Feature-metric Registration: A Fast Semi-supervised Approach for Robust Point Cloud Registration without Correspondences

Xiaoshui Huang
The University of Sydney
Sydney, Australia
Xiaoshui.Huang@sydney.edu.au

Guofeng Mei, Jian Zhang
University of Technology Sydney
Sydney, Australia
Guofeng.Mei@student.uts.edu.au, Jian.Zhang@uts.edu.au

Abstract

We present a fast feature-metric point cloud registration framework, which enforces the optimisation of registration by minimising a feature-metric projection error without correspondences. The advantage of the feature-metric projection error is robust to noise, outliers and density difference in contrast to the geometric projection error. Besides, minimising the feature-metric projection error does not need to search the correspondences so that the optimisation speed is fast. The principle behind the proposed method is that the feature difference is smallest if point clouds are aligned very well. We train the proposed method in a semi-supervised or unsupervised approach, which requires limited or no registration label data. Experiments demonstrate our method obtains higher accuracy and robustness than the state-of-the-art methods. Besides, experimental results show that the proposed method can handle significant noise and density difference, and solve both same-source and cross-source point cloud registration.

1. Introduction

Point cloud registration is a process of transforming different scans of the same 3D scene or object into one coordinate system. Most of the state-of-the-art registration methods [12, 8, 9] minimize a geometric (point coordinate-based) projection error through two processes: correspondences searching and transformation estimation. These two processes alternatively conduct until the geometric projection error is minimum. We strive to solve the point cloud registration because the registration is critical for many tasks such as robotics vision and augment reality.

In the point cloud registration task, looking at Figure 1, the best alignment of point 1 and point 2 is the third column (manually aligned) since the geometric projection error is minimum. However, such best alignment can hardly achieve by using the existing two-process methods since
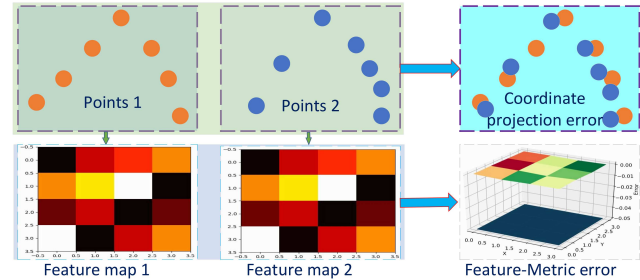

Figure 1. The difference between the coordinate-based registration (top) and feature-metric registration(bottom). The top shows the best alignment can never be achieved since this alignment has no point-point correspondences. The bottom feature-metric registration can achieve the best alignment since it optimizes the feature difference without correspondence and the feature difference is minimum in the best alignment.

there are no point-point correspondences. Even though the soft correspondence [10] provides an alternative way by using weighted point-point correspondences, the behind theory still relies on the point-point correspondences, and it is tricky to decide the weights. Besides, the noise, outliers and density differences existed in the point clouds will make the registration problem more challenging since the point-point correspondences are impaired during the acquisition process [12].

There are plenty of literature focus on solving point cloud registration problem [24, 14, 26, 12, 25, 9]. Roughly, they could generally be divided into two categories: global and local methods. Global methods [24, 14] usually estimate the up-bound and low-bound of the objective function. In contrast, the local methods [26, 12, 25, 9] always care about local alignment. A fundamental block of the existing methods is the iterative computation of correspondences and transformation. Present implementations of this fundamental block have some inherent limitations. (1) They solve the registration tasks by minimizing a geometric projection error. This error metric is sensitive to noise and outliers since the registration relies on point-point correspon-

dences. (2) The pre-trained descriptor training is an independent process to the registration framework. This separated training process can only learn two points which are matched or not matched based on their local regions. This match/non-match descriptor may not be helpful or work for the registration task, especially when the data has noise and outliers. (3) The existing learning-based registration methods [1, 9] rely on huge registration label data which makes the algorithms not practical since the 3D registration label is labour-consuming. These observations indicate that most of the current research work is focusing on optimization strategies or pre-trained match descriptors or supervised learning. Few pieces of research focus on the semi-supervised or unsupervised techniques to smartly transferring the registration ability acquired from aligning one pair to another without searching correspondences.

In this paper, a feature-metric point cloud registration framework is proposed to solve the registration problem by minimizing the projection error on a feature space without the need of searching correspondence. The principle of the proposed method is that the feature difference of two-point clouds should be smallest if they align well. In particular, the proposed method contains two modules: encoder and multi-task branches. The encoder module learns to extract feature for input point clouds. For the multi-task branches module, the first branch is an encoder-decoder neural network which is designed to train the encoder module in an unsupervised manner. The goal of this branch is that the features of two copies of the same point cloud should be different if they have a spatial transformation change and the features must be the same if the transformation change is removed. The second branch is a feature-metric registration which estimates a transformation matrix between two input point clouds by minimizing the feature difference.

The contributions of this work are mainly two aspects:

- (1) A feature-metric point cloud registration framework is proposed to solve the registration without searching correspondences.

- (2) A semi-supervised approach is proposed to train the registration framework. This solution enables the feature network to learn to extract a distinctive feature for registration with limited or without registration label data.

2. Related works

2.1. Optimisation-based registration methods

Before deep learning is prevalent, most of point cloud registration methods [23, 14, 18, 3] use two-stage optimisation-based algorithms: correspondence searching and transformation estimation. Most of these methods focus their research on estimating the transformation under differ-ent variations, such as noise and outliers. [23] uses a truncated least-squares cost to solve the extreme outliers problem in polynomial time. [14] claims that descriptor-based correspondence estimation becomes unreliable in noisy and contaminated setting, and proposes a randomized algorithm for robust point cloud registration without correspondences estimation. [8] proposes a probabilistic registration method to formulate the E step as a filtering problem and solve it using advances in efficient Gaussian filters. [18, 3] provide very good review about point cloud registration algorithms. A discussion of full literature of these methods is beyond the scope of this paper. Although there have endured a great development on optimisation-based registration, most of the existing methods use coordinate-based Euclidean distance metric to evaluate the projection error. The limitation of the coordinate-based Euclidean distance metric is sensitive to noise and outliers.

2.2. Feature learning methods for registration

Different to the classical optimization-based registration methods, numerous recent registration methods [26, 4, 9] are tried to use the deep neural network to learn a robust descriptor. The feature is then used to select K pairs of point correspondences. Finally, the RANSAC is applied to reject the correspondence outliers and solve the point cloud registration problem. [26] uses AlexNet to learn a 3D feature from an RGB-D dataset. [4] proposes a local PPF feature by using the distribution of neighbour points and then input into the network for deep feature learning. [9] proposes a rotation-invariant hand-craft feature and input into a deep neural network for feature learning. All these methods are using deep learning as a feature extraction tool. Their goals are developing sophisticated strategies to learn a robust feature so that the robust correspondences can be estimated. The major limitation of this kind of methods is that they use a separated training process to learn a stand-alone feature extraction network. The learned feature network is to determine point-point matching. Different to this kind of methods, we use the neural network to learn the feature for the final registration directly and use the feature-metric projection error to estimate the transformation relationship.

2.3. End-to-end learning-based registration methods

The basic idea of end-to-end learning methods is to transform the registration problem into a regression problem. The recent methods [25] tries to learn a feature from the point clouds to be aligned and then regresses the transformation parameters from the feature. [21] proposes a registration network to formulate the correlation between source and target point sets and predicts the transformation by using the defined correlation. [5] proposes an auto-encoder registration network for localisation, which
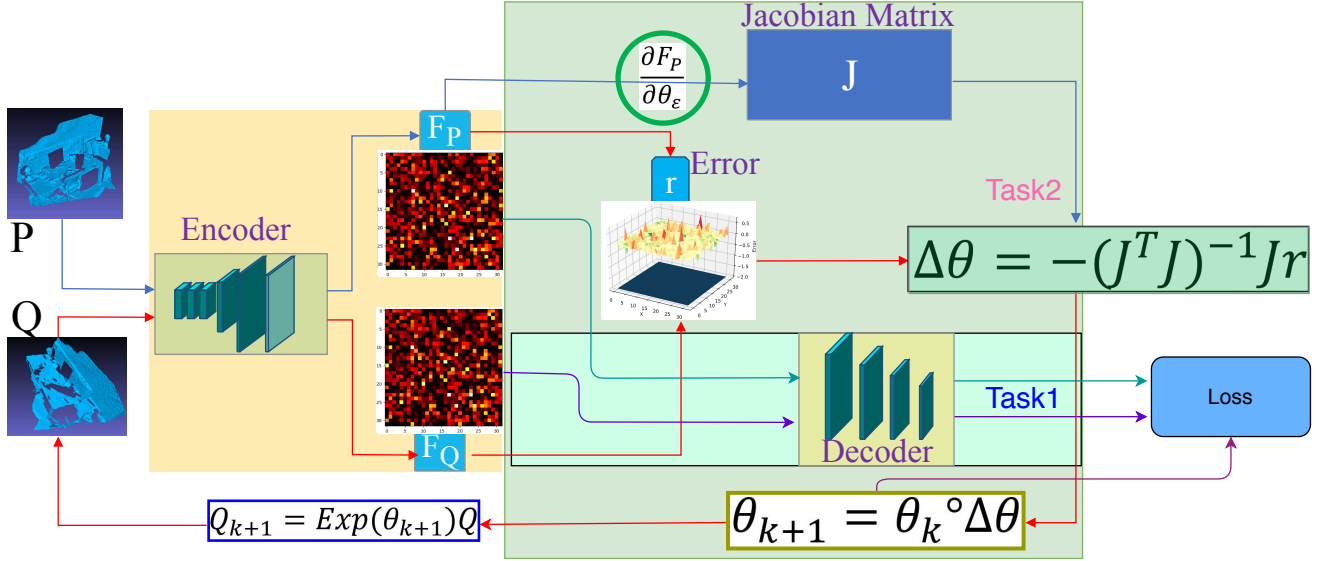
Figure 2. **High-level Overview of the proposed registration framework**. (1)- **Encoder** extracts the features for two input point clouds (**P** and **Q**). (2)- Then, a multi-task semi-supervised neural network aims to solve the registration problem without correspondence. In the *Task1*, **Decoder** decodes the features. The whole encoder-decoder branch trains the encoder network in a unsupervised manner. In the *Task2*, **feature-metric projection error (r)** is calculated based on two input features (F_P and F_Q). The feature error is input into a non-linear optimisation algorithm to estimate transformation increment ($\triangle\theta$) and the transformation parameters are updated. Using the updated transformation parameters (θ_{k+1}), the input point cloud **Q** is transformed and the whole process iteratively runs.

combines super points extraction and unsupervised feature learning. [15] proposes a keypoint detection method and estimates the relative pose simultaneously. [1] proposes a method to use the pointnet to solve the registration task, while its limitation is that category labels are required to initialize the network which is unrealistic in real application. In a small summary, the current end-to-end learning methods regard transformation parameter estimation as a black box, and the distance metric is measured in the coordinate-based Euclidean space, which is sensitive to noise and density difference. Different from the present works, we propose a semi-supervised registration framework to learn to extract distinctive features and solve the transformation parameter estimation by minimising a feature-metric error without correspondences.

In this paper, we investigate how to fused leverage both the classical non-linear optimisation theory and deep learning to directly estimate the transformation parameters without correspondences. In the end, a new semi-supervised feature-metric registration framework is proposed to customise the feature learning for the registration. This method is significantly different from the classical registration pipeline that alternatively solves the correspondence estimation and transformation parameter estimation. At the same time, instead of discarding the conventional mathematical theory and replacing the registration process by a whole black-box neural network, we would like to leverage both advantages by combing the classical

registration mathematical theories with deep learning techniques. Furthermore, unlike conventional registration methods that minimise geometric projection error, our approach minimises feature-metric projection error.

3. Feature-metric Point Cloud Registration

In this section, firstly, the problem formulation and overview of the registration framework are described. Secondly, we explain the *Encoder* module in details. Thirdly, we show how to learn a distinctive feature and solve the registration in multi-task streams. Fourthly, the loss functions are detailed explained.

3.1. Problem Formulation

Given two point clouds $P \in \mathbb{R}^{M \times 3}$ and $Q \in \mathbb{R}^{N \times 3}$, the goal of registration is to find the rigid transformation parameters g (rotation matrix $R \in \mathcal{SO}(3)$ and translation vector $t \in \mathbb{R}^3$) which best aligns the point cloud Q to P as shown below:

$$\underset{R\in\mathcal{SO}(3),t\in\mathbb{R}^3}{\arg\min} \|r(F(P),F(RQ+t))\|_2^2 \quad (1)$$

where $r(F(P),F(RQ+t)) = \|F(P)-F(RQ+t)\|_2$ is the feature-metric projection error between P and transformed Q. $F(P) \in \mathbb{R}^K$ is the feature of the point cloud P, K is the feature dimension (1024 is used in the experiments), F is a feature extraction function learned by a *Encoder* module.

To solve the above equation (1), we propose a feature-metric registration framework to integrate both the merits of the classical non-linear algorithm and unsupervised learning techniques. The framework can be trained in a semi-supervised or unsupervised manner. Figure 2 shows an overview of the algorithm. Firstly, two rotation-attentive features are extracted for two input point clouds. Secondly, the features are input into a multi-task module. In the first branch (Task1), a decoder is designed to train the *Encoder* module in an unsupervised manner. In the second branch, the projection error r is calculated to indicate the difference between two input features, and the best transformation is estimated by minimizing the feature difference. The transformation estimation iteratively runs, and the transformation increment($\triangle\theta$) of each step is estimated by running the inverse compositional (IC) algorithm [2]:

$$\triangle\theta = (J^T J)^{-1}(J^T r) \qquad (2)$$

where r is the feature-metric projection error and $J = \frac{\partial r}{\partial \theta}$ is the Jacobian matrix of r with respect to transformation parameters (θ). Refer to [2], direct computation of J in 2D images is extremely costly. The approach applied in the classical 2D images, the chain rule is utilised to estimate the Jacobian into two partial terms: the gradient of the image and the Jacobian of warp. However, this approach is not working for unordered 3D point clouds since there is not grid structure to allow us to compute the gradient of the 3D point cloud in x, y and z directions [1].

To efficiently compute the Jacobian matrix, instead of calculating the Jacobian matrix by stochastic gradient approach, refer to [1], we utilise a different finite gradient to compute the Jacobian matrix:

$$J_i = \frac{\partial F_P}{\partial \theta_\xi} = \frac{F(R_i P + t_i) - F(P)}{\xi} \qquad (3)$$

where $\xi = (R_i, t_i)$ is the infinitesimal perturbations of the transformation parameters during the iteration. In this work, six motion parameters are given for the perturbations that three angle parameters (v_1, v_2, v_3) for the rotation and three jitter parameters (t_1, t_2, t_3) for the translation. Refer to [1], the small fixed value $(2*e^{-2})$ on all the iteration generate the best results. After several iteration time (10 is used for the experiments), the proposed method outputs a final transformation matrix g_{est} (R and t) and feature-metric projection error r_{est}.

The critical components of the proposed framework are the *Encoder* module and multi-task branches. We will now provide details about these components.

3.2. Encoder

The encoder module aims to learn a feature extraction function F, which can generate a distinctive feature for input point clouds. The main principle of designing the encoder network is that the generated feature should be rotation-attentive so that it can reflect the rotation difference during the transformation estimation. Refer to Point-Net [19], the feature is extracted by two *mlp* layers and a max-pool layer. We discard the input transform and feature transform layers to make the feature aware of the rotation difference.

3.3. Multi-task Branches

After the features extracted, the next steps are feature learning and point cloud transformation estimation. The estimation is finalized by directly minimizing the feature-metric projection error and does not need search correspondences. This design will learn suitable features for iterative optimization-based methods on feature space.

3.3.1 Encoder-Decoder Branch (Task1):

After the encoder module generating distinctive features, inspired by [11], we use a decoder module to recover the features back into 3D point clouds. This encoder-decoder branch can be trained in an unsupervised way which helps the *Encoder* module to learn to generate distinctive feature being aware of rotation difference. For two rotated copies $PC1$ and $PC2$ of a point cloud, the principle of this branch is that the *Encoder* module should generate different features for $P1$ and $P2$ and the decoder can recover the different features back to their correspondent rotated copies of the point cloud. This principle guarantees the success of unsupervised learning in training a distinctive feature extractor for the registration problem.

The decoder block consists of four-layer fully-connected layers and activates by LeakyReLU. The output of decoder module is the same dimension to the input point cloud.

3.3.2 Feature-metric Registration Branch (Task2):

To solve the registration problem, as discussed in the *problem formulation* section and showed in Figure 2, we estimate the transformation parameters by using the inverse compositional algorithm (non-linear optimisation) to minimize a feature-metric projection error. The feature-metric projection error is defined as

$$r = \|F(P) - F(g \cdot Q)\|_2^2 \qquad (4)$$

where $F(.) \in \mathbb{R}^K$ is a global feature of point cloud (P or $g \cdot Q$), and g is the transformation matrix (R and t).

To have a better intuition about what does the feature network extracted during the non-linear optimisation process, we visualise the feature maps and the iterative process in Figure 3. We reshape the global feature maps into a square matrix and show it into as an image. Figure 3 shows the

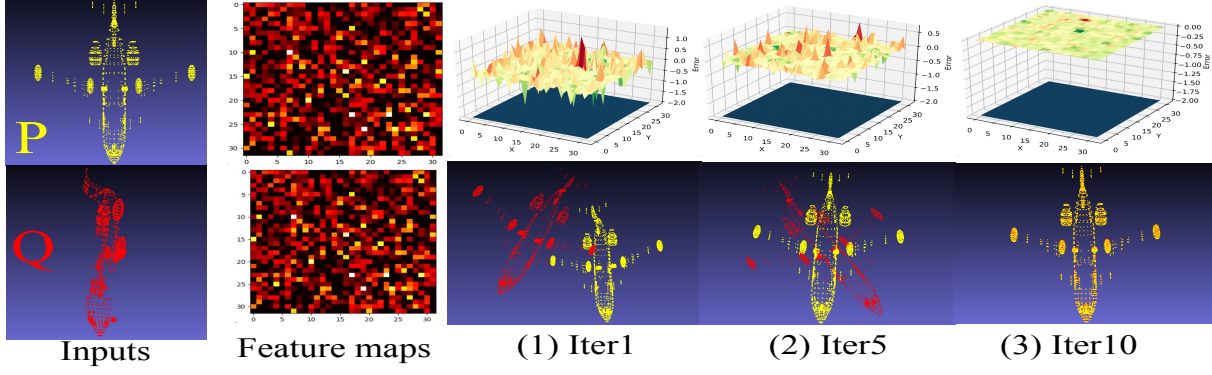| | | (1) Iter1 | (2) Iter5 | (3) Iter10 |
| Inputs | Feature maps | | | |

Figure 3. The feature maps generated by the feature extraction network and the feature difference during the iteration process. At the first iteration, the feature map difference (top) is large between P and transformed Q, and their alignment (bottom) is not well. At the 10th iteration, the feature map difference of point cloud P and transformed Q becomes extremely small and the alignment is almost perfect.

feature maps of point cloud P and transformed point cloud Q, and feature map differences in the first iteration, 5th and the final 10th iteration. Looking at the last three columns of Figure 3, the feature map difference (top row) becomes smaller when the alignment becomes more accurate (bottom row).

3.4. Loss functions

The key task for training is the *Encoder* module which used to extract a rotation-attentive feature. We propose two loss functions to provide a semi-supervised framework. It can be easily extended to unsupervised framework by directly ignoring the supervised geometric loss.

3.4.1 Chamfer loss

The encoder-decoder branch can be trained in an unsupervised manner. Refer to [11], Chamfer distance loss is used:

$$loss_{cf} = \sum_{p \in A} \sum_{i=1}^{N} \min_{q \in S^*} \| \phi_{\theta_i}(p; x) - q \|_2^2 \qquad (5)$$

$$+ \sum_{q \in S^*} \min_{i, i \text{ in } 1..N} \min_{p \in A} \| \phi_{\theta_i}(p; x) - q \|_2^2$$

where $p \in A$ is a set of points sampled from a unit square $[0, 1]^2$, x is a point cloud feature, ϕ_{θ_i} is the ith component in the MLP parameters, S^* is the original input 3D points.

3.4.2 Geometric loss

The goal of the geometric loss function is to minimize the difference between the estimated transformation matrix (g_{est}) and the ground-truth transformation matrix (g_{gt}). To balance the influence of translation and rotation, inspired by [16] in solving 2D image alignment, we exploit the 3D point error (PE) between the different transformation of g_{est} and

g_{gt}. Therefore, the loss function is

$$loss_{pe} = \frac{1}{M} \sum_{i=1}^{M} \| f(g_{est} \cdot P) - f(g_{gt} \cdot P) \|_2^2 \qquad (6)$$

where P is a point cloud, and M is its total point number.

The final loss function of semi-supervised training is:

$$loss = loss_{cf} + loss_{pe} \qquad (7)$$

For unsupervised training, we only use $loss_{cf}$.

4. Experiments

4.1. Datasets

ModelNet40 To systematically evaluate highly varying object motions, we use ModelNet40 dataset. The ModelNet40 [22] contains 3D CAD models from 40 categories, which is a widely used dataset to train 3D deep learning network. The dataset is split into two parts: 20 categories for training and same-category testing, and another 20 categories for cross-category testing. In the same-category experiments, we split the dataset into $8 : 2$, where 80% are used for training and 20% for testing. The source point cloud is the vertices from ModelNet40, which is normalized into a unit box at the origin $[0, 1]^3$. Then, the target point cloud is generated by applying a rigid transformation to the source point cloud. The registration performance is evaluated and compared by using the source and target point clouds. During the training, the rigid transformation T_{gt} is randomly generated, where the rotation is in the range $[0, 45]$ degrees with arbitrarily chosen axes and translation is in the range $[0, 0.8]$. For a fair comparison, initial translations for testing are in the range $[0, 0.8]$ and initial rotations are in the range $[0, 80°]$.

7Scene [20] is a widely used benchmark registration dataset by a Kinect RGB-D camera on the indoor environment. It contains seven scenes including Chess, Fires,
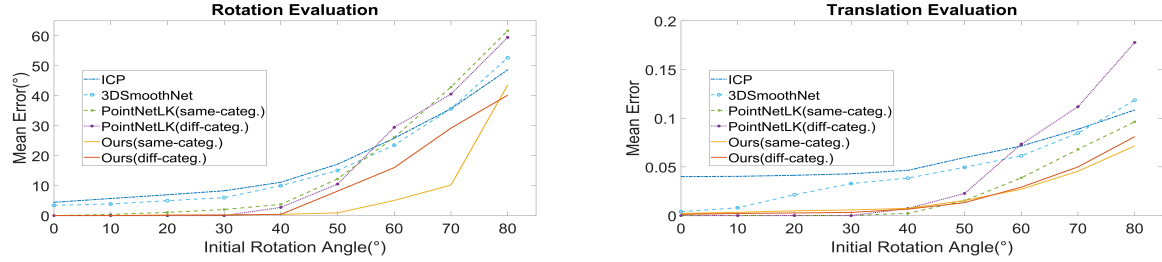
Figure 4. Comparison of the same and different categories under different angles' rotation. The results show our method obtains better accuracy than the classic optimisation method, feature learning method and deep learning registration method.

Heads, Office, Pumpkin, Redkitchen and Stairs. Following 3DMatch [26], we project the depth images into point clouds and fuse multi-frame depth thorough truncated signed distance function(TSDF) fusion. The dataset is divided into 296 scans for training and 57 scans for testing. The rotation is initialized in the range of $[0, 60°]$ and the translation is initialized in the range of $[0, 1.0]$ both for training and testing.

4.2. Baselines

We implemented the following baselines:

(1) **LM-ICP** [7]: To compare with the classic registration method, we use the Levenberg–Marquardt (LM)-based iterative closest point (LM-ICP).

(2) **3DSmoothNet + RANSAC** [9]: To compare with the feature learning methods, which combine a learned feature and a classical registration method, we select the state-of-the-art 3DSmoothNet +RANSAC as our comparison method. In the comparison experiments, 5K points are sampled.

(3) **PointNetLK** [1]: To compare with the registration method of using deep learning, we use PointnetLK, which combines Pointnet with Lucas-Kanade algorithm. We train the PointnetLK using the same way to our method.

Evaluation Criteria. Following the [1][17], the evaluation is performed by calculating the difference of the estimated transformation parameters (g_{est}) against the ground truth (g_{gt}). The angular is calculated as $\theta = log(R)$ and the angular error can be calculated as Euclidean distance between θ_{est} and θ_{gt}. The translational error is calculated as the Euclidean distance between t_{est} and t_{gt}. The $RMSE$ represents the root-mean-square error between g_{est} and g_{gt}.

4.3. Evaluation on ModelNet40.

For network training, PointNetLK initializes the feature extraction network by a standard classification task on ModelNet40 and fine-tune the neural network for 400 epochs. 3DSmoothNet initializes the network using 16-bin pre-trained model and fine-tunes on the training dataset of ModelNet40. Our method does not rely on any classification labels and trains in an end-end stream. The method is

implemented in PyTorch and trained on an NVIDIA P5000.

Comparison Results: Figure 4 summarizes the comparison results on the dataset with different initial rotation angles and random translation. Compared to all baseline methods, our method achieves the overall best performance across different initial angles. Especially, the proposed method shows tiny registration error before $60°$, which is less than 0.02.

Looking deeper into the experimental results in Figure 4, the classical ICP is sensitive to the initial angles so that the error increase when the initial angles grow. Interestingly, although the 3DSmoothNet learns a rotation-invariant descriptor, the registration error still goes up when the initial angle increases. The interpretation contains two folds. 1) The feature learning is separated from the registration goal, and the learned model is only suitable for point cloud matching but not perfect for the registration task. 2) The data in our experiments use the vertexes directly from the CAD models, which contain a lot of similar local structures — the local feature learning for matching faces an ambiguous problem. Instead of learning indirect features for matching, our method solves the registration problem by directly learning a rotation-attentive feature for point cloud registration. Besides, our method obtains apparent better performance than PointNetLK in large angles ($30° - 80°$). The interpretation is that our semi-supervised solution can generate much better features, which can estimate a better registration solution.

Density Difference: To demonstrate our method can handle significant density difference, one of the original point cloud (source) was randomly removed 90% points to generate ten times' density difference. The left column of Figure 5 shows the results of density difference, which demonstrates that the proposed method achieves better registration results in high-density difference. To visually show the results, we select the initial rotation angle as $60°$. The top row of Figure 6 shows that the proposed method can align accurately on the large density difference (See the red points align well with the CAD model). However, 3DSmoothNet (green points) and PointNetLK(light blue) face challenging on aligning these point clouds with a high-
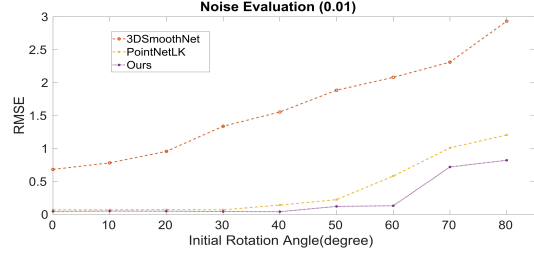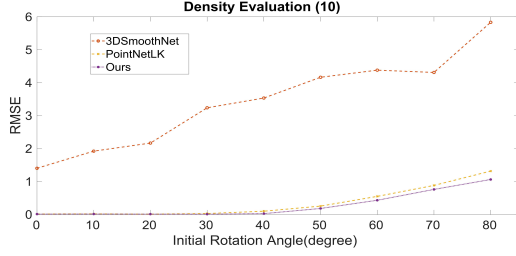
Figure 5. Comparison results with the impact of density difference and noise. The left column shows the comparison results on 10 times density difference. The right column shows the comparison results under Gaussian noise with signal-to-noise ratio(SNR) 0.01.

density difference. The third column of the first row in Figure 6 shows the challenging density difference case.
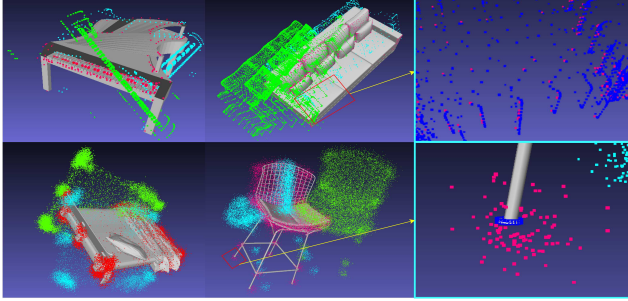

Figure 6. Ablation study on high density difference (Top) and high nosie (bottom). Our method (red) shows obviously better accuracy than 3DSmoothNet (green) and PointnetLK (light blue). The blue points in the last column are the original points without variants.

Large Noise: We also conduct experiments on noisy data. The right column of Figure 5 shows the registration results with Gaussian noise ratio. Although the point clouds are impacted by different noise, our method obtains the lowest root-mean-square error of transformation matrix difference [6], which demonstrates the proposed method always obtain high robustness under noise variation. The bottom row of Figure 6 shows a visual comparison results where the data is impacted by Gaussian noise with a ratio of 0.02. The result shows the proposed method can align accurately on the noisy data. The right column of the bottom row in Figure 6 demonstrates that the noise is large and challenging. Our method can overcome this large noise while the compared methods fail.

Semi-supervised *VS* Unsupervised To investigate the performance of the unsupervised encoder-decoder branch, we train the *Encoder* module using by only considering the *Chamfer loss*. Then, both semi-supervised and unsupervised are evaluated on the ModelNet40 with the same experiment set before. Figure 7 shows that the unsupervised method can always obtain better results than the semi-supervised method. These experimental results show the proposed feature-metric registration framework can have a wide application since no labels are required to train the neural network. In this paper, to make a fair compari-
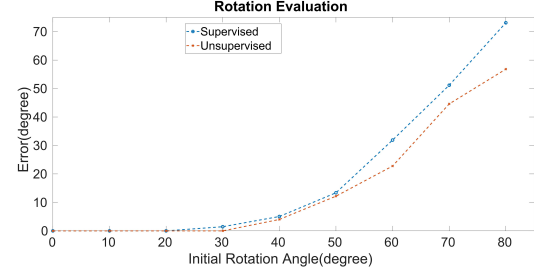

Figure 7. Comparison of training the model in a semi-supervised or unsupervised way. The results show that the unsupervised training obtains better performance.

son with PointNetLK, most experiments are conducted in a semi-supervised manner. The better results also demonstrate the unsupervised branch improve the registration robustness and accuracy.

4.4. Evaluation on 7Scenes

Since there are no category labels available in 7Scenes dataset, we directly train the PointNetLK without feature network initialization. Our method is trained directly in an unsupervised manner. The pre-trained model of 16 bin is selected for 3DSmoothNet since this model is already trained on 3DMatch[26] dataset, which includes *7Scene* dataset. We evaluate the rotation angle and translation difference between algorithm estimation and groundtruth and compare the registration performance with state-of-the-art methods.

Comparison Results: Table 1 summarizes the performance comparison results on 7Scene. Our method outperforms the classic optimization method, feature extraction method and the recent deep learning method. We believe this is due to the unsupervised branch can train the feature extraction network to generate a distinctive feature to understand the point cloud, and the feature-metric registration framework provides an effective way to solve the registration problem.

Partial overlap: To compare the performance on the partial overlap, we manually remove part of the scan (top row of Figure 10) or some objects inside the scans (bottom row of Figure 10). Figure 10 shows that the proposed method can handle partial overlap problem, and obtain bet-
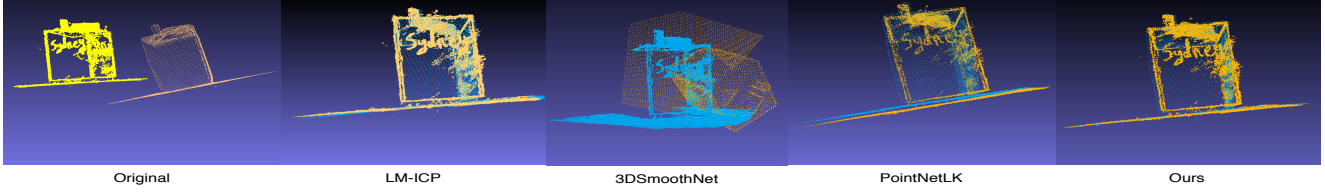
Figure 8. Visually comparison results in solving cross-source point clouds registration.
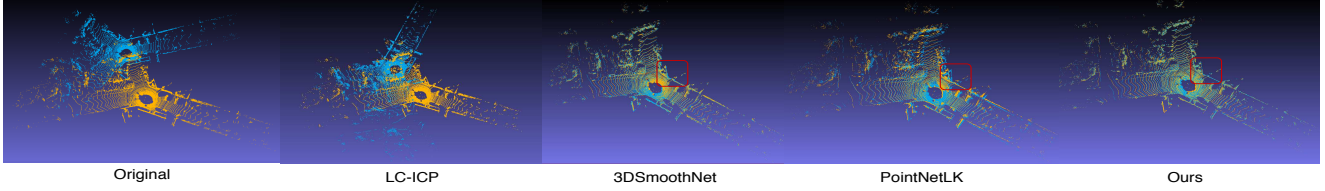

Figure 9. Visually comparison results in solving sparse outdoor Lidar point clouds registration.

Method	LM-ICP	[9]	[1]	Ours
Rot.(°)	12.7	1.2	15.3	0.004
Trans.	0.96	0.05	0.92	0.002

Table 1. Quantitative comparisons on the 7Scene dataset with the classical optimisation method (LM-ICP), learned feature method [9](3DSmoothNet) and recent deep learning method [1](PointNetLK)
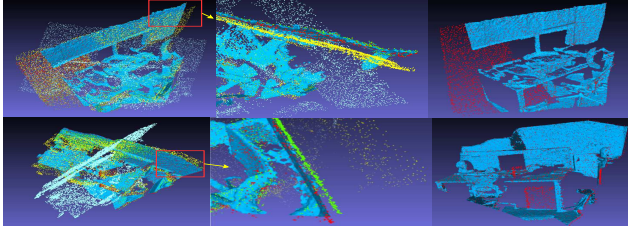
ter performance.

Figure 10. Ablation study on partial overlap. Our method (red) shows obviously better accuracy than LM-ICP (yellow), 3DSmoothNet (green) and PointnetLK (light blue). The last column shows the partial overlap.

Generalize to cross sources: Recently, the 3D sensor hardware and 3D reconstruction software have undergone tremendous changes. Cross-source point clouds are data from different types of image sensors. As 3D data become widely existed, data fusion of cross-source point clouds can leverage different benefits in different image sensors. According to [12, 13], cross-source point cloud registration is challenging since the data contain large variations such as large noise and outliers, density difference. To demonstrate the performance of our method in solving the cross-source point clouds, following [12], we capture cross-source point clouds by using Kinect and RGB camera + 3D reconstruction software and remove the scale variation. Figure 8 shows our method can also handle the cross-source point

cloud registration and obtain a better result than the recent state-of-the-art registration methods.

Generalize to outdoor point clouds: Figure 9 shows results on sparse outdoor Lidar point cloud registration using KITTI point cloud sequences. Our method obtains better results than other methods as shown in the redbox.

4.5. Time complexity

We also run the running time comparison on point clouds with around 12K points. Table 2 shows that our method achieves eight times faster than LM-ICP, seven times faster than 3DSmoothNet and two times faster than PointNetLK. We find that our method can usually achieve the best alignment within five iterations.

Method	LM-ICP	3DSmoothNet	PointNetLK	Ours
Time(s)	10.8	9.6	2.4	1.3

Table 2. Running time comparison on point clouds with around 12K points.

5. Conclusion

In this paper, we propose a feature-metric framework to solve the point cloud registration, and the framework can be trained using a semi-supervised or unsupervised manner. Experiments demonstrate that the proposed method achieves better accuracy than the classical registration method, state-of-the-art feature learning method and deep learning registration method. Our ablation studies show that the proposed method can handle significant noise, density difference and partial overlap. Besides, experiments show that our method can also achieve robust, accurate and fast registration by training in an unsupervised manner. We believe the feature-metric framework will provide high potential for both academic and industrial applications.

References

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019.

[2] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[3] Liang Cheng, Song Chen, Xiaoqiang Liu, Hao Xu, Yang Wu, Manchun Li, and Yanming Chen. Registration of laser scanning point clouds: A review. *Sensors*, 18(5):1641, 2018.

[4] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018.

[5] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640, 2017.

[6] Georgios Dimitrios Evangelidis and Radu Horaud. Joint alignment of multiple point sets with batch and incremental expectation-maximization. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1397–1410, 2017.

[7] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and vision computing*, 21(13-14):1145–1153, 2003.

[8] Wei Gao and Russ Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11095–11104, 2019.

[9] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[10] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998.

[11] Thibault Groueix, Matthew Fisher, Vladimir Kim, Bryan Russell, and Mathieu Aubry. Atlasnet: A papier-mâché approach to learning 3d surface generation. In *CVPR 2018*, 2018.

[12] Xiaoshui Huang, Jian Zhang, Lixin Fan, Qiang Wu, and Chun Yuan. A systematic approach for cross-source point cloud registration by preserving macro and micro structures. *IEEE Transactions on Image Processing*, 26(7):3261–3276, 2017.

[13] Xiaoshui Huang, Jian Zhang, Qiang Wu, Lixin Fan, and Chun Yuan. A coarse-to-fine algorithm for matching and registration in 3d cross-source point clouds. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2965–2977, 2017.

[14] Huu M Le, Thanh-Toan Do, Tuan Hoang, and Ngai-Man Cheung. Sdrsac: Semidefinite-based randomized approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 124–133, 2019.

[15] Weixin Lu, Guowei Wan, Yao Zhou, Xiangyu Fu, Pengfei Yuan, and Shiyu Song. Deepicp: An end-to-end deep neural network for 3d point cloud registration. *arXiv preprint arXiv:1905.04153*, 2019.

[16] Zhaoyang Lv, Frank Dellaert, James M Rehg, and Andreas Geiger. Taking a deeper look at the inverse compositional algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4581–4590, 2019.

[17] G Dias Pais, Pedro Miraldo, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, and Rama Chellappa. 3dregnet: A deep neural network for 3d point registration. *arXiv preprint arXiv:1904.01701*, 2019.

[18] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015.

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

[20] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon. Scene coordinate regression forests for camera relocalization in rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2930–2937, 2013.

[21] Lingjing Wang, Jianchun Chen, Xiang Li, and Yi Fang. Non-rigid point set registration networks. *arXiv preprint arXiv:1904.01428*, 2019.

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

[23] Heng Yang and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. *arXiv preprint arXiv:1903.08588*, 2019.

[24] Jiaolong Yang, Hongdong Li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE transactions on pattern analysis and machine intelligence*, 38(11):2241–2254, 2015.

[25] Zhenpei Yang, Jeffrey Z. Pan, Linjie Luo, Xiaowei Zhou, Kristen Grauman, and Qixing Huang. Extreme relative pose estimation for rgb-d scans via scene completion. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[26] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.