Robust Point-Cloud Registration based on Dense Point Matching and Probabilistic Modeling

Gustavo Marques Netto§, Manuel M. de Oliveira Neto
Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil
gmnetto@inf.ufrgs.br, oliveira@inf.ufrgs.br

***Abstract*—We present techniques for 3D point-cloud registration that are suited for scenarios where robustness to outliers and missing regions is necessary, besides being applicable to both rigid and non-rigid configurations. Our techniques exploit advantages from deep learning models for dense point matching and from recent advances in probabilistic modeling of point-cloud registration. Such a combination produces context awareness and resilience to outliers and missing information. We demonstrate their effectiveness by comparing them to state-of-the-art methods and showing that ours achieves superior results in general. For example, our approaches achieve registration error up to 45% smaller than these methods in partial point clouds for non-rigid registration, and up to 49% smaller on rigid registration.**

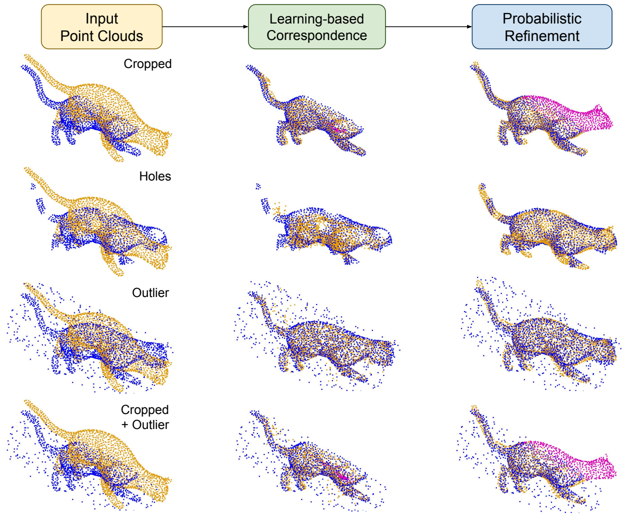


Figure 1: Non-rigid registrations produced by our method for challenging scenarios. We present the results after each stage of our pipeline. The source clouds are in orange and the target ones are in blue. Points without correspondence in the target are shown in magenta.

I. Introduction

The recent popularization of 3D scanners and LIDARs has led to a growing interest on the manipulation and processing of 3D point clouds. A fundamental task when processing point clouds is registration, which aims at aligning different sets of samples. It is a basic block for applications such as SLAM [1], scene/object reconstruction [2], [3], automated driving [4], motion estimation [5], among others.

In *rigid registration*, two point clouds to be aligned only differ by a rotation and/or a translation. ICP [6] and its variants are perhaps the most widely used solutions for this class of problems. In *non-rigid registration*, portions from a source/target point cloud can be deformed independently from others, such as in articulated or organic figures. In this case, one seeks for an as-close-as-possible alignment between the two point clouds. The registration problem becomes considerably harder in the presence of noise, and when portions of the point clouds to be aligned are missing. Approaching these challenges is key for achieving robust registration, able to deal with real-world scenarios.

Recently, deep learning techniques targeted at rigid registration have obtained impressive results [7]. However, little work has been done on non-rigid registration. State-of-the-art works [7], [8] still lack resilience to critical cases, e.g. when dealing with noisy point clouds and/or when there are only partial matches between them. This is mostly due to the difficulty of establishing correspondences among the available parts. The same observation applies to state-of-the-art non-learning-based methods, such as BCPD [9].

We present robust techniques for performing non-rigid as well as rigid registration of point clouds. Our techniques combine the advantages of deep learning models for obtaining dense point correspondences with recent advances in probabilistic modeling. They achieve state-of-the-art results for challenging scenarios involving noisy and missing regions (Fig. 1), leading to better and faster registration. We also present a self-supervised training approach for non-rigid registration with on-par results to supervised training.

Fig. 1 illustrates the use of our technique, trained using self-supervised learning, for non-rigid registration of point clouds under multiple difficult situations. It shows the results produced by the steps of our method (*Learning-based Correspondence* and *Probabilistic Refinement*) for scenarios including: (i) missing a large portion of one of the point clouds (*Cropped* - first row). (ii) several holes across one point cloud (*Holes* - second row); (iii) noisy point cloud (*Outliers* - third row); (iv) a combination of missing large portions and noise (*Cropped* + *Outliers* - fourth row). Note how our method consistently achieves good results across all these situations, properly handling points without correspondences (in pink).

§M.Sc. dissertation

The **contributions** of of this work include:

- Learning-based techniques for non-rigid and rigid registration of point clouds that are robust to large missing portions, as well as to noise. Both techniques outperform state-of-the-art approaches in these challenging scenarios;
- An adaptation of the SuperGlue network to produce correspondences between pairs of 3D point clouds;
- A self-supervised training strategy for robust non-rigid registration of point clouds.

II. Related Works

Rigid Registration: ICP [6] is one of the most popular approaches for rigid registration, and several improvements have been proposed for it (*e.g.*, Tr-ICP [10] and EM-ICP [11]). Probabilistic approaches like CPD [12], FilterReg [13], GMM-Tree [14], and Branch-and-Bound-based methods, such as Go-ICP [15], have shown improved results. Feature matching [16] followed by RANSAC is largely used to better prune the correspondences. In general, these approaches rely on iterative optimization, which becomes expensive when dealing with challenging scenarios. For an in-depth discussion of these techniques, see [17].

Recently, neural networks have been used to address the limitations of traditional approaches. Interestingly, new works tend to build on concepts from traditional ones to achieve state-of-art results. DCP [18] takes the correspondence and transformation estimation from ICP, while RPM-Net [19] builds on DCP by having an iterative process and solving an optimal transport (OT) problem. Predator [20] uses an attention mechanism on features from KPConv convolutions [21] followed by RANSAC. Other works also using deep features plus RANSAC include D3Feat [22] and FCGF [23]. Finally, RGM [24] employs a graph matching network to estimate the correspondences and a binary-assignment loss function, instead of directly outputting a rigid transformation.

Non-Rigid Registration: Non-rigid ICP [25] expands the original algorithm to this domain. CPD [12] models one cloud as a Gaussian Mixture Model (GMM) and solves a likelihood problem through Expectation-Maximization (EM). GMMReg [26] presents a framework that represents both point clouds as GMM's and minimizes their statistical discrepancy. Ma et al. [27] presented a series of papers on the topic. Recently, BCPD [9] extended CPD using variational Bayesian inference and Nyström acceleration, instead of EM. Overall, these approaches have issues with partial clouds as they are unable to deal with points without correspondences.

FlowNet3D [28] uses a cost volume to learn the flow of points in the source point cloud in an end-to-end fashion. This was followed by extensions covering non-supervised learning [29], [30], pyramid refinement with an improved cost volume (PointPWC-Net) [31], and recurrent networks (FlowStep3D) [32]. FLOT [33] estimates the scene flow modeling it as an OT problem using deep features. Ouyang and Raviv [34] expanded PointPWC to explicitly account for occlusions in the clouds by considering them in the cost volume and in the loss function. Although these works present impressive results on LIDAR point clouds, the cost-volume layer has issues learning large deformations as it uses nearest-neighbors similarities.

CPD-Net [35] predicts the complete deformation using deep features and an unsupervised Chamfer distance loss function. PR-Net [36] expands traditional techniques by learning the shape correlation based on features of the voxelized point clouds. RMANet [7] presented an unsupervised learning approach based on a recurrent model that offers superior results to CPD-Net and PR-Net. These works do not handle partial point clouds, nor propose ways to do it. A related problem is shape correspondence [37]. State-of-the-art learning-based works include NeuroMorph [8] which is based on point-cloud correspondence and can be extended to the registration case. Like previously mentioned works, NeuroMorph does not account for partial correspondences.

III. Robust Non-rigid Registration

Our non-rigid registration technique consists of two parts: *learning-based correspondences* followed by *probabilistic refinement* (Fig. 1). To obtain the correspondences among the samples of source and target point clouds, we modified the SuperGlue network [38] (originally designed to establish matches among keypoints in pairs of 2D images). The main changes include defining a feature encoding to support matches between 3D points, and a new Optimal Transport module that improved the registration performance by 43% (more details in the thesis). For the probabilistic refinement, we adapted the BCPD algorithm to use the soft-correspondence assignments produced by our network, thus replacing BCPD's point matching process with a more robust alternative.

BCPD models the registration problem using a GMM. It iteratively optimizes five variables: the single variance of all 3D Gaussian distributions in the GMM, the scale, the rotation, the translation, and the non-rigid deformation vectors required for the matching. While BCPD produces good results, it does not handle large deformations.

Our technique uses the soft assignments produced by our network for 3D point clouds as an approximation to BCPD's correspondence matrix, $\mathbf{P}$ (Fig. 2). BCPD assumes that the probability matrix comes from a GMM. Although this is not the case for our soft assignments, we argue that it is feasible to use it, as both matrices encode correspondence and outlier information through matching probabilities. The key aspect for our technique is then to iteratively update all variables given $\mathbf{P}$, and vice-versa. We achieve this by splitting the BCPD original loop into two nested ones. An outer loop updates the correspondences based on the latest variable values. The inner one optimizes the five variables given the estimated $\mathbf{P}$. This setup reduces the registration error for partial clouds on our *Custom* Dataset by 22% on average, when compared to BCPD.

IV. Robust Rigid Registration

Our rigid-registration approach minimizes the weighted distance between all points from the source and target clouds, where the weights are given by the correspondences generated by our neural network. Since this problem is simpler compared

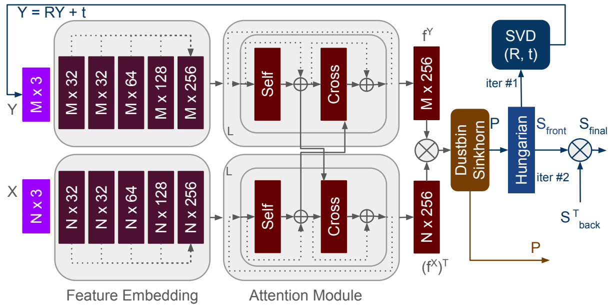


Figure 2: Adapted SuperGlue network for learning-based correspondences. Non-rigid registration uses a single iteration of the purple, red, and brown modules, producing a soft-assignment matrix P (brown) as output. For rigid registration, the network includes all modules, and it performs two iterations, producing a hard assignment matrix S_{final} as output.

to non-rigid registration, we only use the learning-based correspondences, dropping the probabilistic refinement step. We slightly modify our network used for the non-rigid case, and we iterate twice (as opposed to a single iteration used for the non-rigid case). This is illustrated in Fig. 2 by the blue lines and modules (Hungarian and SVD). The Hungarian algorithm converts the soft-assignment matrix $\mathbf{P}$ into a hard assignment matrix $\mathbf{S}$, which undergoes an SVD decomposition to estimate the rotation and translation ($\mathbf{R}$ and $\mathbf{t}$). Finally, we also use cyclic consistency to estimate the final registration.

V. Learning-based Correspondence Network

Our proposed network learns the probability matrix $\mathbf{P}$ that indicates the dense correspondences between source and target point clouds, Y and X (Fig. 2). Thus, it estimates dense correspondences in a pair of 3D point clouds. Next we describe each module of the network.

Feature Embedding: Differently from SuperGlue, we densely estimate the initial point-cloud features using the DGCNN [39] network and we focus on considering local geometry information. The embeddings for the different layers, l, are given by:

$$\begin{aligned} \mathsf{x}_m^0 &= max_{j:(m,j)} h_\theta^0(\{x_m, x_j\})\ \forall j \ \in \ \mathsf{N}_m,\\ \mathsf{x}_m^l &= max_{j:(m,j)} h_\theta^l(h_\theta^{l-1}),\\ \mathsf{x}_m^{final} &= h_\theta^l(\{\mathsf{x}_m^{l-1},...,\mathsf{x}_m^0\}),\end{aligned} \tag{1}$$

where N_m are the k nearest neighbors of x_m, and $\{,\}$ is the concatenation operator. h_θ^l is a non-linear function consisting of a multi-layer perceptron (MLP: 2D Convolution with kernel size of 1 + Batch Normalization + ReLU) which is followed by maxpolling across the neighbors. We use five layers (Fig. 2); the first computes the embedding from the point cloud coordinates (x_m^0). The next three layers iteratively refine the embedding (x_m^l). The last one takes the concatenation of all previous four layers and does not max-pool (x_m^{final}).

Attention Module: We directly use the attention mechanism of SuperGlue to include more contextual information in the deep features. As a result, the deep feature x_n receives contributions from X (*self*) and Y (*cross*). Attention enhances the dense estimated features by alternatively applying multiple independent layers of *self* and *cross* modules in a residual manner

$$\mathsf{x}_n^{l+1} = \mathsf{x}_n^l + \delta_{x_n}^l, \ \delta_{x_n}^l = \begin{cases} attention(\mathsf{x}_n^l, \mathsf{x}_n^l), & \text{if } self, \\ attention(\mathsf{x}_n^l, \mathsf{y}_m^l), & \text{if } cross, \end{cases} \tag{2}$$

where x_n^l is the current feature values and $attention$ is the actual module. The $attention$ function is a neural network that aggregates features from one of its inputs, x_n^l or y_m^l, based on a learned similarity between this same input and the queried one, x_n^l in this case. The same is done to y_m^l and we apply alternately *self* and *cross* L times, resulting in $\mathsf{f}_m^Y, \mathsf{f}_n^X$ (Fig. 2).

Sinkhorn Operator with Dustbins: Given the refined descriptors, a direct way to establish correspondences is through scores given by the dot products $\tilde{\mathbf{P}}_{m,n} = \langle \mathsf{f}_m^Y, \mathsf{f}_n^X \rangle$, where $\mathsf{f}_m^Y, \mathsf{f}_n^X$ are 1×256 feature vectors corresponding to the rows of the tensors f^Y and f^X. Although this produces good results for one-to-one correspondences, it is unable to directly account for points without correspondences. In addition, the dot product by itself does not produce a valid probability matrix, which in our case should be a relaxed doubly stochastic matrix. This means the sum of the probabilities in the rows and columns should be less or equal to one. To tackle this situation, we use the Sinkhorn operator proposed by Mena *et al* [40]. The possible lack of correspondences is handled by an extra possibility of assignment in both sets of descriptors, called dustbins (Fig. 2). The resulting Sinkhorn operator is iterative and it converges to the desired probability matrix in the limit.

Loss Function:We introduce a new term to the cross-entropy loss used by RGM to compare the estimated matrix $\mathbf{P}$ and the ground truth $\bar{\mathbf{P}}$ (a hard-assignment matrix). This aims at classifying if a source point $y_m \in Y$ has a correspondence to any target point $x_n \in X$. The new loss term compares the sum of the matching probabilities associated with each source point, $\nu = \mathbf{P}\mathbf{1}_{N_c}$, to the ground truth $\bar{\nu} = \bar{\mathbf{P}}\mathbf{1}_{N_c}$. Even though this new term is indirectly covered by the original loss function, we justify it because we want to minimize the occurrence of false matches. False matches tend to create artifacts as the whole neighborhood around each incorrectly matched points is deformed as well. The complete loss function is given by:

$$\begin{aligned} Loss = &- \sum_{n=1}^{N} \sum_{m=1}^{M} (\bar{\mathbf{P}}_{m,n} \log \mathbf{P}_{mn} + \\ &\qquad (1 - \bar{\mathbf{P}}_{m,n}) \log(1 - \mathbf{P}_{m,n})) \\ &- \sum_{m=1}^{M} (\bar{\nu}_m \log \nu_m + (1 - \bar{\nu}_m) \log(1 - \nu_m)). \end{aligned} \tag{3}$$

VI. Experiments

We implemented our models using PyTorch [41] and Python and used them to register a large number of point clouds under challenging configurations.

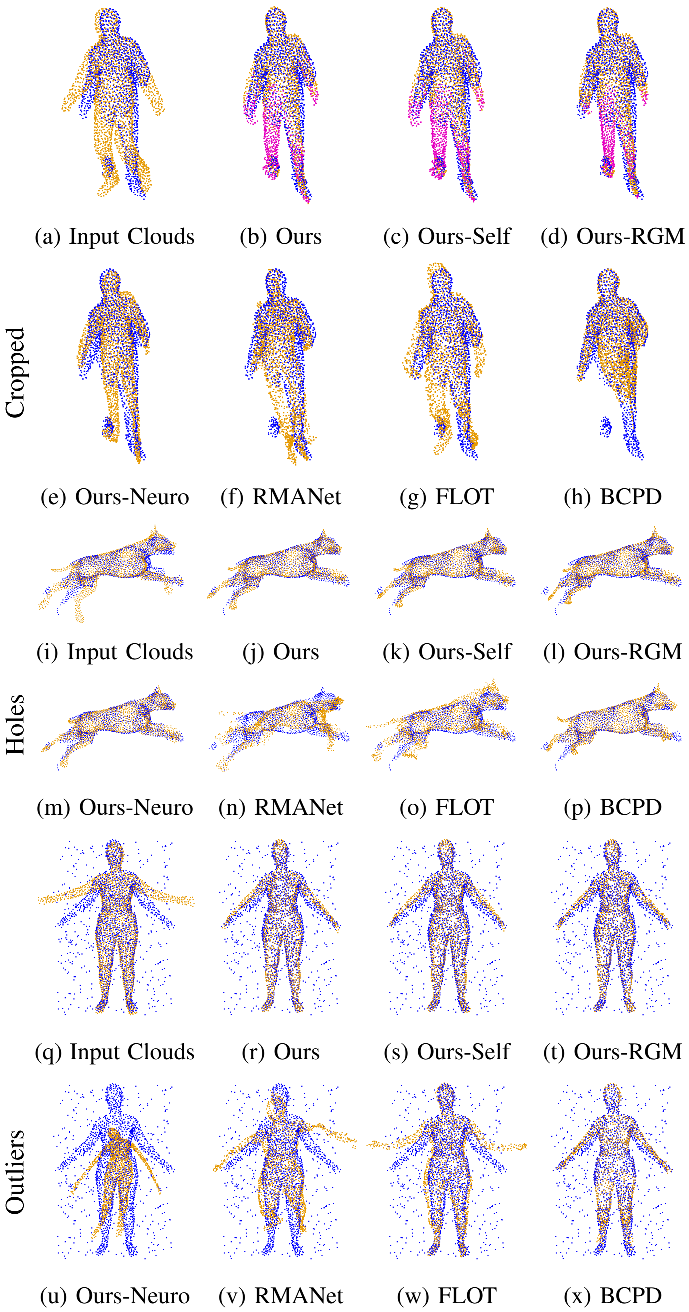
A. Non-Rigid Registration

The main metric used to assess the registration quality of the evaluated algorithms is the end-to-end point error (EPE), which is the mean Euclidean distance between the deformed points and their correspondences. The experiments included four types of point clouds: (i) *Clean*; (ii) *Cropped*, where contiguous regions covering 30% of the points were removed; (iii) *Outliers*, where 20% of target points consist of random uniformly distributed points; and (iv) *Holes*, where 25% of the points in the target point cloud were removed creating holes around random seed points. For experiments involving *Cropped* point clouds, we report Precision and Recall for the classification of points without correspondences.

We compare our approach to five others designed to, or adapted by us to, perform non-rigid registration of point clouds: RMANet, RGM, NeuroMorph, FLOT, and BCPD. We used three datasets for training and testing. The first one is the dataset proposed by RMANet. We compiled the second dataset (referred to as *Custom*) sampling from different datasets (more details in the thesis). The third dataset was built from face meshes [42].

An existing problem when training networks on non-rigid deformation is creating datasets. Rigid-transformation approaches can randomly sample rotations, translations, and crops on the fly. Learning-based non-rigid techniques, on the other hand, generally rely on non-supervised learning, using loss functions based on Chamfer and Earth Mover's distance, for example. However, these are unable to explicitly account for points with no correspondences. To address this issue, we trained our model adding noise to point clouds, and mainly using the artificial deformation method proposed by Hirose [43] to generate dense correspondences. We refer to this dataset as *Self* in the experiments (Fig. 4).

Fig. 3 compares several techniques considering point clouds of type *Cropped*, *Holes*, and *Outliers*. For each type, the original source and target point clouds (Input) are shown on the top left. These are followed by the results produced by: our method (*Ours*), our method trained using self-supervised learning (*Ours-Self*), Ours-RGM, Ours-Neuro, RMANet, FLOT, and BCPD. For the *Cropped* example, our method and its variants are able to map source points without correspondences in the target cloud to plausible locations. FLOT produces a reasonable result, although one observes misalignment in the head, arms and legs. RMANet does a good job for the upper body, but the legs are off. BCPD was unable to map points without correspondences. For the *Holes* example, RMANet, followed by FLOT, are the ones for which misalignment is most noticeable. The *Outlier* test case also illustrates the effectiveness of our method and its variants Ours-Self and Ours-RGM. Ours-Neuro did not produce a satisfactory result, indicating that the $\mathbf{P}$ matrix generated by NeuroMorph does define reliable correspondences in the presence of noise. Both RMANet and FLOT produced unsatisfactory results, while BCPD achieved good registration. These results highlight that our method is consistently robust across a range of challenging scenarios. It also shows that the $\mathbf{P}$ matrix generated by our model leads to better results than RGM's (Fig. 3l) and NeuroMorph's (Fig. 3u). Moreover, it shows that although BCPD nicely complements our model, it cannot, just by itself,



(a) Input Clouds (b) Ours (c) Ours-Self (d) Ours-RGM

(e) Ours-Neuro (f) RMANet (g) FLOT (h) BCPD

(i) Input Clouds (j) Ours (k) Ours-Self (l) Ours-RGM

(m) Ours-Neuro (n) RMANet (o) FLOT (p) BCPD

(q) Input Clouds (r) Ours (s) Ours-Self (t) Ours-RGM

(u) Ours-Neuro (v) RMANet (w) FLOT (x) BCPD

Figure 3: Results on *Cropped*, *Holes*, and Outlier types of point clouds on top of the RMANet dataset. Source cloud is in orange and target is in blue. Source points classified as having no matches are shown in magenta.

Method	RMANet							Custom				
	Clean		Outliers	Holes	Cropped			Clean		Cropped		
	P/D	BCPD				Prec.	Recall	P/D	BCPD		Prec.	Recall
Ours	0.020 ^{P}	**0.020**	**0.023**	**0.028**	**0.018**	**0.98**	**0.98**	0.060 ^{P}	**0.036**	0.040	**0.90**	0.92
Ours-Self	0.026 ^{P}	**0.020**	0.026	**0.029**	0.020	0.93	**0.98**	0.052 ^{P}	**0.035**	**0.037**	**0.90**	**0.94**
Ours-RGM	0.027 ^{P}	**0.021**	0.027	0.033	0.021	0.92	0.94	0.084 ^{P}	0.047	0.073	0.84	0.85
Ours-Neuro	**0.008** ^{P}	0.023	0.160	0.042	0.140	0.46	0.22	**0.019** ^{P}	0.038	0.206	0.52	0.32
RMANet	0.012 ^{D}	-	0.101	0.149	✗	✗	✗	0.131 ^{D}	-	✗	✗	✗
FLOT	0.043 ^{D}	-	0.053	0.042	0.045	-	-	0.111 ^{D}	-	0.110	-	-
BCPD	-	0.042	0.045	0.057	✗	✗	✗	-	0.159	✗	✗	✗

$^{P, D}$ Projection Operation; Direct Generation.

Table I: Performance comparison using EPE as well as Precision and Recall metrics for several methods on different datasets. *BCPD* indicates the errors after the probabilistic refinement. We further indicate the results with a projection operation using the correspondence matrix (P) or with the direct generation (D), in the column P/D. (✗) means the technique did not handle the given dataset, (-) means the comparison is not applicable. The best results are shown in **bold**.

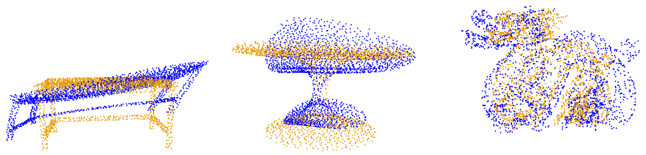
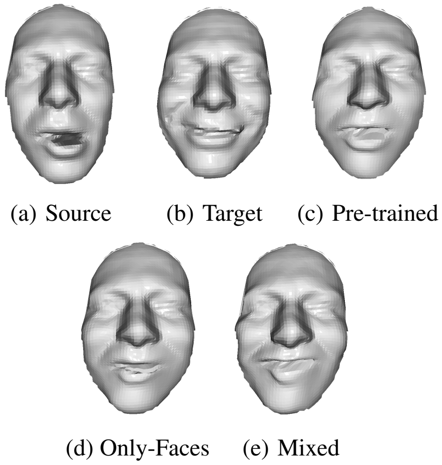


Figure 4: Examples of artificial deformations created by our adopted self-suppervised training approach. The original point cloud is in blue while the deformed one is in orange. Notice that by keeping a pointwise correspondence, we can generate *Cropped*, *Holes*, and *Outliers* point clouds.

handle large missing regions (Fig. 3h)

Table I summarizes the results comparing the different approaches for non-rigid registration with *Clean*, *Cropped*, *Holes*, and *Outlier* configurations on the two datasets. Our method (Ours) yields the best general results when considering all point cloud types. The use of self-supervised learning leads to comparable results with respect to supervised learning, especially on the *Custom* dataset. For *Clean* point clouds, our implementation of NeuroMorph obtained the best scores. However, NeuroMorph cannot handle missing parts, and could not be trained with such datasets. The second closest to ours is Ours-RGM followed by BCPD. BCPD obtained scores close to ours in the RMANet dataset, but could not handle *Cropped* clouds. Finally, we highlight how the BCPD-based probabilitic refinement step improves the results of ours and RGM's network, as can be observed by comparing P/D and BCPD results under the *Clean* column.



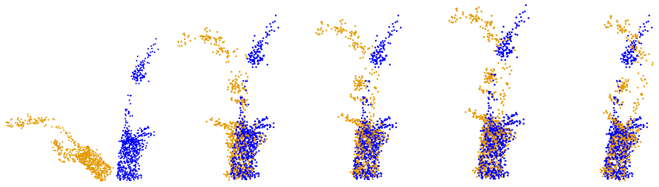
(a) Source (b) Target (c) Pre-trained

(d) Only-Faces (e) Mixed

Figure 5: Registering facial expressions using source (a) and target (b) point clouds. Registration results produced by our model trained on: the Custom dataset (c), fine tuning the weights on faces (d), and fine-tuning on faces plus some clouds from Custom (e). The latter keeps the original performance while better dealing with faces.

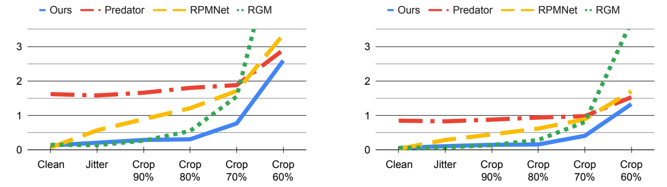
Fig. 5 shows registration results on a real dataset of different facial expressions from TOSCA [44]. Starting from only 7 pairs of scanned faces, we generated 1,000 face pairs using self-supervised learning. In Fig. 5c we used the model pre-trained with the Custom dataset, which can handle the registration but misses details mostly in the mouth and around eyes. We attribute this to the lack of instances of faces (or similar) in the dataset. To improve the results we fine-tuned our model using the *Face* dataset (Fig. 5d). We also fine-tuned it using the *Face* dataset plus some clouds from the *Custom* dataset (Fig. 5e). Although they produce similar results, the latter maintains its performance on the Custom dataset. This shows how the proposed model can be fine-tuned without becoming too specialized on the new dataset.

We present a set of extra experiments in the thesis. They include a sensitivity and ablation study that justifies our choice of architecture and parameters for the proposed neural network. This is followed by discussions about the use of multiple outer-loop iterations and cyclic consistency, the selection of BCPD parameters, training with mixed types of noise, and



(a) Input Clouds (b) Ours (c) Predator (d) RPMNet (e) RGM

Figure 6: Rigid registration produced by various models.



(a) Mean Isometric Rotation Error (MIE)

(b) Mean Anisotropic Rotation Error (MAE)

Figure 7: Rotation errors in degrees, MIE and MAE, comparing different approaches for rigid registration on ModelNet40 dataset on *Clean*, *Jitter*, and different percentages of samples kept for *Crop-Noise*.

generalization to a different number of points during training and evaluation.

B. Rigid Registration

For the rigid registration experiments, we used the ModelNet40 [45] dataset which consists of point clouds from CAD models. We compare against state-of-art works RPMNet [19], RGM [24], and Predator [20]. For each cloud, we randomly rotate by up to 45°around each of the axes and also translate by some random amount in the interval [-0.5, 0.5] along each of the three axes during training and evaluation. The set of point clouds with these random transformations is called *Clean*. The experiments also include a set of *Jitter* point clouds, obtained by adding random Gaussian noise to *Clean* point clouds. The last set of point clouds, *Crop-Noise*, builds on *Jitter* by keeping only a percentage of the source and target points. To evaluate rigid registration, we adopt the mean isotropic error (MIE) and the mean anisotropic error (MAE) largely adoped in the literature (with $\text{MIE}(\mathbf{R})$ and $\text{MAE}(\mathbf{R})$ are in degrees).

Fig. 6 compares the results produced by our method (*Ours*), Predator, RPMNet, and RGM for the rigid alignment of incomplete point clouds with partial overlap. Our method produces better registration of the vase and plant, while other methods cause the partial clouds to cross each other. Fig. 7 compares the rotation errors in all test cases for Predator, RPMNet, and RGM. It shows that our method has smaller errors for all tested cropping levels. We present further analysis and a sensitity study in the thesis.

VII. Conclusion

We presented a learning-based model that can be applied for both non-rigid and rigid registration of point clouds while being robust to noise and missing portions. This was accomplished by adapting and improving a network designed to match sparse features in 2D images to densely match features in 3D point clouds. This is a considerably harder problem performed on unstructured data, as opposed to on a regular image grid. Additionally, the proposed work combines the advantages of deep-learning and probabilistic modeling. To the best of our knowledge, this is the first time such setup is explored in the context of non-rigid registration of 3D point clouds. Our experiments show that our models outperform previous techniques in terms of robustness in a series of challenging scenarios. Improving our model for handling large deformations and point clouds at different scales is an important direction for future exploration. Better integration of the neural network with the probabilistic refinement can further improve performance.

VIII. Publication

The results of this work covering rigid and non-rigid registration were reported in an article entitled *Robust Point-Cloud Registration based on Dense Point Matching and Probabilistic Modeling* published in the *The Visual Computer* journal. The paper can be found here. We encourage the reader to access our source code repository, which provides our official implementation, example results, and the pre-trained models.

References

[1] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, “Simultaneous localization and mapping: A survey of current trends in autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[2] M. Naseer, S. Khan, and F. Porikli, “Indoor scene understanding in 2.5/3d for autonomous agents: A survey,” *IEEE access*, vol. 7, pp. 1859–1887, 2018.

[3] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, “A survey of surface reconstruction from point clouds,” in *Comput. Graph. Forum*, vol. 36, no. 1. Wiley Online Library, 2017, pp. 301–329.

[4] Z. Wang, Y. Wu, and Q. Niu, “Multi-sensor fusion in automated driving: A survey,” *IEEE Access*, vol. 8, pp. 2847–2868, 2019.

[5] J. Liu, H. Li, R. Wu, Q. Zhao, Y. Guo, and L. Chen, “A survey on deep learning methods for scene flow estimation,” *Pattern Recognition*, vol. 106, p. 107378, 2020.

[6] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–606.

[7] W. Feng, J. Zhang, H. Cai, H. Xu, J. Hou, and H. Bao, “Recurrent multi-view alignment network for unsupervised surface registration,” in *CVPR*, 2021, pp. 10 297–10 307.

[8] M. Eisenberger, D. Novotny, G. Kerchenbaum, P. Labatut, N. Neverova, D. Cremers, and A. Vedaldi, “Neuromorph: Unsupervised shape interpolation and correspondence in one go,” in *CVPR*, 2021, pp. 7473–7483.

[9] O. Hirose, “A bayesian formulation of coherent point drift,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[10] A. W. Fitzgibbon, “Robust registration of 2d and 3d point sets,” *Image and vision computing*, vol. 21, no. 13-14, pp. 1145–1153, 2003.

[11] S. Granger and X. Pennec, “Multi-scale em-icp: A fast and robust approach for surface registration,” in *European Conference on Computer Vision*. Springer, 2002, pp. 418–432.

[12] A. Myronenko and X. Song, “Point set registration: Coherent point drift,” *IEEE TPAMI*, vol. 32, no. 12, pp. 2262–2275, 2010.

[13] W. Gao and R. Tedrake, “Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization,” in *CVPR*, 2019, pp. 11 095–11 104.

[14] B. Eckart, K. Kim, and J. Kautz, “Fast and accurate point cloud registration using trees of gaussian mixtures,” *arXiv preprint arXiv:1807.02587*, 2018.

[15] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-icp: A globally optimal solution to 3d icp point-set registration," *IEEE TPAMI*.

[16] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*. Springer, 2010, pp. 356–369.

[17] F. Pomerleau, F. Colas, and R. Siegwart, "A review of point cloud registration algorithms for mobile robotics," 2015.

[18] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *ICCV*, 2019, pp. 3523–3532.

[19] Z. J. Yew and G. H. Lee, "Rpm-net: Robust point matching using learned features," in *CVPR*, 2020, pp. 11 824–11 833.

[20] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, "Predator: Registration of 3d point clouds with low overlap," in *CVPR*, 2021, pp. 4267–4276.

[21] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *ICCV*, 2019, pp. 6411–6420.

[22] X. Bai, Z. Luo, L. Zhou, H. Fu, L. Quan, and C.-L. Tai, "D3feat: Joint learning of dense detection and description of 3d local features," in *CVPR*, 2020, pp. 6359–6367.

[23] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *ICCV*, 2019, pp. 8958–8966.

[24] K. Fu, S. Liu, X. Luo, and M. Wang, "Robust point cloud registration framework based on deep graph matching," in *CVPR*, 2021, pp. 8893–8902.

[25] B. Amberg, S. Romdhani, and T. Vetter, "Optimal step nonrigid icp algorithms for surface registration," in *CVPR*. IEEE, 2007, pp. 1–8.

[26] B. Jian and B. C. Vemuri, "Robust point set registration using gaussian mixture models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1633–1645, 2010.

[27] J. Ma, J. Wu, J. Zhao, J. Jiang, H. Zhou, and Q. Z. Sheng, "Nonrigid point set registration with robust transformation learning under manifold regularization," *IEEE transactions on neural networks and learning systems*, 2018.

[28] X. Liu, C. R. Qi, and L. J. Guibas, "Flownet3d: Learning scene flow in 3d point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 529–537.

[29] H. Mittal, B. Okorn, and D. Held, "Just go with the flow: Self-supervised scene flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 177–11 185.

[30] I. Tishchenko, S. Lombardi, M. R. Oswald, and M. Pollefeys, "Self-supervised learning of non-rigid residual flow and ego-motion," in *Int. Conf. on 3D Vision (3DV)*. IEEE, 2020, pp. 150–159.

[31] W. Wu, Z. Y. Wang, Z. Li, W. Liu, and L. Fuxin, "Pointpwc-net: Cost volume on point clouds for (self-) supervised scene flow estimation," in *ECCV*. Springer, 2020, pp. 88–107.

[32] Y. Kittenplon, Y. C. Eldar, and D. Raviv, "Flowstep3d: Model unrolling for self-supervised scene flow estimation," in *CVPR*, 2021, pp. 4114–4123.

[33] G. Puy, A. Boulch, and R. Marlet, "Flot: Scene flow on point clouds guided by optimal transport," in *ECCV*. Springer, 2020, pp. 527–544.

[34] B. Ouyang and D. Raviv, "Occlusion guided scene flow estimation on 3d point clouds," in *CVPR*, 2021, pp. 2805–2814.

[35] L. Wang, X. Li, J. Chen, and Y. Fang, "Coherent point drift networks: Unsupervised learning of non-rigid point set registration," *arXiv preprint arXiv:1906.03039*, 2019.

[36] L. Wang, J. Chen, X. Li, and Y. Fang, "Non-rigid point set registration networks," *arXiv preprint arXiv:1904.01428*, 2019.

[37] Y. Sahillioğlu, "Recent advances in shape correspondence," *The Visual Computer*, vol. 36, no. 8, pp. 1705–1721, 2020.

[38] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *CVPR*, 2020, pp. 4938–4947.

[39] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM TOG*, vol. 38, no. 5, pp. 1–12, 2019.

[40] G. Mena, J. Snoek, S. Linderman, and D. Belanger, "Learning latent permutations with gumbel-sinkhorn networks," in *ICLR 2018 Conference Track*, vol. 2018. OpenReview, 2018.

[41] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8026–8037.

[42] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *ECCV*, 2018, pp. 725–741. [Online]. Available: http://coma.is.tue.mpg.de/

[43] O. Hirose, "Acceleration of non-rigid point set registration with downsampling and gaussian process regression," *IEEE TPAMI*, vol. 43, no. 8, pp. 2858–2865, 2020.

[44] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Calculus of nonrigid surfaces for geometry and texture manipulation," *IEEE TVCG*, vol. 13, no. 5, pp. 902–913, 2007.

[45] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015, pp. 1912–1920.