MaGo-I2P: Image-to-Point Cloud Registration with Mamba and Geometry Recovery

Yunda Sun
School of Computer Science and Technology
Tongji University
Shanghai, China
sunyunda@tongji.edu.cn

Lin Zhang*
School of Computer Science and Technology
Tongji University
Shanghai, China
cslinzhang@tongji.edu.cn

Abstract

Estimating the relative poses between images and point clouds is a fundamental problem in multi-sensor fusion, with extensive applications in tasks such as robot localization and navigation. However, existing methods fall short in registration accuracy and efficiency due to the modality gaps and resource-consuming backbones. To address these issues, we propose the first Mamba-based I2P registration framework called MaGo-I2P. On the one hand, MaGo-I2P recovers the geometric structure of images through depth estimation, thereby constructing an implicit 3D representation of the image scene to alleviate the modality gap between images and point clouds, facilitating cross-modal feature extraction. On the other hand, unlike transformer-based backbones applied in existing methods, a Mamba-based backbone with linear time complexity is utilized in our MaGo-I2P. Such a backbone allows our method to possess both context-aware capability and fast inference speed. In addition, by adopting a coarse-to-fine matching strategy, MaGo-I2P eliminates outlier matches by progressively narrowing the matching region, establishing more accurate 2D-3D correspondences. Experiments on KITTI Odometry and Oxford Robotcar datasets suggest that our method achieves state-of-the-art registration accuracy while maintaining high-efficiency. Meanwhile, we also demonstrate the application potential of MaGo-I2P in LiDAR-camera calibration through qualitative experiments. The source code will be released at https://cslinzhang.github.io/MaGo-I2P.

CCS Concepts

• **Computing methodologies** → **Vision for robotics**.

Keywords

Image-to-point cloud registration, Cross-modal learning, Multi-sensor fusion, Cross-modality correspondence retrieval.

ACM Reference Format:
Yunda Sun and Lin Zhang. 2025. MaGo-I2P: Image-to-Point Cloud Registration with Mamba and Geometry Recovery. In *Proceedings of the 2025 International Conference on Multimedia Retrieval (ICMR '25), June 30-July 3, 2025, Chicago, IL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3731715.3733379

*Corresponding author

ICMR '25, Chicago, IL, USA

ACM ISBN 979-8-4007-1877-9/2025/06
https://doi.org/10.1145/3731715.3733379

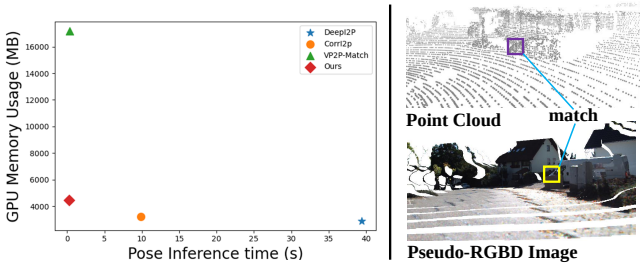
1 Introduction

Image-to-point cloud (I2P) registration is a fundamental and challenging problem in robot localization and navigation. It refers to determining the relative poses between images and LiDAR point clouds [1]. In robot systems with limited computational resources, it is crucial for an I2P registration framework to possess not only fast and accurate pose inference but also be compatible with low-resource devices.

Modality gaps between images and LiDAR point clouds are the key factor hindering the improvement of I2P registration accuracy. Specifically, images cannot capture the scene's geometric structure, whereas LiDAR point clouds provide rich 3D information. Such modality gaps pose a challenge in 2D-3D matching [16, 35]. To alleviate these gaps, some methods [14, 26] attempt to construct an aligned feature space to extract cross-modality features of images and point clouds, thereby enhancing the semantic correlation between features. Nevertheless, the alignment of the feature space still cannot mitigate the geometric misalignment between images and point clouds.

The selection of backbones determines the efficiency and accuracy of establishing 2D-3D matching. Generally, existing methods model the I2P registration as a PnP problem [31]. They establish 2D-3D correspondences between images and point clouds and estimate the poses based on these correspondences [6, 13]. Usually, 2D-3D correspondences are established based on 2D/3D features extracted by CNN/Transformer-based backbone [12, 14, 26, 39]. However, both CNN-based and Transformer-based methods have their drawbacks [20, 37]. The former's weak feature extraction capability leads to time-consuming pose optimization, while the latter requires a large amount of GPU memory, as illustrated in Fig. 1 (a). Neither type of backbone is friendly to low-resource devices, making it important to design an inference-efficient backbone that does not demand a large amount of memory for the I2P registration task.

To address these challenges, we propose **MaGo-I2P**, an **I**mage-to-**P**oint cloud registration framework with **Ma**mba and **Geo**metry recovery. MaGo-I2P combines the efficient inference of the Mamba [9] with the geometric recovery capability of the image depth estimation [33], enabling fast and accurate pose estimation without excessive computational resource consumption. Specifically, by utilizing Mamba with linear time complexity, our MaGo-I2P can



(a) Efficiency of Different Methods (b) Geometry Recovery

Figure 1: The characteristics of MaGo-I2P. (a) Compared to other methods, our MaGo-I2P strikes a balance between the pose estimation speed and the GPU memory usage, making it a low-resource device-friendly approach. (b) By employing geometric recovery, the modality differences between images and point clouds are alleviated, enhancing pixel-to-point matching.

achieve more efficient feature extraction with lower GPU memory. Additionally, in order to align the image and point cloud in terms of geometric structure, MaGo-I2P proposes to leverage the image depth estimation module to generate pseudo-RGBD data, which effectively alleviates the modality gaps, as illustrated in Fig. 1 (b). Our MaGo-I2P is evaluated on two large outdoor datasets of different point cloud densities, and its efficiency and accuracy are demonstrated.

Our main contributions are summarized as follows:

- We propose the first Mamba-based I2P registration framework named MaGo-I2P. By introducing a Mamba-based backbone, our method can extract features with a linear time complexity.
- We propose to alleviate the modality gaps by generating pseudo-RGBD data, which aligns the geometric structure of images and point clouds, thereby improving the 2D-3D matching accuracy.
- We conducted extensive experiments on KITTI Odometry and Oxford Robotcar, demonstrating that MaGo-I2P outperforms other state-of-the-art methods in registration accuracy, pose inference speed, and memory usage. Additionally, we collected data using a custom-built device, showcasing the application potential of MaGo-I2P in LiDAR-camera calibration.

2 Related Work

2.1 CNN-based I2P Registration

Driven by intra-modality registration (image registration [8, 27, 32] and point cloud registration [3, 18]), CNN-based I2P registration methods utilize convolutional neural networks for cross-modal representation of images and point clouds [5, 31]. They first extract keypoints [21, 38] from images and point clouds separately, then employ neural networks to learn cross-modal descriptors of these keypoints. Based on these descriptors, the 2D-3D correspondences are established via feature matching. These methods rely on the complex hand-crafted features, without mitigating the modality gaps between images and point clouds, hence resulting in poor registration performance.

2.2 PointNet-based I2P Registration

Various PointNet-based I2P registration approaches [11, 14, 26] have been proposed recently. Typically, PointNet [2] and its variants [25] are the dominant methods for point cloud feature extraction. By leveraging 2D backbone [10] and PointNet [2], PointNet-based I2P registration approaches avoid hand-crafted feature extraction and incorporate both feature extracting and matching into the deep learning framework. Moreover, these methods utilize attention [30] to fuse the features of images and point clouds, allowing for the acquisition of cross-modal feature representations. Although these methods alleviate modality differences to some extent and ensure a sufficient number of 2D-3D matches, their registration accuracy and computational speed remain unsatisfactory due to time-consuming pose estimation and matching outliers.

2.3 Transformer-based I2P Registration

Inspired by the progress of transformers [30] in point cloud analysis, some researchers attempt to apply the transformer [30, 37] to I2P registration tasks. Transformer-based I2P registration methods benefit from the contextual awareness of transformers and improve the quality of 2D-3D matches. Some representative studies are reviewed here. VP2P-Match [39] uses point transformer [37] as the 3D backbone and contracts a triplet network to learn a structured cross-modality latent space. Furthermore, building upon an encoder-decoder architecture, 2D3D-MATR [15] introduces a multi-scale matching framework for 2D-3D matches and achieves excellent performance in matching images with dense RGBD point clouds. EP2P-Loc [12] leverages swin transformer [20] and fast point transformer [23] as the 2D and 3D backbones, taking in LiDAR sub-maps and images as input to achieve visual localization. However, the complexity of the transformer is quadratic, bringing significant computational cost, which is not friendly to low-resource devices.

Therefore, in this work, we focus on designing an I2P registration framework that is not only friendly to low-resource devices but also possesses high-precision 2D-3D matching capabilities.

3 Method

3.1 Overview

Given an RGB image $\mathbf{I} \in \mathbb{R}^{W\times H\times 3}$ and a LiDAR point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ which are collected from the same scene, the objective of I2P registration is to estimate the relative rigid transformation $\{\mathbf{T} = [\mathbf{R}|\mathbf{t}]|\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3\}$ between them. A traditional I2P registration pipeline first establishes correspondences $\mathcal{M} = \{(\mathbf{u}_i, \mathbf{p}_i)|\mathbf{u}_i \in \mathbb{R}^2, \mathbf{p}_i \in \mathbb{R}^3\}$ between 3D points and 2D pixels, and then estimates the transformation by minimizing the 2D projection error:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}} \sum_{\mathbf{u}_i,\mathbf{p}_i\in\mathcal{M}} \|O(\mathbf{K}, \mathbf{T}, \mathbf{p}_i) - \mathbf{u}_i\|^2, \tag{1}$$

where O is the projection function from 3D space to image plane and $\mathbf{K} \in \mathbb{R}^{3\times 3}$ is the intrinsic matrix of the camera. Although Eq. 1 can be solved by PnP-RANSAC algorithm [6, 13], the solution can be erroneous due to inaccurate correspondences.

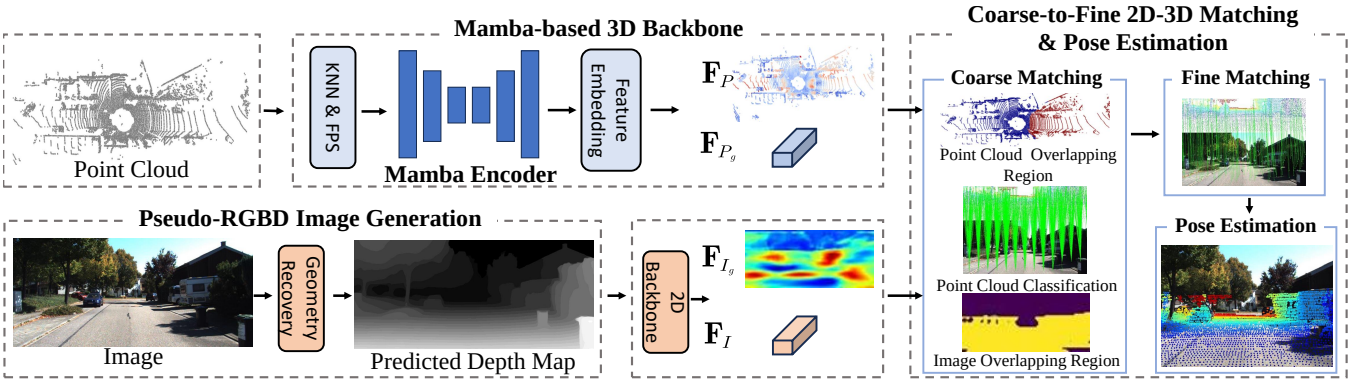


Figure 2: Overview of MaGo-I2P. MaGo-I2P comprises three components: the pseudo-RGBD image generation module, the Mamba-based 3D backbone and the coarse-to-fine 2D-3D matching strategy. First, the modality gaps between images and point clouds are alleviated by generating pseudo-RGBD images. Next, a 2D backbone and Mamba-based 3D backbone are used to extract features of images and point clouds, respectively, and cross-modal feature embedding is performed. Finally, pixel-to-point correspondences are obtained through coarse-to-fine 2D-3D matching. Based on these correspondences, the pose is iteratively solved by PnP with RANSAC.

To establish accurate 2D-3D correspondences and be compatible with low-resource devices, we propose MaGo-I2P. First, it recovers image depth using an off-the-shelf image depth estimation module to alleviate the modality gaps between images and point clouds (Sec. 3.2). Next, an effective Mamba-based backbone is employed to extract cross-modal features from images and point clouds (Sec. 3.3). Finally, accurate 2D-3D correspondences are established based on a coarse-to-fine matching strategy (Sec. 3.4). Fig. 2 provides an overview of our framework.

3.2 Pseudo-RGBD Image Generation

Due to perspective projection, the 2D images captured by the camera lose depth information, which poses a challenge for establishing pixel-to-point correspondences. To address this issue, we propose recovering the depth of the image to alleviate the geometric misalignment between the point cloud and the image. Specifically, an off-the-shelf depth estimation model $\mathcal{F}_d$ is leveraged to generate pseudo-RGBD images $\mathbf{I}_d \in \mathbb{R}^{4\times W\times H}$.

$$\mathbf{I}_d = \begin{pmatrix} \mathbf{I} \\ \mathcal{F}_d(\mathbf{I}) \end{pmatrix}. \tag{2}$$

Then, pseudo-RGBD features are extracted through a 2D backbone [10] and the details are introduced in Sec. 3.3.

Fig. 3 illustrates the features extracted from RGB images and pseudo-RGBD images. The warmth or coldness of the colors in the image indicates the probability of matching the point cloud. It can be seen that after introducing the depth estimation, the pseudo-RGBD features has a stronger response to the point cloud.

3.3 Cross-modality Feature Embedding

Given a pair of pseudo-RGBD image and point cloud, two modality-specific backbone networks are adopted for feature extraction. Then feature fusion is used to enhance the cross-modal representation ability of features.

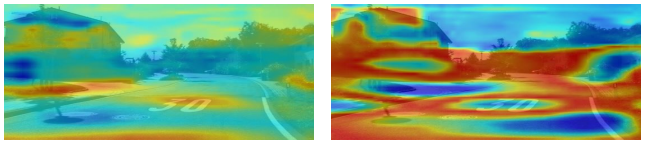


Image features without geometry recovery | Image features with geometry recovery

Figure 3: Comparison of extracted image features before and after geometry recovery. The warmer the color, the higher the probability of matching.

2D Backbone. For the image, ResNet [10] is used as the 2D backbone $\mathcal{F}_{2D}$ to generate multi-scale image features,

$$\mathbf{F}_I, \mathbf{F}_{I_g} = \mathcal{F}_{2D}(\mathbf{I}_d), \tag{3}$$

where $\mathbf{F}_I \in \mathbb{R}^{H'\times W'\times C}$ represents the image features at matching resolution, while $\mathbf{F}_{I_g} \in \mathbb{R}^{1\times \hat{C}}$ represents the global features. The former is used for matching at fine levels, while the latter is used for coarse-level matching.

Mamba-based 3D Backbone. Building on the pioneer works [4, 9, 17, 19], a Mamba-based 3D backbone is designed for extracting point cloud features. Specifically, this backbone consists of three main components: point tokens generation, Mamba encoder, and point-wise feature extraction.

For point tokens generation, we follow the approach of previous work [17], using FPS (Farthest Point Sampling) and KNN to generate keypoints from $\mathbf{P}$. Then, these keypoints are mapped into feature space by a lightweight PointNet [2] to generate the final point tokens $\mathbf{Z} \in \mathbb{R}^{2n\times \tilde{C}}$, where n is the number of tokens.

After obtaining $\mathbf{Z}$, Mamba encoder [9] $\mathcal{F}_m$ is leveraged to extract the features of input tokens,

$$\mathbf{F}_z = \mathcal{F}_m(\mathbf{Z}), \tag{4}$$

where $\mathbf{F}_z \in \mathbb{R}^{2n\times \tilde{C}}$.

Based on point cloud $\mathbf{P}$, point tokens $\mathbf{Z}$ and token fetures $\mathbf{F}_z$, the point-wise features $\mathbf{F}_P \in \mathbb{R}^{N\times C}$ are obtained by feature propagation. First, the feature weight of each point $\mathbf{w}' \in \mathbb{R}^{N\times 1}$ is determined by the reciprocal of the distance between that point and the corresponding token center point. Next, the initial point-wise features $\mathbf{F}'_z \in \mathbb{R}^{N\times \tilde{C}}$ are set as the features of the token to which each point belongs. For computational convenience, we replicate $\mathbf{w}'$ to match the size of $\mathbf{F}'_z$, resulting in $\mathbf{w} \in \mathbb{R}^{N\times \tilde{C}}$. Finally, the point-wise features $\mathbf{F}_P$ and the point cloud global features $\mathbf{F}_{P_g} \in \mathbb{R}^{1\times \hat{C}}$ can be obtained by a feature embedding function $\mathcal{F}_w$ [2],

$$\mathbf{F}_P, \mathbf{F}_{P_g} = \mathcal{F}_w[(\mathbf{w} \otimes \mathbf{F}'_z) \oplus \mathbf{P}], \tag{5}$$

where $\otimes$ denotes the Hadamard product and $\oplus$ denotes the concatenation operation for matrix.

Cross-modality Feature Embedding. To enhance the repeatability of features extracted from modality-specific backbones and establish more accurate 2D-3D correspondences, a simple yet effective feature fusion scheme is adopted. Specifically, it extends $\mathbf{F}_{I_g}$ to the size of $\mathbf{F}_P$, and concatenate the extended features with $\mathbf{F}_P$ to obtain $\mathbf{F}'_P \in \mathbb{R}^{N\times(\tilde{C}+\hat{C})}$. Then, a 3D feature embedding function $\mathcal{F}_{w,3D}$[2] is utilized to extract the cross-modal features of point clouds ($\tilde{\mathbf{F}}_P \in \mathbb{R}^{N\times\frac{C}{2}}$),

$$\tilde{\mathbf{F}}_P = \mathcal{F}_{w,3D}(\mathbf{F}'_P), \tag{6}$$

In a similar way, $\mathbf{F}_{P_g}$ is duplicated as $\mathbf{F}'_{P_g} \in \mathbb{R}^{H'\times W'\times \hat{C}}$. Then,$\mathbf{F}'_{P_g}$ is concatenated with $\mathbf{F}_I$ to obtain $\tilde{\mathbf{F}}'_I \in \mathbb{R}^{H'\times W'\times(C+\hat{C})}$. A 2D feature embedding function $\mathcal{F}_{w,2D}$[10] is applied to extract the cross-modal features of images ($\tilde{\mathbf{F}}_I \in \mathbb{R}^{H'\times W'\times\frac{C}{2}}$),

$$\tilde{\mathbf{F}}_I = \mathcal{F}_{w,2D}(\tilde{\mathbf{F}}'_I). \tag{7}$$

3.4 Coarse-to-Fine 2D-3D Matching

Generally, by calculating the cosine distance between $\tilde{\mathbf{F}}_P$ and $\tilde{\mathbf{F}}_I$, pixel-point matching can be established. However, this method requires maintaining an all-pair matching matrix, implying significant computational and storage costs. Therefore, a coarse-to-fine matching strategy is designed to reduce computational costs and improve the quality of feature matching. Fig. 2 illustrates the matching pipeline.

Overlapping Region Detection. Due to differences in the observation range between cameras and LiDARs, a significant portion of points and pixels are not within the overlapping observation regions, rendering them invalid matching candidates. To filter out these invalid matching candidates, overlapping region detectors for images (ORD_I) and point clouds (ORD_P) are designed to determine whether points/pixels are within the overlapping regions. Specifically, the overlapping scores $\mathbf{S}_P \in \mathbb{R}^{N\times 1}$ and $\mathbf{S}_I \in \mathbb{R}^{H'\times W'\times 1}$ can be derived as follows,

$$\mathbf{S}_P = \mathrm{ORD}_P(\tilde{\mathbf{F}}'_P), \tag{8}$$

$$\mathbf{S}_I = \mathrm{ORD}_I(\tilde{\mathbf{F}}_I). \tag{9}$$

A point/pixel is considered within the overlapping region if the value of $\mathbf{S}_P/\mathbf{S}_I$ is greater than threshold γ_P/γ_I. In this way, points and pixels within the overlapping region can be selected, denoted by $\mathbf{P}_o$ and $\mathbf{I}_o$ respectively.

Point Cloud Classification. Considering the different data densities of images and point clouds, establishing a strict one-to-one correspondence between pixels and points is non-trivial. Therefore, we downsample the image, divide it into M image blocks, use the coordinates of the image blocks as labels, and classify the point cloud accordingly. Based on the classification results $\mathbf{C}_a \in \mathbb{R}^{N\times M}$, the matching of each point with image blocks can be obtained.

Pixel-to-Point Similarity. With $\mathbf{P}_o$, $\mathbf{I}_o$ and $\mathbf{C}_a$, final pixel-point matches can be derived. First, points that match with the image block and are located in the overlapping region are selected and represented as $\mathcal{P}_j = \{\mathbf{p}_i | \mathbf{p}_i \in \mathbb{R}^3, i = 1\cdots N'\}$ (j means the j^{th} block). Similarly, pixels within the overlapping region of the image block are also selected and represented as $\mathcal{I}_j = \{\mathbf{u}_i | \mathbf{u} \in \mathbb{R}^2, i = 1\cdots m'\}$. Next, the cosine distance between the features of $\mathcal{P}$ and $\mathcal{I}$ are computed, and the pixel-to-point correspondences $\mathcal{M}_j$ are established by selecting the pixel-point pairs with minimum distance,

$$\mathcal{M}_j = \begin{bmatrix} \left(\mathbf{u}_1, \quad \underset{\mathbf{p}\in\{\mathbf{p}_1,\cdots,\mathbf{p}_{N'}\}}{\arg\min} \quad \delta(\mathbf{u}_1, \mathbf{p})\right) \\ \vdots \\ \left(\mathbf{u}_{m'}, \quad \underset{\mathbf{p}\in\{\mathbf{p}_1,\cdots,\mathbf{p}_{N'}\}}{\arg\min} \quad \delta(\mathbf{u}_{m'}, \mathbf{p})\right) \end{bmatrix}, \tag{10}$$

where δ is the function that calculates the cosine distance. For brevity, the process of selecting the corresponding features of $\mathbf{u}$ and $\mathbf{p}$ is omitted. Finally, the 2D-3D correspondences between $\mathbf{I}$ and $\mathbf{P}$ can be derived,

$$\mathcal{M} = [\mathcal{M}_1, \cdots, \mathcal{M}_j, \cdots, \mathcal{M}_M]^T. \tag{11}$$

With $\mathcal{M}$ and Eq. 1, the I2P registration can be modeled as a PnP problem, and EPnP with RANSAC [6, 13] can be utilized for iteratively optimizing the pose.

3.5 Loss Function

Our MaGo-I2P is trained in a metric learning paradigm [15]. It is expected to perform overlapping region detection, point classification, and pixel-point similarity estimation simultaneously. With $\tilde{\mathbf{F}}_I$, $\tilde{\mathbf{F}}_P$, $\mathbf{S}_P$ $\mathbf{S}_I$ and $\mathbf{C}_a$ a joint loss function $\mathcal{L}$ is designed to optimize the network, which consists of the overlapping loss $\mathcal{L}_o$, the point classification loss $\mathcal{L}_c$, and similarity loss $\mathcal{L}_s$,

$$\mathcal{L} = \alpha_1\mathcal{L}_o + \alpha_2\mathcal{L}_c + \alpha_3\mathcal{L}_s, \tag{12}$$

where α_1, α_2, and α_3 are weight coefficients.

Specifically, we define the overlapping loss $\mathcal{L}_o$ [26] as follows,

$$\mathcal{L}_o = \frac{1}{H}\sum_{h=1}^{H}((1 - s_{P,pos,h}) + (1 - s_{I,pos,h}) + s_{P,neg,h} + s_{I,neg,h}), \tag{13}$$

where $s_{P,pos,h}$ and $s_{I,pos,h}$ are the scores of h^{th} points/pixels within the overlapping region, $s_{P,neg,h}$ and $s_{I,neg,h}$ are the scores of h^{th} points/pixels out of the overlapping region, H is the number of sampled points/pixels. To enhance the discriminative capacity of the model, the similarity loss $\mathcal{L}_s$ is defined as a circle loss [28],

$$\mathcal{L}_s = \log[1 + \sum e^{\mu_p(\delta^p - \eta_p)} \cdot \sum e^{\mu_n(\eta_n - \delta^n)}], \tag{14}$$

where δ^p represents the cosine distance between a matched pixel-point pair, while δ^n represents an unmatched one. η_p and η_n are the margins. $\mu_p = \lambda(\delta^p - M_p)$ is the weighting factor for a positive pair, while $\mu_n = \lambda(M_n - \delta^n)$ for a negative pair. As for $\mathcal{L}_c$, it is defined as the cross-entry loss [36].

4 Experiments

In this section, we validate our MaGo-I2P on two large-scale outdoor benchmarks: KITTI Odometry [7] and Oxford Robotcar [22] datasets. We also compare our method with existing I2P registration methods. Besides, real-device experiments were conducted to demonstrate the application potential of MaGO-I2P in LiDAR-camera calibration tasks.

4.1 Experimental Setup

4.1.1 Datasets. our MaGo-I2P was validated on two large-scale outdoor benchmarks: KITTI Odometry [7] and Oxford Robotcar [22] datasets. The former contains point clouds captured by a 3D LiDAR, while the latter's point clouds are scanned by a 2D LiDAR. Also, both of them provide RGB images of real streets. To ensure fairness in the experimental setup and generate diverse image-to-point cloud scenes, the data processing method in previous studies [14, 26] is applied to process the KITTI Odometry [7] and Oxford Robotcar datasets [22].

4.1.2 Baselines. Our MaGo-I2P is compared with existing state-of-the-art I2P registration methods: DeepI2P (2D) [14], DeepI2P (3D) [14], CorrI2P [26], EFGHNet [11], VP2P-Match [39], EP2P-Loc [12] and $\text{I2P}_{\text{ppsim}}$ [29]. DeepI2P, CorrI2P and $\text{I2P}_{\text{ppsim}}$ employ PointNet [2] as their 3D backbone, while VP2P-Match and EP2P-Loc utilize transformer-based 3D backbones [23, 37]. Unfortunately, EP2P-Loc does not have open-source code, and VP2P-Match lacks open-source training code. Therefore, we simply listed the results reported in their papers for comparison.

4.1.3 Metrics. To evaluate the registration accuracy, RTE (Relative Translation Error), RRE (Relative Rotation Error) and RR (Registration Recall) are adopted as evaluation metrics, where the metrics are defined as follows:

$$\text{RTE} = \|\boldsymbol{t}_{pred} - \boldsymbol{t}_g\|_2, \quad \text{RRE} = \sum_{i=1}^{3} |\theta(i)|, \tag{15}$$

$$\text{RR} = \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} [\![\text{RTE}_i < \tau_t \wedge \text{RRE}_i < \tau_r]\!]. \tag{16}$$

In Eq. 15 and 16, the predicted translation vector and ground truth are represented by $\boldsymbol{t}_{pred}$ and $\boldsymbol{t}_g$, respectively. $\theta(\cdot)$ denotes the Euler angle of $\boldsymbol{R}_{pred}^{-1}\boldsymbol{R}_g$, where $\boldsymbol{R}_{pred}$ represents the predicted rotation matrix, $\boldsymbol{R}_g$ represents the ground truth rotation matrix. $\theta(1)$, $\theta(2)$, and $\theta(3)$ are roll, pitch, and yaw, respectively. $\tilde{M}$ represents the total number of data samples, $[\![\cdot]\!]$ is the Iverson bracket, and τ_t is set 5 m and τ_r is set to 2 °.

In addition to registration accuracy, IR (Inlier Ratio) and FMR (Feature Matching Recall) were applied to evaluate the matching accuracy of point-to-pixel correspondences. They are defined as follows:

$$\text{IR} = \frac{1}{|\mathcal{M}|} \sum_{(\mathbf{u}_i, \mathbf{p}_i) \in \mathcal{M}} [\![\|\mathbf{u}_i - O(\mathbf{K}, \mathbf{T}_g, \mathbf{p}_i)\|_2 < \tau_d]\!], \tag{17}$$

$$\text{FMR} = \frac{1}{|\mathcal{M}|} \sum_{i=1}^{|\mathcal{M}|} [\![\text{IR}_i > \tau_m]\!], \tag{18}$$

where $\mathcal{M}$ is the estimated correspondences set, $|\tilde{M}|$ represents the number of correspondences, τ_d and τ_m are the thresholds, and $\mathbf{T}_g$ is the ground truth.

4.1.4 Implementation Details. The experiments were conducted on a workstation that was equipped with an AMD Ryzen 9 5900X processor and an NVIDIA GeForce RTX 3090 GPU. PyTorch [24] was used for network implementation and the Adam optimizer was employed for network training. Our network was trained for 25 epochs on each dataset. The optimizer's learning rate was initialized to 10^{-3} and decayed by 75% every 5 epochs. During training, we set $\alpha_1 = \alpha_3 = 2, \alpha_2 = 1$.

4.2 Registration Accuracy

The registration accuracy of different I2P registration methods was evaluated on both datasets, and the results are provided in TABLE 1. It can be observed that our MaGo-I2P not only outperforms other competitors on both datasets but also demonstrates superior adaptability to different point cloud densities. This is mainly attributed to the generalization ability of the geometry restoration module in MaGo-I2P. It can predict image depth in an unsupervised manner, thereby constructing an implicit 3D representation of the scene, which facilitates the extraction of cross-modal features. Meanwhile, the results also indicate that the registration performance of the Mamba-based method (Ours) outperforms that of PointNet-based methods [14, 26, 29] and transformer-based methods [12, 39], further confirming the contribution of the Mamba-based 3D backbone to cross-modal feature extraction from point clouds. Compared to transformer-based methods, MaGo-I2P demonstrates superior performance in the RR metric, which is primarily attributed to the coarse-to-fine 2D-3D matching strategy, which effectively filters out mismatched outliers.

Qualitatively, the registration visualizations of different methods are illustrated in Fig. 4 and Fig. 5. The sparse point clouds in the Oxford Robotcar hinder the I2P methods from effectively extracting cross-modal features, resulting in suboptimal performance. Our MaGo-I2P constructs the 3D implicit space of images through depth estimation, rendering the image modality akin to the point cloud modality, thereby reducing the dependency on point cloud density. This enhancement facilitates the extraction of cross-modal features between images and point clouds, enabling more precise pose predictions and achieving superior alignment.

4.3 Pixel-to-Point Correspondences

To investigate the impact of proposed pseudo-RGBD generation and coarse-to-fine 2D-3D Matching strategy on registration performance, we evaluated the accuracy of pixel-to-point correspondences established by different methods. The quantitative results are presented in Table 2, where $\tau_d = 1, 2, 3$, and $\tau_m = 0.2$. Higher values indicate stronger 2D-3D matching performance and better outlier

Table 1: Registration accuracy on KITTI Odometry and Oxford Robotcar. The best results are highlighted in bold.

	KITTI Odometry			Oxford Robotcar		
	RTE (m) ↓	RRE (°) ↓	RR (%) ↑	RTE (m) ↓	RRE (°) ↓	RR (%) ↑
DeepI2P-3D (CVPR'21) [14]	3.17 ± 3.22	15.52 ± 12.73	3.77	2.27 ± 2.19	15.00 ± 13.64	62.35
DeepI2P-2D (CVPR'21) [14]	3.28 ± 3.09	7.56 ± 7.63	25.95	1.65 ± 1.36	4.14 ± 4.90	69.54
CorrI2P (TCSVT'23) [26]	2.32 ± 9.74	4.66 ± 6.79	72.42	3.20 ± 3.14	2.49 ± 8.51	40.64
EFGHNet (RA-L'22) [11]	4.83 ± 2.92	4.58 ± 8.67	5.65	3.78 ± 3.48	4.76 ± 5.69	20.33
VP2P-Match (NeurIPS'23) [39]	0.75 ± 1.13	3.29 ± 7.99	83.04	-	-	-
EP2P-Loc (ICCV'23) [12]	1.32 ± 1.13	4.11 ± 5.46	-	-	-	-
$I2P_{ppsim}$ (TOMM'24) [29]	1.18 ± 1.48	4.08 ± 4.46	78.49	2.95 ± 2.66	2.26 ± 5.12	52.33
MaGo-I2P (Ours)	**0.66 ± 0.75**	**3.04 ± 4.25**	**96.99**	**1.59 ± 1.26**	**2.03 ± 4.61**	**70.12**

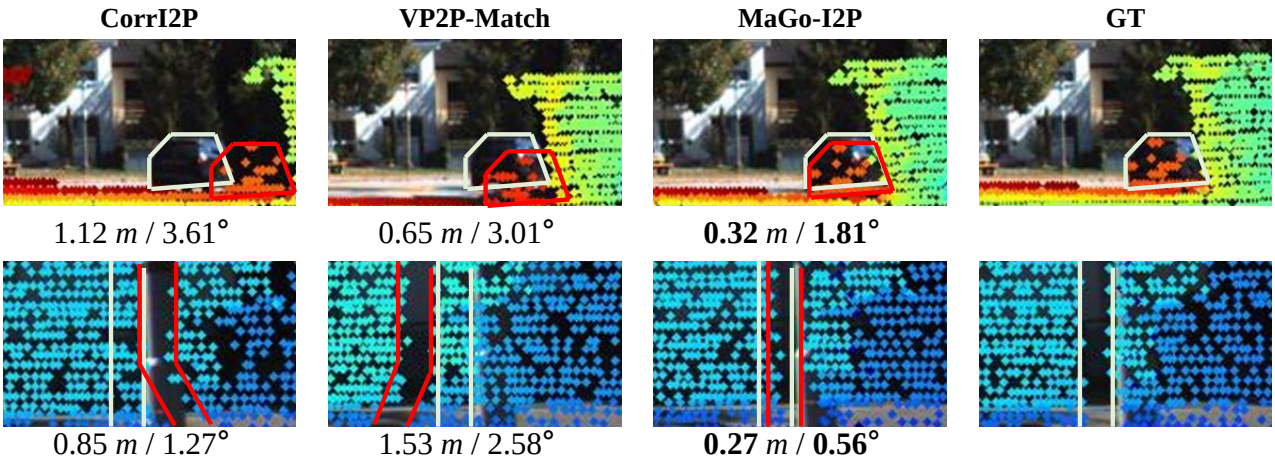


Figure 4: Visual comparison of Image-to-Point Cloud registration results under KITTI Odometry [7], where the red boxes represent the predicted results and the white boxes represent the ground truth. RTE (m) and RRE (°) are labeled below the results for different methods.

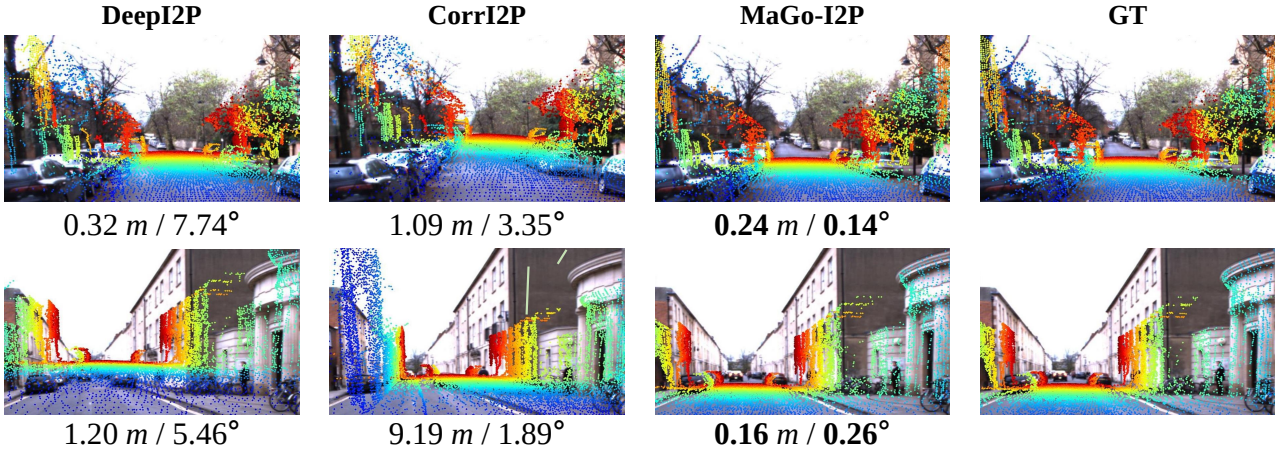


Figure 5: Visual comparison of Image-to-Point Cloud registration results on Oxford Robotcar [7]. RTE (m) and RRE (°) are labeled below the results for different methods.

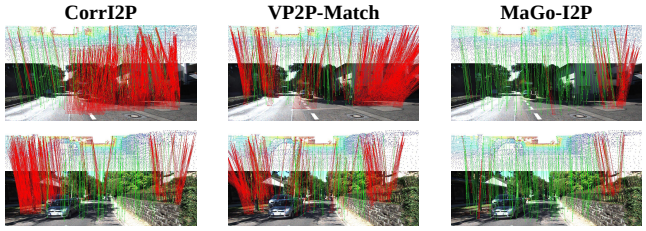


Figure 6: Pixel-to-Piont correspondences established by different methods on KITTI Odometry. The line segments represent correspondences, where red indicates false ones and green indicates correct ones.

rejection capabilities. It is important to note that other competitors did not specifically optimize their strategies for 2D-3D matching, whereas our MaGo-I2P leverages pseudo-RGBD generation to facilitate cross-modal feature extraction and introduces a coarse-to-fine 2D-3D Matching strategy to eliminate outlier matches. Additionally, the design of multiple weak classifiers (ORD, point cloud classifier) in MaGo-I2P enhances the network's representational capacity.

Overall, our method demonstrates superior performance in terms of both IR and FMR under various threshold settings. Furthermore, qualitative experimental results illustrated in Fig. 6 show correct matches represented by green lines and incorrect matches by red lines. Compared to CorrI2P and VP2P-Match, the pixel-to-point correspondences established by MaGo-I2P are more accurate, further confirming the effectiveness of the aforementioned components. Notably, while CorrI2P and VP2P-Match adopt a similar pose estimation paradigm as our method, the utilization of the geometry recovery module and the Mamba-based backbone enables our method to exhibit state-of-the-art 2D-3D matching accuracy.

Table 2: Quantitative results of point-to-pixel correspondences on KITTI Odometry. The best results are highlighted in bold. Here, we report the IR (%)/ FMR (%) with different thresholds τ_d(pixel) / τ_m(%)

	IR_1 / $FMR_{0.2}$ ↑	IR_2 / $FMR_{0.2}$ ↑	IR_3 / $FMR_{0.2}$ ↑
CorrI2P [26]	10.84 / 18.12	27.69 / 64.60	42.18 / 82.38
VP2P [39]	21.23 / 68.49	44.39 / 96.54	69.95 / 97.87
MaGo-I2P	**39.58 / 91.26**	**67.67 / 98.94**	**81.34 / 99.73**

4.4 Runtime and GPU Memory Usage

The pose inference time and GPU memory usage of different I2P registration methods were compared, and the quantitative results are reported in TABLE 3. It can be seen that our MaGo-I2P is the best trade-off between the pose inference time and GPU memory usage. Specifically, our MaGo-I2P is superior to all competitors in terms of pose inference time while taking up acceptable GPU memory usage. Notably, the pose inference time of DeepI2P and CorrI2P is significantly higher than other methods. The main reason is that the former only establishes ambiguous 2D-3D matching and uses time-consuming inverse camera projection to optimize pose. The latter is limited by a large number of matching external points, which increases the cost of iterative optimization. Besides, VP2P-Match relies on a transformer-based 3D backbone, which requires a large amount of GPU memory. In contrast, thanks to the hardware-aware algorithm of Mamba [9], our MaGo-I2P can quickly infer poses with lower GPU memory usage, making it a suitable algorithm for low-resource devices.

Table 3: Runtime & GPU Memory usage.

	Pose infer. (s) ↓	GPU Mem. (MB) ↓
DeepI2P (3D) [14]	39.39	**2906**
DeepI2P (2D) [14]	26.01	**2906**
CorrI2P [26]	9.86	3208
VP2P-Match [39]	0.31	17169
MaGo-I2P	**0.29**	4422

Table 4: Ablation studies of the components in MaGo-I2P.

GR	Ma	CF	RTE (m) ↓	RRE (°) ↓
			2.12 ± 8.36	5.46 ± 6.48
✓			1.56 ± 1.94	3.88 ± 4.92
✓		✓	0.84 ± 1.19	3.30 ± 5.12
✓	✓		0.94 ± 1.16	3.49 ± 6.56
✓	✓	✓	**0.66 ± 0.75**	**3.04 ± 5.25**

4.5 Ablation Study

To validate the effectiveness of the proposed components, we conducted ablation experiments on the KITTI Odometry dataset, and the quantitative results are shown in TABLE 4. The components examined are detailed as follows:

- **GR**: Geometry recovery module for images;
- **Ma**: Mamba-based 3D backbone for point clouds;
- **CF**: Coarse-to-fine matching strategy.

To guarantee the proper operation of the I2P framework, **Ma** and **CF** were replaced by other modules with the same functions. Specifically, we replaced the coarse-to-fine matching strategy with the matching strategy adopted in CorrI2P [26]. The Mamba-based 3D backbone was substituted with a transformer-based 3D backbone [37]. As for **GR**, it can be removed, meaning that the 2D backbone only extracted features from RGB images. TABLE 4 indicates that each module contributes positively to the I2P task, demonstrating the effectiveness of our proposed geometry recovery module and the Mamba-based 3D backbone. We believe this is attributed to the global modeling ability of **Ma** and the geometric alignment ability of **GR**.

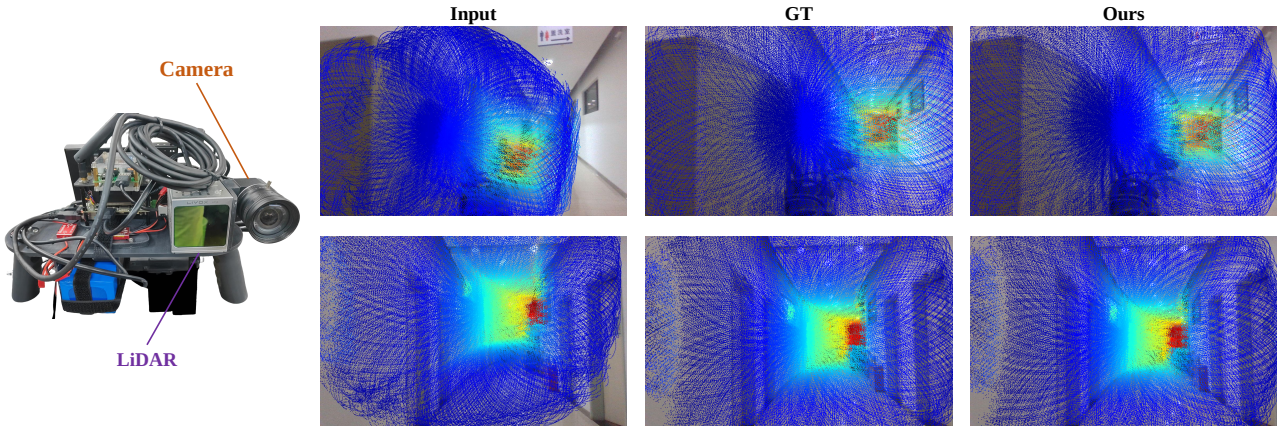


Figure 7: Visualization of extrinsic correction. On the left is the custom-built device, which is equipped with a camera and a LiDAR. On the right are the visualization results of the extrinsic calibration correction.

4.6 Application in LiDAR - Camera Extrinsic Correction

In this section, we demonstrate the potential of MaGo-I2P when applied to the LiDAR-camera calibration task.

As shown in Fig. 7, a custom-built device equipped with a monocular camera (MV-CA050-12UC) and LiDAR (Livox Avia) was used to validate the effectiveness of MaGo-I2P in the LiDAR-camera calibration task. First, the extrinsics were calibrated offline and time synchronization was completed[34]. Then, 562 pairs of image-point cloud data were collected, with 462 pairs used as the training set and 100 pairs used as the testing set. Next, the self-collected data was used to fine-tune the pre-trained MaGo-I2P model. During the testing phase, extrinsic parameter noises ranging from 0-5°and 0-10cm were randomly applied to the point clouds.

Fig. 7 qualitatively shows the extrinsic correction performance of MaGo-I2p. The visualization shows that the misalignment of the input data is diverse. Despite the different misalignments, the extrinsic estimation made by MaGo-I2P have few differences with ground truth. These results demonstrate the potential of MaGo-I2P for application in the extrinsic parameter correction task.

5 Conclusion

In this paper, we propose a novel I2P registration framework with the Mamba-based backbone and geometry recovery, MaGo-I2P. It proposes to alleviate the modality gaps between images and point clouds by recovering image depth, thereby improving the accuracy of 2D-3D matching. Moreover, by introducing the Mamba-based 3D backbone, MaGo-I2P can quickly extract features from images and point clouds with lower GPU memory usage. As a result, our approach demonstrates outstanding registration accuracy, fast pose inference, and lower GPU memory dependency, which indicates that MaGo-I2P is a suitable method for low-resource devices. We believe that the proposed MaGo-I2P not only contributes to the I2P registration task but is also applicable to other tasks requiring cross-modal data association, such as sensor calibration and multi-robot cooperative localization.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62272343.

References

[1] Lin Bie, Siqi Li, and Kai Cheng. 2024. Image-to-Point Registration via Cross-Modality Correspondence Retrieval. In *Proc. ACM Int. Conf. Multimedia Retrieval.* 266–274.
[2] 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.
[3] Guangyan Chen, Meiling Wang, Yi Yang, Li Yuan, and Yufeng Yue. 2024. Fast and Robust Point Cloud Registration with Tree-based Transformer. In *Proc. IEEE Int. Conf. Robot. Autom.* 773–780.
[4] Tri Dao and Albert Gu. 2024. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. In *Proc. Int. Conf. Mach. Learn.*
[5] 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.
[6] Martin A. Fischler and Robert C. Bolles. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* 24, 6 (1981), 381–395.
[7] 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.
[8] Marcel Geppert, Peidong Liu, Zhaopeng Cui, Marc Pollefeys, and Torsten Sattler. 2019. Efficient 2D-3D Matching for Multi-Camera Visual Localization. In *Proc. IEEE Int. Conf. Robot. Autom.* 5972–5978.
[9] Albert Gu and Tri Dao. 2023. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv:2312.00752* (2023).
[10] 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.
[11] Yurim Jeon and SeungWoo Seo. 2022. EFGHNet: A Versatile Image-to-Point Cloud Registration Network for Extreme Outdoor Environment. *IEEE Robot. Autom. Letters* 7, 3 (2022), 7511–7517.
[12] 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.
[13] 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.
[14] 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.
[15] Minhao Li, Zheng Qin, Zhirui Gao, Renjiao Yi, Chenyang Zhu, Yulan Guo, and Kai Xu. 2023. 2D3D-MATR: 2D-3D Matching Transformer for Detection-Free Registration Between Images and Point Clouds. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 14128–14138.

[16] Yaoqing Li, Sheng-Hua Zhong, Shuai Li, and Yan Liu. 2023. A Robust Deep Learning Enhanced Monocular SLAM System for Dynamic Environments. In *Proc. ACM Int. Conf. Multimedia Retrieval*. 508–515.
[17] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. 2024. PointMamba: A Simple State Space Model for Point Cloud Analysis. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.*
[18] Li Ling, Jun Zhang, Nils Bore, John Folkesson, and Anna Wåhlin. 2024. Benchmarking Classical and Learning-Based Multibeam Point Cloud Registration. In *Proc. IEEE Int. Conf. Robot. Autom.* 6118–6125.
[19] Jiuming Liu, Ruiji Yu, Yian Wang, Yu Zheng, Tianchen Deng, Weicai Ye, and Hesheng Wang. 2024. Point Mamba: A Novel Point Cloud Backbone Based on State Space Model with Octree-Based Ordering Strategy. *arXiv:2403.06467* (2024).
[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 9992–10002.
[21] David G. Lowe. 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* 60, 2 (2004), 91–110.
[22] 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.
[23] Chunghyun Park, Yoonwoo Jeong, Minsu Cho, and Jaesik Park. 2022. Fast Point Transformer. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 16928–16937.
[24] 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.
[25] 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.
[26] 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.
[27] 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.
[28] Yifan Sun, Changmao Cheng, Yuhan Zhang, Chi Zhang, Liang Zheng, Zhongdao Wang, and Yichen Wei. 2020. Circle Loss: A Unified Perspective of Pair Similarity Optimization. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 6397–6406.
[29] 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. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 12, Article 388 (2024), 21 pages.
[30] 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.
[31] Bing Wang, Changhao Chen, Zhaopeng Cui, Jie Qin, Chris Xiaoxuan Lu, Zhengdi Yu, Peijun Zhao, Zhen Dong, Fan Zhu, Niki Trigoni, and Andrew Markham. 2021. P2-Net: Joint Description and Detection of Local Features for Pixel and Point Matching. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 15984–15993.
[32] Yifan Wang, Xingyi He, Sida Peng, Dongli Tan, and Xiaowei Zhou. 2024. Efficient LoFTR: Semi-Dense Local Feature Matching with Sparse-Like Speed. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 21666–21675.
[33] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data. In *Proc. IEEE Conf. Comput. Vis. Patt. Recog.* 10371–10381.
[34] Chongjian Yuan, Xiyuan Liu, Xiaoping Hong, and Fu Zhang. 2021. Pixel-Level Extrinsic Self Calibration of High Resolution LiDAR and Camera in Targetless Environments. *IEEE Robot. Autom. Letters* 6, 4 (2021), 7517–7524.
[35] Ruonan Zhang, Xiaohang Liu, Ge Li, Thomas H. Li, and Pengjun Zhao. 2024. Sketch-aided Interactive Fusion Point Cloud Place Recognition. In *Proc. ACM Int. Conf. Multimedia Retrieval*. 1115–1119.
[36] Tong Zhang. 2004. Statistical behavior and consistency of classification methods based on convex risk minimization. *The Ann. Statistics* 32, 1 (2004), 56–85.
[37] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. 2021. Point Transformer. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.* 16239–16248.
[38] Yu Zhong. 2009. Intrinsic shape signatures: A shape descriptor for 3D object recognition. In *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop*. 689–696.
[39] Junsheng Zhou, Baorui Ma, Wenyuan Zhang, Yi Fang, Yu-Shen Liu, and Zhizhong Han. 2023. Differentiable Registration of Images and LiDAR Point Clouds with VoxelPoint-to-Pixel Matching. In *Proc. Adv. Neural Inf. Process. Syst.*, Vol. 36. 51166–51177.