

ImLoveNet: Misaligned Image-supported Registration Network for Low-overlap Point Cloud Pairs

Honghua Chen
chenhonghuacn@gmail.com
Nanjing University of Aeronautics
and Astronautics
Nanjing, China

Zeyong Wei
weizeyong1@gmail.com
Nanjing University of Aeronautics
and Astronautics
Nanjing, China

Yabin Xu
yabinxu007@gmail.com
Nanjing University of Aeronautics
and Astronautics
Nanjing, China

Mingqiang Wei
mingqiang.wei@gmail.com
Nanjing University of Aeronautics
and Astronautics
Nanjing, China

Jun Wang
wjun@nuaa.edu.cn
Nanjing University of Aeronautics
and Astronautics
Nanjing, China

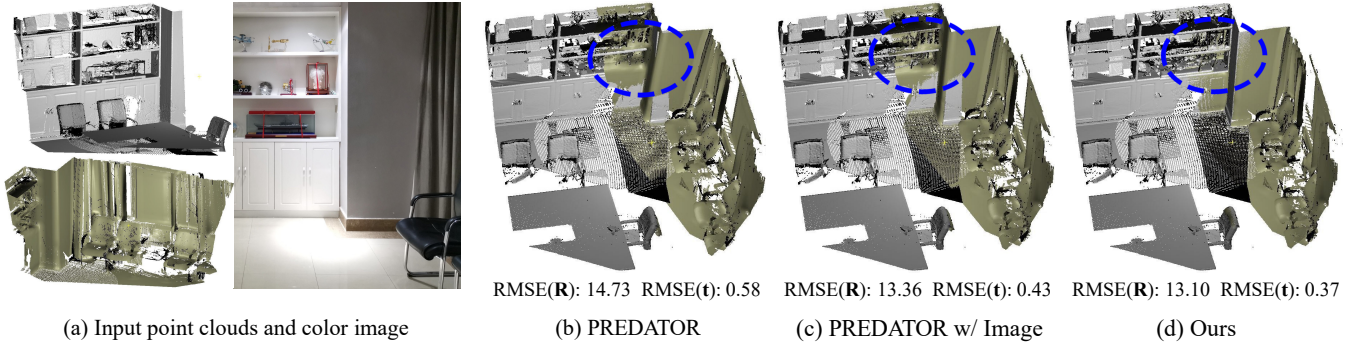


Figure 1: Our method is able to well align low-overlap point cloud pairs with the assistance of a misaligned intermediate color image. Such a new configuration for the point cloud registration is also useful to enhance the performance of other registration model, like PREDATOR [Huang et al. 2021].

ABSTRACT

Low-overlap regions between paired point clouds make the captured features very low-confidence, leading cutting edge models to point cloud registration with poor quality. Beyond the traditional wisdom, we raise an intriguing question: Is it possible to exploit an intermediate yet misaligned image between two low-overlap point clouds to enhance the performance of cutting-edge registration models? To answer it, we propose a misaligned image supported registration network for low-overlap point cloud pairs, dubbed ImLoveNet. ImLoveNet first learns triple deep features across different modalities and then exports these features to a two-stage classifier, for progressively obtaining the high-confidence overlap

region between the two point clouds. Therefore, soft correspondences are well established on the predicted overlap region, resulting in accurate rigid transformations for registration. ImLoveNet is simple to implement yet effective, since 1) the misaligned image provides clearer overlap information for the two low-overlap point clouds to better locate overlap parts; 2) it contains certain geometry knowledge to extract better deep features; and 3) it does not require the extrinsic parameters of the imaging device with respect to the reference frame of the 3D point cloud. Extensive qualitative and quantitative evaluations on different kinds of benchmarks demonstrate the effectiveness and superiority of our ImLoveNet over state-of-the-art approaches.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SIGGRAPH '22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada

© 2022 Association for Computing Machinery.

ACM ISBN 978-1-4503-9337-9/22/08...\$15.00

<https://doi.org/10.1145/3528233.3530744>

CCS CONCEPTS

• Computing methodologies → Shape modeling; Point-based models.

KEYWORDS

Point cloud registration, low overlap, cross-modality feature, deep learning

ACM Reference Format:

Honghua Chen, Zeyong Wei, Yabin Xu, Mingqiang Wei, and Jun Wang. 2022. ImLoveNet: Misaligned Image-supported Registration Network for Low-overlap Point Cloud Pairs. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings)*, August 7–11, 2022, Vancouver, BC, Canada. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3528233.3530744>

1 INTRODUCTION

With the rapid development of new 3D acquisition technologies, 3D sensors are becoming increasingly available and affordable, including various types of 3D laser scanners (or LiDAR), and RGB-D cameras (such as Microsoft Kinect, Intel RealSense, and Apple Truth Depth Camera). This benefits to acquire more reliable 3D information, enabling better understanding of surrounding large-scale environment for machines. Consequently, these sensors greatly increase the expectations on the performance of point cloud registration. Point cloud registration aims at finding a rigid transformation to align a pair of point clouds. It has wide applications from low-level 3D reconstruction to higher-level scene analysis or applied robotics.

Recent work has made substantial and impressive progress in automatic point cloud registration with deep learning, like [Choy et al. 2020; Huang et al. 2021; Pais et al. 2020; Wang and Solomon 2019a,b; Xu et al. 2021a]. However, when the overlap-region ratio between two point clouds falls below 30% or less, the registration performance of these methods deteriorates rapidly [Huang et al. 2021; Xu et al. 2021a]. This is because it is difficult to perceive reliable corresponding information from the entire point cloud pairs with limited overlap area, which is very common in many practical scenarios. For example, data acquisition for large-scale objects is often time-consuming (Completely scanning an object of size $10m \times 5m$ consumes over 12 hours, by a structured-light scanner), so practitioners aim for a low number of scans with only the necessary overlap. Also, it may be difficult to ensure a high overlap ratio for a moving scanner, when suffering from occlusions, missing frames, or large deviation of the scanning angle of view.

One interesting thing is that if an intermediate color image depicting the rough overlapping region is given, it will be easier for a human operator to register two low-overlap scans. Intuitively, with the help of the intermediate image, i) overlapping information is significantly clearer and ii) 2D image can provide certain underlying 3D-aware geometry information, even though the information is cross-modality. For example, we have witnessed many impressive 3D-related tasks on a single image [Qi et al. 2018; Saha et al. 2021; Wang et al. 2020]. Such kind of 3D-aware information is consistent with the real 3D feature from the point cloud to a certain extent. In addition, an intermediate image between two point clouds is easy to obtain, and we do not need to know extrinsic parameters of the imaging device with respect to the reference frame of the 3D point cloud, namely a misaligned intermediate image. All these encourage us to employ such an auxiliary image to enhance pairwise 3D point cloud registration.

To this end, we propose a misaligned image-supported registration network for low-overlap point cloud pairs, dubbed ImLoveNet, as illustrated in the example shown in Figure 2. In order to fully utilize both color image and point cloud information, our network learns triple deep features, composed of the 3D feature for the

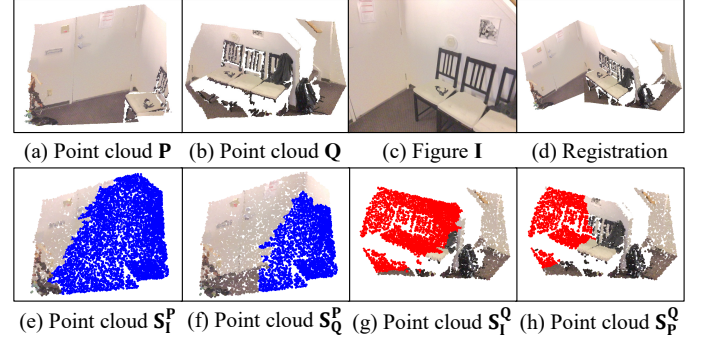


Figure 2: Illustration of low-overlap point cloud registration (overlap ratio $< 20\%$) with an intermediate image. (a) and (b) are two point clouds to register. (c) is an intermediate color image. (d) is the registration result of our ImLoveNet. S_I^P (blue points) and S_I^Q (red points) denote the points which can be projected to the image space. S_Q^P (blue points) and S_P^Q (red points) are overlap region between the two point clouds. We append texture color on each point only for better visualization.

point cloud, and 2D feature and simulated 3D feature for image. These learned features are then fused and applied to progressively detect the intersecting area between the input two point clouds. Finally, soft correspondences can be well established on the predicted overlap region, which leads to the quality rigid transformation parameters for final registration. Experiments and detailed analysis show that our approach achieves state-of-the-art performance (see from Figure. 1) compared with previous algorithms.

Our main contributions are three-fold:

- We design a new point cloud registration network by collaborating cross-modality information, which shows clear improvements over the state-of-the-art methods.
- We extract triple features from the 2D domain, 3D domain, and mimicked 3D domain, and fuse them with attention modules, which can output more reliable features as an input of the following classification network.
- We propose a two-stage classifier, which can progressively locate the overlapping regions among three inputs.

2 RELATED WORK

We start this section by reviewing the feature-based point cloud registration methods to newer end-to-end point cloud registration algorithms. Finally, we briefly cover recent advances in using cross-modality information to guide feature extraction and matching.

2.1 3D Features for Point Cloud Registration

Traditional methods tend to use hand-crafted 3D features that characterize the local geometry for point cloud registration, such as FPFH [Rusu et al. 2009], SHOT [Tombari et al. 2011], or PPF [Drost and Ilic 2012]. Although lacking robustness in cluttered or occluded scenes, they have been widely employed in downstream applications owing to their generality across different datasets [Guo

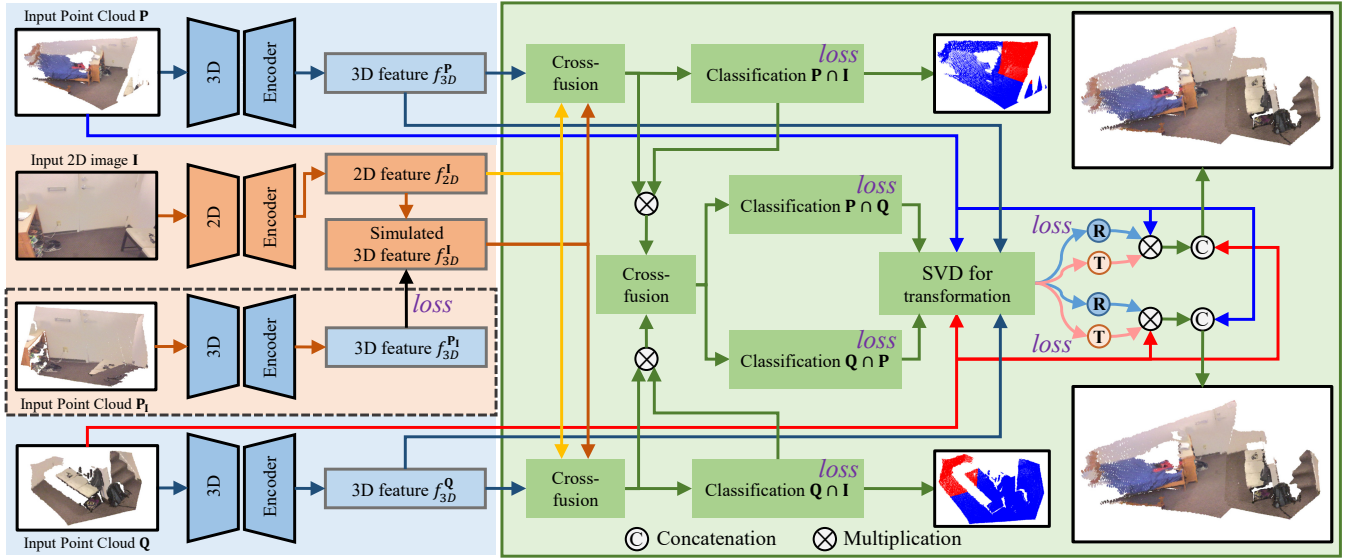


Figure 3: Network architecture of ImLoveNet. Note that we append texture colors on each point only for better visualization. We do not use the point cloud color information during both training and testing stages. Besides, P_1 is only used during the training stage.

et al. 2014]. More recently, with the advances in deep neural networks, there is also a growing trend that utilizes learned 3D descriptors in point cloud registration. For instance, the pioneering work 3DMatch [Zeng et al. 2017] employed a siamese deep learning architecture to extract local 3D descriptors. PPFNet [Deng et al. 2018b] and PPF-FoldNet [Deng et al. 2018a] proposed to combine PointNet [Qi et al. 2017a] and PPF to extract descriptors that are aware of the global context. To improve the robustness against noise and voxelization, [Gojcic et al. 2019] proposed to learn 3D descriptors based on a voxelized smoothed density value (SDV) representation. D3Feat [Bai et al. 2020] proposed a joint learning of keypoint detector and descriptor, which provides descriptors and keypoint scores for all points with extra cost during inference. [Wang et al. 2021] devised a new descriptor that simultaneously has rotation invariance and rotation equivalence. Although promising results have been achieved by these methods, limitations still occur on low-overlap regions due to their locality and extracted single-modality information on point clouds.

2.2 End-to-End Point Cloud Registration

Apart from introducing learned keypoints and descriptors, methods [Aoki et al. 2019; Qi et al. 2017a; Wang and Solomon 2019a; Wang et al. 2019] have also been proposed to embed the differentiable pose optimization to the registration pipeline to form an end-to-end framework. [Avetisyan et al. 2019] formulated a differentiable Procrustes alignment paired with a symmetry-aware dense object correspondence prediction to align CAD models to RGB-D scans. PointNetLK [Aoki et al. 2019] designed a Lucas/Kanade like optimization algorithm that tailored to a PointNet-based [Qi et al. 2017a] descriptor to estimate the relative transformation in an iterative manner. DCP [Wang and Solomon 2019a] utilized a

DGCNN network for correspondence matching, and used a differentiable SVD module for transformation estimation. PRNet [Wang and Solomon 2019b] extended DCP by including a keypoint detection step and allowed for aligning partially overlapping point clouds without the need for strict one-to-one correspondence. RPM-Net [Yew and Lee 2020] used the differentiable Sinkhorn layer and annealing to get soft assignments of point correspondences from hybrid features. Later, [Huang et al. 2020] proposed FMR, which achieved pleasing results by constraining the similarity between point cloud pairs. [Bai et al. 2021] designed a novel deep neural network that explicitly incorporates spatial consistency for pruning outlier correspondences. To alleviate the low-overlap registration problem, PREDATOR [Huang et al. 2021] and OMNet [Xu et al. 2021a] were both designed to focus more on learning the low-overlap regions. An outlier filtering network is embedded into a learned feature descriptor [Choy et al. 2020; Gojcic et al. 2020] to imply the weights of the correspondence in the Kabsch algorithm. At last, [Yan et al. 2021] proposed to solve the tele-registration problem, by combining the registration and completion tasks in a way that reinforces each other.

2.3 Cross-modality Feature Extraction and Fusion

Recently, several algorithm have been proposed to leverage multiple sources from different channels (i.e. geometry and color information) to enhance the content of feature extraction for subsequent tasks. 3D-to-2D Distillation [Liu et al. 2021] uses an additional 3D network in the training phase to leverage 3D features to complements the RGB inputs for 2D feature extraction. Pri3D [Hou et al. 2021] tried to imbue image-based perception with learned view-invariant, geometry-aware representations based on multi-view RGB-D data for 2D downstream tasks. To overcome the difficulty

of cross-modality feature association, DeepI2P [Li and Lee 2021] designed a neural network to covert the registration problem into a classification and inverse camera projection optimization problem. [Xu et al. 2021b] investigated the potential to transfer a pretrained 2D ConvNet to point cloud model for 3D point-cloud classification or segmentation. However, existing works tend to utilize cross-modality for enhanced segmentation or 2D-to-3D matching. In contrast, our work is the first to explore the possibility to leverage geometry and color information for point cloud registration with low-overlap region, by fusing cross-modality features.

3 METHOD

3.1 Problem description and overview

We denote $\mathbf{P} = \{\mathbf{p}_i \in \mathbb{R}^3 \mid i = 1 \dots N\}$ and $\mathbf{Q} = \{\mathbf{q}_i \in \mathbb{R}^3 \mid i = 1 \dots M\}$ as the two point clouds to be registered. $\mathbf{I} \in \mathbb{R}^{W \times H \times 3}$ is the intermediate color image between \mathbf{P} and \mathbf{Q} , and \mathbf{P}_I is the corresponding point cloud of \mathbf{I} , which is only used during the training stage. The overall goal is to locate the overlap region of paired point clouds and recover the rigid transformation parameters, \mathbf{R} and \mathbf{t} . Figure 3 illustrates the overall architecture of ImLoveNet, which can be decomposed into following four main steps:

- Triple features extraction: extracting 3D point cloud features $f_{3D}^{\mathbf{P}} \in \mathbb{R}^{N \times C}$, $f_{3D}^{\mathbf{Q}} \in \mathbb{R}^{M \times C}$ for \mathbf{P} and \mathbf{Q} , and the image feature $f_{2D}^{\mathbf{I}} \in \mathbb{R}^{W_I \times H_I \times C}$ for \mathbf{I} ; in particular, a mimicked 3D feature $f_{3D}^{\mathbf{I}} \in \mathbb{R}^{W_I \times H_I \times C}$ for \mathbf{I} is additionally generated with the assistance of its paired point cloud \mathbf{P}_I (Sec. 3.2);
- Cross-modality feature fusion: effectively combining the extracted triple features, e.g., $f_{3D}^{\mathbf{P}}$ (or $f_{3D}^{\mathbf{Q}}$), image feature $f_{2D}^{\mathbf{I}}$, and simulated feature $f_{3D}^{\mathbf{I}}$ together (Sec. 3.3);
- Two-stage classification: fed with the hybrid feature, detecting the points located in the camera frustum of \mathbf{I} from \mathbf{P} and \mathbf{Q} , and then fusing the hybrid features of above detected points again to further identify the overlap region between \mathbf{P} and \mathbf{Q} (Sec. 3.4);
- SVD for transformation: based on the predicted high-confidence overlap region between the two point clouds, computing final rigid transformation parameters via a differentiable SVD module.

3.2 Triple features extraction

We first embed the input point cloud pair and the image into their respective feature spaces, to obtain point-wise or pixel-wise features. Observing that many existing work achieves substantial results on some 3D-related tasks, e.g. normal estimation and depth prediction [Qi et al. 2018; Saha et al. 2021], we think certain 3D-aware geometry information also exists in the image space. Inspired by [Liu et al. 2021] and [Xu et al. 2021b], our network generates a simulated 3D feature to provide additional information. Such features are utilized with other features to collaboratively classify whether a point can be projected to the image space and whether a point is within the intersection part of the two input point clouds.

Specifically, we mimic the 3D feature $f_{3D}^{\mathbf{I}}$, by a small sub-module, which consists of a convolutional layer, a batch normalization layer, and an additional convolutional layer. Based on the corresponding point cloud \mathbf{P}_I , the real 3D feature $f_{3D}^{\mathbf{P}_I}$ is obtained via the 3D encoder and an extra batch normalization layer. The two batch normalization layers in 2D encoder and 3D encoder are used to roughly unify

the feature distribution of two modalities. A feature loss is formulated to constrain the generation of $f_{3D}^{\mathbf{I}}$. We explain how to compute it in Sec. 3.6. Since $f_{3D}^{\mathbf{P}_I}$ is only involved in calculating the feature loss, \mathbf{P}_I is required in the training phase, testing unnecessary. This is consistent with our design intuition: only using a single readily-available image to enhance the registration performance. Note that our default implementation uses PointNet++ [Qi et al. 2017b] for 3D encoder, and PSPNet [Zhao et al. 2017] for 2D encoder. These two encoders can be replaced with other state-of-the-art models.

3.3 Cross-modality feature fusion

Since the first stage of the classification is to determine the points that can be projected to the image plane, the information from different modalities should be taken into consideration. To this end, we introduce a cross-modality feature fusion module to effectively mix the above triple deep features. We take the mixture of \mathbf{P} and \mathbf{I} as an example. The input to this module consists of three parts: $f_{3D}^{\mathbf{P}}$, $f_{3D}^{\mathbf{I}}$, and $f_{2D}^{\mathbf{I}}$. The output is hierarchically computed by:

$$\begin{aligned} f_{cm1}^{\mathbf{P}} &= f_{3D}^{\mathbf{P}} + \text{MLP} \left(\text{cat} \left[f_{3D}^{\mathbf{P}}, \text{att}(f_{3D}^{\mathbf{P}}, f_{2D}^{\mathbf{I}}, f_{3D}^{\mathbf{I}}) \right] \right) \\ f_{cm}^{\mathbf{P}} &= f_{cm1}^{\mathbf{P}} + \text{MLP} \left(\text{cat} \left[f_{cm1}^{\mathbf{P}}, \text{att}(f_{cm1}^{\mathbf{P}}, f_{3D}^{\mathbf{I}}, f_{3D}^{\mathbf{I}}) \right] \right), \end{aligned} \quad (1)$$

where $\text{MLP}(\cdot)$ denotes a three-layer fully connected network, $\text{cat}(\cdot, \cdot)$ is concatenation, and $\text{att}(\cdot, \cdot, \cdot)$ means the attention model, which weights the image feature using learned weights. We reshape the 2D image feature as $f_{2D}^{\mathbf{I}} \in \mathbb{R}^{(W_I \cdot H_I) \times C}$ before feeding it to the fusion module. Similarly, we can obtain the fused feature $f_{cm}^{\mathbf{Q}}$. Detailed model structure is illustrated in Figure 4.

3.4 Two-stage classification

We conducted a statistical analysis on the Bundlefusion dataset [Dai et al. 2017] over 800 point cloud-image-point cloud triplets, with a 40% overlap ratio between two point clouds. We observed that about 80% of points of the overlapping area between two input point clouds are also located in the intermediate image space. Moreover, the image data in Bundlefusion dataset was captured close to the target, with a small resolution of 640×480 . If we use some other photo-taking devices, e.g. our cellphone, we will capture a larger intermediate image that contains more overlapping points. Hence, instead of directly learning the overlap parts from point clouds, we adopt a two-stage coarse-to-fine classification strategy to detect the overlap region between the point cloud pairs. First, the fused feature $f_{cm}^{\mathbf{P}}$ (or $f_{cm}^{\mathbf{Q}}$) is fed into a classifier, which is composed of a two-layer MLP and a Softmax layer, to compute the probability that each point can be projected into the image, or not. So we can obtain the subset $\mathbf{S}_I^{\mathbf{P}}$ (or $\mathbf{S}_I^{\mathbf{Q}}$), which are very likely to be projected to the image \mathbf{I} (see from Figure 2 (e) and (g)). Then, the features of the points from $\mathbf{S}_I^{\mathbf{P}}$ and $\mathbf{S}_I^{\mathbf{Q}}$ are fused again to further segment the potential overlapping regions $\mathbf{S}_O^{\mathbf{P}}$ and $\mathbf{S}_O^{\mathbf{Q}}$. The two subsets have similar shapes, but slightly different in point resolution and distribution (see from Figure 2 (f) and (h)). The second feature fusion module is similar to that used in the first classification stage, but we only need to fuse once, namely use the half (left or right) part of Figure 4.

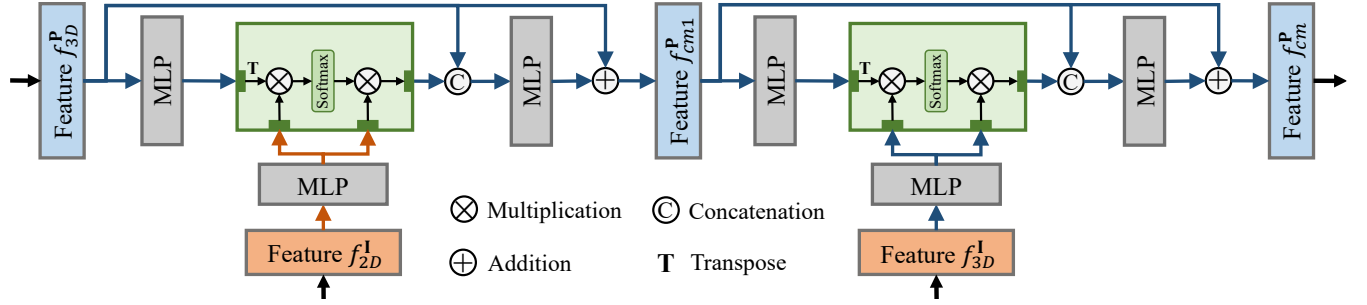


Figure 4: An illustration of cross feature fusion module.

3.5 SVD for transformation

In this section, we calculate the transformation parameters \mathbf{R} and \mathbf{t} . In the early test, we provided the features of points from the overlapping area to an MLP to regress a 7D vector (a 3D translation and a 4D quaternion) as the rigid transformation. However, we found that the transformation parameters are difficult to regress, even if we design various losses [Xiang et al. 2017; Xu et al. 2021a] to supervise them. Inspired by [Wang and Solomon 2019a], we believe that establishing suitable point-to-point correspondences is more important for solving transformation parameters, while a general approximator, e.g., MLP has poor interpretability and poor stability. Specifically, we establish a soft corresponding relation matrix among the point in \mathbf{S}_Q^P and \mathbf{S}_P^Q , according to their similarities in the embedded feature space. The weighted SVD is then used to solve for the rigid transformation, which has been shown to be differentiable in [Wang and Solomon 2019a].

3.6 Joint loss function

The proposed network is trained end-to-end, using multiple losses w.r.t. the real 3D feature f_{3D}^P from \mathbf{P}_I , the ground truth classification labels for points in image (denoted as $\hat{l}(\mathbf{S}_I^P)$ and $\hat{l}(\mathbf{S}_I^Q)$) and points in another point cloud (denoted as $\hat{l}(\mathbf{S}_P^Q)$ and $\hat{l}(\mathbf{S}_Q^P)$), and the ground-truth rigid transformation ($\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$) as supervisions.

Feature loss. To supervise the generation of f_{3D}^I in the 2D network, we use an L_2 loss:

$$L_F = \|\mathbb{T}(f_{3D}^P) - f_{3D}^I\|_2^2, \quad (2)$$

where $\mathbb{T}(\cdot)$ means the operator of projecting the real 3D feature into 2D domain. As the dimension of f_{3D}^I is $W_1 \times H_1 \times C$ and f_{3D}^P is $N \times C$, we cannot directly compute their differences. For each point in \mathbf{P}_I , we project it to the 2D image space via the camera intrinsic parameters, and treat the nearest pixel as its corresponding pixel.

Classification loss. The goal of two-stage classification is to progressively detect the overlapping region between the input two point clouds \mathbf{P} , \mathbf{Q} , via an intermediate image \mathbf{I} . Therefore, we formulate four constraint terms:

$$L_C = \text{CE}(\hat{l}(\mathbf{S}_I^P), l(\mathbf{S}_I^P)) + \text{CE}(\hat{l}(\mathbf{S}_I^Q), l(\mathbf{S}_I^Q)) + \text{CE}(\hat{l}(\mathbf{S}_Q^P), l(\mathbf{S}_Q^P)) + \text{CE}(\hat{l}(\mathbf{S}_P^Q), l(\mathbf{S}_P^Q)), \quad (3)$$

Table 1: Quantitative comparison of different methods on three datasets.

	Method	RMSE(\mathbf{R})	MAE(\mathbf{R})	RMSE(\mathbf{t})	MAE(\mathbf{t})	MIE(\mathbf{R})	MIE(\mathbf{t})
BundleFusion	DCP-v2	28.61	7.56	0.76	0.53	26.99	1.53
	PREDATOR	14.87	4.34	0.59	0.43	14.92	1.03
	OMNet	15.09	4.89	0.65	0.57	12.74	1.17
	FMR	23.62	7.91	1.01	0.70	20.96	1.98
	Ours	13.21	3.31	0.51	0.39	10.80	0.94
KITTI Odometry	DCP-v2	39.22	14.97	2.47	1.90	34.50	4.02
	PREDATOR	13.69	4.30	1.11	0.97	10.13	2.06
	OMNet	15.54	5.25	1.89	1.38	13.60	3.99
	FMR	21.11	8.38	2.27	1.87	19.29	4.25
	Ours	16.34	6.71	1.66	1.59	10.33	3.33
LiDAR_Ours	DCP-v2	17.01	6.31	0.45	0.34	16.09	1.01
	PREDATOR	6.45	3.01	0.17	0.13	6.79	0.60
	OMNet	6.91	2.90	0.20	0.17	6.55	0.58
	FMR	12.87	4.77	0.59	0.51	10.21	1.18
	Ours	6.40	2.98	0.19	0.12	5.80	0.39

where $\text{CE}(\cdot, \cdot)$ denotes cross entropy loss and $l(\cdot)$ means the predicted point label.

Transformation Loss. In our early testing, we try to represent both the ground-truth and predicted rotation matrices in the format of quaternion, and calculate their difference, by L_2 loss, as follows:

$$L_{T1} = \|\mathbf{q} - \hat{\mathbf{q}}\|_2^2 + \lambda \|\mathbf{t} - \hat{\mathbf{t}}\|_2^2, \quad (4)$$

where $\hat{\mathbf{q}}$ and $\hat{\mathbf{t}}$ are ground-truth quaternion and translation, \mathbf{q} and \mathbf{t} are the predicted results. However, we found the results is less pleasing. For a neural network, it may be hard to control the rotation accuracy only by relying on four parameters. Inspired by [Xiang et al. 2017], we indirectly compute the pose loss to supervise the rotation parameters, as follows:

$$L_{T2} = \sum_{i=1}^K \|\hat{\mathbf{R}}\mathbf{s}_i - \mathbf{R}\mathbf{s}_i\|_2^2 + \lambda \|\hat{\mathbf{t}} - \mathbf{t}\|_2^2, \quad (5)$$

where \mathbf{s}_i is the ground truth overlap point between the two input point clouds, and K is the number of overlapping points. Finally, the total loss is formulated as:

$$L = L_F + \alpha L_C + \beta L_{T2}. \quad (6)$$

Note that we compute two pairs of transformation parameters during training, while using the one with less error in testing stage. The entire network can be run in an iterative manner, so the loss is accumulated over iterations. We experimentally find that setting α , β , and λ as 1 works well.

4 EXPERIMENTS

We evaluate our ImLoveNet on three datasets with different kinds of 3D data: 1) BundleFusion dataset [Dai et al. 2017], which is an indoor scene RGB-D benchmark; 2) KITTI Odometry dataset [Geiger et al. 2013], which is an outdoor scene LiDAR benchmark; 3) a small indoor scene dataset acquired by a commercial LiDAR-based scanner, which is built by ourselves.

4.1 Dataset and implementation details

BundleFusion. The dataset was captured using a depth sensor coupled with an iPad color camera, and consists of 7 large indoor scenes (60m average trajectory length, 5833 average number of frames). Each sequence contains continuous color images and depth maps with a resolution of 640×480 , as well as the camera’s intrinsic and extrinsic parameters. The corresponding point clouds can be re-constructed by the depth maps and the camera intrinsic parameters. The ground-truth rotation and registration can be computed via camera extrinsic parameters. We choose 0-3 sequences for training, 4 for validation, and 5-6 for testing. In training and inference, detailed point cloud-image-point cloud triplets are formed as follows: 1) Randomly select a frame i and the corresponding point cloud is P . 2) The point cloud in the frame $i + 100$ is then regarded as Q ; We set the frame interval as 100, according to the overlap ratio. The overlap ratio is computed by the Eq. 1 in [Huang et al. 2021] and the mean overlap ratio is around 30% with the distance threshold equal to 0.05. 3) Choose the image in the frame $i + 50$ as the input intermediate image I . Also, the point cloud P_I in frame $i + 50$ is used in the training phase; 4) The supervision labels are easily computed by the given camera’s intrinsic and extrinsic parameters. Finally, we generate 800 pairs for each sequence, with totally 3200 triplets for training, 800 triplets for validation, and 1000 triplets for testing.

KITTI Odometry. Point clouds in this dataset are directly acquired from a 3D Lidar. There are 11 sequences (00-10) with ground-truth trajectories. We observe that: i) most of the points are located on the ground, while sparse above the ground; ii) the intermediate image should be behind of two point cloud frames, for achieving a high overlap between the point cloud and the image; and iii) point cloud-image-point cloud triplets with rich features, such as static cars or buildings, can provide more reliable information. Hence, we do some pre-processing steps (see from the supplemental file) and select 1000 triplets for training and 100 for testing.

LiDAR_Ours. This dataset was constructed by ourselves, which contains 3 indoor scenes. We build it by using a commercial LiDAR-based 3D scanner, i.e., Leica ScanStation P20 with the precision of 3 mm@50 m. This scanner captures the 3D data of a large-scale scene station by station. Point clouds from different stations are then registered together via at least three pairs of static markers. We cropped 40 pairs of point clouds, and took intermediate photos with our cellphone. The overlap ratio is also around 30%. We use this dataset only for testing.

Implementation details. We run our network 3 iterations during both training and test. The 3D encoder is shared by different input point clouds within each iteration. All sub-modules are shared during iterations. The input point clouds are downsampled and the size is fixed to 6000 ($M = N$). The feature channel C is 256. The leveraged

Table 2: Comparison of registration errors on the BundleFusion dataset of different ablative settings.

Model	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	MIE(R)	MIE(t)
B	18.99	5.50	1.19	1.02	11.02	2.04
B+I	23.22	6.13	1.34	1.21	15.78	2.37
B+I+MF	20.62	4.89	1.01	0.88	12.98	1.88
B+I+MF+CF	16.79	4.47	0.81	0.74	12.01	1.68
B+I+MF+CF+TC	14.08	4.10	0.77	0.62	11.26	1.04
B+I+MF+CF+TC+PL	13.21	3.31	0.63	0.39	10.80	0.94

2D feature map size is $1/8$ of the input image, namely $W_1 = 1/8W$ and $H_1 = 1/8H$. Our network is implemented on Pytorch and is trained for 200 epochs with the Adam optimizer [Kingma and Ba 2015]. The initial learning rate is 10^{-5} and is multiplied by 0.9 every two epochs. The network is trained with BundleFusion and KITTI individually.

4.2 Competitors

Since we focus more on registering low-overlap point clouds, two most advanced and related methods, OMNet [Xu et al. 2021a] and PREDATOR [Huang et al. 2021], are chosen as competitors. We also compared with the supervised version of FMR [Huang et al. 2020]. Beside, considering that DCP-v2 [Wang and Solomon 2019a] is better than many ICP-like traditional techniques, we here only compare with DCP-v2, without comparing to those traditional methods. We retrain all compared networks for fair comparison.

Evaluation metrics. Thanks for the released codes of OMNet [Xu et al. 2021a] and DCP [Wang and Solomon 2019a], we carefully checked their respective evaluation metrics. Although they use several of the same metrics, they evaluate on different targets. DCP evaluates on the registered point clouds, while OMNet on the Euler angles. To avoid unnecessary misunderstanding, we specify that our metrics are the same as OMNet. These used metrics are root mean squared error (RMSE) and mean absolute error (MAE), and mean isotropic error (MIE) (Please refer to [Xu et al. 2021a] for more details). The smaller these metric values, the better the results.

4.3 Evaluation on three datasets

We quantitatively evaluate the effectiveness of our method on three different datasets, in which the point clouds are generated by different sensors. Table 1 is the predicted rotation and translation errors of different methods. As observed, since DCP-v2 and FMR do not pay more attention on the low-overlap region, their results are less pleasing. Our method achieves better results on BundleFusion and LiDAR_Ours, while performing less better than PREDATOR and OMNet on KITTI Odometry. The main reason is that PREDATOR does not directly learn the transformation parameters, but employs the RANSAC scheme to compute them based on the learned overlapping probabilities. This scheme is more stable when dealing with sparse and noisy KITTI data.

4.4 Analysis

Ablation study. We ablate the following five main contributions: i) the intermediate image input (I), ii) the mimicked 3D feature (MF), iii) the cross-modality feature fusion (CF); iv) the two-stage classification (TC), and v) the pose loss (PL). Detailedly, if we do not use CF, we replace it with concatenation after necessary projection

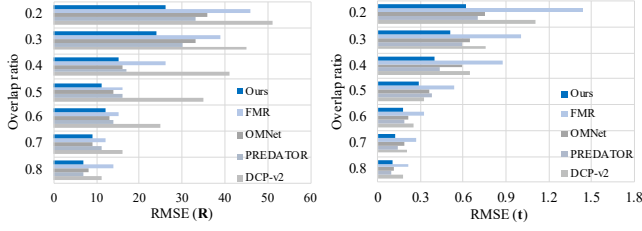


Figure 5: Quantitative comparison of different methods with 7 varying overlap ratios.



Figure 6: Registration with two different input images captured from different views and positions. The testing data is from LiDAR_ours. Although the two images (b) and (e) are different, they both contain the majority part of the overlapping region between the two input point clouds ((a) and (d)).

operation; if we do not use TC, we directly classify overlapping points; if we do not use PL, we use the loss L_{T1} in Eq. 4. Table 2 clearly reports the contribution of each module on the BundleFusion dataset. Interestingly, we found that directly using image information without any design significantly worsens the final result. This shows the necessity of our subsequent network design.

Classification accuracy. We perform overlapping classification evaluation on the selected BundleFusion dataset and KITTI Odometry dataset. The *precision* for point-in-image detection is 94% (BundleFusion) and 90% (KITTI Odometry), while 81% and 74% for the overlap region detection on two point clouds, which means that there are sufficient points to solve the transformation parameters (see visual detection results from the supplemental file).

Different overlap ratios. We evaluate the performance of our model on different overlap ratios. We additional extracted 1,000 test triplets from the BundleFusion dataset. By randomly rejecting the points within the point cloud-point cloud overlap region, we obtain 7 groups of test sets, with varying overlap ratios from 20% to 80%. Figure 5 shows that the distributions of RMSE(R) and RMSE(t) for different approaches. As observed, when the overlap ratio is lower, our network produces more stable results.

Table 3: Comparison of registration errors by using different intermediate images.

Method	I_1	I_2	I_3	I_4	I_5	I_6
RMSE(R)	11.77	5.58	5.46	5.73	11.74	11.88
RMSE(t)	1.03	0.92	0.94	0.92	0.97	1.01

Table 4: Performance comparison of original PREDATOR model and its two variants on the BundleFusion dataset.

Model	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)
PREDATOR	14.87	4.34	0.59	0.43
PREDATOR w/ image	14.29	5.26	0.63	0.57
PREDATOR w/ image and simulated 3D feature	13.42	4.03	0.45	0.39

Table 5: Quantitative comparison between DeepI2P (twice registration) and our network.

Method	RMSE(R)	MAE(R)	RMSE(t)	MAE(t)	MIE(R)	MIE(t)
DeepI2P	34.78	14.19	1.23	0.85	43.32	1.23
Ours	24	8.70	0.51	0.29	20.78	0.37

Different image positions. We testify our method with the input image of different image positions and views, as shown in Figure 6. The testing data is from LiDAR_ours and the used model is trained on the BundleFusion. Although I_1 and I_2 (Figure 6 (b) and (e)) are captured in different views, they both contain most of the overlapping area between the two input point clouds (Figure 6 (a) and (d)). As a result, they both contribute to the registration results. Moreover, we select six kinds of image frame as the intermediate image: one is near the source point cloud (I_1), three are very close to the intermediate frame (I_2 , I_3 , and I_4), one is near the target point cloud (I_5), and the last one is a randomly-selected image (I_6). We collect twenty groups of data from BundleFusion. Quantitative statistics reported in Table 3 show that if the intermediate image contains a higher overlap rate with both two input point clouds, it will assist to produce a better result.

Effect of image for other registration model. In fact, it is very interesting to see whether the input image contributes to other registration models. This is also meaningful to clarify the contribution of our work. In this section, we try to integrate the image information, as well as the simulated 3D feature into the PREDATOR [Huang et al. 2021]. Specifically, we feed an extra image into PREDATOR, and fuse the extracted image deep features in the encoding step of PREDATOR. Hence, we can obtain two variants: 1) only incorporate image information; 2) incorporate both image information and the simulated 3D feature. We compare the performance of the original PREDATOR model and its two variants on the BundleFusion dataset, as shown in Table. 4. We observe that directly encoding the image information into the network may not produce better results, while using the simulated 3D feature can help to yield more accurate registration results. Figure 1 also shows the visualization results.

Comparison with DeepI2P [Li and Lee 2021]. DeepI2P is designed for registering an image to a point cloud, namely computing the extrinsic parameters of the imaging device with respect to the reference frame of the 3D point cloud. Intuitively, we can register P

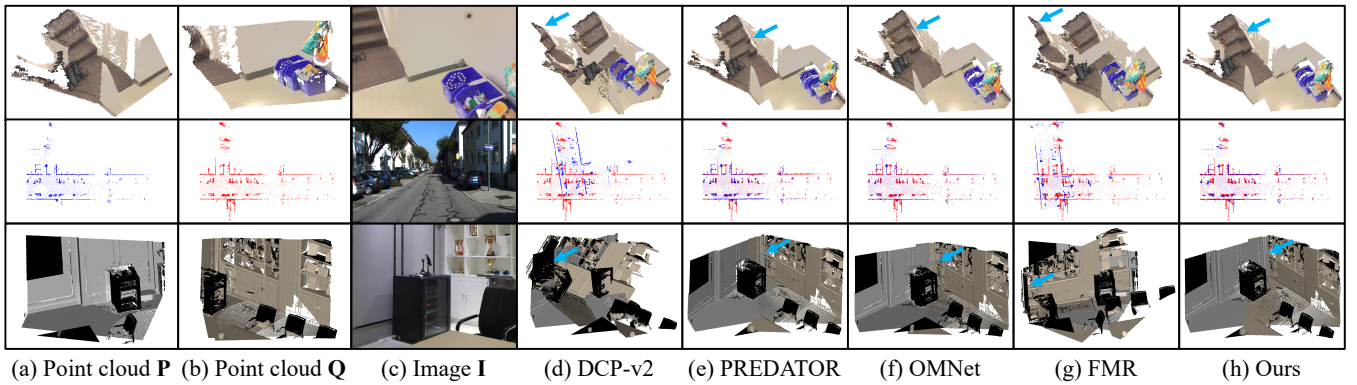


Figure 7: Visual comparisons of registration results of different methods. The inputs are from BundleFusion, KITTI Odometry, and LiDAR_Ours. The overlap ratios are set around 30%. Our method yields significantly better results than DCP-v2 and FMR, and achieves comparable visual results with OMNet and PREDATOR. Blue arrows indicate some registration details for better observation.

and Q to I , respectively. Then, the relative pose between P and Q is achieved. We give an objective evaluation between DeepI2P and our model in Table 5. The test data is from the selected KITTI Odometry dataset. Our method produces better results, while DeepI2P suffers from the accumulated errors of twice registration. Note that DeepI2P is retrained on the KITTI dataset.

Generalization ability. Indeed, the generalization ability of a network can alleviate the domain gap problem. Our network is able to generalize smoothly on the LiDAR_Ours dataset, which is composed of unseen indoor scenes. To further validate it, we attempt to test on the KITTI dataset with the network trained on BundleFusion. Although the scene objects, like cars, buildings, and trees, are never seen by the model, we observed that the registration result is visually satisfactory (see from the supplemental file).

Timing. When the input point cloud size is 6000 and the image resolution is 640×480 , the inference time of our model is around 5 seconds with 3 iterations.

4.5 Visualizations

We show visual comparisons of registration results in Figure 7. The inputs are from different datasets: BundleFusion, KITTI Odometry, and LiDAR_Ours. The overlap ratios are all around 30%. Our method yields significantly better results than DCP-v2 and FMR, and achieves slight improvement than OMNet and PREDATOR.

5 CONCLUSIONS

We have introduced, ImLoveNet, a new deep model designed for pairwise registration of low-overlap point clouds, with the assistance of misaligned intermediate images, which can be easily captured via a readily camera device. ImLoveNet tries to fully utilize and collaborate cross-modality information, to faithfully learn the most reliable overlapping regions for robust registration. Compared with common registration models, one obvious limitation of this method is that it should have additional paired point cloud and image as well as the camera intrinsic parameter for training. In the

future, we will try to transfer well-trained 3D or 2D models, to further boost the performance of low-overlap point cloud registration.

ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China (No. 2019YFB1707501), National Natural Science Foundation of China (No. 62172218, No. 62032011), and Natural Science Foundation of Jiangsu Province (No. BK20190016). The corresponding author for this paper is Jun Wang.

REFERENCES

- Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. 2019. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7163–7172.
- Armen Avetisyan, Angela Dai, and Matthias Niessner. 2019. End-to-End CAD Model Retrieval and 9DoF Alignment in 3D Scans. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2551–2560. <https://doi.org/10.1109/ICCV.2019.00264>
- Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. 2021. Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15859–15869.
- Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. 2020. D3feat: Joint learning of dense detection and description of 3d local features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6359–6367.
- Christopher Choy, Wei Dong, and Vladlen Koltun. 2020. Deep global registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2514–2523.
- Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1.
- Haowen Deng, Tolga Birdal, and Slobodan Ilic. 2018a. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 602–618.
- Haowen Deng, Tolga Birdal, and Slobodan Ilic. 2018b. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 195–205.
- Bertram Drost and Slobodan Ilic. 2012. 3D Object Detection and Localization Using Multimodal Point Pair Features. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*. 9–16. <https://doi.org/10.1109/3DIMPVT.2012.53>
- Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research* 32, 11 (2013), 1231–1237.

- Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. 2020. Learning multiview 3d point cloud registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 1759–1769.
- Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. 2019. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5545–5554.
- Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, Jianwei Wan, and Jun Zhang. 2014. Performance evaluation of 3D local feature descriptors. In *Asian Conference on Computer Vision*. Springer, 178–194.
- Ji Hou, Saining Xie, Benjamin Graham, Angela Dai, and Matthias Nießner. 2021. Pri3D: Can 3D Priors Help 2D Representation Learning? *arXiv preprint arXiv:2104.11225* (2021).
- Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. 2021. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4267–4276.
- Xiaoshui Huang, Guofeng Mei, and Jian Zhang. 2020. Feature-metric registration: A fast semi-supervised approach for robust point cloud registration without correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11366–11374.
- Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).
- Jiaxin Li and Gim Hee Lee. 2021. DeepI2P: Image-to-Point Cloud Registration via Deep Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15960–15969.
- Zhengzhe Liu, Xiaojuan Qi, and Chi-Wing Fu. 2021. 3d-to-2d distillation for indoor scene parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4464–4474.
- G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 2020. 3dregnet: A deep neural network for 3d point registration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7193–7203.
- Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 652–660.
- Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems* 30 (2017).
- Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. 2018. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 283–291.
- Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. 2009. Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE international conference on robotics and automation*. IEEE, 3212–3217.
- Suman Saha, Anton Obukhov, Danda Pani Paudel, Menelaos Kanakis, Yuhua Chen, Stamatios Georgoulis, and Luc Van Gool. 2021. Learning to Relate Depth and Semantics for Unsupervised Domain Adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8197–8207.
- Federico Tombari, Samuele Salti, and Luigi Di Stefano. 2011. A combined texture-shape descriptor for enhanced 3D feature matching. In *2011 18th IEEE International Conference on Image Processing*. 809–812. <https://doi.org/10.1109/ICIP.2011.6116679>
- Haiping Wang, Yuan Liu, Zhen Dong, Wenping Wang, and Bisheng Yang. 2021. You Only Hypothesize Once: Point Cloud Registration with Rotation-equivariant Descriptors. *arXiv preprint arXiv:2109.00182* (2021).
- Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. 2020. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 541–550.
- Yue Wang and Justin M Solomon. 2019a. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3523–3532.
- Yue Wang and Justin M Solomon. 2019b. Prnet: Self-supervised learning for partial-to-partial registration. *arXiv preprint arXiv:1910.12240* (2019).
- Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. 2019. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)* 38, 5 (2019), 1–12.
- Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. 2017. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199* (2017).
- Chenfeng Xu, Shijia Yang, Bohan Zhai, Bichen Wu, Xiangyu Yue, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. 2021b. Image2Point: 3D Point-Cloud Understanding with Pretrained 2D ConvNets. *arXiv preprint arXiv:2106.04180* (2021).
- Hao Xu, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. 2021a. OMNet: Learning Overlapping Mask for Partial-to-Partial Point Cloud Registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3132–3141.
- Zihao Yan, Zimu Yi, Ruizhen Hu, Niloy Jyoti Mitra, Daniel Cohen-Or, and Hui Huang. 2021. Consistent Two-Flow Network for Tele-Registration of Point Clouds. *IEEE Transactions on Visualization and Computer Graphics* (2021).
- Zi Jian Yew and Gim Hee Lee. 2020. Rpm-net: Robust point matching using learned features. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11824–11833.
- Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 2017. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1802–1811.
- Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2881–2890.