

IGICP: Intensity and Geometry Enhanced LiDAR Odometry

Li He ^{ID}, Member, IEEE, Wen Li ^{ID}, Yisheng Guan ^{ID}, Member, IEEE, and Hong Zhang ^{ID}, Fellow, IEEE

Abstract—Point matching and pose optimization are two important processes in LiDAR odometry. The former one is prone to noise and initial pose estimation, while the latter often falls into local minima due to improper matched points. In this paper, we propose a new point pair similarity method in the combination of the normal vector, the smallest eigenvalue of the spatial covariance matrix, and the KL divergence of local intensity values. In pose optimization step, we use both the proposed point pair similarity and planarity as the weight. Experimental results show that it may guarantee higher accuracy with our method and is able to run at 27 FPS on a common PC.

Index Terms—LiDAR odometry, intensity, SLAM, ICP, intelligent vehicle.

I. INTRODUCTION

B EING one of the fundamental technologies of intelligent vehicles, odometry is widely used in the poses estimation which is essential for many tasks such as planning and control. It is frequently utilized in automatic driving [1], automatic mobile robots [2], and SLAM [3], [4]. One of the most commonly used sensors for intelligent vehicles is Light Detection And Ranging (LiDAR), by which the LiDAR sensors directly provides accurate range measurement, typically with the centimeter-level error. Compared with visual odometry, LiDAR odometry is accurate and robust to illumination changes [5], [6]. Given two sets of points in different coordinate systems, the goal of LiDAR odometry is to find the transformation that best aligns the two point sets, as shown in Fig. 1.

Manuscript received 3 November 2023; accepted 18 November 2023. Date of publication 24 November 2023; date of current version 23 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62173096, in part by the Pearl River Talent Recruitment Program under Grant 2019QN01X761, in part by the Shenzhen Key Laboratory of Robotics and Computer Vision under Grant ZDSYS20220330160557001, in part by the Open Projects Program of State Key Laboratory of Multimodal Artificial Intelligence Systems under Grant MAIS-2023-08, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515140044. (Corresponding author: Hong Zhang.)

Li He and Hong Zhang are with the Shenzhen Key Laboratory of Robotics and Computer Vision, Southern University of Science and Technology, Shenzhen 518055, China, and also with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: hel@sustech.edu.cn; hzhang@sustech.edu.cn).

Wen Li and Yisheng Guan are with the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou 510006, China (e-mail: 2112101119@mail2.gdtu.edu.cn; ysguan@gdtu.edu.cn).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIV.2023.3336376>.

Digital Object Identifier 10.1109/TIV.2023.3336376

The source code is available at <https://github.com/LiHeUA/IGICP>.

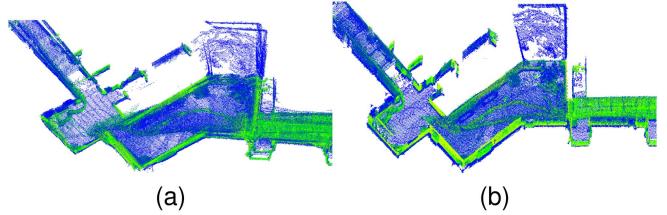


Fig. 1. Aligned point clouds with estimated poses. (a) Pose estimated by GICP, and (b) Pose estimated by IGICP.

Although a lot of research studies have been carried out in this field, point matching and minimization have become the most challenging problems in LiDAR odometry. A good set of matched points will significantly accelerate odometry with high accuracy. Due to its conceptual simplicity and good performance, nearest neighbor is very popular in practice, successfully applying in numerous real-world tasks. However, this approach is known for its tendency of falling into local minima. To improve point matching, previous efforts have been devoted to employing more efficient features [7], [8] or utilizing line or plane constraints [9], [10], [11]. Additional features, such as the normal vector and curvature of one point, are widely adopted to evaluate the similarity of point candidates. However, the normal vectors or the curvature values cannot distinguish points on the same plane because of the identical local structure in this case. Furthermore, some methods, such as those in the application to point matching with plane (or line) points, are not always reliable when the point cloud contains insufficient planes.

Given the matched points, many minimization methods adopt an equal weight for all point pairs. Nevertheless, the equal weight fails to represent the matching quality and the uncertainties of the linked points position. On the one hand, there will exist potentially wrong matching, of which the weight should be low due to these poorly matched points. On the other hand, even with the correct matching, the positions may be noisy in the two matched points, thus regrettably reaching the same result of being low weights.

In this article, we propose an intensity-spatial LiDAR descriptor, called IGICP, for LiDAR odometry. The point representation is extended by augmenting the Euclidean coordinates with the surface normal and LiDAR intensity. The proposed representation is to calculate the similarity of point candidates. In the minimization step, the receptive weight for each represents the

pair-wise reliability and point-wise uncertainty. The reliability of matching is measured via the proposed similarity, meanwhile point uncertainty by point planarity. The main contributions of this article are as follows:

- We propose an efficient and robust point matching method. In the process, the normal vector, the corresponding eigenvalue, and the KL divergence of reflection intensity are jointly applied to match points. The employment of both local geometry features and intensity provides a high correct match rate in return. It also shows that the intensity of the KL divergence is related to the range and the incidence angle. In general, the KL divergence grows in quartic with the range factor. To ensure a range-free intensity measurement, we propose to normalize the raw input intensity.
- We propose a method to calculate the weight for each matching, which takes both pair-wise uncertainty and point-wise uncertainty into account. Meanwhile, the proposed similarity is employed to evaluate the pair wise uncertainty of matching, with the average planarity representing the uncertainty of a single point.
- Experimental results on KITTI, M2DGR and self-collecting data sets show that the proposed IGICP outperforms the SOTA methods on most KITTI sequences and all M2DGR sequences in terms of ATE. Compared with GICP, IGICP achieves convergence with fewer iterations hence requiring lower time cost than GICP.

The rest of this article is organized as follows. Section II gives the literature review. Section III derives our main contribution to the intensity and geometry-enhanced ICP. Experimental results are presented in Section IV, and conclusions are shown in Section V.

II. RELATED WORK

A. Point-Based Methods

LiDAR Odometry can be divided into three branches: point-based methods, distribution-based methods, and learning-based methods. ICP [12] is the exemplar of point-based methods and among the earliest odometry work. ICP directly establishes the matches between the nearest points and constructs the cost function according to the Euclidean distance. However, it is still a challenging problem in finding one-to-one correspondences between two point clouds. Therefore, to improve the suitability of ICP, point-to-line ICP [13] and point-to-plane ICP [14] are proposed, which estimate the transformation by minimizing the point-to-line distance or the point-to-plane distance. NICP [7] takes further advantage of the local features (normals and curvatures) for each point to improve point matching. Extracting corner points and plane points according to the curvature of points on each scan line, LOAM [9] estimates the transformation in the combination of both point-to-line and point-to-plane models. LEGO-LOAM [15] filters noise points with the point cloud segmentation and divides them into ground and non-ground points. As a variant of LOAM, LOAM-livox [16] realizes point selection and iterative pose optimization for LiDAR with a small field of view (FoV). R-LOAM [17] extends LOAM with point-to-mesh correspondences. MULLS [10] extracts more features

(ground, facade, pillar, beam, etc.) to improve the robustness against noise. Point-based methods always show high accuracy in pose estimation, but ICP and its variants sometimes may fall into local minima with an improper initial pose.

B. Distribution-Based and Learning-Based Methods

Distribution-based methods are also widely applied in LiDAR odometry. NDT [18], [19], [20], [21] is one of the earliest and most popular methods. The target cloud in NDT is first voted into voxel grids where in-grid points are described by normal distributions eventually. Then, each point of the source cloud gets a score by voting it into these distributions, and the transformation can be estimated by optimizing the point scores. NDT describes the target point cloud in the way of normal distributions, while GICP [22] does both the source point cloud and the target point cloud instead. VGICP [23] extends GICP with voxelization to avoid costly nearest neighbor search while retaining its accuracy. LITAMIN [24] is stabilized with normalization of the cost function by the Frobenius norm and a regularized covariance matrix. LITAMIN2 [25] further introduces symmetric KL-divergence [26] to measure the difference between local distributions, and the cost function includes both distances among points and differences in distributions. NDT and its variants are shown to be robust for the initial poses in many applications, but NDT-estimated pose is often less accurate compared with point-based methods. So, it is common in practice to apply NDT as the coarse pose estimator and adopt other accurate pose estimation methods, such ICP, to fine-tune the pose.

In recent years, learning-based methods drawn more attention in the field. LO-Net [27] is an end-to-end method with a new mask weighted geometry constraint loss. LodoNet [28] transfers LiDAR scans to images and reforms the LiDAR place description problem in the pursuit of image feature extraction. DeepCLR [29] provides an end-to-end architecture based on radius search and mini-PointNet without extracting explicit point correspondences. PWCLO-Net [30] proposes to use hierarchical embedding mask optimization. Recently, HPPLO-Net [31] has utilized a differentiable weighted Point-to-Plane SVD to solve the pose matrix. The quadratic growth in time complexity prevents the popular transformer module for point cloud processing. TransLO [32] projects 3D points to a 2D surface and feeds the 2D projections to a local transformer with linear complexity. EfficientLO-Net [33] is an end-to-end framework and employs an hierarchical embedding mask to filter mismatched points. Learning-based methods have their advantages in accurate pose estimation. However, due to the lack of a large-scale LiDAR data set, the generalization ability of learning-based methods is still questionable. In addition, such work requires high computing resources, typically with the acceleration by GPUs.

C. Point Matching

The existing methods for selecting correspondences can be categorized into three classes: the nearest neighbor (NN), NN with plane or line constraints, and feature matching, as shown in Table I. It is the most common and simplest method to take

TABLE I
METHODS OF POINT MATCHING

Method	Main idea	Problem
The nearest point [12] [22]	The nearest point in the target scan	Prone to noise, voxel downsampling, and transformation
Plane or line constraint [9] [10] [11]	NN for plane (or line) points only	Accurate plane or line extraction required
Features-based [7] [34] [8] [35] [36]	Feature matching via local structure/property, such as normal vectors, curvature or intensity	Low accuracy for points with similar local structure, intensity prone to noise, range and view angle

the nearest neighbor point in the target scan as the correspondence [12] [22]. Being the flaw of NN in Euclidean space, it is proneness to noise, voxel downsampling, and transformation. Plane or line constraint methods extract different kinds of key points [9] [10] [11], such as corner points, line points or plane points, and then find the nearest neighbors among points within the same kind. These methods require accurate point classification which is often time-consuming. Feature-based methods match the points by using local structure or point features, such as normal vector, curvature [7], [8], [34], [35], [36] or intensity. However, this kind of method may fail to work when the local geometry features are similar, for example some points on the same plane.

D. Intensity-Based Methods

In the last decade, intensity of a LiDAR sensor have shown its success in many SLAM tasks [37] [38]. Intensity-SLAM [39] utilizes a weighted sum of both spatial and intensity distances among the KNNs which are embodied in back-end graph optimization, while we focus on the front-end LiDAR odometry. Other than processing on the raw point cloud, several methods project the point cloud into a cylindrical image and hence benefit from the plentiful tools of 2D image processing. SLO [40] extracts keypoints of the cylindrical image via SuperPoint and performs an exhaustive search among keypoints by matching their descriptors. Then, SLO iteratively excludes outliers determined by RANSAC and exploits new keypoints in a self-supervised manner. Adopting a cylindrical image to process the point cloud, InTEnLOAM [41] additionally calculates the intensity difference image, the normal image, and the curvature image, alongside with the intensity histogram. According to the features, InTEn-LOAM categorizes each key point into four classes. MCGICP extends GICP with additional input channels, such as color or intensity. MCGICP [42] first projects a 3D point to its local 2D surface, and then calculates the weighted covariance with weights generated by intensity or color. Intensity-based methods always assume a consistent intensity of a local subset, but in practice, the intensity values are prone to noise, range, and view angle.

E. Weighted Cost Function

Given matched points, the minimization step in ICP and its variants attempts to optimize rotation \mathbf{R} and translation \mathbf{t} from

$$\arg \min_{\mathbf{R}, \mathbf{t}} \sum w_i \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_j\|^2 \quad (1)$$

where \mathbf{x}_i and \mathbf{y}_j form a match, and w_i is the weight of one match.

The design of weight can be categorized into three classes: equal weight, pair-wise weight, and point-wise weight, as shown in Table II. Equal weight is a popular choice due to its simplicity, but is easily affected by incorrect matches, noise points or clutter points. The pair-wise weight usually takes the local geometry features as the weight by the calculated similarity. The weights in [34], [36] are inversely correlated to the distance between the two points of a pair. The difference of eigenvalues of the local covariance matrix is used to form a geometry weight in [43]. Pair-wise weight does not take into account the quality of points themselves, therefore always needs to filter out noisy points or cluttered points in advance. The point-wise weight methods focus on measuring the quality of the points to form the matching. Researchers in [44] design the weight by the depth of points. In [45], the weight is assigned according to the angle of incidence of a LiDAR beam. The weight in [46] is formed according to the sensor error model. Point-wise weight ignores the quality of the correspondences and may fail to work in the case of a high incorrect match rate.

III. INTENSITY AND GEOMETRY BASED ICP

Point matching and minimization are two important steps in ICP-based odometry. In this section, we propose our intensity and geometry-enhanced ICP. First, we present the proposed similarity of the points in details. Then, we discuss the design of weights in ICP minimization.

A. An Overview of Intensity and Geometry Based Similarity Matching

Due to its simplicity and efficiency, the nearest neighbor in the Euclidean space is common in selecting correspondences. NN often fails to obtain correct matches for points with a complex structure, such as corners. Fig. 2 shows two cases extracted from the first and second scans in KITTI 00. In Fig. 2(a), most of the NN-selected correspondences are incorrect where the points on the vertical plane are matched to the horizontal plane. Our goal is to improve the matches, as shown in Fig. 2(b), by introducing reliable features.

We use a two-step point matching strategy by first finding match candidates via KNN search and then determining the final correspondence by feature matching. For each source point, we search its K nearest neighbours in the target scan as the candidates for the first place. Then, for each candidate point pair, we define the geometry similarity S_G and the intensity similarity S_I and calculate the final similarity accordingly. We

TABLE II
METHOD OF WEIGHTING

Method	Main idea	Problem
Equal weight [12] [22]	Equal weight	Wrong matches, noisy points or cluttered points in the environment will reduce the accuracy
Pair-wise weight [34] [43] [36]	Weight calculated from pair-wise properties, such as distance, eigenvalues or local structure	Not referring to the quality of points, prone to noisy points or cluttered points
Point-wise weight [44] [45] [46]	Weight calculated from the quality of one single point, such as angle of incidence or range	Not referring to the quality of point pairs

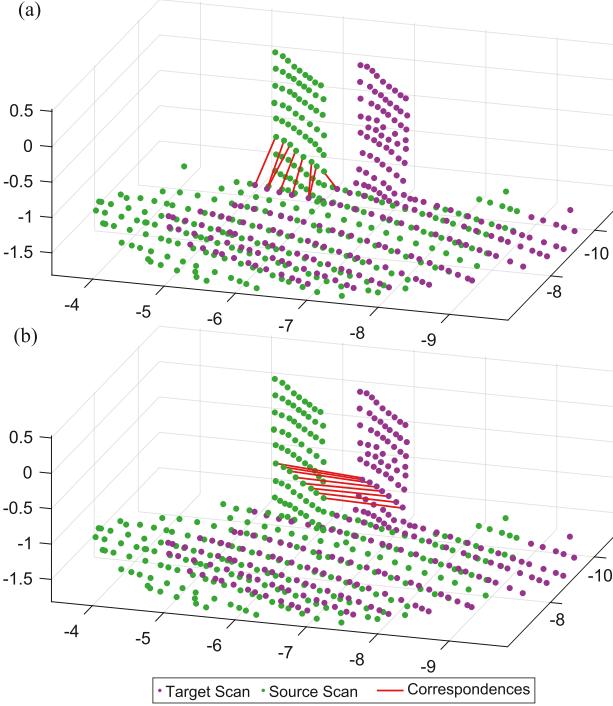


Fig. 2. Example of point matching in KITTI 00. The lines link the matched points. (a) Matches obtained by NN. (b) Matches obtained by the proposed similarity.

choose the pair with the highest similarity among the candidates as the matched pair. KNN search for candidates can be done by kd-tree searching and hence maintains the advantage of speed and simplicity of ICP. The follow-up feature matching then ensures it more accurate and reliable than NN only.

Given both the geometry and intensity similarities, the final similarity S of one match is defined as

$$S = S_G \cdot S_I \quad (2)$$

In the next two sections, the geometry similarity S_G and the intensity similarity S_I will be discussed in details.

B. Geometric Similarity S_G of a Point Pair

Other than Euclidean coordinates, many other features are introduced to represent points, including normal vectors, curvature or intensity values. The normal vector of a point is widely used in previous work indicating the local structure. In our geometry similarity, we adopt the normal vector as one of the features. For one point p_i , we first calculate its covariance matrix \mathbf{C}_i . Then, we get the eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ of \mathbf{C}_i and the corresponding

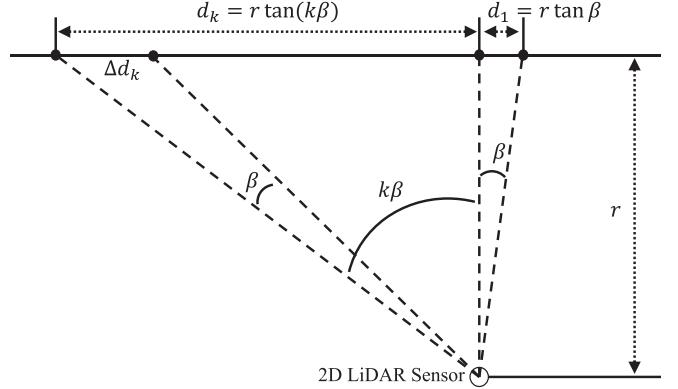


Fig. 3. Demonstration of view angle and its corresponding nearest neighbor distance. The NN distances are related to the view angles, resulting in various local variances of points on the same plane. The instability of variance may lead to unstable curvature estimation.

eigenvectors $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, where $\mathbf{n} = \mathbf{v}_3$ is the normal vector of p_i and λ_3 is the eigenvalue which represents the dispersion of points in the direction of the normal vector.

In many previous researches [7], [8], [34], curvature κ is used to describe the local structure of a point,

$$\kappa = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (3)$$

Despite its simplicity and efficiency, we argue that the calculation of curvature is not robust under view changes. To illustrate this, an example is shown in Fig. 3 with a 2D LiDAR scanning. In Fig. 3, a 2D LiDAR sensor observes several points on a plane. Let r be the distance between the sensor to the plane, β be the angular resolution of the 2D LiDAR sensor, and d_k be the distance of the k -th point to the perpendicular line. Then, the distance Δd_k between the k -th point to the $(k-1)$ -th is

$$\begin{aligned} \Delta d_k &= r[\tan(k\beta) - \tan((k-1)\beta)] \\ &= r \left[\frac{\sin \beta}{\cos(k\beta) \cos((k-1)\beta)} \right] \\ &\approx \frac{r\beta}{\cos^2(k\beta)} \end{aligned} \quad (4)$$

where in the approximate equality we assume a small enough angular resolution $\beta \rightarrow 0$ and $\cos(k\beta) \approx \cos((k-1)\beta)$.

Since most existing methods adopt KNNs for covariance calculation, (4) then implies that the dominant variance of a side-view point is quite different from that of a front-view one. The variance, or the leading eigenvalue of the covariance

matrix, grows in proportional to $1/\cos^4(k\beta)$ as shown in (4), and consequently leading to unstable estimates of curvature.

Other than the problem of unstable variances of side-view points, (4) also implies the inequality of the two leading variances of a planar point which ideally are equal, $\lambda_1 = \lambda_2$. The horizontal and vertical angular resolutions of a LiDAR sensor may be quite different.¹ So, the variance in the vertical direction is typically larger than that in the horizontal one due to the unbalanced density in different directions.

To handle this problem, we substitute the smallest eigenvalue λ_3 for curvature. For planar points mentioned in this article, it shows that λ_3 is a proper approximation to the curvature. For a planar point, λ_1 and λ_2 , or the variances of the two leading dominant directions, are much larger than the variance on the third direction, $\lambda_1, \lambda_2 \gg \lambda_3$, and $\lambda_1 = \lambda_2$ is a constant scalar, in which the equality is always difficult to be obtained due to the unbalanced angular resolutions and view changes as discussed in Fig. 3 and (4). By substituting the turbulent λ_1 and λ_2 with a constant, (3) becomes

$$\kappa \approx \frac{\lambda_3}{2\lambda_1} \quad (5)$$

Equation (5) shows that for planar points, their curvature κ is approximately equal to λ_3 , up to a constant factor. So, we replace the curvature with the smallest eigenvalue λ_3 , to avoid the unstable calculation of both λ_1 and λ_2 , as the feature. It is easy to verify that λ_3 is robust to view changes as discussed in Fig. 3.

We use a four-dimensional vector v , consisting of the normal vector n and its corresponding eigenvalue λ_3 , for one point as the local geometry features,

$$v = [n, \alpha\lambda_3] \quad (6)$$

where α is a gain constant to balance the scale of λ_3 with the normal vector n . For the case of a planar point, the proposed geometry features v are dominated by the normal vector due to a small λ_3 . For a large λ_3 where no dominant direction occurs, λ_3 provides additional constraints on the local shape. The geometry similarity S_G of a point pair is defined as the cosine distance between their geometry features,

$$S_G = \frac{v_i v_j^T}{\|v_i\| \|v_j\|} \quad (7)$$

C. Intensity Similarity S_I

In addition to local structure, LiDAR sensors also take the reflection intensity of one beam as the default output. Intensity is usually used for global scan matching [47] [48] which focuses on the whole point cloud rather than individual points. Studies in [38] directly concatenate the intensity as an additional dimension in the features. It is known that the prime weakness of intensity is its proneness to noise, range, and view angle. So, a direct utility of single-point intensity always turns poor matching results.

¹For example, the azimuth angular resolution of a Velodyne HDL-64 LiDAR sensor is 0.08 ° but the elevation angular resolution is approximately 0.4 °.

To deal with the problem of unstable intensity, we employ the intensity values statistics for a local subset rather than a single point. We assume a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ for intensity values around one point p_i , where μ_i is the mean and σ_i^2 is the variance. We evaluate the intensity difference between two points with KL-divergence,

$$KL(\mathcal{N}_i || \mathcal{N}_j) = \log \frac{\sigma_j}{\sigma_i} + \frac{\sigma_i^2 + (\mu_i - \mu_j)^2}{2\sigma_j^2} - \frac{1}{2} \quad (8)$$

In point matching, the intensity similarity should be symmetric, $KL(\mathcal{N}_i || \mathcal{N}_j) = KL(\mathcal{N}_j || \mathcal{N}_i)$, but the original KL divergence fails to meet this requirement. So, we use the symmetric KL-divergence instead,

$$KL^{sym} = \frac{1}{2} [KL(\mathcal{N}_i || \mathcal{N}_j) + KL(\mathcal{N}_j || \mathcal{N}_i)] \\ = \frac{\sigma_i^2 + (\mu_i - \mu_j)^2}{4\sigma_i^2} + \frac{\sigma_j^2 + (\mu_i - \mu_j)^2}{4\sigma_j^2} - \frac{1}{2} \quad (9)$$

The range of KL^{sym} in (9) is $[0, +\infty)$. We normalize KL^{sym} to the range $[0, 1]$ by a Gaussian function to obtain the intensity similarity S_I ,

$$S_I = e^{-\frac{(KL^{sym})^2}{2\tau^2}} \quad (10)$$

where τ is a user-designed Gaussian scale parameter.

D. Intensity Normalization

The intensity collected by a LiDAR sensor is influenced by many factors and hence is not stable in practice. Before calculating the intensity similarity as described in Section III-C, we first normalize the received intensity values. We use the Lambertian model to construct the sensor intensity model [49],

$$I_r = I_o \times \frac{\phi^2}{4r^2} \eta_{sys} \eta_{atm} \rho \cos \theta \quad (11)$$

where I_o and I_r are the transmitted and received intensity, respectively, ϕ is the receiver aperture diameter, η_{sys} and η_{atm} are the system and atmospheric attenuation, respectively, r is the range from the sensor to the target, θ is the angle of incidence, and ρ is the target reflectance.

Assuming a constant I_o , ϕ , η_{sys} and η_{atm} , then the reflection ρ , which is an important property in shaping the material of the target point, is

$$\rho \propto \frac{r^2 I_r}{\cos \theta} \quad (12)$$

So, we normalize the received intensity by

$$I = I_r \frac{r^2}{\cos \theta} \quad (13)$$

and use I in the intensity similarity calculation. Note that many sensors come with a partial calibration function as the default [39]. For this kind of data, we can selectively normalize the distance or angle of incidence according to (12). Since the normal vector is also required in our construction of the geometry similarity, the intensity normalization of (13), especially

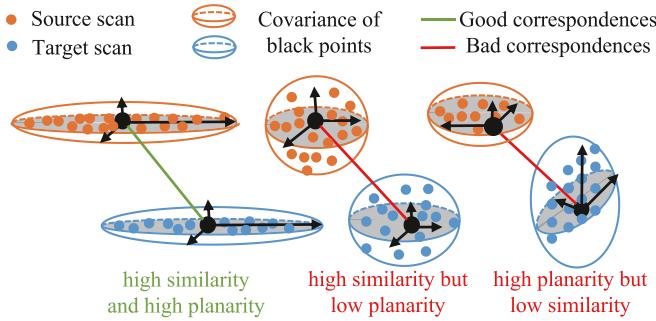


Fig. 4. Illustration of the proposed weight. Green lines indicate correspondences with high weights, which are high in both similarity and planarity, while red lines indicate correspondences with low weights, which are either low in similarity and/or in planarity.

the calculation of the incident angle θ , introduces very limited additional computational burden.

The intensity normalization is essential in our similarity calculation. Considering an ideal case of matching two scans represented respectively by their intensity normal distributions \mathcal{N}_i and \mathcal{N}_j , with only a slight translation between the two scans. We assume a constant range r_i for all KNNs in forming \mathcal{N}_i , and r_j for \mathcal{N}_j , and let $r_i = ar_j$. Then, according to the Lambertian model in (11), $\mu_j = a^2\mu_i$, $\sigma_j^2 = a^4\sigma_i^2$, and KL^{sym} in (9) becomes

$$KL_{r_i=ar_j}^{sym} = \frac{1}{4a^4} + \frac{a^4}{4} + \frac{\mu_i^2}{4\sigma_i^2} \left(\frac{1}{a^4} + 1 \right) (1 - a^2)^2 - \frac{1}{2} \quad (14)$$

Equation (14) implies that the KL divergence of two ideal scans grows in quartic concerning for the range factor a . The normalization in (12) eliminates the effect of range changes, ensuring a constant KL divergence with translations.

E. Weighted Cost Function

The weighted cost function has been successfully applied in many ICP-based methods. However, it is still a challenging problem to shape both pair-wise uncertainty and point-wise one for a matched pair with a proper weight. In this section, we propose our weight taking both the similarity between point pairs and the quality of each single point into account. Fig. 4 demonstrates the main idea of the proposed weight method and Fig. 5 displays the proposed weight on matching the first and second scans of KITTI 00.

The sparsity of LiDAR data makes it impossible to get exact overlap of the individual points even after ICP convergence. Point-to-plane match is a promising solution to overcome the LiDAR sparsity problem by introducing the plane constraint. Motivated by the plane constraint, we use the average planarity by eigenvalues of the covariance [50] to evaluate the point-wise uncertainty. For a pair of points p_i and p_j and their covariance eigenvalues $\lambda_1^i \geq \lambda_2^i \geq \lambda_3^i$ and $\lambda_1^j \geq \lambda_2^j \geq \lambda_3^j$, the average planarity is defined as

$$P = \frac{1}{2} \times \left(\frac{\lambda_2^i - \lambda_3^i}{\lambda_1^i} + \frac{\lambda_2^j - \lambda_3^j}{\lambda_1^j} \right) \quad (15)$$

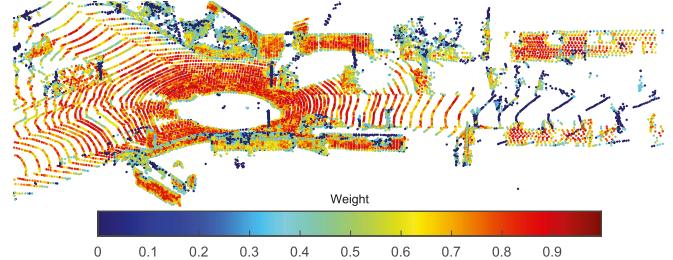


Fig. 5. Proposed pair-wise weights between the first and the second scans in KITTI 00. For viewing purposes, only the first scan is visualized. It can be seen that nearby planar points are given high weights because they are stable in minimization. Points at edges or corners are given low weights because of the confusion of point types. The weights of non-planar points, such as points from trees, are very low.

As to the pair-wise uncertainty, we employ S in (2) to represent the quality of a match. Finally, the weight of one matched point pair is

$$w_i = S_F \cdot P \quad (16)$$

Given the weight for each matched point pair, we minimize the following weighted cost function,

$$\mathbf{T} = \arg \min_{\mathbf{T}} \sum_i \left[w_i \mathbf{d}_i^T \left(\tilde{\mathbf{C}}_i^B + \mathbf{T} \tilde{\mathbf{C}}_i^A \mathbf{T}^T \right)^{-1} \mathbf{d}_i \right] \quad (17)$$

where

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix}, \tilde{\mathbf{C}}_i = \begin{bmatrix} \mathbf{C}_i & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \mathbf{d}_i = \mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_j \quad (18)$$

When w_i all equal to the unit, (17) is identical to GICP. In our article, we use (16) to calculate w_i . Note that, we assume that all the points are planar with the given weight, which therefore regularize the covariance matrix of each point by replacing its eigenvalues with $(1, 1, \epsilon)$, where ϵ is a small positive constant. To do this, let the 3×3 diagonal matrix Λ containing the eigenvalues of a covariance matrix \mathbf{C}_i on the diagonal, and \mathbf{V} be the corresponding eigenvectors, $\mathbf{C}_i = \mathbf{V}\Lambda\mathbf{V}^T$. Then, we substitute the diagonal elements of Λ with $(1, 1, \epsilon)$ and regularize \mathbf{C}_i by $\mathbf{C}_i \leftarrow \mathbf{V}\Lambda\mathbf{V}^T$. By regularizing the covariance matrix, non-planar points will produce big errors in (17). A similar approach can be found in FasterGICP [51] where points with large errors are filtered out directly.

F. Overview of IGICP

The framework of our method is shown in Fig. 6. Given two input point clouds, we first downsample both point clouds and find the KNNs for each point. Despite some improvement in the downsampling methods [52], [53], we use the classic version for simplification. We then calculate three eigenvalues of the covariance matrix centered at one point and build the point geometry features by (6). The geometry similarity then is the cosine of two geometry features. We then calculate the symmetric KL-divergence of intensity and take the scaled KL-divergence as the intensity similarity. The overall pair-wise similarity then is a simple multiplication of both geometry and

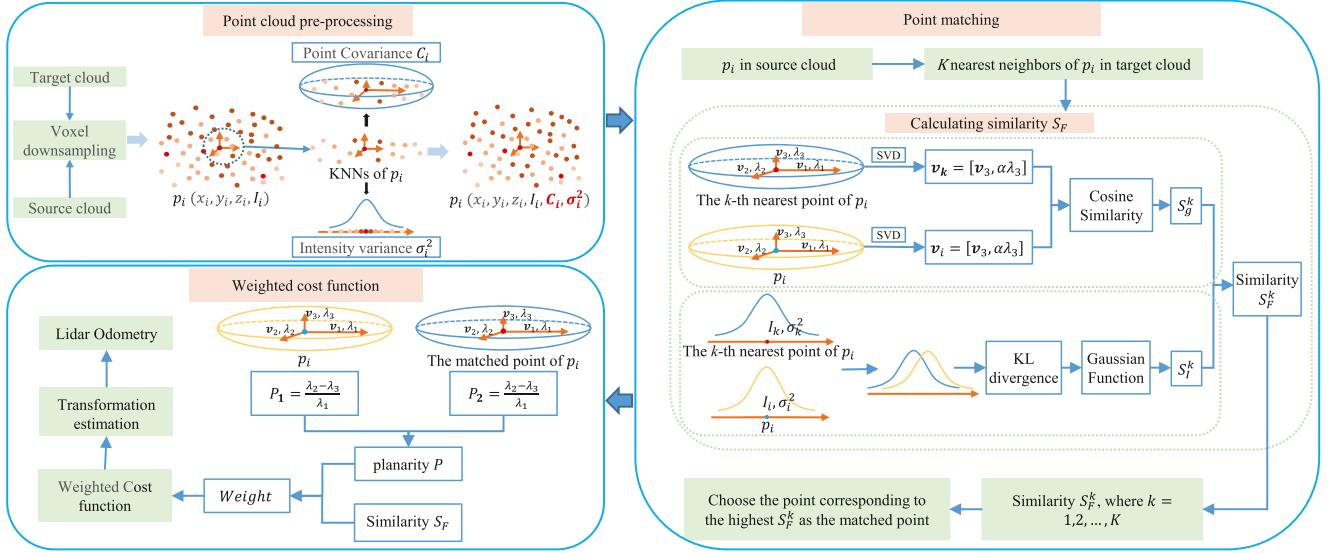


Fig. 6. Overview of IGICP.

intensity similarities, as in (2). For each point in the source scan, we find its KNNs in the target scan and choose the point with the highest similarity among the KNNs as the matched point.

Given the matched point pairs, we assign each pair two weights, the point-wise planarity weight P as in (15) and the pair-wise similarity weight S_F in (2). Then, we optimize the weighted ICP-like cost function to obtain the final odometry.

IV. EXPERIMENT

In this section, we present experimental results that illustrate the performance of the proposed IGICP. First, we compare the proposed method with several state-of-the-art methods on different datasets. Second, we perform an ablation study focusing on several key parts of IGICP. Finally, we analyze the iteration and time cost of IGICP. All experiments are conducted on a PC equipped with an Intel i7-12700HQ@2.30 GHz CPU and a 16 GB RAM.

A. Competing Methods and Data Sets

To evaluate the performance of our method, we compare it with ten state-of-the-art methods: LOAM [9], A-LOAM, LeGO-LOAM [15], SUMA [54], SUMA++ [55], LiTAMIN [24], LiTAMIN2 [25], MULLS-LO [10], MULLS-SLAM [10] and FasterGICP [51]. We adopt the average translational error (ATE) [56] for localization accuracy evaluation. We test the competitors on KITTI, M2DGR and VLP-16 data sets.

1) *KITTI*: KITTI Odometry dataset [56] is an autonomous driving dataset. The point cloud is collected by a Velodyne HDL-64E 3D laser scanner (10 Hz, 64 laser beams, range 100 m). KITTI Odometry provides 11 sequences (00–10) with ground truth trajectories, including various outdoor scenarios including urban roads (Seq. 00, 06, 07, 08), country roads (Seq. 02, 03, 04, 05, 09, 10), and highway (Seq. 01).

2) *M2DGR*: M2DGR dataset [57] is collected by a ground robot with a Velodyne VLP-32 C sensor and includes both indoor and outdoor environments.

3) *Sustech-Corridor*: SUSTech-Corridor dataset is collected by the authors and the point clouds are captured by a AGV with a Velodyne VLP-16 sensor. It contains 7562 scans along a corridor with a total distance of about 300 meters.

Given an input point cloud, we first downsample the point cloud with a 0.25 m voxel grid. We use $k = 5$ nearest neighbors in both point matching candidate selection and intensity KL divergence, and $k = 20$ for local covariance matrix C_i . We fine-tuned on the KITTI 00 sequence to find the optimal gain constant α in (6) and τ in (10). In return, $\alpha = 5$ and $\tau = 60$. We freeze all the parameters in the whole tests.

B. Accuracy Evaluation and Comparison

1) *Experiment on KITTI Dataset*: The results are displayed in Table III. Our method outperforms other competing methods on all the datasets except for sequences 01 and 04, and there is a significant improvement, especially on datasets with more planar features, such as sequences 05, 08 and 09. Sequence 01 is captured on a highway and sequence 04 on a straight road with trees on both sides. Both sequences 01 and 04 have limited planes exists a lot of dynamic objects. In this case, unstable point filtering, as adopted by MULLS and FasterGICP, is critical in selecting reliable point candidates. Although there is no direct point filter in IGICP, our method is still highly accurate on 01 and 04 and outperforms most other methods. IGICP ranks the third after the two MULLS methods on sequence 01 and is the second best next to FasterGICP on sequence 04. We further plot the trajectories estimated by our method and the ground truth in Fig. 7. IGICP provides accurate pose estimation even without any back-end processing, such as loop closure detection and g2o optimization.

TABLE III
ATE (%) EVALUATION AND COMPARISON ON KITTI DATASET

Sequence	00	01	02	03	04	05	06	07	08	09	10	avg.
Environments	urban	highway	country	country	country	country	urban	urban	country	country	country	
Num of frames	4541	1101	4661	801	271	2761	1101	1101	4071	1591	1201	
Total lenth(m)	3724	2453	5067	566	393	2205	1232	694	3222	1705	919	
LOAM [9]	0.78	1.43	0.92	0.86	0.71	0.57	0.65	0.63	1.12	0.77	0.80	0.84
A-LOAM [9]	0.79	1.96	4.57	0.95	0.77	0.50	0.62	0.45	1.11	0.74	1.01	1.22
LeGO-LOAM [15]	1.38	28.03	2.14	1.21	1.27	0.91	0.80	0.74	1.40	1.25	1.70	3.71
SUMA [55]	0.68	1.70	1.20	0.74	0.44	0.43	0.54	0.74	1.20	0.62	0.72	0.82
SUMA++ [56]	0.64	1.60	1.00	0.67	0.37	0.40	0.46	0.34	1.10	0.47	0.66	0.7
LiTAMIN [24]	0.95	11.30	1.25	1.17	18.70	0.70	0.63	0.45	1.03	1.06	3.78	3.73
LiTAMIN2 [25]	0.70	2.10	0.98	0.96	1.05	0.45	0.59	0.44	0.95	0.69	0.80	0.88
MULLS-LO [10]	0.51	0.62	0.55	0.61	0.35	0.28	0.24	0.29	0.80	0.49	0.61	0.49
MULLS-SLAM [10]	0.54	0.62	0.69	0.61	0.35	0.29	0.29	0.27	0.83	0.51	0.61	0.51
FasterGICP [52]	0.15	0.95	0.24	0.24	0.07	0.24	0.13	0.12	0.30	0.27	0.18	0.26
GICP [22]	0.19	1.22	0.37	0.26	0.30	0.17	0.13	0.14	0.26	0.17	0.19	0.31
IGICP (Ours)	0.13	0.94	0.18	0.21	0.12	0.13	0.08	0.07	0.17	0.14	0.14	0.21

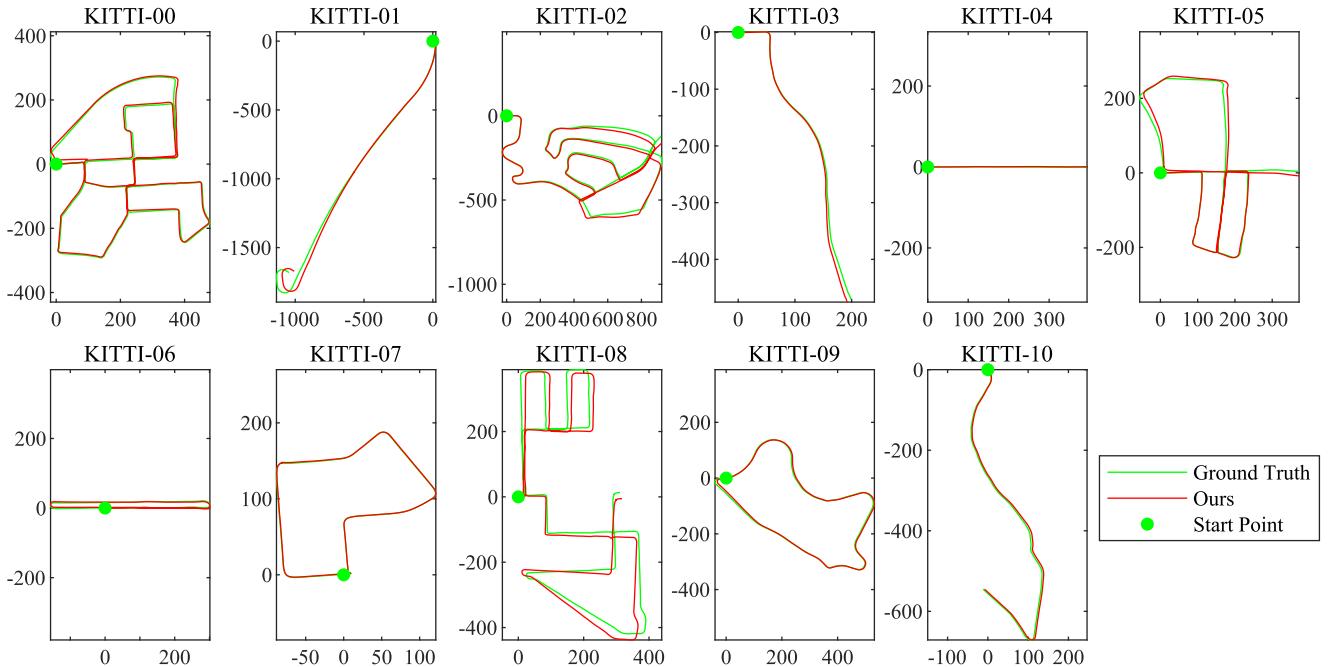


Fig. 7. Trajectory comparison on the KITTI dataset.

TABLE IV
ATE (%) EVALUATION AND COMPARISON ON M2DGR DATASET

Sequences	door02	lift04	room01	street06
length	57.36	79.28	27.17	479.63
Frames	1271	1262	728	4936
GICP [22]	1.01	3.80	1.43	0.79
VGICP [23]	0.94	7.52	3.48	1.11
FasterGICP [52]	1.36	5.71	1.45	0.77
IGICP (ours)	0.45	3.51	0.85	0.54

2) *Experiment on M2DGR Dataset:* We select four sequences from M2DGR under different scenarios: Room01 and Lift04 are collected indoors, Street06 is collected on a road, and Door02 is collected by moving from outdoors to indoors. Since some competing methods do not provide the source codes, in this test we compare GICP, VGICP, FasterGICP and IGICP only. The results, in terms of ATE, are shown in Table IV

and the trajectories in Fig. 8. It can be seen that our method performs best among these algorithms. Notably, the ATE values on Lift04 of all methods are much worse than that of the other sequences. This is due to the challenging vertical movement in Lift04 where the robot has taken a lift to move across different floors.

3) *Experiment on SUSTech-Corridor Dataset:* Besides testing on public datasets, we also evaluate on our SUSTech-Corridor dataset collected by the authors. Since there is no ground truth pose for this dataset, we build a map with the estimated poses and evaluate the accuracy of odometry by comparing the quality of the built map. After comparing our method with GICP, the results are shown in Fig. 9. We can see that the GICP map bends upwards and is poorly aligned at a pivot plot where a sharp rotation occurs. IGICP map is clean with very limited ghost points, indicating a small drift of the odometry estimated by IGICP.

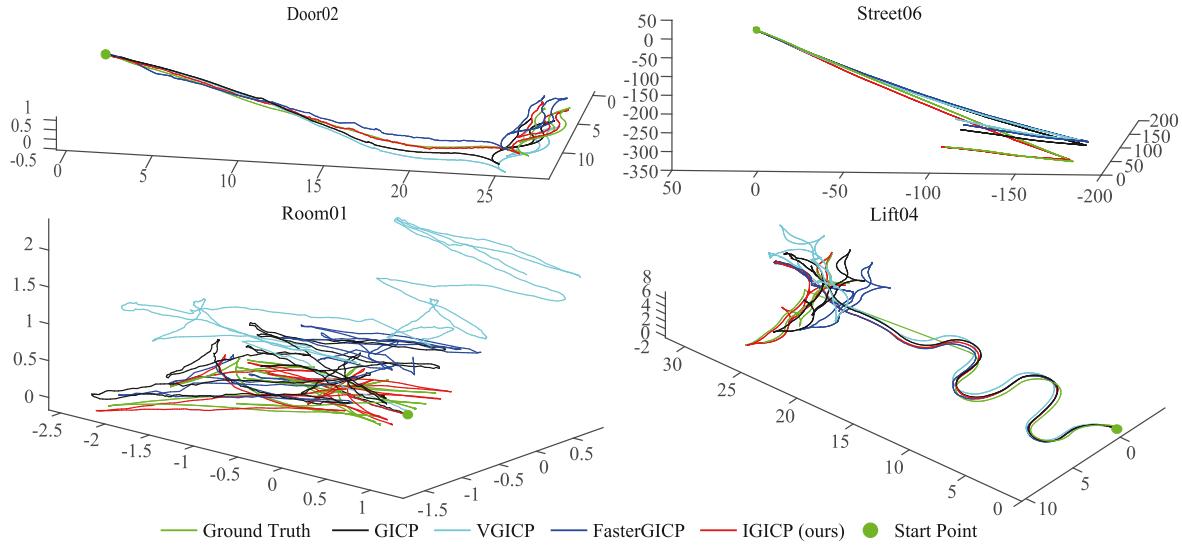


Fig. 8. Trajectory comparison on the M2DGR dataset.

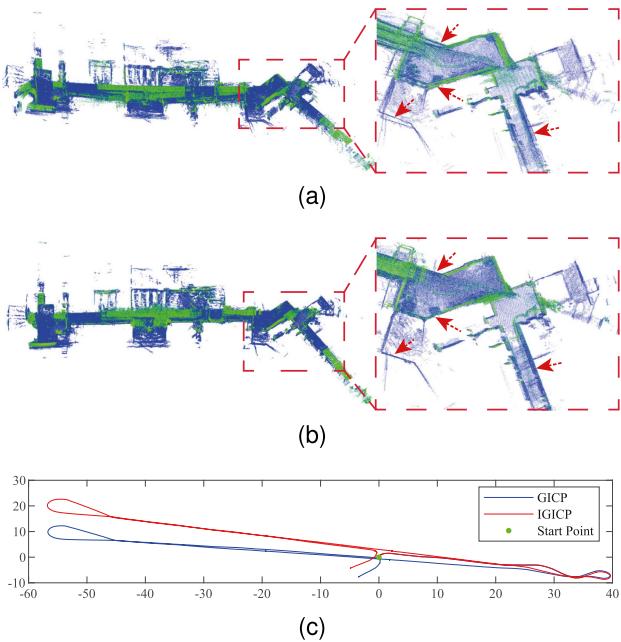


Fig. 9. Mapping results on SUSTech-Corridor dataset. (a) GICP, (b) IGICP and (c) Estimated trajectories.

C. Ablation Study

The main contribution of our work consists of two parts: point matching and weighted cost function. In particular, we introduce two features, geometry and intensity features, in point matching. In the weighted cost function, we use both pair-wise weight and point-wise weight to generate the overall weight. For the validity and necessity of each component, we test a series of ablation studies on the KITTI dataset. For a fair comparison, we add one component to GICP, which is used as the baseline, per test for validation. The results are shown in Table V.

Point matching consists of the intensity and geometry features. To test their effectiveness, we conduct three experiments: Euclidean coordinates plus geometry features (*GICP + G*), Euclidean plus intensity (*GICP + I*), and Euclidean plus both geometry and intensity (*GICP + IG*), as shown in columns 3 to 5 of Table V. Compared with the popular Euclidean coordinates, it can be seen that the additional employment of either geometry or intensity features can improve the odometry accuracy, and a joint of both features results in the lowest ATE.

The weight of cost function is obtained by the pair-wise weight and the point-wise weight, or the point matching similarity and the average planarity of one match. Three different weights are tested for their effects: pair-wise similarity only (*GICP + C*), planarity only (*GICP + P*), and both (*GICP + CP*), as shown in columns 6 to 8 of Table V. It can be seen that both similarity and planarity weights may improve the accuracy on most sequences except for 09 and 10.

The last column of Table V shows the result of IGICP, which combines the proposed point matching and weighted cost function. The employment of both parts achieves the lowest ATE on all tests. It is interesting that sequence 09 and 10 see the limited improvements in individual proposed parts, while the overall improvement is significantly distinct.

D. Robustness Against Initial Pose and Dynamic Occlusion

As shown in many previous works, a poor initial pose may significantly downgrade the performance of ICP and its variants, including the proposed IGICP. To verify the robustness under varying initial poses, we compare IGICP and GICP on KITTI 07.

In KITTI 07, we compute the pose between a scan and its previous one with ICICP and GICP, respectively. With a fixed initial pose as the input to both IGICP and GICP, we rotate one point cloud around the z-axis by a distortion angle $\phi = [0^\circ, 1^\circ, \dots, 90^\circ]$. We consider it a successful registration

TABLE V
ATE (%) EVALUATION RESULTS OF THE ABLATION EXPERIMENT ON KITTI

Sequence	GICP [22]	GICP + Point Matching			GICP + Weighted Cost Function			GICP + IGCP (IGICP)
		GICP + G	GICP + I	GICP + IG	GICP + C	GICP + P	GICP + CP	
00	0.19	0.18	0.16	0.15	0.18	0.13	0.15	0.13
01	1.22	0.94	0.98	0.96	0.95	1.20	0.96	0.94
02	0.37	0.31	0.21	0.21	0.25	0.25	0.20	0.18
03	0.26	0.26	0.22	0.24	0.24	0.22	0.21	0.21
04	0.30	0.23	0.14	0.14	0.11	0.17	0.13	0.12
05	0.17	0.16	0.15	0.14	0.17	0.16	0.16	0.13
06	0.13	0.09	0.11	0.10	0.12	0.10	0.10	0.08
07	0.14	0.07	0.07	0.06	0.08	0.08	0.07	0.07
08	0.26	0.21	0.22	0.22	0.24	0.19	0.20	0.17
09	0.17	0.19	0.20	0.18	0.18	0.18	0.18	0.14
10	0.19	0.17	0.17	0.16	0.21	0.24	0.20	0.14
Average	0.31	0.26	0.24	0.23	0.25	0.27	0.23	0.21

if the difference between the output rotation angle and the ground truth is less than 1° and the displacement less than 1 cm. Let ϕ_{GICP}^i be the highest distortion angle corresponding to a successful registration of GICP at the i -th scan, and ϕ_{IGICP}^i for IGICP. ϕ_{GICP}^i and ϕ_{IGICP}^i represent the robustness of one competing method against a noisy initial pose. We subtract ϕ_{GICP}^i from ϕ_{IGICP}^i and count the number of scan pairs with different subtraction results, as shown in Fig. 10(a). A positive value in the x-axis of Fig. 10(a) means the proposed IGICP performs better than GICP and a negative value refers to the superior of GICP.

The differences on time-consuming are shown respectively in Fig. 10(b) and (c), in terms of number of iterations until convergence and elapsed time. Similar to the distortion angle experiment, for each scan in KITTI 07, we subtract the time cost (number of iterations) of GICP from that of IGICP, and show the histogram in Fig. 10 for comparison. In Fig. 10(b) and (c), a positive value in the x-axis indicates a high time cost of the proposed method. The average maximum distortion angle with a successful registration of GICP is 30.19° , the average number of used iterations 39.5 and the average time cost 79.5 ms. The average angle, number of iterations and time cost of IGICP are 34.53° , 25.82 and 57.44 ms, respectively.

Occlusion always results in a performance drop of LiDAR odometry. In this experiment, we verify the robustness of the proposed method against dynamic occlusion. We run an AGV (Clearpath Dingo) equipped with a Velodyne VLP-16 LiDAR sensor having the same trajectory repeatedly and adopt a Motion Capture system, which provides accurate global localization with less than 1 mm in error, to record the real trajectories the AGV moved. We then invite five volunteers to walk near the AGV, and compare the LiDAR odometry obtained by GICP and IGICP with the ground truth. An overview of this test is shown in Fig. 11(a).

The ATE of all the six tests, with 0 to 5 pedestrians, are shown in Fig. 11(b). The average ATE among the six tests of GICP is 1.24% and that of the proposed IGICP is 0.66%.

E. Efficiency Analysis

Despite the advantages of our method on accuracy, the proposed point matching and weighted cost function minimization may bring additional computational burden for LiDAR

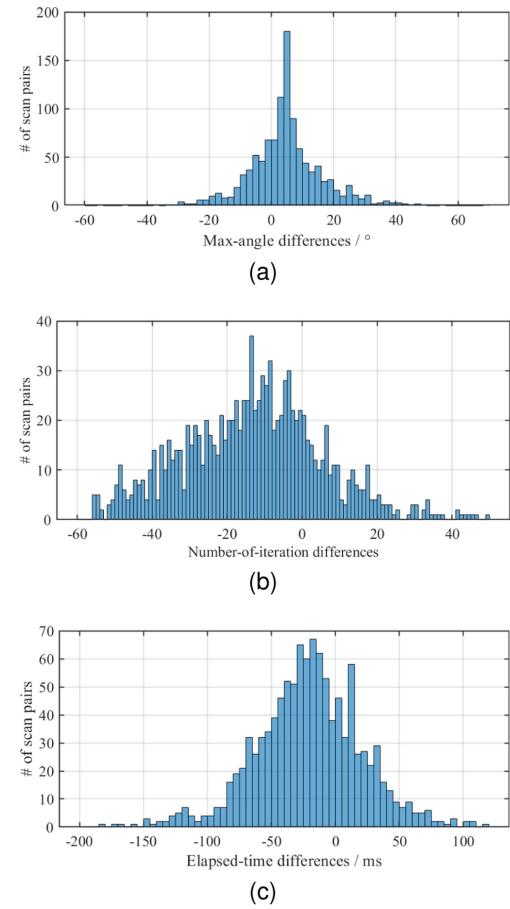
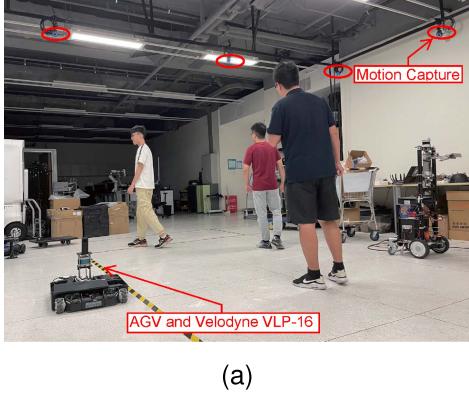
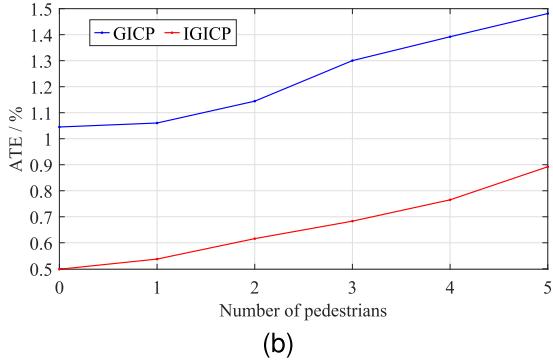


Fig. 10. Robustness against noisy initial poses, KITTI 07. (a) Histogram of maximum angles of successful registration, (b) Histogram of number of iterations, and (c) Histogram of elapsed time. We rotate one point cloud around the z-axis and ask both GICP and IGICP to align the rotated point cloud with its previous one. We record the maximum distortion angle corresponding to a small enough estimated pose. We then subtract the maximum distortion angle of GICP from that of IGICP and show the histogram in (a). The positive value in the x-axis indicates the superiority of IGICP. We process the number iterations and elapsed time in the same way in (b) and (c), respectively. The positive value in (b) and (c) means a high time cost of IGICP.

odometry, where most odometry-related tasks are time-sensitive and demand real-time processing. Compared with the NN search in the Euclidean space, the normal vector calculation introduces more operations regarding local neighbor search and is known



(a)



(b)

Fig. 11. Overview of the dynamic occlusion test and the ATE. (a) An overview of this test and (b) The ATE of all six tests.

TABLE VI
AVERAGE NUMBER OF ITERATIONS UNTIL CONVERGENCE, KITTI

Sequence	GICP [22]	GICP + IG	GICP + CP	GICP + IGCP (IGICP)
00	4.67	3.43	4.07	3.27
01	13.88	7.77	8.61	7.61
02	6.91	5.07	5.95	4.82
03	5.54	4.19	5.09	4.01
04	10.42	6.78	7.80	6.37
05	4.54	3.44	4.01	3.21
06	5.13	3.62	4.01	3.30
07	4.23	3.30	3.78	3.10
08	5.46	4.07	4.81	3.83
09	6.42	4.61	5.41	4.28
10	5.04	3.81	4.51	3.60
Average	6.57	4.55	5.28	4.31

time-consuming. We ease this burden by first downsampling the input point cloud and then searching in a kd-tree for normal estimation, which can significantly reduce the time cost. As to the weight, the λ s in calculating the average planarity are already obtained in normal estimation, therefore the average planarity calculation introduces limited additional time cost in general. In addition, since our method is superior to others in the matching accuracy rate, IGICP may require fewer iterations and consequently achieve faster convergence speed than its competitors. To verify the speed of our method, we compare IGICP with the popular GICP and show the results in Table VII. Although our method introduces additional feature extraction

TABLE VII
FPS COMPARISON BETWEEN GICP AND IGICP, KITTI

Sequences	GICP [22]	Ours
00	32.61	32.77
01	15.19	17.10
02	23.76	30.08
03	22.45	25.37
04	15.47	18.00
05	28.81	31.80
06	17.22	27.85
07	25.90	35.09
08	20.86	26.19
09	16.71	24.37
10	24.20	28.40
Average	22.11	27.00

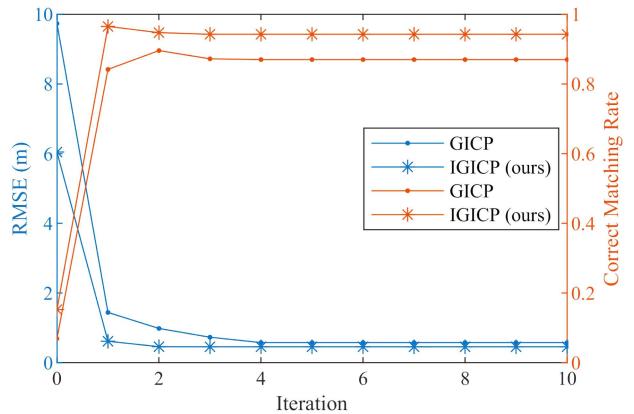


Fig. 12. Comparison of convergence of scan 100 and 104 of KITTI 00. GICP completes the optimization with 4 iterations, while IGICP takes only 2 iterations to achieve convergence. IGICP also obtains low RMSE and a high matching rate at the first iteration, indicating its good ability in dealing with improper initial poses.

and weight calculation operations, the average FPS of IGICP is 27 FPS, which is 22.12% higher than that of GICP.

To further analyze the efficiency of IGICP, we estimate the poses between scan 100 and 104 of KITTI 00 by running both IGICP and GICP, in which both RMSE and the correct matching rate are shown in Fig. 12. The aligned point clouds with poses estimated from the four leading iteration are displayed in Fig. 13.

Since there is no point-to-point ground truth matches in KITTI, we first translate scan 104 to the coordinates of scan 100 by the ground truth pose provided by KITTI. Then, for each point in scan 100, we find the nearest neighbor in scan 104, and take this point pair as the ground truth match of GICP. A similar approach is done for IGICP where the nearest neighbor is defined by the highest similarity pair. In other words, we find the optimal correspondences which is consistent with the ground truth pose in the best way. Given the generated point-to-point ground truth matches, we obtain the correct matching rate per iteration in Fig. 12.

As in Fig. 12, IGICP uses two iterations to achieve convergence with a low RMSE while GICP needs four iterations to obtain a similar RMSE. The correct matching rate of IGICP is also higher than that of GICP in all iterations, indicating a better point matching of IGICP.

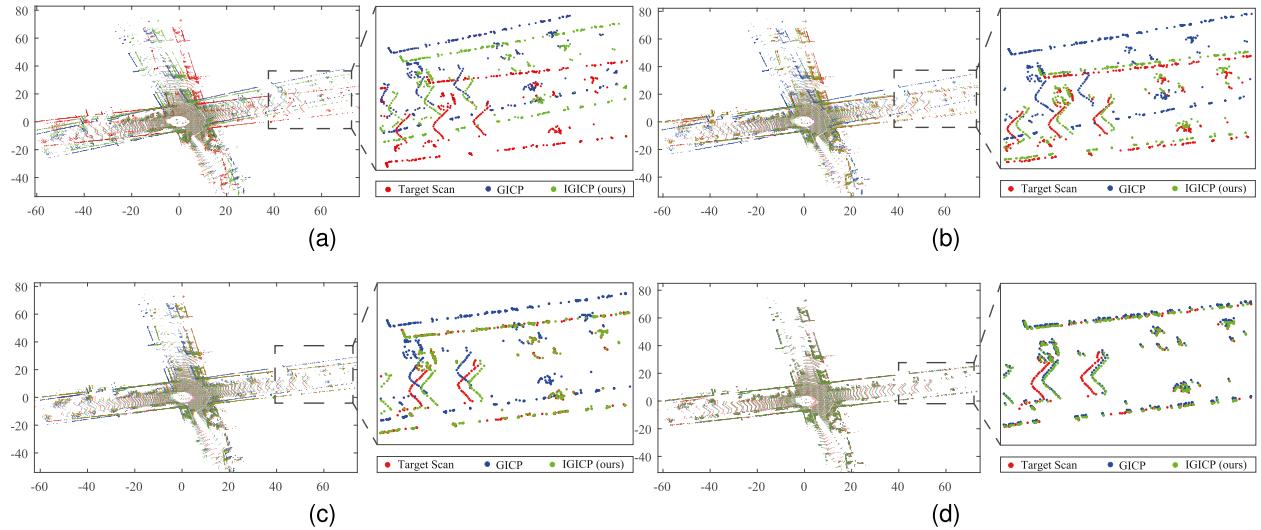


Fig. 13. Aligned point clouds with poses of the first four iterations, scan 100 and 104 of KITTI 00. (a) Aligned point clouds of the first iteration, (b) the second iteration, (c) the third iteration and (d) the fourth iteration.

We also perform an ablation experiment to test the effect of point matching and weighted cost function on convergence. We count the number of iterations until convergence by using GICP as the baseline. Three methods have been tested: GICP + IG, GICP + CP and GICP + IGCP, with the abbreviations described in Section IV-C. The results are shown in Table VI. It can be seen that the employment of point matching reduces the average iteration by 30.63%, and the weighted cost function by 19.61%. The number of iterations of IGICP is 34.37% lower than that of GICP.

V. CONCLUSION

In this article, we propose an intensity and geometry-enhanced LiDAR odometry method called IGICP. We evaluate the similarity of point pairs with the normal vector, the corresponding eigenvalue and the KL divergence of local intensity values, in which we select the matched point according to the highest similarity. In addition, we evaluate the quality of each point match with the obtained similarity and point planarity which takes both pair-wise uncertainty and point-wise uncertainty into account, and assign accordingly the weight for one match in optimization. Experiments on different real-world datasets show that higher accuracy have been improved with our method which achieves the lowest ATE on most tests. Our method also runs at 27 FPS on a common PC due to the improved point matching and consequently fast convergence.

REFERENCES

- [1] M. Maaref, J. Khalife, and Z. M. Kassas, “Lane-level localization and mapping in GNSS-challenged environments by fusing lidar data and cellular pseudoranges,” *IEEE Trans. Intell. Veh.*, vol. 4, no. 1, pp. 73–89, Mar. 2019.
- [2] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong, “3D LiDAR-based global localization using siamese neural network,” *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1380–1392, Apr. 2020.
- [3] C. Wang, Z. Cao, J. Li, J. Yu, and S. Wang, “Hierarchical distribution-based tightly-coupled LiDAR inertial odometry,” *IEEE Trans. Intell. Veh.*, early access, May 5, 2023, doi: [10.1109/TIV.2023.3273288](https://doi.org/10.1109/TIV.2023.3273288).
- [4] C. Shu and Y. Luo, “Multi-modal feature constraint based tightly coupled monocular visual-lidar odometry and mapping,” *IEEE Trans. Intell. Veh.*, vol. 8, no. 5, pp. 3384–3393, May 2023.
- [5] L. He, X. Wang, and H. Zhang, “M2DP: A novel 3D point cloud descriptor and its application in loop closure detection,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 231–237.
- [6] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong, “DISCO: Differentiable scan context with orientation,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2791–2798, Apr. 2021.
- [7] J. Serafin and G. Grisetti, “NICP: Dense normal based point cloud registration,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 742–749.
- [8] Z. Yao, Q. Zhao, X. Li, and Q. Bi, “Point cloud registration algorithm based on curvature feature similarity,” *Measurement*, vol. 177, 2021, Art. no. 109274.
- [9] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time,” in *Proc. Robot.: Sci. Syst.*, 2014, pp. 1–9.
- [10] Y. Pan, P. Xiao, Y. He, Z. Shao, and Z. Li, “MULLS: Versatile LiDAR SLAM via multi-metric linear least square,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11633–11640.
- [11] L. Liu, J. Xiao, Y. Wang, Z. Lu, and Y. Wang, “A novel rock-mass point cloud registration method based on feature line extraction and feature point matching,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–17, 2022.
- [12] P. J. Besl and N. D. McKay, “Method for registration of 3-D shapes,” *Proc. SPIE*, vol. 1611, pp. 586–606, 1992.
- [13] A. Censi, “An ICP variant using a point-to-line metric,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 19–25.
- [14] K.-L. Low, “Linear least-squares optimization for point-to-plane ICP surface registration,” Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, USA, Tech. Rep. TR04-004, 2004.
- [15] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4758–4765.
- [16] J. Lin and F. Zhang, “Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for lidars of small FOV,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 3126–3131.
- [17] M. Oelsch, M. Karimi, and E. Steinbach, “R-LOAM: Improving LiDAR odometry and mapping with point-to-mesh features of a known 3D reference object,” *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2068–2075, Apr. 2021.
- [18] P. Biber and W. Straßer, “The normal distributions transform: A new approach to laser scan matching,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2003, pp. 2743–2748.
- [19] E. Takeuchi and T. Tsubouchi, “A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2006, pp. 3068–3073.
- [20] M. Magnusson, A. Nuchter, C. Lorken, A. J. Lilienthal, and J. Hertzberg, “Evaluation of 3D registration reliability and speed-a comparison of ICP and NDT,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2009, pp. 3907–3912.

- [21] H. Courtois, N. Aouf, K. Ahiska, and M. Cecotti, "NDT RC: Normal distribution transform occupancy 3D mapping with recentering," *IEEE Trans. Intell. Veh.*, early access, Feb. 28, 2023, doi: [10.1109/TIV2023.3250326](https://doi.org/10.1109/TIV2023.3250326).
- [22] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst.*, 2009, Art. no. 435.
- [23] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11054–11059.
- [24] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "Litamin: Lidar-based tracking and mapping by stabilized ICP for geometry approximation with normal distributions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 5143–5150.
- [25] M. Yokozuka, K. Koide, S. Oishi, and A. Banno, "LITAMIN2: Ultra light LiDAR-based SLAM using geometric approximation applied with kl-divergence," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 11619–11625.
- [26] Goldberger, Gordon, and Greenspa, "An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures," in *Proc. IEEE 9th Int. Conf. Comput. Vis.*, 2003, pp. 487–493.
- [27] Q. Li et al., "LO-Net: Deep real-time LiDAR odometry," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8473–8482.
- [28] C. Zheng, Y. Lyu, M. Li, and Z. Zhang, "LodoNet: A deep neural network with 2D keypoint matching for 3D LiDAR odometry estimation," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 2391–2399.
- [29] M. Horn, N. Engel, V. Belagiannis, M. Buchholz, and K. Dietmayer, "DeepCLR: Correspondence-less architecture for deep end-to-end point cloud registration," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst.*, 2020, pp. 1–7.
- [30] G. Wang, X. Wu, Z. Liu, and H. Wang, "PWCLONet: Deep LiDAR odometry in 3D point clouds using hierarchical embedding mask optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15910–15919.
- [31] B. Zhou, Y. Tu, Z. Jin, C. Xu, and H. Kong, "HPPLO-Net: Unsupervised LiDAR odometry using a hierarchical point-to-plane solver," *IEEE Trans. Intell. Veh.*, early access, Jun. 23, 2023, doi: [10.1109/TIV2023.3288943](https://doi.org/10.1109/TIV2023.3288943).
- [32] J. Liu, G. Wang, C. Jiang, Z. Liu, and H. Wang, "TransLO: A window-based masked point transformer framework for large-scale LiDAR odometry," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 1683–1691.
- [33] G. Wang, X. Wu, S. Jiang, Z. Liu, and H. Wang, "Efficient 3D deep LiDAR odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 5, pp. 5749–5765, May 2023.
- [34] B. Liu, X. Gao, H. Liu, X. Wang, and B. Liang, "A fast weighted registration method of 3D point cloud based on curvature feature," in *Proc. 3rd Int. Conf. Multimedia Image Process.*, 2018, pp. 83–87.
- [35] L. Li, X. Cao, and J. Sun, "Three-dimensional point cloud registration based on normal vector angle," *J. Indian Soc. Remote Sens.*, vol. 47, pp. 585–593, 2019.
- [36] R. Wang and Z. Geng, "WA-ICP algorithm for tackling ambiguous correspondence," in *Proc. IEEE 3rd IAPR Asian Conf. Pattern Recognit.*, 2015, pp. 076–080.
- [37] G. Wu, J. Wang, H. Wang, L. Xie, and P. Li, "An improved scan matching method based on laser reflection intensity," in *Proc. IEEE 10th Int. Conf. Intell. Control Inf. Process.*, 2019, pp. 51–58.
- [38] B. Lohani and A. Sasidharan, "An evaluation of intensity augmented icp for terrestrial lidar data registration," *J. Geomatics*, vol. 11, no. 2, pp. 139–148, 2017.
- [39] H. Wang, C. Wang, and L. Xie, "Intensity-SLAM: Intensity assisted localization and mapping for large scale environment," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 1715–1721, Apr. 2021.
- [40] T. Guadagnino, X. Chen, M. Sodano, J. Behley, G. Grisetti, and C. Stachniss, "Fast sparse LiDAR odometry using self-supervised feature selection on intensity images," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7597–7604, Jul. 2022.
- [41] S. Li, B. Tian, X. Zhu, J. Gui, W. Yao, and G. Li, "InTEn-LOAM: Intensity and temporal enhanced LiDAR odometry and mapping," *Remote Sens.*, vol. 15, no. 1, 2023, Art. no. 242.
- [42] J. Servos and S. L. Waslander, "Multi-channel generalized-icp: A robust framework for multi-channel scan registration," *Robot. Auton. Syst.*, vol. 87, pp. 247–257, 2017.
- [43] A. Gressin, C. Mallet, J. Demantké, and N. David, "Towards 3D LiDAR point cloud registration improvement using optimal neighborhood knowledge," *ISPRS J. Photogrammetry Remote Sens.*, vol. 79, pp. 240–251, 2013.
- [44] E. Wakisaka et al., "TLS point cloud registration based on ICP algorithm using point quality," *Int. Arch. the Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 42, pp. 963–968, 2019.
- [45] S. Liu, D. Gao, P. Wang, X. Guo, J. Xu, and D.-X. Liu, "A depth-based weighted point cloud registration for indoor scene," *Sensors*, vol. 18, no. 11, 2018, Art. no. 3608.
- [46] K. Naus and L. Marchel, "Use of a weighted ICP algorithm to precisely determine usv movement parameters," *Appl. Sci.*, vol. 9, no. 17, 2019, Art. no. 3530.
- [47] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2095–2101.
- [48] H. Yoshitaka, K. Hirohiko, O. Akihisa, and Y. Shin'ichi, "Mobile robot localization and mapping by scan matching using laser reflection intensity of the sokuiki sensor," in *Proc. IEEE 32nd Annu. Conf. Ind. Electron.*, 2006, pp. 3018–3023.
- [49] B. Höfle and N. Pfeifer, "Correction of laser scanning intensity data: Data and model-driven approaches," *ISPRS J. Photogrammetry Remote Sens.*, vol. 62, no. 6, pp. 415–433, 2007.
- [50] J. Demantké, C. Mallet, N. David, and B. Vallet, "Dimensionality based scale selection in 3D LiDAR point clouds," *Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, vol. 38, pp. 97–102, 2012.
- [51] J. Wang, M. Xu, F. Foroughi, D. Dai, and Z. Chen, "FasterGICP: Acceptance-rejection sampling based 3D Lidar odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 1, pp. 255–262, Jun. 2022.
- [52] C. Yuan, W. Xu, X. Liu, X. Hong, and F. Zhang, "Efficient and probabilistic adaptive voxel mapping for accurate online LiDAR odometry," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 8518–8525, Jul. 2022.
- [53] L. Zhu, W. Chen, X. Lin, L. He, and Y. Guan, "Curvature-variation-inspired sampling for point cloud classification and segmentation," *IEEE Signal Process. Lett.*, vol. 29, pp. 1868–1872, 2022.
- [54] J. Behley and C. Stachniss, "Efficient surfel-based SLAM using 3D laser range data in urban environments," in *Proc. Robot.: Sci. Syst.*, 2018, pp. 59–69.
- [55] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa++ : Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 4530–4537.
- [56] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
- [57] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A multi-sensor and multi-scenario SLAM dataset for ground robots," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2266–2273, Apr. 2022.



Li He (Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees from the Department of Automation, Northwestern Polytechnical University, Xi'an, China. He was a Visiting Student with the Department of Computing Science, University of Alberta, Edmonton, AB, Canada, from 2010 to 2011, and then a Postdoctoral Fellow from 2014 to 2017. In 2017, he joined the School of Electromechanical Engineering, Guangdong University of Technology, China, as an Associate Professor. Since 2021, he has been a Research Associate Professor with the Department

of Electrical and Electronic Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include machine learning, image analysis, and SLAM. He is an Associate Editor for the IEEE ACCESS and an Associate Editor/Managing Guest Editor of *Computers and Electrical Engineering*.



Wen Li received the bachelor's degree in 2021 from the Guangdong University of Technology, Guangzhou, China, where he is currently working toward the M.S. degree. He is a Visiting Student with the Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include LiDAR SLAM and point cloud registration.



Yisheng Guan (Member, IEEE) received the master's degree in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 1990, and the Ph.D. degree in mechanical engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 1998. He conducted research as a Postdoctoral Fellow with the Department of Computing Science, University of Alberta, Edmonton, AB, Canada, from 1998 to 2000, and with the Intelligent Systems Institute, AIST, Tsukuba, Japan, with a fellowship from the Japan Society for the Promotion of Science, from 2003 to 2005. In 2007, he joined the School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China, and later became a Professor till 2012. He is currently a Professor with the School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include biomimetic robotics, modular robotics, humanoid robotics, and medical robotics.



Hong Zhang (Fellow, IEEE) received the B.Sc. degree in electrical and computer engineering from Northeastern University, Boston, MA, USA, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA. In 1988, he became a faculty Member with the Department of Computing Science, University of Alberta, Edmonton, AB, Canada, where he worked till 2020. He is currently a Chair Professor with the Department of Electrical and Electronic Engineering, Southern University of Science and Technology, Shenzhen, China. His research interests include robotics, computer vision, and image processing. Dr. Zhang is a Fellow of Canadian Academy of Engineering.