kittenplon, Y., Eldar, Y.C., Raviv, D.: Flowstep3d: Model unrolling for self-supervised scene flow estimation. CoRR **abs/2011.10147** (2020)

[32] Mittal, H., Okorn, B., Held, D.: Just go with the flow: Self-supervised scene flow estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11174–11182 (2020)

[33] Shimada, S., Golyanik, V., Tretschk, E., Stricker, D., Theobalt, C.: Dispvoxnets: Non-rigid point set alignment with supervised learning proxies. In: International Conference on 3D Vision, pp. 27–36 (2019)

[34] Wang, L., Chen, J., Li, X., Fang, Y.: Non-rigid point set registration networks. CoRR **abs/1904.01428** (2019)

[35] Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 648–655 (1998)

[36] Hurlburt, N.E., Jaffey, S.: A spectral optical flow method for determining velocities from digital imagery. Earth Sci. Informatics **8**(4), 959–965 (2015)

[37] Liu, P., Lyu, M.R., King, I., Xu, J.: Selflow: Self-supervised learning of optical flow. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4571–4580 (2019)

[38] Teed, Z., Deng, J.: RAFT: recurrent all-pairs field transforms for optical flow. In: European Conference on Computer Vision (ECCV), pp. 402–419 (2020)

[39] Sun, D., Yang, X., Liu, M., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8934–8943 (2018)

[40] Bozic, A., Palafox, P.R., Zollhöfer, M., Dai, A., Thies, J., Nießner, M.: Neural non-rigid tracking. In: Annual Conference on Neural Information Processing Systems (NeurIPS) (2020)

[41] Aubry, M., Schlickewei, U., Cremers, D.: The wave kernel signature: A quantum mechanical approach to shape analysis. In: IEEE International Conference on Computer Vision (ICCV), pp. 1626–1633 (2011)

[42] Bronstein, M.M., Kokkinos, I.: Scale-invariant heat kernel signatures for non-rigid shape recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1704–1711 (2010)

[43] Coifman, R.R., Lafon, S., Lee, A.B., Maggioni, M., Nadler, B., Warner, F., Zucker, S.W.: Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proceedings of the national academy of sciences, 7426–7431 (2005)

[44] Mémoli, F., Sapiro, G.: A theoretical and computational framework for isometry invariant recognition of point cloud data. Found. Comput. Math., 313–347 (2005)

[45] Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. Proceedings of the National Academy of Sciences, 1168–1172 (2006)

[46] Lipman, Y., Funkhouser, T.A.: Möbius voting for surface correspondence. ACM Trans. Graph., 72 (2009)

[47] Nguyen, A., Ben-Chen, M., Welnicka, K., Ye, Y., Guibas, L.J.: An optimization approach to improving collections of shape maps. Comput. Graph. Forum, 1481–1491 (2011)

[48] Kim, V.G., Lipman, Y., Funkhouser, T.A.: Blended intrinsic maps. ACM Trans. Graph., 79 (2011)

[49] Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.J.: Functional maps: a flexible representation of maps between shapes. ACM Trans. Graph., 30–13011 (2012)

[50] Rodolà, E., Möller, M., Cremers, D.: Point-wise map recovery and refinement from functional correspondence. In: International Symposium on Vision Modeling and Visualization, pp. 25–32 (2015)

[51] Tombari, F., Salti, S., di Stefano, L.: Unique signatures of histograms for local surface description. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) European Conference on Computer Vision (ECCV). Lecture Notes in Computer Science, vol. 6313, pp. 356–369 (2010)

[52] Guo, K., Xu, F., Wang, Y., Liu, Y., Dai, Q.: Robust non-rigid motion tracking and surface reconstruction using L0 regularization. In: IEEE International Conference on Computer Vision (ICCV), pp. 3083–3091 (2015)

[53] Li, H., Adams, B., Guibas, L.J., Pauly, M.: Robust single-view geometry and motion reconstruction. ACM Trans. Graph. **28**, 175 (2009)

[54] Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. **38**(5), 146–114612 (2019)

[55] Teed, Z., Deng, J.: RAFT-3D: scene flow using rigid-motion embeddings. CoRR **abs/2012.00726** (2020)

[56] Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Monographs in Computer Science, (2009)

[57] Bogo, F., Romero, J., Pons-Moll, G., Black, M.J.: Dynamic FAUST: Registering human bodies in motion. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017)

[58] Smith, L.N., Topin, N.: Super-convergence: Very fast training of residual networks using large learning rates. CoRR **abs/1708.07120** (2017)

[59] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (ICLR) (2015)

[60] Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: 3d-coded: 3d correspondences by deep deformation. In: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II, vol. 11206 (2018)

[61] Li, C., Simon, T., Saragih, J.M., Póczos, B., Sheikh, Y.: LBS autoencoder: Self-supervised fitting of articulated meshes to point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11967–11976 (2019)

[62] Litany, O., Remez, T., Rodolà, E., Bronstein, A.M., Bronstein, M.M.: Deep functional maps: Structured prediction for dense shape correspondence. In: IEEE International Conference on Computer Vision (ICCV) (2017)

[63] Yu, T., Zheng, Z., Guo, K., Zhao, J., Dai, Q., Li, H., Pons-Moll, G., Liu, Y.: Doublefusion: Real-time capture of human performances with inner body shapes from a single depth sensor. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)