I2P Registration by Learning the Underlying Alignment Feature Space from Pixel-to-Point Similarities

YUNDA SUN and LIN ZHANG*, School of Software Engineering, Tongji University, China
ZHONG WANG, Department of Automation, Shanghai Jiao Tong University, China
YANG CHEN and SHENGJIE ZHAO, School of Software Engineering, Tongji University, China
YICONG ZHOU, Department of Computer and Information Science, University of Macau, China

Estimating the relative pose between a camera and a LiDAR holds paramount importance in facilitating complex task execution within multi-agent systems. Nonetheless, current methodologies encounter two primary limitations. First, amid the cross-modal feature extraction, they typically employ separate modal branches to extract cross-modal features from images and point clouds. This approach results in the feature spaces of images and point clouds being misaligned, thereby reducing the robustness of establishing correspondences. Second, due to the scale differences between images and point clouds, one-to-many pixel-point correspondences are inevitably encountered, which will mislead the pose optimization. To address these challenges, we propose a framework named **I2P$_{\mathbf{ppsim}}$** (**I**mage to **P**oint cloud registration by learning the underlying alignment feature space from **P**ixel-to-**P**oint **SIM**ilarities). Central to I2P$_{\text{ppsim}}$ is a shared feature alignment module (SFAM). It is designed under on a coarse-to-fine architecture and uses a weight-sharing network to construct an alignment feature space. Benefiting from SFAM, I2P$_{\text{ppsim}}$ can effectively identify the co-view regions between images and point clouds and establish high-reliability 2D-3D correspondences. Moreover, to mitigate the one-to-many correspondence issue, we introduce a similarity maximization strategy termed point-max. This strategy effectively filters out outliers, thereby establishing accurate 2D-3D correspondences. To evaluate the efficacy of our framework, we conduct extensive experiments on KITTI Odometry and Oxford Robotcar. The results corroborate the effectiveness of our framework in improving image-to-point cloud registration. To make our results reproducible, the source codes have been released at https://cslinzhang.github.io/I2P.

CCS Concepts: • **Computing methodologies → Vision for robotics**.

Additional Key Words and Phrases: image-to-point cloud registration, Data association, Cross-modal learning.

ACM Reference Format:
Yunda Sun, Lin Zhang, Zhong Wang, Yang Chen, Shengjie Zhao, and Yicong Zhou. 2024. I2P Registration by Learning the Underlying Alignment Feature Space from Pixel-to-Point Similarities. *J. ACM* 1, 1, Article 1 (April 2024), 21 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*Corresponding author: Lin Zhang

This work was supported in part by the National Natural Science Foundation of China under Grant 62272343; in part by the Shuguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission under Grant 21SG23; and in part by the Fundamental Research Funds for the Central Universities.

Authors' addresses: Yunda Sun, 2110850@tongji.edu.cn; Lin Zhang, cslinzhang@tongji.edu.cn, School of Software Engineering, Tongji University, Shanghai 200092, China; Zhong Wang, cszhongwang@sjtu.edu.cn, Department of Automation, Shanghai Jiao Tong University, Shanghai 200092, China; Yang Chen, 2011439@tongji.edu.cn; Shengjie Zhao, shengjiezhao@tongji.edu.cn, School of Software Engineering, Tongji University, Shanghai 200092, China; Yicong Zhou, yicongzhou@um.edu.mo, Department of Computer and Information Science, University of Macau, Macau 999078, China.

0004-5411/2024/4-ART1 $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

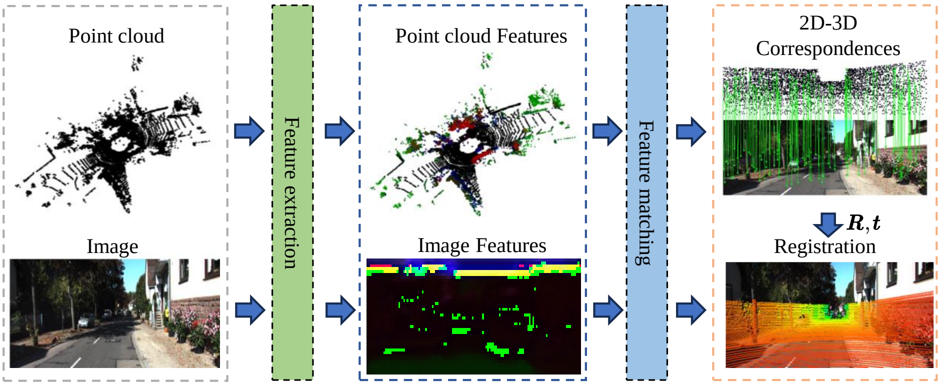


Fig. 1. General pipeline of image-to-point cloud (I2P) registration. I2P registration first extracts the common features and then conducts the feature matching to build the 2D-3D correspondences. After that, a PnP solver is used to estimate the relative pose.

Image-to-point cloud (I2P) registration refers to estimating the relative pose between a LiDAR and a camera via their measurements (point clouds and images), where the image and the point cloud are captured from the same scene. This task is widely used in many robotics and computer vision applications, such as SLAM (Simultaneous Localization and Mapping), robot navigation, and scene understanding [16, 18, 31, 42, 52].

The key to I2P registration is feature matching between images and point clouds. Unlike the widely studied homologous registration (image-to-image registration [4, 5, 47, 51], point cloud-to-point cloud registration [7, 46, 50]), image-to-point cloud registration is limited exploration due to the challenging modality differences between images and point clouds.

As shown in Fig. 1, the pipeline of I2P involves feature extraction, feature matching, correspondence establishment, and pose estimation. Previous studies relied on complex cross-modal manual feature designs or time-consuming optimization algorithms [10, 23], overlooking the differences between images and point clouds in feature space, perceptual range, and scale. Therefore, the performance of these studies is unsatisfactory. Specifically, to improve the performance of I2P registration, there are still three challenges to be faced with:

1) **Misalignment feature space.** Existing methods utilize separate modal branches to extract cross-modal features from images and point clouds [10], which poses a challenge in feature matching. Specifically, as different modalities of data, images and point clouds have significant differences in data structure and information content. Due to the use of different modal branches, current approaches cannot effectively alleviate such modal differences, but instead lead to feature space misalignment. This misalignment in the feature space reduces the performance of feature matching.
2) **Insufficient feature fusion.** The current single-stage feature fusion scheme fails to meet the different requirements of feature receptive fields for image-point cloud feature matching [23, 30]. Generally, cross-modal features with global receptive fields are suitable for detecting co-view regions between images and point clouds, while those with local receptive fields are suitable for predicting pixel-point matches. Therefore, it is of necessity to extract cross-modal features with different receptive fields for co-view region detection and matching pixel-point prediction. Unfortunately, current methods lack this capability.

3) **Matching ambiguity.** The one-to-many correspondence problem caused by the scale difference between images and point clouds misleads pose optimization. For example, taking the camera center as the origin, as the perception distance gets farther, for one pixel, there are usually multiple points within the frustum. Moreover, the Euclidean distance among those points may vary greatly. These seemingly "correct" one-to-many correspondences can mislead the pose optimization.

To deal with the aforementioned challenges, we propose a novel I2P registration framework, called **I2P$_{\mathbf{ppsim}}$** (**I**mage to **P**oint cloud registration by learning the underlying alignment feature space from **P**ixel-to-**P**oint **SIM**ilarities). I2P$_{\text{ppsim}}$ learns the underlying feature alignment space between images and point clouds via a shared feature alignment module (SFAM), and designs a matching constraint called point-max based on feature similarity to alleviate the one-to-many correspondence dilemma. The characteristics of I2P$_{\text{ppsim}}$ and our contributions are as follows:

1) The first feature space alignment-based image-to-point cloud (I2P) registration framework is proposed, named **I2P$_{\mathbf{ppsim}}$**. It mines pixel-point similarities by learning aligned cross-modal feature spaces. Based on the cross-modal representation, co-view regions are detected and pixel-point correspondences are directly predicted. Extensive experiments demonstrate that our I2P$_{\text{ppsim}}$ achieves state-of-the-art (SOTA).
2) A novel **S**hared **F**eature **A**lignment **M**odule (**SFAM**) is designed. Benefiting from the coarse-to-fine architecture, SFAM can extract coarse-grained features focusing on global expression and fine-grained features focusing on local expression. The former is helpful for co-view region detection and the latter is suitable for pixel-point matching estimation. In addition, SFAM uses a weight-sharing network to construct an aligned cross-modal feature space, which effectively alleviates the modality difference.
3) A plug-and-play matching strategy named **point-max** is introduced to solve the one-to-many correspondences. It does not rely on the feature learning ability of the network and aims to identify the best matching point for each pixel in the co-view region. By using point-max, a significant improvement in registration accuracy is achieved. Moreover, point-max can be seamlessly integrated as a plug-and-play module for other I2P registration methods, thereby improving their performance.

2 RELATED WORK

2.1 Image Registration

Image registration is usually treated as a pre-processing step for applications such as Structure from Motion (SfM) and image stitching [13, 20, 29]. The key to registration is to establish an accurate image matching in the $\mathbb{R}^2$ space. Current studies on image matching can be divided into two categories: feature-based ones and matching-based ones. The typical pipeline of the former is to extract the feature descriptors of the image [24], then calculate the distance between those descriptors, and determine the matching feature pairs. Recent approaches expect to obtain better visual feature descriptions through Convolutional Neural Networks (CNNs)[4, 5, 8], and further improve the correct rate and number of matching pairs.

Most feature-based methods determine the matching pairs through the neighbor search among features[9], while the matching-based ones no longer focus on the extraction of image features, but model the image matching as a learning problem. For example, SuperGlue[32] uses attention to aggregate the global and local features, formulates the image matching problem as a graph matching problem, and determines the matching pairs by approximately linear distribution. Furthermore, LoFTR[34] completely abandons the learning of image features and directly predicts dense pixel-by-pixel matches in an end-to-end manner.

2.2 Point Cloud Registration

Point cloud registration (PCR) methods can be broadly categorized into two groups. The first category is characterized by its emphasis on the extraction of point cloud features, aiming to establish correspondences mainly based on feature matching [1, 15, 28, 33, 36–40, 44, 49]. These methods primarily rely on the feature extraction capabilities of neural networks. Early studies mainly use PointNet [3] to extract point clouds' global descriptors and optimize the network by minimizing distances between global descriptors [1, 33]. Recently, some approaches replaced PointNet with transformer [35]. Benefiting from the expanded receptive fields and enhanced contextual association capabilities brought by the transformer, these methods achieved impressive performance [28, 36].

The above-mentioned PCR methods are sensitive to noise, and another category of methods introduces additional geometric or optimization constraints to enhance the robustness of PCR [2, 6, 11, 41, 45, 48]. PointDSC [2] introduces spatial consistency to eliminate inaccurate matching pairs. RGM [11] uses deep map matching to implement PCR. MAC [48] searches for the maximum clique subsets among the matching pairs and selects the optimal transformation guided by the reprojection error.

2.3 Image-to-Point Cloud Registration

Compared with image registration and point cloud registration, there are few studies on I2P registration. According to the ways of correspondence establishment, these I2P methods are mainly categorized into two classes: keypoint-based methods and keypoint-free ones. The core idea of the keypoint-based methods is to measure the distance and establish correspondences based on the keypoint descriptors extracted from images and point clouds [10].

In order to avoid complex cross-modal keypoint design, keypoint-free methods aim to learn point/pixel-wise features with strong repetitiveness [17, 19, 23, 30, 43]. DeepI2P [23] utilizes cross-attention to fuse the features of images and point clouds and proposes inverse camera projection for relative pose estimation. Building upon DeepI2P, CorrI2P establishes 2D-3D correspondences based on feature similarity metrics [30]. Similar to CorrI2P, EP2P-Loc achieves visual localization using images and point cloud submaps as inputs [19].

Although keypoint-free methods improve registration performance, they overlook the one-to-many correspondence dilemma caused by scale ambiguity. Furthermore, these methods extract features from images and point clouds through different modal branches, indicating that the feature spaces of images and point clouds are not aligned. The misalignment of feature spaces further reduces the repeatability of cross-modal features.

3 METHODOLOGY

3.1 Problem Definition and Framework Overview

Given an image $\mathcal{I} \in \mathbb{R}^{3\times W\times H}$ (W and H represent the width and height of the image) and a point cloud $\mathcal{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_{\boldsymbol{N}} \in \mathbb{R}^3\}$ (N is the number of points), the task of I2P registration is to estimate the relative rigid transformation $\boldsymbol{T} = [\boldsymbol{R}|\boldsymbol{t}] \in \boldsymbol{SE}(3)$ ($\boldsymbol{R} \in \boldsymbol{SO}(3)$, $\boldsymbol{t} \in \mathbb{R}^3$) from the LiDAR frame to the camera frame. Generally, a standard registration problem is modeled as a PnP (Perspective-n-Point) or ICP (Iterative Closest Point) problem. However, the point cloud collected by LiDAR has little geometric and appearance similarity with the RGB image. Also, to establish the correspondences among pixels and points is non-trivial. We expect to represent the two kinds of data in a higher-dimensional feature space through information fusion, build 2D-3D correspondences, and then regard I2P registration as a PnP problem. To this end, I2P$_{\text{ppsim}}$ is designed to comprise two parts: a correlation map estimation module and a pose estimator (as

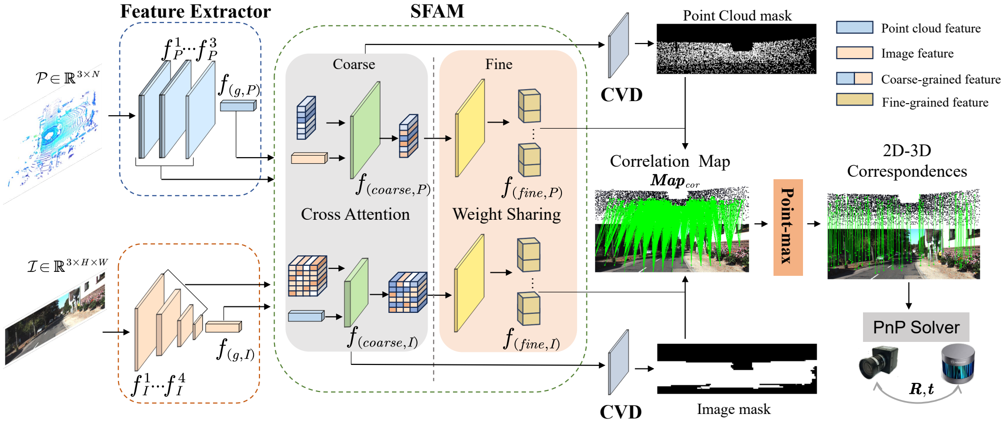


Fig. 2. Overview of I2P$_{\text{ppsim}}$. I2P$_{\text{ppsim}}$ is composed of five modules: Feature Extractor, SFAM, CVD, Point-max, and PnP Solver. Firstly, the high-dimensional features of images and point clouds are separately extracted by the two branches of Feature Extractor. Such features are then fed into SFAM to obtain cross-modal features in a coarse-to-fine architecture. With the coarse-grained features from SFAM, the co-view region of the image and point cloud is determined by CVD. Further, based on the fine-grained features from SFAM and the outputs of CVD, a correlation map is predicted, where 2D-3D correspondences are selected by Point-max. At last, the relative pose of the LiDAR to the camera is estimated via the PnP solver.

shown in Fig. 2). In I2P$_{\text{ppsim}}$, Feature Extractor, Shared Feature Alignment Module (SFAM), and **C**o-**V**iew **D**etector (**CVD**) are used to estimate the correlation map. When performing inference, given a pair of $\mathcal{I}$ and $\mathcal{P}$, the image and point cloud are first mapped to the high-dimensional space by Feature Extractor, and then SFAM is used to perform feature fusion and feature space alignment. Subsequently, the pixel-point correlation map is calculated based on the aligned cross-modal features in the co-view region and is further fed into the pose estimator. Based on our point-max strategy, we select matching candidates obtained from the correlation map. In this way, the 2D-3D correspondences can be established. Finally, EPnP [21] and RANSAC are employed to iteratively optimize the pose.

3.2 Feature Extractor

In view of the inherent dissimilarities in the properties of images and point clouds, employing the same feature extraction network to process both of them is impractical. Inspired by DeepI2P, we resort to ResNet [14] to encode the image features, while a modified PointNet++ [22, 27] serves as the feature encoder for the point cloud. Through these feature encoders, the multi-scale features of images and point clouds can be obtained, expressed as $\boldsymbol{F}_I^i \in \mathbb{R}^{c_i\times W_i\times H_i}$, $i \in \{1,2,3,4\}$, and $\boldsymbol{F}_P^j \in \mathbb{R}^{c_j\times N_j}$, $j \in \{1,2,3\}$, respectively, where c_i/c_j denotes the i-th/j-th features. Then, the global descriptions of the scene from images and point clouds can be obtained by performing max pooling, which are denoted by $\boldsymbol{F}_{(g,I)} \in \mathbb{R}^{c_I}$ and $\boldsymbol{F}_{(g,P)} \in \mathbb{R}^{c_P}$, where c_I/c_P is the dimension of image/point cloud global feature vector. We posit that taking into account both local and global features is more conducive to enhancing the ability of feature representation. Consequently, images and point clouds of various scales are employed for subsequent feature fusion.

3.3 Shared Feature Alignment Module

SFAM utilizes multi-scale features $\boldsymbol{F}_I^i$ and $\boldsymbol{F}_P^j$ as inputs and generates per-pixel and per-point cross-modal features $\boldsymbol{F}_{(fine,I)}$ and $\boldsymbol{F}_{(fine,P)}$. Considering that forcibly aligning the visual-laser feature

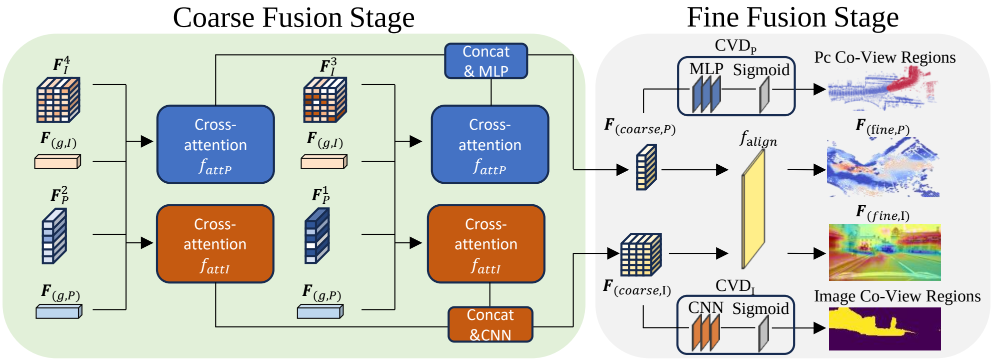


Fig. 3. Overall architecture of SFAM. SFAM fuses image and point cloud features in a coarse-to-fine manner. In the coarse fusion stage, symmetric cross-attention is leveraged to fuse multi-scale image and point cloud features. In the fine-grained fusion stage, fine-grained features $F_{(fine,P)}$ and $F_{(fine,I)}$ are extracted by a weight-sharing network f_{align}. In addition, coarse-grained features $F_{(coarse,P)}$ and $F_{(coarse,I)}$ are used as inputs of the co-view detectors (CVD), where $\mathrm{CVD_I}$ is the image co-view detector and $\mathrm{CVD_P}$ is the point cloud co-view detector.

spaces is rude and meaningless, SFAM is designed as a coarse-to-fine architecture, employing a two-stage approach to extract common features, as illustrated in Fig. 3.

In the coarse fusion stage, we concatenate the global features of the image ($F_{(g,I)}$) and point cloud ($F_{(g,P)}$) with multi-scale point cloud features ($\{F_P^j\}_{j=1,2}$) and image features ($\{F_I^i\}_{i=3,4}$), respectively. Next, fused features at different scales are obtained via symmetrical cross-attention ($f_{att_I} : \mathbb{R}^{c\times W\times H} \to \mathbb{R}^{c\times W\times H}$ and $f_{att_P} : \mathbb{R}^{c\times N} \to \mathbb{R}^{c\times N}$),

$$\begin{aligned} f'_{(coarse,I)} &= f_{att_I}(F_{(g,P)}, F_I^4, F_P^2, F_{(g,I)}), \\ f'_{(coarse,P)} &= f_{att_P}(F_{(g,P)}, F_I^4, F_P^2, F_{(g,I)}), \end{aligned} \tag{1}$$

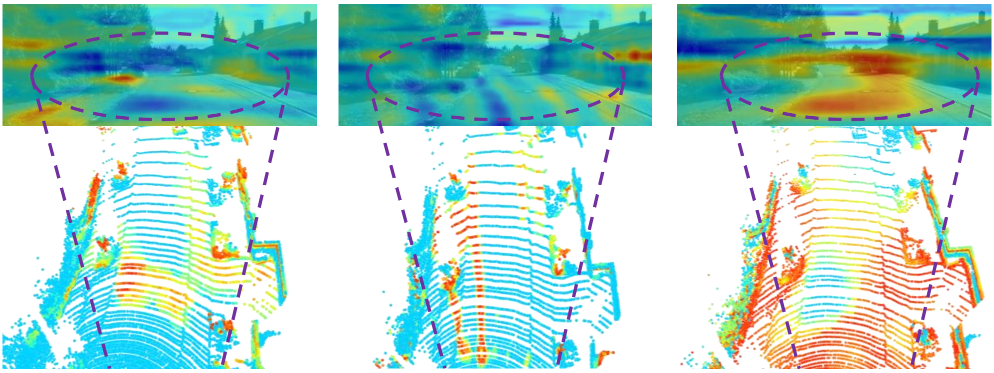
where $f'_{(coarse,I)}$ and $f'_{(coarse,P)}$ are the fused features of the image and the point cloud respectively. Similarly, by replacing F_I^4 and F_P^2 in Eq. 2 with F_I^3 and F_P^1, fused features of another scale can be obtained, which are indicated as $f''_{(coarse,I)}$ and $f''_{(coarse,P)}$. Then the fused features from different scales are concatenated, and the feature encoding functions are used to further extract coarse-grained features $f_{(coarse,I)}$ and $f_{(coarse,P)}$,

$$\begin{aligned} f_{(coarse,I)} &= CNN(f'_{(coarse,I)}, f''_{(coarse,I)}), \\ f_{(coarse,P)} &= MLP(f'_{(coarse,P)}, f''_{(coarse,P)}). \end{aligned} \tag{2}$$

Coarse-grained features focus more on global information and provide a broader perspective of the correspondence between the two modalities. In $\mathrm{I2P_{ppsim}}$, they are regarded as the input of co-view detectors.

In the fine fusion stage, an aligned feature space is constructed to mine the consistent features of both images and point clouds. Considering that CNN is difficult to handle unordered point cloud features, while unordered network structures still have the ability to handle ordered features, we resort to a variant of pointnet [23] to construct a weight-sharing network to model the feature space. With $f_{(coarse,I)}$ and $f_{(coarse,P)}$ as the network inputs, the fine-grained features $f_{(fine,I)}$ and $f_{(fine,P)}$ are obtained by,

$$\begin{aligned} f_{(fine,P)} &= f_{align}(f_{(coarse,P)}), \\ f_{(fine,I)} &= f_{align}(f_{(coarse,I)}), \end{aligned} \tag{3}$$



Features of image and point cloud obtained by CorrI2P

Features of image and point cloud obtained by DeepI2P

Features of image and point cloud obtained by I2P$_{\text{ppsim}}$

Fig. 4. Comparison of cross-modal features extracted by different feature fusion schemes. The co-view regions are marked, the warmer the feature, the higher the probability of a match.

where $f_{align} : \mathbb{R}^{c\times(W\times H)} \longrightarrow \mathbb{R}^{c\times(W\times H)}$ for images and $\mathbb{R}^{c\times N} \longrightarrow \mathbb{R}^{c\times N}$ for point clouds. These features focus on local similarity, enabling SFAM to capture delicate correspondences among pixels and points. After acquiring the fine-grained features, we expect to measure the similarity among them to build correspondences. With $\boldsymbol{f}_{(fine,I)}$, $\boldsymbol{f}_{(fine,P)}$, we can calculate the correlation map $\boldsymbol{Map}_{cor} \in \mathbb{R}^{(W\times H)\times N}$ between the image and the point cloud by,

$$\boldsymbol{Map}_{cor} = \boldsymbol{f}^{T}_{(fine,I)}\boldsymbol{f}_{(fine,P)}. \tag{4}$$

$\boldsymbol{Map}_{cor}$ reflects the similarity of visual-laser data, which enables I2P$_{\text{ppsim}}$ to learn the feature matching process. We optimize SFAM by minimizing the similarity loss of $\boldsymbol{Map}_{cor}$. The specific loss function design and analysis will be presented in Sec. 3.6.

The motivation behind the weight-sharing network. Existing feature fusion scheme for I2P registration extracts cross-modal features of images and point clouds through different modal branches. This results in these cross-modal descriptions being in different feature spaces, which in turn hinders the prediction of 2D-3D correspondences. To address this issue, we design a weight-sharing network in SFAM. By utilizing this network, features from different modal branches are mapped to the same aligned feature space, which assists feature matching.

The advantages of SFAM. Compared to other methods, our SFAM takes into account both global and local information, which enables I2P$_{\text{ppsim}}$ to better focus on semantic objects in the scene (such as cars, houses, etc.). Moreover, while other methods use separate modal branches to extract cross-modal features from images and point clouds, SFAM employs a weight-sharing network to directly construct an aligned feature space. As a result, the cross-modal features of images and point clouds are mapped to the same feature space, which enhances the reliability of feature similarity. Fig. 4 illustrates the similarities of cross-modal features extracted by different methods. It can be seen that our I2P$_{\text{ppsim}}$ outperforms the others.

3.4 Co-View Detector

The number of pixels or points in an image or point cloud typically ranges from thousands to tens of thousands. In the case where the number of pixels is M and the number of points is N, the size of the correlation map would be $M \times N$. The computation and storage requirements for such a large-scale correlation map are substantial. However, in I2P registration, the associated data is typically concentrated in a fan-shaped area, occupying only a small portion of the image and point

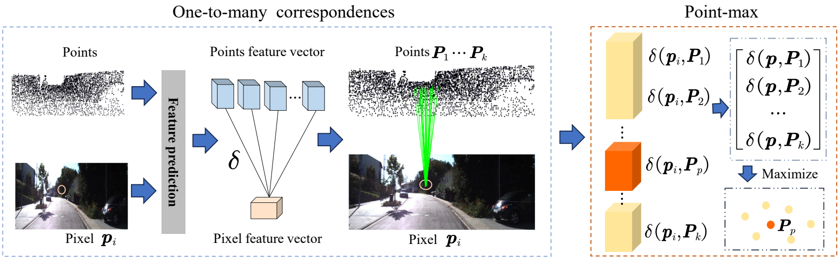

One-to-many correspondences
Points
Feature prediction
Points feature vector
δ
Points $\boldsymbol{P}_1 \cdots \boldsymbol{P}_k$
Pixel $\boldsymbol{p}_i$
Pixel feature vector
Pixel $\boldsymbol{p}_i$
Point-max
$\delta(\boldsymbol{p}_i,\boldsymbol{P}_1)$
$\delta(\boldsymbol{p}_i,\boldsymbol{P}_2)$
$\delta(\boldsymbol{p}_i,\boldsymbol{P}_p)$
$\delta(\boldsymbol{p}_i,\boldsymbol{P}_k)$
$\delta(\boldsymbol{p},\boldsymbol{P}_1)$
$\delta(\boldsymbol{p},\boldsymbol{P}_2)$
$\delta(\boldsymbol{p},\boldsymbol{P}_k)$
Maximize
$\boldsymbol{P}_p$

Fig. 5. The phenomenon of one-to-many correspondences between a pixel $\boldsymbol{p_i}$ and multiple points $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_k$. Point-max is designed to eliminate those outliers.

cloud. Leveraging this characteristic, Co-View Detectors (CVD) for images and point clouds are designed to determine whether a pixel or point belongs to the co-view region, they are denoted by CVD_I and CVD_P respectively.

During the calculation of the correlation map, only the pixels and points in the co-view region are considered. In this way, the data scale of the correlation map is significantly reduced, leading to accelerated network inference. To accomplish co-view region detection, we treat it as a binary classification problem. We employ two classification heads to analyze the coarse-grained features $\boldsymbol{f}_{(coarse,I)}$ and $\boldsymbol{f}_{(coarse,P)}$ separately. The outputs of the CVD correspond to the co-view scores for each pixel or point, indicating their likelihood of belonging to the co-view region.

3.5 Pose Estimation

Following the network prediction, notable dissimilarities are expected to exist between the matched and unmatched pixel-point pairs within the correlation map $\boldsymbol{Map}_{cor}$. With this in mind, one intuitive approach is to employ threshold screening to determine the correspondences between pixels and points. Alternatively, some methods establish 2D-3D associations by employing a classification-based approach. These methods aim to achieve the closest possible one-to-one correspondence between points and pixels. However, in practice, due to the different sensor measurement manners, the matching between point and pixel is not strictly one-to-one correspondence. As shown in Fig. 5, in the reprojection of the point cloud, each pixel in the image corresponds to a frustum in the real world, and multiple points are distributed in the frustum. So, one pixel in the image often has high similarity with multiple points. Furthermore, as the image downscaling and visual depth increase, the adverse impact caused by one-to-many correspondences will become more serious. Those seemingly "correct" correspondences indeed do harm to the calculation of reprojection errors in pose optimization. For example, four adjacent pixels $[u_i, v_i]^T, i \in \{1, 2, 3, 4\}$ are downscaled to one pixel $[u_s, v_s]^T$, which corresponds to k points $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_k$, where $\boldsymbol{P}_j = [x_j, y_j, z_j]^T, j = 1 \sim k$. Before downscaling, for each pixel $[u_i, v_i]^T$ and its matching points $\{\boldsymbol{P}_c = [x_c, y_c, z_c]^T : |c = 1 \sim k_i\}$ (where k_i is the number of matching points for $[u_i, v_i]^T$), the reprojection error is,

$$e = \sum_i e_i = \sum_i \sum_c \| \begin{bmatrix} u_i \\ v_i \end{bmatrix} - \pi(\frac{1}{z_c} \boldsymbol{KT}\bar{\boldsymbol{P}}_c)\|_2, \tag{5}$$

where $\boldsymbol{K}$ refers to the intrinsic matrix of the camera and $\bar{\boldsymbol{P}}_c$ is the homogeneous coordinate of $\boldsymbol{P}_c$, $\pi(\cdot)$ is an operator that takes the first two dimensions of a vector. After scaling, the reprojection error is reduced to,

$$e_s = \sum_{j=1}^{k} \| \begin{bmatrix} u_s \\ v_s \end{bmatrix} - \frac{1}{4}\pi(\frac{1}{z_j} \boldsymbol{KT}\bar{\boldsymbol{P}}_j)\|_2. \tag{6}$$

Such a loss will undoubtedly lead to performance degradation of the network which regards the reprojection error as its supervision. Unfortunately, it is challenging for classification or threshold screening to eliminate those seemingly "correct" points. Besides, these points are indeed observed by the corresponding pixels. Consequently, relying solely on cross-modal learning is also insufficient for removing such correspondences, and it is necessary to design a matching constraint to further refine the matching based on the correlation map.

In image matching, the matching relationship satisfies the following constraint: one pixel in an image can at most match with one pixel in another image [32]. Inspired by that, we appropriately relax this constraint criterion, apply it to I2P registration, and propose the point-max matching constraints: a pixel matches at most one point. Considering the data structure of the correlation map, point-max is performed via maximizing similarity in practice. Specifically, given a pixel $\boldsymbol{p}$, its receptive field contains k candidate matching points $\boldsymbol{P}_1, \cdots, \boldsymbol{P}_k$. Define the vector correlation operator as δ. The fine-grained feature vectors of $\boldsymbol{p}$ and $\boldsymbol{P}$ are $f_{(fine,\boldsymbol{p})}$ and $f_{(fine,\boldsymbol{P})}$, respectively. Then the correlation between pixel $\boldsymbol{p}$ and point $\boldsymbol{P}$ is,

$$\delta(\boldsymbol{p}, \boldsymbol{P}) = f_{(fine,\boldsymbol{p})} \cdot f_{(fine,\boldsymbol{P})} \tag{7}$$

Thus, the point $\boldsymbol{P_p}$ matching with the pixel $\boldsymbol{p}$ is,

$$\boldsymbol{P_p} = \underset{\boldsymbol{P}_j}{\arg\max}\{(\delta(\boldsymbol{p}, \boldsymbol{P}_1), \cdots, \delta(\boldsymbol{p}, \boldsymbol{P}_k)), j = 1, \cdots, k\}. \tag{8}$$

Eq. 8 is similar to performing max pooling among the matching candidates and selecting the candidate point with the highest correlation for the pixel. By applying Eq. 8 to the correlation map $\boldsymbol{Map_{cor}} \in \mathbb{R}^{M \times N}$, we can get the 2D-3D correspondences $\boldsymbol{Match_I}$, where map $\boldsymbol{Map_{cor}} \in \mathbb{R}^{M \times N}$ is expressed as,

$$\boldsymbol{Map_{cor}} = \begin{bmatrix} \delta(\boldsymbol{p}_1, \boldsymbol{P}_1) & \cdots & \delta(\boldsymbol{p}_1, \boldsymbol{P}_N) \\ \vdots & \ddots & \vdots \\ \delta(\boldsymbol{p}_M, \boldsymbol{P}_1) & \cdots & \delta(\boldsymbol{p}_M, \boldsymbol{P}_N) \end{bmatrix}, \tag{9}$$

and the final correspondence $\boldsymbol{Match_I}$ is,

$$\boldsymbol{Match_I} = \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{p}_1} \\ \vdots \\ \boldsymbol{P}_{\boldsymbol{p}_i} \\ \vdots \\ \boldsymbol{P}_{\boldsymbol{p}_M} \end{bmatrix}. \tag{10}$$

$\boldsymbol{Match_I}$ is the 2D-3D correspondence derived from $\boldsymbol{Map_{cor}} \in \mathbb{R}^{M \times N}$ by point-max selection, where each element embeds a pair of corresponding 2D pixel and the 3D point. Through point-max, the most similar match is selected in the one-to-many pixel-point correspondences, which alleviates the adverse impact of the outliers. With $\boldsymbol{Match_I}$, a set of equations that relate the observed 2D image coordinates to their corresponding 3D point coordinates can be established. In this way, the I2P registration problem becomes a PnP problem, and we iteratively optimize the pose through EPnP under RANSAC.

3.6 Loss Function

In I2P$_{\text{ppsim}}$, the performance of SFAM and CVD is particularly important. In order to obtain better cross-modal descriptions and more accurate co-view region detection results, we propose a joint loss function, which consists of the correlation loss and the co-view loss. For the correlation loss, the matched pixel-point pairs are expected to have cross-modal features with higher similarity, and vice versa for unmatched pairs. It implies that the correlations between matched pairs and unmatched pairs should have significant differences. For the co-view loss, based on the output scores of CVD, we expect that the points and pixels in the co-view region have higher scores, and the scores for outliers should be lower.

Correlation Loss. For the input image-point cloud pair $(\mathcal{I}, \mathcal{P})$, with the ground truth of relative pose $\boldsymbol{T} \in SE(3)$ and intrinsic matrix $\boldsymbol{K}$ of the camera, the reprojection error e_{pro} between the pixel $\boldsymbol{p}_i$ and point $\boldsymbol{P}_j = [x_j, y_j, z_j]^T$ can be calculated by,

$$e_{pro}(\boldsymbol{p}_i, \boldsymbol{P}_j) = \|\boldsymbol{p}_i - \pi(\frac{1}{z_j}\boldsymbol{K}\boldsymbol{T}\bar{\boldsymbol{P}}_j)\|_2. \tag{11}$$

When e_{pro} is less than a safety threshold e_t, $\boldsymbol{p}_i$ and $\boldsymbol{P}_i$ can be considered as a matched pair, otherwise an unmatched pair. The matching ground truth is denoted by $\boldsymbol{G}(\boldsymbol{p}_i, \boldsymbol{P}_j)$,

$$\boldsymbol{G}(\boldsymbol{p}_i, \boldsymbol{P}_j) = \begin{cases} 1, & \text{if } e_{pro}(\boldsymbol{p}_i, \boldsymbol{P}_j) < e_t \\ 0, & \text{otherwise.} \end{cases}. \tag{12}$$

With $\boldsymbol{G}(\boldsymbol{p}_i, \boldsymbol{P}_j)$, the correlation loss for a predicted pixel-point pair can be calculated according to the correlation between the pixel-point feature vectors. As mentioned in Eq. 7, the correlation between the feature vectors is denoted by $\delta(\boldsymbol{p}_i, \boldsymbol{P}_j)$, where $\boldsymbol{p}_i$ and $\boldsymbol{P}_j$ stand for the pixel and point, respectively. Defining the logits function as ξ, the loss of each pixel-point pair can be given as,

$$\begin{aligned} \mathcal{L}(\boldsymbol{p}_i,\boldsymbol{P}_j) = -w[&\boldsymbol{G}(\boldsymbol{p}_i, \boldsymbol{P}_j) \cdot \log \xi(\delta(\boldsymbol{p}_i, \boldsymbol{P}_j)) + \\ &(1 - \boldsymbol{G}(\boldsymbol{p}_i, \boldsymbol{P}_j)) \cdot \log(1 - \xi(\delta(\boldsymbol{p}_i, \boldsymbol{P}_j))], \end{aligned} \tag{13}$$

where w is the weight parameter.

There are $M \times N$ pixel-point pairs in $\boldsymbol{Map}_{cor} \in \mathbb{R}^{M \times N}$, and most of them are unmatched pairs. In order to speed up the optimization and balance the sample distribution, during training, we randomly select n pixels and n points in the co-view region for loss calculation and construct a correlation map with the size of $n \times n$. Finally, based on $\mathcal{L}(\boldsymbol{p}_i, \boldsymbol{P}_j)$, the correlation loss is defined as,

$$\mathcal{L}_c = \frac{1}{n^2} \sum_{i,j=0}^{n} \mathcal{L}(\boldsymbol{p}_i, \boldsymbol{P}_j). \tag{14}$$

Co-View Loss. Similar to the correlation loss, we sample n pixels $\boldsymbol{I}_{pos}$ and n points $\boldsymbol{P}_{pos}$ in the co-view region, and n pixels $\boldsymbol{I}_{neg}$ and n points $\boldsymbol{P}_{neg}$ out of the co-view region when calculating the co-view loss. Instead of focusing on the correlation between pixels and points, the co-view loss concerns whether the pixel or point belongs to the co-view region, which is a binary classification problem. The classification scores of each pixel or point can be obtained by CVD, denoted by $S_{I,pos}, S_{P,pos}, S_{I,neg}$, and $S_{P,pos}$. We expect that CVD can make the pixels and points in the co-view region have higher scores and vice versa. So the co-view loss is defined as,

$$\mathcal{L}_{cv} = \frac{1}{n} \sum (S_{I,neg} + S_{P,neg} - S_{I,pos} - S_{P,pos}). \tag{15}$$

Combining Eq. 14 and Eq. 15 yields the final joint loss function,

$$\mathcal{L}_{i2p} = \mathcal{L}_c + \mathcal{L}_{cv} \tag{16}$$

Table 1. Network hyperparameters of I2P$_{\text{ppsim}}$, where H_4/W_4 denotes a quarter of image height/width. *scale* represents the reduction factor during the network inference.

Module	Layer Type	K	Channel Dimensions
Point cloud Feature Extractor	Layer 1	-	[32, 128, 256]
	Layer 2	-	[64, 64]
	KNN Layer	32	[256, 256], [512, 256]
	Layer 3	-	[256, 512]
SFAM	Coarse layer Pa	-	[256, $H_4/scale \times W_4/scale$]
	Coarse fusion layer Pa	-	[1024, 512, 512]
	Coarse layer Pb	-	[256, $H_4/scale \times W_4/scale$]
	Coarse fusion layer Pb	-	[512,128,128]
	Fine Layer P	-	[128, 256, 128, 64]
	Fine Layer I	-	[128, 256, 128, 64]
CVD	Detector Head P	-	[128,128,64, 1]
	Detector Head I	-	[64,64,64, 1]

4 EXPERIMENTS

4.1 Setup

4.1.1 Dataset.

Our I2P$_{\text{ppsim}}$ was evaluated on KITTI Odometry [12] and Oxford Robotcar [25].

KITTI Odometry. In KITTI Odometry, the images and point clouds were acquired from an RGB camera and a 3D LiDAR. The camera and LiDAR had fixed extrinsics $\boldsymbol{T}_{cam}^{pc} \in \boldsymbol{SE}(3)$. This fixed relative pose in training and testing would be prone to cause network overfitting. On this account, it was necessary to perform data augmentation. Therefore, we followed the design of CorrI2P, using a random pose $\boldsymbol{T}_r$ to transform $\boldsymbol{T}_{cam}^{pc}$ and the point cloud. After taking augmentation, the relative pose of image-point cloud pair became $\boldsymbol{T}_{gt} = \boldsymbol{T}_{cam}^{pc}\boldsymbol{T}_r^{-1}$. Besides, the relative translation between the image and point cloud was guaranteed less than 10m. We followed Ren *et al.*'s [30] and Li *et al.*'s [23] settings to use the 0th~8th sequences for training, and the 9th~10th ones for testing. During training and testing, the size of the image was set to 160×512, and the number of points was 20,480. In total, there were 40,818 image-point cloud pairs used for training, and 5,584 pairs for testing.

Oxford Robotcar. Different from the acquisition method of point cloud in KITTI Odometry, the point clouds in Oxford Robotcar were captured by 2D scanning using a 2D LiDAR. To make the point clouds more dense, following DeepI2P, we spliced the adjacent point clouds at an interval of 2m, and finally merged all the point clouds in an area within the radius of 50m. 34 sequences were used for training and 4 sequences for testing. During training and testing, the image size was set to 384×640, and the number of points was the same as that in KITTI Odometry. Finally, 109,398 image-point cloud pairs were used for training and 13,545 pairs for testing.

4.1.2 Implementation Details. We conducted all experiments on a workstation equipped with an AMD Ryzen9 5900X processor and an NVIDIA GeForce RTX 3090 GPU. I2P$_{\text{ppsim}}$ was implemented by Pytorch [26]. The Adam optimizer was used for network training. We trained our network over 25 epochs on each dataset. The batch size for training was 16, and 8 for testing. The learning rate

Table 2. Registration accuracy on KITTI Odometry and Oxford Robotcar.

	KITTI Odometry		Oxford Robotcar	
	RTE (m)	RRE (°)	RTE (m)	RRE (°)
Grid Cls. + EPnP	3.22 ± 3.58	10 ± 13.74	1.91 ± 1.56	2.94 ± 10.72
DeepI2P (3D)	3.17 ± 3.22	15.52 ± 12.73	2.27 ± 2.19	15.00 ± 13.64
DeepI2P (2D)	3.28 ± 3.09	7.56 ± 7.63	**1.65 ± 1.36**	4.14 ± 4.90
CorrI2P	2.32 ± 9.74	4.66 ± 6.79	3.20 ± 3.14	2.49 ± 8.51
EFGHNet	4.83 ± 2.92	4.58 ± 8.67	3.78 ± 3.48	4.76 ± 5.69
EP2P-Loc	1.32 ± 1.13	4.11 ± 5.46	3.56 ± 3.79	8.65 ± 9.81
Ours	**1.18 ± 1.48**	**4.08 ± 4.46**	2.95 ± 2.66	**2.26 ± 5.12**

of the optimizer was initialized as 10^{-3}, and decayed by 75% every 5 epochs. During training, we set the safe threshold of the reprojection error (e_t) to 1 pixel.

Some important hyperparameters in $\text{I2P}_{\text{ppsim}}$ are reported in Table 1. The feature extractor of image is ResNet34. It outputs four different scale feature maps ($f_I^1 \sim f_I^4$) of SFAM as mentioned in Sec. 3.3.

In pose estimation, we experimentally set the co-view threshold of CVD as 0.9. The relative pose was estimated by EPnP under the RANSAC framework. The number of iterations was 500, and the reprojection error threshold was set to 1 pixel.

4.1.3 Evaluation Metrics. The Relative Rotation Error (RRE) and the Relative Translation Error (RTE) are adopted to evaluate the performance of the registration, which are formulated as,

$$\begin{aligned} \text{RRE} &= \sum_{i=1}^{3} |\theta(i)| \\ \text{RTE} &= \|\boldsymbol{t}_{pred} - \boldsymbol{t}_g\|_2, \end{aligned} \tag{17}$$

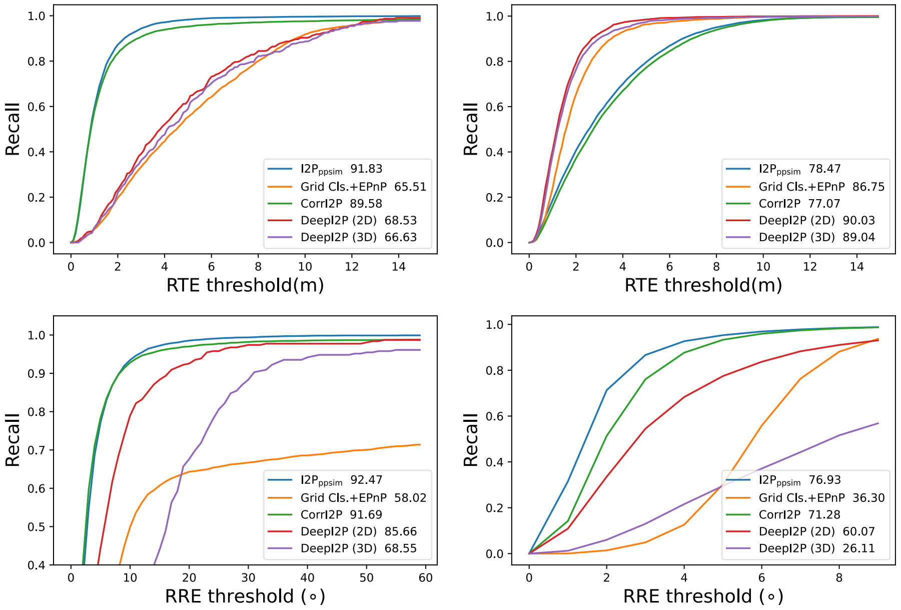
where $\theta(\cdot)$ denotes the Euler angle of $\boldsymbol{R}_{pred}^{-1}\boldsymbol{R}_g$, and $\theta(1)$, $\theta(2)$, and $\theta(3)$ are roll, pitch, and yaw, respectively. We denote the ground truth of the rotation matrix and relative translation vector by $\boldsymbol{R}_g$, $\boldsymbol{t}_g$, and the predicted ones by $\boldsymbol{R}_{pred}$ and $\boldsymbol{t}_{pred}$.

4.1.4 Compared Methods. We compared our $\text{I2P}_{\text{ppsim}}$ with 4 other approaches, which were Grid Cls.+PnP [23], DeepI2P (2D) [23], DeepI2P (3D) [23], CorrI2P [30], EFGHNet [17], and EP2P-Loc [19]:

- **Grid Cls.+EPnP.** This method was proposed in DeepI2P [23]. It divides the image into grids with the same size. For example, in the evaluation on KITTI Odometry, an image is divided into 80 grids, with a size of 5 × 16. Grid Cls.+EPnP predicts which grid the point cloud belongs to through a classification network. The 2D-3D correspondences are built from the classification results. With such correspondences, a PnP solver is used to estimate the relative pose.
- **DeepI2P.** Based on the idea of frustum binary classification, DeepI2P trains a frustum classifier to judge whether the point cloud is within the field of view of the camera. With the

classification results, it proposes 2D / 3D inverse camera projection to estimate the relative pose, called DeepI2P (2D) / DeepI2P (3D), respectively.

- **CorrI2P.** This method builds 2D-3D correspondences using the outputs of multi-modal branches. Based on those correspondences, the relative pose is estimated accordingly. We used the same network settings as those in the paper to reproduce the work. To our knowledge, CorrI2P is the state-of-the-art (SOTA) approach for image-to-point cloud registration.
- **EFGHNet.** This method adopts a divide-and-conquer strategy to decouple feature alignment and feature matching, and estimates the pose based on the feature matching results.
- **EP2P-Loc.** EP2P-Loc establishes 2D-3D correspondences through multi-scale matching and uses a differentiable PnP layer to directly estimate the relative pose. Due to the lack of open-source code, we reproduced this method as described in the paper.



(a) KITTI Odometry

(b) Oxford Robotcar

Fig. 6. Comparison of the registration recall of different methods with various RTE (m) and RRE (°) thresholds on KITTI Odometry and Oxford Robotcar. The area under each curve is presented behind the corresponding method's name.

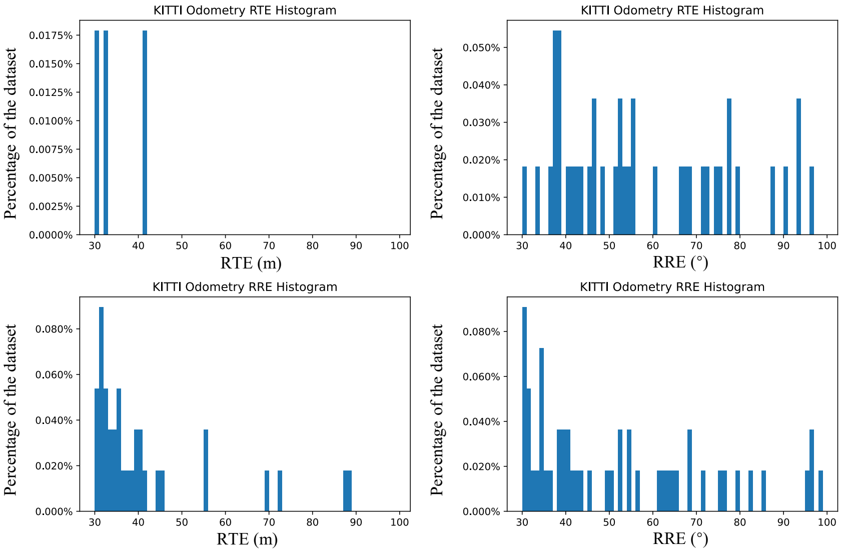
4.2 Quantitative Experiments

Different from CorrI2P which uses the error threshold to eliminate data with large RTE and RRE, we believe that using all test data to evaluate the accuracy of the algorithm can better reflect the robustness of the registration method. Therefore, we followed DeepI2P and compared I2P$_{\text{ppsim}}$ with competing methods on all test data of KITTI Odometry and Oxford Robotcar. The results achieved are reported in Table 2. It can be seen that I2P$_{\text{ppsim}}$ achieves significantly better performance over other counterparts on KITTI Odometry. As for Oxford Robotcar, although I2P$_{\text{ppsim}}$ achieves the best performance on RRE, it did not perform as well as other SOTA approaches on RTE. The main reason for this phenomenon is the preprocessing method of the point cloud. The point cloud in

Oxford Robotcar is formed by accumulating the 2D LiDAR scanning results from nearby areas. As a result, ghosting and blurring of many dynamic objects may emerge, which makes it difficult for I2P$_{\text{ppsim}}$ to predict correct correspondences among pixels and points. However, Grid Cls.+EPnP, DeepI2P (2D), and DeepI2P (3D) only need to predict rough grid classification or point cloud visibility results, without establishing strict pixel-point correspondences, so their performance on translation estimation is relatively better.

To compare the registration performance in more detail, we drew the registration recall curve under different RRE and RTE thresholds on the two datasets and calculated the area under the curve in Fig. 6. It can be seen that the performance of each approach is basically the same as Table 2 shows. However, we find that the leading edge of I2P$_{\text{ppsim}}$ over CorrI2P is narrow, which is different from the significant advantage of I2P$_{\text{ppsim}}$ shown in Table 2.

To dive deep into this phenomenon, we plotted the error distributions of the two approaches with a higher error threshold on KITTI Odometry in Fig. 7. It can be found that many large errors emerge in the results of CorrI2P. In contrast, I2P$_{\text{ppsim}}$ performs more stably, and there are few cases where the errors are extremely large.



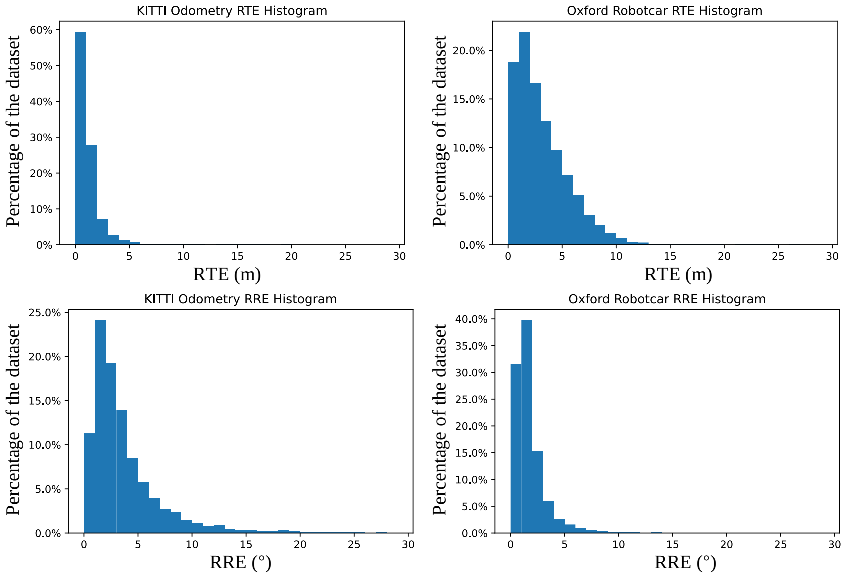
I2P$_{\text{ppsim}}$ CorrI2P

Fig. 7. Histograms of error distributions of I2P$_{\text{ppsim}}$ and CorrI2P at high RTE and RRE thresholds.

Error Distribution. I2P$_{\text{ppsim}}$'s error distributions of RRE (°) and RTE (m) on KITTI Odometry and Oxford Robotcar are shown in Fig. 8. Obviously, the translation estimation ability of I2P$_{\text{ppsim}}$ on KITTI Odometry is stronger than that on Oxford Robotcar, while the rotation estimation ability is weaker than that on Oxford Robotcar. The mode of RTE/RRE is $\sim 0.5m/1^{\circ}$on KITTI Odometry and $\sim 2m/2^{\circ}$on Oxford Robotcar.

4.3 Qualitative Experiments

To compare the performance of different methods more intuitively, we demonstrated the 2D-3D correspondences they generated. When generating the correspondences, the reprojection error threshold was set to 1 pixel. We show the 2D-3D correspondences in Fig. 9, where the wrong matches are in red and the correct ones in green.



(a) KITTI Odometry

(b) Oxford Robotcar

Fig. 8. Histograms of RTE (m) and RRE (°) on KITTI Odometry and Oxford Robotcar obtained by I2P$_{\text{ppsim}}$.

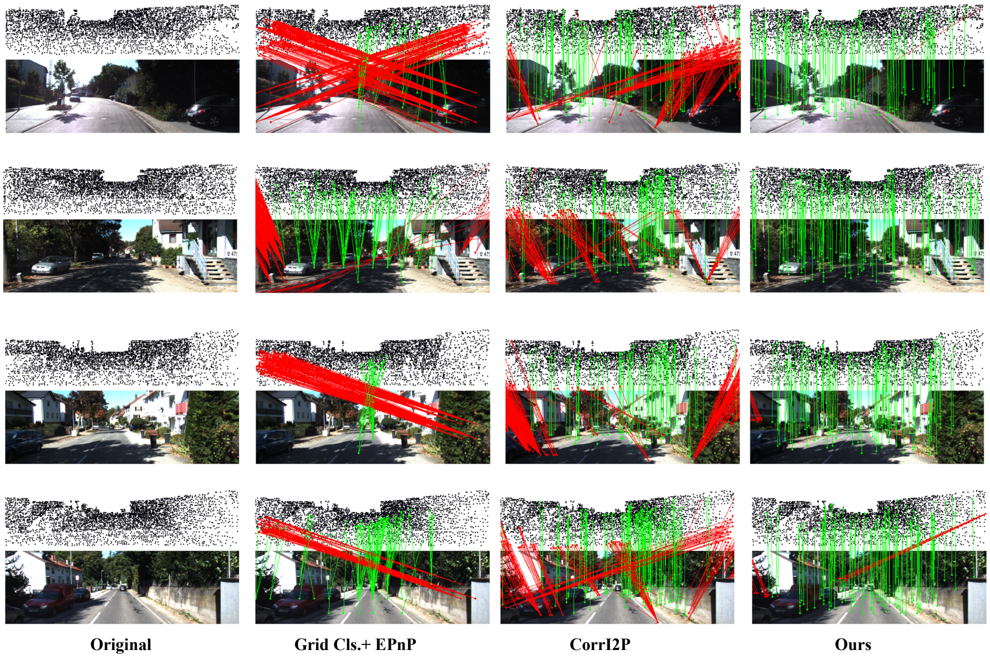


Fig. 9. 2D-3D correspondences established by different methods. The line segments represent 2D-3D correspondences, where red indicates false ones and green indicates correct ones.

Obviously, the “slopes” of the correspondence line segments are positively correlated with the reprojection errors. As mentioned in Sec. 3.5, as the visual depth grows, one pixel often corresponds to multiple points, which is particularly evident in Grid Cls.+EPnP and CorrI2P. For Grid Cls.+EPnP, amid the network’s inference, the final feature map’s size is downscaled at least 64 times compared with the raw input. After image downscaling, in KITTI Odometry, there are only 5×16 image grids and 20,480 points classified into these grids. As shown in the column “Grid Cls.+EPnP” of Fig. 9, one pixel may correspond to several or even dozens of points. Under the framework of grid classification of Grid Cls.+EPnP, even if the correspondences are correctly matched, their “slopes” can still be large. Such rough correspondences make it difficult to estimate the accurate pose. CorrI2P alleviates this phenomenon by reducing the times of image downscaling. However, there are still a lot of one-to-many correspondences, which mislead the pose estimation. Compared with the above methods, I2P$_{\text{ppsim}}$ handles the one-to-many correspondences through point-max. With such a strategy, the 2D-3D correspondences distribute more uniformly. Moreover, the wrong matches will be reduced from one cluster to one. These advantages are reflected in the notable improvement of the registration accuracy.

In addition, benefiting from SFAM proposed in Sec. 3.3, the 2D-3D correspondences generated by I2P$_{\text{ppsim}}$ are also more accurate than the competing methods. In Fig. 9, to make the visualization clearer, we downsampled the correspondences and filtered out the wrong ones with reprojection errors of less than 15 pixels. It is notable that the correlation prediction of I2P$_{\text{ppsim}}$ is more accurate, and there is basically no significant reprojection errors. We believe that the alignment feature space in SFAM plays an active role. Amid the coarse-to-fine feature alignment of I2P$_{\text{ppsim}}$, the fine-grained features are utilized to predict pixel-point correspondences, while the counterparts only conduct feature fusion in a one-stage pattern. Also, the effective guidance of the correlation loss enables I2P$_{\text{ppsim}}$ to more directly predict the pixel-point correspondences, thus further improving the ability of correspondence prediction.

4.4 Ablation Study

To verify the effectiveness of each module in our approach, we conducted ablation studies on our I2P$_{\text{ppsim}}$ using KITTI Odometry. The baselines involved in the ablation study are elaborated as follows,

- **PC-CVD** : CVD for point clouds;
- **IMG-CVD** : CVD for images;
- **w/o CVD** : I2P$_{\text{ppsim}}$ without CVD;
- **Direct Regression** : I2P$_{\text{ppsim}}$ without SFAM and CVD;
- **CorrI2P (point-max)** : CorrI2P with point-max;
- **I2P$_{\text{ppsim}}$ (w/o point-max)** : I2P$_{\text{ppsim}}$ without point-max;
- **I2P$_{\text{ppsim}}$ (mutual check)** : I2P$_{\text{ppsim}}$ with mutual check.
- **I2P$_{\text{ppsim}}$ (w/o fine)** : I2P$_{\text{ppsim}}$ without fine stage in SFAM.
- **I2P$_{\text{ppsim}}$ (w/o coarse)** : I2P$_{\text{ppsim}}$ without coarse stage in SFAM.

Table 3. Accuracy of CVD on KITTI Odometry.

	Recall	Precision	F2-Score	Accuracy
PC-CVD	0.962	0.901	0.949	0.973
IMG-CVD	0.782	0.732	0.771	0.791

Analysis on CVD. I2P$_{\text{ppsim}}$ detects the co-view region between the camera and the LiDAR through CVD and establishes 2D-3D correspondences based on the points and pixels in that region. In this way, we can effectively reduce the time cost and storage space of $\boldsymbol{Map_{cor}}$. Before CVD is used, the storage space of $\boldsymbol{Map_{cor}}$ is 12.5MB. After CVD selecting, $\boldsymbol{Map_{cor}}$ is notably compressed to 1.15MB. Also, the time cost of the correlation map computing is largely reduced from 33.3 seconds to 0.51 seconds. Furthermore, we evaluated CVD's recall, precision, F2-score, and accuracy, and the results are reported in Table 3. It can be seen that CVD has advanced co-view region detection performance, achieving accuracies of 97.3% and 79.1% for images and point clouds, respectively.

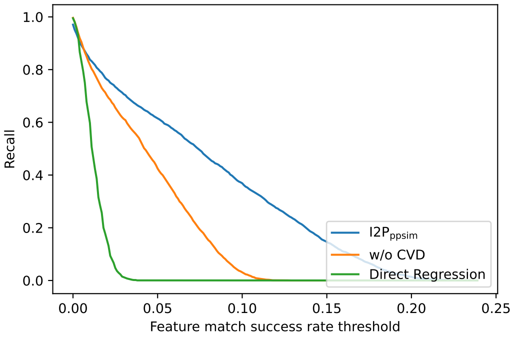


Fig. 10. Ablation study of SFAM on KITTI Odometry.

Analysis on SFAM. SFAM can fully integrate the multi-scale features of images and point clouds and mine the correlation between them. It enables I2P$_{\text{ppsim}}$ to perform co-view region detection, improves the ability to extract cross-modal features, and enhances the accuracy of 2D-3D matching. To verify such a claim, we carried out ablation experiments to analyze the importance of SFAM. Specifically, we designed a network without SFAM and CVD, called Direct Regression. Direct Regression only extracts the respective features of the image and the point cloud and predicts the matching relationship between pixel and point via direct regression. Feature matching recall was utilized to analyze the performance of Direct Regression and the results are shown in Fig. 10. It can be observed that Dircet Regression without feature fusion has a poor ability to predict correct feature matching. This is mainly due to the huge modal difference between the image and the point cloud, which makes the network unable to extract effective common features. Consequently, the poor matching prediction ability of Dircet Regression further justifies the necessity of feature fusion.

In addition, to explore the impact of the coarse-to-fine architecture on feature matching, we constructed a network without CVD (w/o CVD) which only uses fine-grained features to generate $\boldsymbol{Map_{cor}}$. Also, the feature matching recall was utilized as the metric to compare the impact of using coarse-grained features for co-view region judgment on the establishment of 2D-3D correspondences. The corresponding I2P$_{\text{ppsim}}$ uses coarse-grained features for co-view region screening and then generates 2D-3D correspondences by fined-grained features. The relevant experimental results are shown in Fig. 10. Compared with "w/o CVD", it is evident that the introduction of coarse-grained features has a significant positive effect on the establishment of 2D-3D correspondences.

To analyze the impact of each stage of SFAM, we conducted ablation experiments on SFAM, and the results are presented in Table 4. We removed the fine stage of SFAM (I2P$_{\text{ppsim}}$ (w/o fine)) to analyze the impact of the weight-sharing network. Similarly, we removed the coarse stage of SFAM (I2P$_{\text{ppsim}}$ (w/o coarse)), which means that SFAM only adopts the weight-sharing network to fuse the features of images and point clouds. From Table 4, it can be seen that both the coarse stage and fine stage of SFAM have a positive effect on registration performance.

Table 4. Ablation results of point-max and SFAM.

Method	RTE (*m*)	RRE (°)
CorrI2P	2.32 ± 9.74	4.66 ± 6.79
CorrI2P (point-max)	1.45 ± 1.62	4.32 ± 5.03
I2P$_{\text{ppsim}}$ (w/o point-max)	1.95 ± 2.97	4.31 ± 6.41
I2P$_{\text{ppsim}}$ (mutual check)	1.44 ± 2.19	4.26 ± 6.13
I2P$_{\text{ppsim}}$ (w/o fine)	2.12 ± 2.67	4.41 ± 6.26
I2P$_{\text{ppsim}}$ (w/o coarse)	2.46 ± 2.80	4.82 ± 6.84
I2P$_{\text{ppsim}}$	**1.18 ± 1.48**	**4.08 ± 4.46**

Analysis on Point-max. Based on $\boldsymbol{Map_{cor}}$, point-max is employed to find the matching point for each pixel. In this way, I2P$_{\text{ppsim}}$ filters out a large number of outliers that are difficult to eliminate only via the feature distances, therefore improving the registration accuracy.

To demonstrate the effectiveness of point-max, we compared the registration performance of I2P$_{\text{ppsim}}$, I2P$_{\text{ppsim}}$ without point-max, and I2P$_{\text{ppsim}}$ with mutual check. Among them, I2P$_{\text{ppsim}}$ with mutual check was achieved by applying point-max to both pixels and point clouds and conducting consistency check on the point-max results. In addition, since point-max is a feature-independent matching constraint strategy that can be considered as a plug-and-play module for other approaches, we also applied it to CorrI2P and compared the registration performance of point-max before and after use. All the above experimental results are reported in Table 4. It is notable that the introduction of point-max significantly improves the performance of the I2P registration approaches, especially in terms of translation estimation. Such a result also confirms the negative impact of the one-to-many correspondences on registration. Specifically, the introduction of mutual check led to another serious problem, i.e, the number of available 2D-3D correspondences will sharply decrease. The average number of matching pairs is only 307 for I2P$_{\text{ppsim}}$ with mutual check, while the count for the original I2P$_{\text{ppsim}}$ is 2,264. This sparsity of matching pairs has a detrimental effect on the performance of subsequent pose estimation which relies on nonlinear optimization techniques.

5 CONCLUSION

In order to fulfill the task of image-to-point cloud registration, this paper presents a novel framework based on alignment feature space learning, namely I2P$_{\text{ppsim}}$. I2P$_{\text{ppsim}}$ leverages SFAM (Shared Feature Aggregation Module) to enhance the network's ability to extract cross-modal features. Moreover, a matching strategy called point-max is proposed to address the one-to-many correspondences caused by scale ambiguity. Extensive experiments were conducted on benchmark datasets, namely KITTI Odometry and Oxford Robotcar, to demonstrate the outstanding performance of I2P$_{\text{ppsim}}$. Furthermore, we conducted ablation studies to validate the efficacy of each module within the framework. The promising results obtained from these experiments suggest that I2P$_{\text{ppsim}}$ holds potential for utilization in other tasks that require cross-modal fusion, such as multi-sensor calibration. In future work, we will devote our efforts to further improve the scalability of our framework, e.g., to make it leverage the pose as supervisory information and support multi-agent systems.

REFERENCES

[1] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. 2019. PointNetLK: Robust Efficient Point Cloud Registration Using PointNet. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 7156–7165.

[2] 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 *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 15854–15864.

[3] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 77–85.

[4] Hongkai Chen, Zixin Luo, Jiahui Zhang, Lei Zhou, Xuyang Bai, Zeyu Hu, Chiew-Lan Tai, and Long Quan. 2021. Learning to Match Features with Seeded Graph Matching Network. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 6281–6290.

[5] Hongkai Chen, Zixin Luo, Lei Zhou, Yurun Tian, Mingmin Zhen, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. 2022. ASpanFormer: Detector-Free Image Matching with Adaptive Span Transformer. In *Proc. Euro. Conf. Comput. Vis.* 20–36.

[6] Zhi Chen, Kun Sun, Fan Yang, and Wenbing Tao. 2022. SC2-PCR: A Second Order Spatial Compatibility for Efficient and Robust Point Cloud Registration. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 13211–13221.

[7] Christopher Choy, Jaesik Park, and Vladlen Koltun. 2019. Fully Convolutional Geometric Features. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 8957–8965.

[8] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2018. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog. Workshop.* 337–349.

[9] Patrick Ebel, Eduard Trulls, Kwang Moo Yi, Pascal Fua, and Anastasiia Mishchuk. 2019. Beyond Cartesian Representations for Local Descriptors. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 253–262.

[10] Mengdan Feng, Sixing Hu, Marcelo H Ang, and Gim Hee Lee. 2019. 2D3D-Matchnet: Learning To Match Keypoints Across 2D Image and 3D Point Cloud. In *Proc. IEEE Int. Conf. Robot. Autom.* 4790–4796.

[11] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. 2021. Robust Point Cloud Registration Framework Based on Deep Graph Matching. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 8889–8898.

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 3354–3361.

[13] Md. Nazmul Haque, Moyuresh Biswas, Mark R. Pickering, and Michael R. Frater. 2012. A low-Complexity Image Registration Algorithm for Global Motion estimation. *IEEE Trans. Circuits Syst. Video Tech.* 22, 3 (2012), 426–433.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 770–778.

[15] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. 2021. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 4265–4274.

[16] Youngseok Jang, Changsuk Oh, Yunwoo Lee, and H. Jin Kim. 2021. Multirobot Collaborative Monocular SLAM Utilizing Rendezvous. *IEEE Trans. Robot.* 37, 5 (2021), 1469–1486.

[17] Yurim Jeon and Seung-Woo Seo. 2022. EFGHNet: A Versatile Image-to-Point Cloud Registration Network for Extreme Outdoor Environment. *IEEE Robot. Autom. Letters* 7, 3 (2022), 7511–7517.

[18] Marco Karrer, Patrik Schmuck, and Margarita Chli. 2018. CVI-SLAM-Collaborative Visual-Inertial SLAM. *IEEE Robot. Autom. Letters* 3, 4 (2018), 2762–2769.

[19] Minjung Kim, Junseo Koo, and Gunhee Kim. 2023. EP2P-Loc: End-to-End 3D Point to 2D Pixel Localization for Large-Scale Visual Localization. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 21470–21480.

[20] Ik-Hyun Lee and Tae-Sun Choi. 2013. Accurate Registration using Adaptive Block Processing for Multispectral Images. *IEEE Trans. Circuits Syst. Video Tech.* 23, 9 (2013), 1491–1501.

[21] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. 2009. EPnP: An accurate O(n) solution to the PnP problem. *Int. J. Comput. Vis.* 81 (2009), 155–166.

[22] Jiaxin Li, Ben M. Chen, and Gim Hee Lee. 2018. SO-Net: Self-Organizing Network for Point Cloud Analysis. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 9397–9406.

[23] Jiaxin Li and Gim Hee Lee. 2021. DeepI2P: Image-to-Point Cloud Registration via Deep Classification. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 15955–15964.

[24] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* 60, 2 (2004), 91–110.

[25] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 2017. 1 Year, 1000km: the Oxford RobotCar dataset. *Int. J. Robot. Res.* 36, 1 (2017), 3–15.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proc. Adv. Neural Inf. Process. Syst.* 8024–8035.

[27] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. Adv. Neural Inf. Process. Syst.* 5105–5114.

[28] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. 2023. GeoTransformer: Fast and Robust Point Cloud Registration With Geometric Transformer. *IEEE Trans. Patt. Anal. Mach. Intell.* 45, 8 (2023), 9806–9821.

[29] B.S. Reddy and B.N. Chatterji. 1996. An FFT-based Technique for Translation, Rotation, and Scale-invariant Image Registration. *IEEE Trans. Imag. Process.* 5, 8 (1996), 1266–1271.

[30] Siyu Ren, Yiming Zeng, Junhui Hou, and Xiaodong Chen. 2023. CorrI2P: Deep Image-to-Point Cloud Registration via Dense Correspondence. *IEEE Trans. Circuits Syst. Video Tech.* 33, 3 (2023), 1198–1208.

[31] Sajad Saeedi, Michael Trentini, Mae Seto, and Howard Li. 2016. Multiple-robot simultaneous localization and mapping: A review. *J. Field Robot.* 33, 1 (2016), 3–46.

[32] PaulEdouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. 2020. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 4937–4946.

[33] Vinit Sarode, Xueqian Li, Hunter Goforth, Yasuhiro Aoki, Rangaprasad Arun Srivatsan, Simon Lucey, and Howie Choset. 2019. PCRNet: Point Cloud Registration Network using PointNet Encoding. arXiv:1908.07906. arXiv:1908.07906 [cs.CV]

[34] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. 2021. LoFTR: Detector-Free Local Feature Matching with Transformers. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 8918–8927.

[35] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Proc. Adv. Neural Inf. Process. Syst.* 6000–6010.

[36] Yue Wang and Justin Solomon. 2019. Deep Closest Point: Learning Representations for Point Cloud Registration. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 3522–3531.

[37] Yue Wu, Xidao Hu, Yue Zhang, Maoguo Gong, Wenping Ma, and Qiguang Miao. 2023. SACF-Net: Skip-Attention Based Correspondence Filtering Network for Point Cloud Registration. *IEEE Trans. Circuits Syst. Video Tech.* 33, 8 (2023), 3585–3595.

[38] Yue Wu, Jiaming Liu, Maoguo Gong, Peiran Gong, Xiaolong Fan, A. K. Qin, Qiguang Miao, and Wenping Ma. 2024. Self-Supervised Intra-Modal and Cross-Modal Contrastive Learning for Point Cloud Understanding. *IEEE Trans. Multimedia* 26 (2024), 1626–1638.

[39] Yue Wu, Jiaming Liu, Yongzhe Yuan, Xidao Hu, Xiaolong Fan, Kunkun Tu, Maoguo Gong, Qiguang Miao, and Wenping Ma. 2023. Correspondence-Free Point Cloud Registration Via Feature Interaction and Dual Branch. *IEEE Comput. Intell. Mag.* 18, 4 (2023), 66–79.

[40] Yue Wu, Yue Zhang, Xiaolong Fan, Maoguo Gong, Qiguang Miao, and Wenping Ma. 2023. INENet: Inliers Estimation Network With Similarity Learning for Partial Overlapping Registration. *IEEE Trans. Circuits Syst. Video Tech.* 33, 3 (2023), 1413–1426.

[41] Yue Wu, Yue Zhang, Wenping Ma, Maoguo Gong, Xiaolong Fan, Mingyang Zhang, A. K. Qin, and Qiguang Miao. 2023. RORNet: Partial-to-Partial Registration Network With Reliable Overlapping Representations. *IEEE Trans. Neural Netw. Learn. Syst.* Early Access (2023).

[42] Yuting Xie, Yachen Zhang, Long Chen, Hui Cheng, Wei Tu, Dongpu Cao, and Qingquan Li. 2022. RDC-SLAM: A Real-time Distributed Cooperative SLAM System Based on 3D LiDAR. *IEEE Trans. Intell. Transp. Syst.* 23, 9 (2022), 14721–14730.

[43] Shen Yan, Maojun Zhang, Yang Peng, Yu Liu, and Hanlin Tan. 2022. AgentI2P: Optimizing Image-to-Point Cloud Registration via Behaviour Cloning and Reinforcement Learning. *Remote Sens.* 14, 24:6301 (2022).

[44] Jiaqi Yang, Zhiqiang Huang, Siwen Quan, Qian Zhang, Yanning Zhang, and Zhiguo Cao. 2022. Toward efficient and robust metrics for RANSAC hypotheses and 3D rigid registration. *IEEE Trans. Circuits Syst. Video Tech.* 32, 2 (2022), 893–906.

[45] Yongzhe Yuan, Yue Wu, Xiaolong Fan, Maoguo Gong, Wenping Ma, and Qiguang Miao. 2023. EGST: Enhanced Geometric Structure Transformer for Point Cloud Registration. *IEEE Trans. Visualization Comput. Graph.* Early Access (2023).

[46] Ji Zhang and Sanjiv Singh. 2014. LOAM: Lidar Odometry and Mapping in Real-time. In *Proc. Robot. Sci. Syst.*

[47] Tianjun Zhang, Hao Deng, Lin Zhang, Shengjie Zhao, Xiao Liu, and Yicong Zhou. 2022. Online Correction of Camera Poses for the Surround-view System: A Sparse Direct Approach. *ACM Trans. Multimedia Comput. Commun. Appl.* 18,

4, Article 106 (2022), 24 pages.

[48] Xiyu Zhang, Jiaqi Yang, Shikun Zhang, and Yanning Zhang. 2023. 3D Registration with Maximal Cliques. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 17745–17754.

[49] Qianyi Zhou, Jaesik Park, and Vladlen Koltun. 2016. Fast Global Registration. In *Proc. Euro. Conf. Comput. Vis.* 766–782.

[50] Guanyu Zhu, Yong Zhou, Rui Yao, Hancheng Zhu, and Jiaqi Zhao. 2023. Cyclic Self-attention for Point Cloud Recognition. 19, 1s (2023).

[51] Jianke Zhu, Steven C. H. Hoi, Michael R. Lyu, and Shuicheng Yan. 2011. Near-Duplicate Keyframe Retrieval by Semi-supervised Learning and Nonrigid Image Matching. *ACM Trans. Multimedia Comput. Commun. Appl.* 7, 1, Article 4 (2011), 24 pages.

[52] Danping Zou and Ping Tan. 2013. CoSLAM: Collaborative visual SLAM in dynamic environments. *IEEE Trans. Patt. Anal. Mach. Intell.* 35, 2 (2013), 354–366.