Intensity-SLAM: Intensity Assisted Localization and Mapping for Large Scale Environment

Han Wang ⬥, Chen Wang ⬥, and Lihua Xie ⬥

Abstract—Simultaneous Localization And Mapping (SLAM) is a task to estimate the robot location and to reconstruct the environment based on observation from sensors such as LIght Detection And Ranging (LiDAR) and camera. It is widely used in robotic applications such as autonomous driving and drone delivery. Traditional LiDAR-based SLAM algorithms mainly leverage the geometric features from the scene context, while the intensity information from LiDAR is ignored. Some recent deep-learning-based SLAM algorithms consider intensity features and train the pose estimation network in an end-to-end manner. However, they require significant data collection effort and their generalizability to environments other than the trained one remains unclear. In this letter we introduce intensity features to a SLAM system. And we propose a novel full SLAM framework that leverages both geometry and intensity features. The proposed SLAM involves both intensity-based front-end odometry estimation and intensity-based back-end optimization. Thorough experiments are performed including both outdoor autonomous driving and indoor warehouse robot manipulation. The results show that the proposed method outperforms existing geometric-only LiDAR SLAM methods.

Index Terms—Industrial robots, localization, SLAM.

I. INTRODUCTION

LOCALIZATION is one of the fundamental and essential topics in robotics. With the development of the robotic industry, robot localization has become more challenging in the past decades: from known to unknown environments, from simple to complex environments, from static to dynamic environments [1], and from short-term to long-term localization [2]. Traditionally, fixed anchors (or routers) are set up in a predefined area and the robot pose is obtained by using the distances of the robot from multiple anchors, e.g., Ultra-Wide Band (UWB) [3] and Wi-Fi localization [4]. However, these methods rely on external setup and they are mainly used in a small scale environment. To tackle the limitation of traditional localization methods, Simultaneous Localization And Mapping (SLAM) is

Fig. 1. Example of localization and mapping from the proposed method on KITTI sequence 00. The localization result is shown on the right corner with the ground truth plotted in red color and the trajectory plotted in green color. Our method achieves much higher localization accuracy compared to geometric-only methods.

introduced to estimate the robot pose from on-board sensors [5]. It is independent of an external setup and hence becomes promising in robotic applications. Based on the perception system used, SLAM can be further divided into Visual SLAM (V-SLAM) and LiDAR SLAM. Compared to Visual SLAM, LiDAR SLAM is more accurate and robust to environmental changes such as weather and illumination [6].

Conventional LiDAR SLAM methods such as LOAM [7] and HDL-Graph-SLAM [10] mainly focus on the geometric-only information to minimize point cloud differences. In this way, the information from the intensity channel is often ignored. Note that the intensity information is related to the reflectivity of material and is different for different objects, which can be useful for localization and object recognition. Therefore, we argue that a robust LiDAR SLAM system should take such intensity information into account. Although recently there are some works aiming to use intensity information to improve the accuracy, they mainly adopt the Convolutional Neural Networks (CNN) to fit the raw laser scan data without any specific intensity analysis or formulation. Moreover, collection of large data is required in order to train neural networks, which can be troublesome in practical applications.

In the previous work [8], we have demonstrated that intensity information is very useful for robust loop closure detection. In

this work we further utilize the intensity information to improve the localization accuracy of a SLAM system. We propose a novel SLAM framework that uses both geometry and intensity information for odometry estimation. We first analyze the physical model of intensity measurement. Then we introduce an extra intensity cost to the existing geometric-only cost in the odometry estimation formula. Finally, we combine the intensity-based loop closure detection and back-end optimization to further improve the performance. The proposed framework is tested using an indoor warehouse environment robot as well as an outdoor autonomous driving car. The results indicate that our method provides reliable and accurate localization in multiple environments and outperforms geometric-only methods. The main contributions of this letter are as below:

- We propose a novel SLAM framework that uses both intensity and geometry information for localization estimation, which involves front-end odometry estimation and back-end factor graph optimization. The proposed method is open sourced.[1]
- We propose to build an intensity map to reveal the intensity distribution and introduce an intensity cost to the existing geometry-only LiDAR SLAM to estimate the robot's location.
- A thorough evaluation on the proposed method is presented. More specifically, our method is tested in both warehouse and autonomous driving scenarios. The results show that our method outperforms the existing geometric-only methods for multiple scenarios.

This letter is organized as follows: Section II reviews the related work on existing LiDAR SLAM approaches. Section III describes the details of the proposed approach, including intensity calibration, feature selection, odometry estimation, and Intensity Scan Context (ISC) based graph optimization. Section IV shows the experimental results and comparison with some existing works, followed by the conclusion in Section V.

II. RELATED WORK

Majority of the existing works on LiDAR SLAM focus on geometry information of the environment. One of the most popular methods for point cloud matching is the Iterative Closest Point (ICP) method which matches the current point to the nearest point in the target frame [9]. ICP iteratively searches for the optimal point correspondence by minimizing the Euclidean distance between point pairs until the transformation matrix converges. The algorithm is popularly used in SLAM systems such as HDL-Graph SLAM [10] and LiDAR-Only Odometry and Localization (LOL) [11]. However, all points are used for calculation, which is computationally expensive for a LiDAR with tens of thousands of points per scan. ICP is also sensitive to noise. In real applications such as autonomous driving, the measurement noise (e.g., measurement from trees alongside a road) can be significant and subsequently cause localization drifts. A more robust and computationally efficient way is to leverage geometric features. In LiDAR Odometry And Mapping (LOAM) [7], a simple feature extraction strategy is introduced by analyzing the local smoothness. The feature points are segmented into edge features and planar features based on local

smoothness, and the robot pose is subsequently calculated by doing edge-to-edge and planar-to-planar matching respectively. Similar idea is also used in Lightweight and Ground-Optimized LiDAR Odometry And Mapping (LeGO-LOAM) [12], which targets at providing a computationally efficient LiDAR SLAM for Unmanned Ground Vehicles (UGVs). A laser scan is first segmented into ground points and non-ground points. The planar features are extracted from ground points while the edge features are extracted from non-ground points. The planar features are used to estimate the roll and pitch angle as well as z-translation. The results are subsequently used in the matching of non-ground points to calculate x and y translation as well as yaw angle. LeGO-LOAM has achieved higher localization accuracy than LOAM for grounded robots. However, the intensity channel is ignored and only the geometry channel is used in those methods.

In recent years, some works attempt to introduce the intensity channel into SLAM via deep learning. In DeepICP, an end-to-end learning-based 3D point cloud registration is proposed to find the robot pose [13]. Both intensity channel and geometry channel are introduced into a Deep Feature Extraction (DFE) layer to find the keypoints. Rather than searching for the closest point, a Corresponding Points Generation (CPG) layer is used to generate keypoint correspondences based on learned matching probabilities among a group of candidates, which makes back-propagation possible. Deep-ICP is validated using both KITTI [14] and Apollo-SouthBay dataset [15] and outperforms existing methods such as Generalized-ICP [16] and Normal Distribution Transform (NDT) [17]. Chen et al. also introduced an end-to-end CNN framework to identify the overlap of two laser scans for back-end SLAM [18]. The geometry channel as well as intensity channel are trained through a 11-layer network to generate a 3-dimensional feature. The overlap is estimated by applying another smaller layer to the extracted features, and the yaw angle change is estimated by the cross correlation of the trained features. The experiment on the KITTI dataset shows that the yaw angle estimated from both intensity and geometry channels is better than the geometric-only deep-learning-based framework. The deep-learning-based method adopts end-to-end training without further analysis of intensity information. However, it is often difficult and time consuming to collect, label and train data in practice. Moreover, the performance under environment changes may not be consistent, i.e., the trained network may perform well in the environment similar to the training data, but fails when transferring to another environment.

Some other works attempt to use the intensity channel and integrate intensity into pose estimation without end-to-end training. Tian et al. propose an intensity-assisted ICP for fast registration of 2D-LiDAR [19]. Compared to the traditional ICP algorithm, a target function is introduced to determine the initial rigid-body transformation estimation based on the spatial distance and intensity residual. The number of iterations is reduced by 10 times so that the proposed algorithm is able to operate in real time with a single core Central Processing Unit (CPU). In [20], an intensity model is incorporated into the Sparse Bundle Adjustment (SBA) [21] estimation problem. The authors analyze the physical model of intensity measurement and propose a new measurement model that augments 3D localization, intensity and surface normal. Simulation results show that the addition of intensity measurements achieves similar
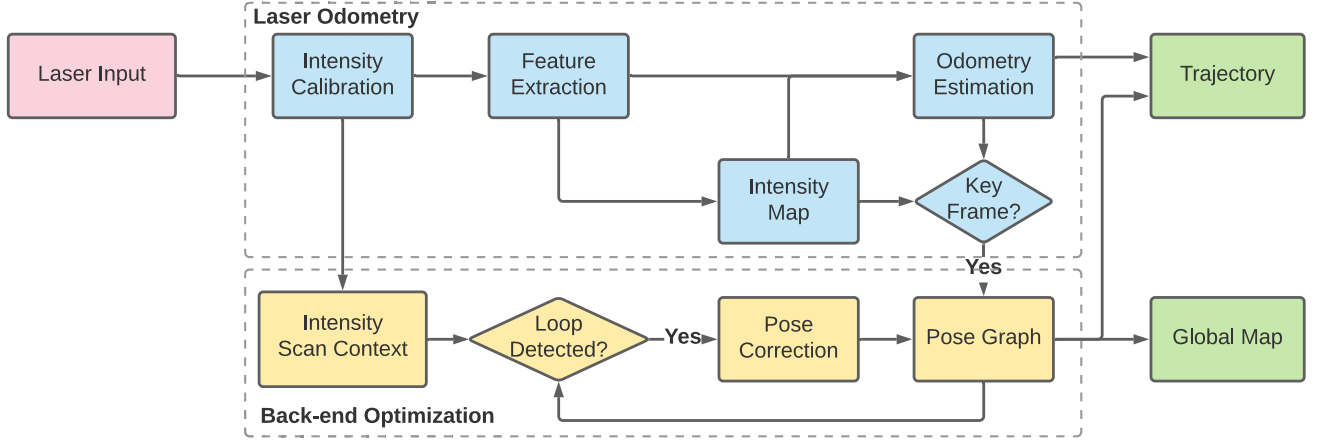
Fig. 2. System overview of proposed Intensity-SLAM. In the front-end, we analyze both geometry and intensity features to estimate odometry. In the back-end, we use intensity scan context for loop closure detection and global pose graph optimization.

SLAM accuracy with fewer landmarks. This work is further extended in [22] with the experiment performed using a Time of Flight (ToF) camera. The key-points are selected and tracked via Scale Invariant Feature Transform (SIFT) [23] or Binary Robust Invariant Scalable Keypoints (BRISK) [24] feature extraction from an intensity imaginary. The experiment mainly focuses on Visual Odometry (VO) via computer vision techniques but the performance of LiDAR SLAM in large scale environments is not demonstrated. Khan *et al.* propose a data driven approach to calibrate the intensity information [25]. Then the scan matching is solved by minimizing the intensity residual of the point pairs rather than geometry residual. It is subsequently integrated into Hector SLAM [26], which significantly reduces the drifting error compared to the original Hector SLAM. However, it is limited to 2D localization and mapping in a small scale environment.

III. METHODOLOGY

The proposed framework of Intensity-SLAM consists of both front-end laser odometry and back-end loop closure detection. The system overview is shown in Fig. 2. The front-end includes intensity calibration, feature extraction and odometry estimation, while the back-end includes loop closure detection and pose graph optimization.

A. Intensity Calibration

LiDAR emits laser beams and measures the arrival time as well as the energy of reflected signals. The object's position in sensor coordinates is determined by emitting angle and distance (i.e., arrival time). The intensity value is determined by the ratio between the energy received from the reflected signals and the laser power emitted. The physical principles of the received power P_r can be determined as [21]:

$$P_r = \frac{P_e D_r^2 \rho}{4R^2} \eta_{sys} \eta_{atm} \cos \alpha, \tag{1}$$

where P_e is the power of emitted laser beam, D_r is the receiver aperture diameter, η_{sys} is the system transmission factor, η_{atm} is the atmospheric transmission factor, α is the incident angle between the object surface and laser beam, and ρ is the material reflectivity of the object. Specifically, the distance measurement

R and the incident angle α are extrinsic parameters, while η_{sys} and η_{atm} are constant parameters. Therefore, the intensity measurement I is determined by [21]:

$$I = \frac{P_r}{P_e} = \frac{D_r^2 \rho}{4R^2} \eta_{sys} \eta_{atm} \cos \alpha$$
$$= \eta_{all} \frac{\rho \cos \alpha}{R^2}, \tag{2}$$

where η_{all} is a constant. Therefore, the surface reflectivity ρ is only related to incident angle α and measured distance R by

$$\rho \propto \frac{IR^2}{\cos \alpha}. \tag{3}$$

For a LiDAR scan, the distance can be easily measured. Hence the incident angle can be estimated by analyzing the local normal. For each point $\mathbf{p} \in \mathcal{R}^3$ we can search for the nearest two points $\mathbf{p}_1$ and $\mathbf{p}_2$, so that the local surface normal $\mathbf{n}$ can be expressed as:

$$\mathbf{n} = \frac{(\mathbf{p} - \mathbf{p}_1) \times (\mathbf{p} - \mathbf{p}_2)}{|\mathbf{p} - \mathbf{p}_1| \cdot |\mathbf{p} - \mathbf{p}_2|}, \tag{4}$$

where $\times$ is the cross product. The incident angle is

$$\cos \alpha = \frac{\mathbf{p}^T \cdot \mathbf{n}}{|\mathbf{p}|}. \tag{5}$$

Therefore, we can derive the calibrated intensity scan $\tilde{\mathcal{I}}$ from a raw LiDAR scan with geometric reading $\mathcal{P}$ and intensity reading $\mathcal{I}$. Specifically, the intensity is partially calibrated with distance measurement by default in some sensors, hence we only apply the calibration to the incident angle. Moreover, low intensity value often leads to a lower Signal-to-Noise Ratio (SNR), which also deteriorates the ranging accuracy for intensity measurement with low value [27]. Points with low intensity are ignored in practice.

B. Salient Point Selection and Feature Extraction

A LiDAR scan often consists of tens of thousands of points. It is less computationally efficient to use raw point cloud matching methods such as ICP. Moreover, the raw data contains noisy measurements that can reduce the matching accuracy,

e.g., measurements from trees alongside a road in autonomous driving. Hence it is more robust and computationally efficient to match the points clouds with features [7], [13]. Instead of using geometric-only features, in this letter we utilize the feature based on both geometry and intensity information. The calibrated intensity information contains the reflectivity profile of the environment that reveals the distribution of different objects. Therefore, the intensity information also helps to recognize the identical features across multiple frames. For each point $\mathbf{p}_i \in \mathcal{P}$ and its intensity value $\tilde{\eta}_i \in \tilde{\mathcal{I}}$, we search for the nearby points $\mathcal{N}_i \in \mathcal{P}$ and calculate the local distance distribution $\sigma_i^{\mathcal{G}}$ and intensity distribution $\sigma_i^{\mathcal{I}}$ by:

$$\sigma_i^{\mathcal{G}} = \frac{1}{|\mathcal{N}_i|} \cdot \sum_{p_j \in \mathcal{N}_i} (|\mathbf{p}_i - \mathbf{p}_j|)$$

$$\sigma_i^{\mathcal{I}} = \frac{1}{|\mathcal{N}_i|} \cdot \sum_{p_j \in \mathcal{N}_i} (|\eta_i - \eta_j|), \qquad (6)$$

where $|\mathcal{N}|$ is the number of points. The features are selected based on the weighted summation:

$$\sigma_i = w^{\mathcal{G}} \cdot \sigma_i^{\mathcal{G}} + w^{\mathcal{I}} \cdot \sigma_i^{\mathcal{I}}, \qquad (7)$$

where $w^{\mathcal{G}}$ and $w^{\mathcal{I}}$ are the weights of geometry and intensity distribution, respectively. Note that for flat surfaces such as wall, the smoothness values are small; while for corner or edge points, the smoothness values are large. Thus, for each scan, edge features $\mathcal{P}_{\mathcal{E}} \in \mathcal{P}$ are selected from the points with large σ, and planar features $\mathcal{P}_{\mathcal{S}} \in \mathcal{P}$ are selected from the points with low σ.

C. Intensity Map Building

The intensity map $\mathcal{M}$ contains the reflectivity distribution of the surrounding environment. For most geometric-only SLAM, the map is maintained and updated via occupancy grid [28] or Octomap [29]. The 3-D space is segmented into grid cells and each cell is represented by a probability function. Similar ideas can be used to construct and update an intensity map. Instead of a probability function, we use the intensity observation $\mathcal{I}(\eta_i | z_{1:t})$ to represent each grid cell m_i [26]. More specifically, for each observation of grid cell at time t, the surface reflectivity can be updated by:

$$\mathcal{M}(m_i | z_{1:t}) = \mathcal{M}(m_i | z_{1:t-1}) + \frac{\eta_{m_i} - \mathcal{M}(m_i | z_{1:t-1})}{n_{m_i}}, \quad (8)$$

where $\mathcal{M}(m_i | z_{1:t})$ is the current intensity observation and n_{m_i} is the total number of observation times on cell m_i. To note that, if the grid contains no object, the intensity is marked as 0 since there is no signal reflected.

D. Scan to Map Matching

The laser odometry is the task to estimate the transformation matrix $\mathbf{T} \in SE(3)$ between the current frame to the global map. The optimal pose estimation is calculated by minimizing the geometry error and intensity error.

1) Geometry Residual: Similar to LOAM [7], the geometric error is calculated by matching the current edge and planar features with the global map. It can be achieved by minimizing
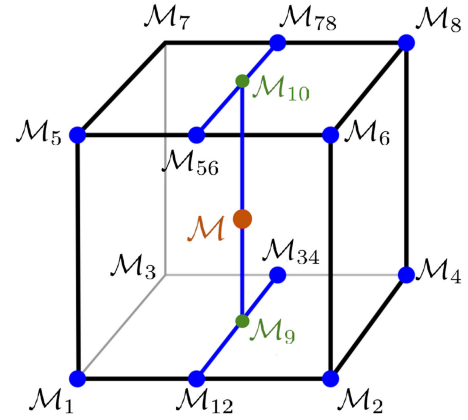

Fig. 3.   Trilinear interpolation.

the point-to-edge and point-to-plane residuals. Given an edge feature $\mathbf{p}_i \in \mathcal{P}_{\mathcal{E}}$ and the transformed point $\hat{\mathbf{p}}_i = \mathbf{T}\mathbf{p}_i$, we can search for two nearest points $\mathbf{p}_1^{\mathcal{E}}$ and $\mathbf{p}_2^{\mathcal{E}}$ from the global map. The point-to-edge residual is defined by:

$$f_{\mathcal{E}}(\hat{\mathbf{p}}_i) = \frac{(\hat{\mathbf{p}}_i - \mathbf{p}_1^{\mathcal{E}}) \times (\hat{\mathbf{p}}_i - \mathbf{p}_2^{\mathcal{E}})}{|\mathbf{p}_1^{\mathcal{E}} - \mathbf{p}_2^{\mathcal{E}}|}. \qquad (9)$$

Similarly, given a planar feature point $\mathbf{p}_i \in \mathcal{P}_{\mathcal{S}}$ and the transformed point $\hat{\mathbf{p}}_i$, we can search for three nearest points $\mathbf{p}_1^{\mathcal{S}}$, $\mathbf{p}_2^{\mathcal{S}}$, and $\mathbf{p}_3^{\mathcal{S}}$ from the global map. The point-to-plane residual is defined by:

$$f_{\mathcal{S}}(\hat{\mathbf{p}}_k) = (\hat{\mathbf{p}}_k - \mathbf{p}_1^{\mathcal{S}})^T \cdot \left(\frac{(\mathbf{p}_1^{\mathcal{S}} - \mathbf{p}_2^{\mathcal{S}}) \times (\mathbf{p}_1^{\mathcal{S}} - \mathbf{p}_3^{\mathcal{S}})}{|(\mathbf{p}_1^{\mathcal{S}} - \mathbf{p}_2^{\mathcal{S}}) \times (\mathbf{p}_1^{\mathcal{S}} - \mathbf{p}_3^{\mathcal{S}})|} \right). \quad (10)$$

2) Intensity Residual: The intensity residual is calculated by matching the features with the intensity map. It can be achieved by minimizing the intensity residual between the current point $\mathbf{p}_i$ (including both edge features and planar features) and the transformed point $\hat{\mathbf{p}}_i$ in the intensity map:

$$f_{\mathcal{I}}(\hat{\mathbf{p}}_i) = \eta_i - \mathcal{M}(\hat{\mathbf{p}}_i), \qquad (11)$$

where $\mathcal{M}(\hat{\mathbf{p}}_i)$ is the point intensity value in the intensity map $\mathcal{M}$. To search for the intensity information in the intensity map, we introduce trilinear interpolation. Although it is more straightforward to represent the intensity by the nearest grid cell, the intensity information is less accurate especially for large scale mapping where grid resolution is low. For each transformed point $\hat{\mathbf{p}}_i = [x_i, y_i, z_i]^T$, we can find the surrounding eight grid cells. As shown in Fig. 3, the intensity measurements of these grid cells are noted as $\mathcal{M}_1(\hat{\mathbf{p}}_i), \mathcal{M}_2(\hat{\mathbf{p}}_i), \ldots, \mathcal{M}_8(\hat{\mathbf{p}}_i)$. Let the center position of cell 1 (closest to the origin) be $\mathbf{p}_1 = [x_1, y_1, z_1]^T$ and the center position of cell 8 (farthest to the origin) be $\mathbf{p}_2 = [x_2, y_2, z_2]^T$, where $x_2 - x_1$, $y_2 - y_1$, and $z_2 - z_1$ are the width, height and depth of each grid cell, respectively. The intensity estimation of the target point is calculated as:

$$\mathcal{M}_{12}(\hat{\mathbf{p}}_i) = \frac{x_i - x_1}{x_2 - x_1} \cdot \mathcal{M}_1(\hat{\mathbf{p}}_i) + \frac{x_i - x_1}{x_2 - x_1} \cdot \mathcal{M}_2(\hat{\mathbf{p}}_i). \quad (12)$$

Similarly, we can derive $\mathcal{M}_{34}$, $\mathcal{M}_{56}$ and $\mathcal{M}_{78}$. The intensity value is estimated as

$$\mathcal{M}_9(\hat{\mathbf{p}}_i) = \frac{y_i - y_1}{y_2 - y_1} \cdot \mathcal{M}_{12}(\hat{\mathbf{p}}_i) + \frac{y_i - y_1}{y_2 - y_1} \cdot \mathcal{M}_{34}(\hat{\mathbf{p}}_i)$$

$$\mathcal{M}_{10}(\hat{\mathbf{p}}_i) = \frac{y_i - y_1}{y_2 - y_1} \cdot \mathcal{M}_{56}(\hat{\mathbf{p}}_i) + \frac{y_i - y_1}{y_2 - y_1} \cdot \mathcal{M}_{78}(\hat{\mathbf{p}}_i)$$

$$\mathcal{M}(\hat{\mathbf{p}}_i) = \frac{z_i - z_1}{z_2 - z_1} \cdot \mathcal{M}_9(\hat{\mathbf{p}}_i) + \frac{z_i - z_1}{z_2 - z_1} \cdot \mathcal{M}_{10}(\hat{\mathbf{p}}_i).$$

3) Pose Estimation: The final pose can be estimated by minimizing both geometry residual and intensity residual:

$$\mathbf{T}^* = \arg\min_{\mathbf{T}^*} \sum_{\mathbf{p}_i \in \mathcal{P}_\mathcal{E}} |f_\mathcal{E}(\hat{\mathbf{p}}_i)| + \sum_{\mathbf{p}_i \in \mathcal{P}_\mathcal{S}} |f_\mathcal{S}(\hat{\mathbf{p}}_i)| + \sum_{\mathbf{p}_i \in \mathcal{P}} |f_\mathcal{I}(\hat{\mathbf{p}}_i)|.$$
(13)

It can be solved by the Levenberg-Marquardt algorithm [30]. Note that the initial pose alignment is estimated by assuming constant angular velocity and linear velocity which can increase convergence speed compared to assuming an identical transformation matrix.

E. Loop Closure Detection & Global Optimization

The goal of loop closure detection is to identify a revisited scene from historical data. For a front-end only SLAM system, it is inevitable to have drifts from the measurement noises. To reduce the localization drifts, the back-end SLAM detects the loops by recognizing identical places. We use key frame selection to reduce the retrieval time since the computational cost of loop closure detection is often high. The key frames are selected based on the following criteria: (1) The displacement of the robot is significant, i.e., greater than a pre-defined threshold; (2) The change of rotation angles is significant; (3) The time elapsed is more than a certain period. For large scale environments, the thresholds are set to be higher to reduce the computational cost. All key frames are stored into a pose graph maintained in the back-end.

For each keyframe, we use Intensity Scan Context (ISC) [9] to extract the current frame into a global signature. Compared to geometric-only descriptors such as GLAROT3D [31], NBLD [32] and Scan Context [33], the ISC is robust to rotation change in identifying a loop closure. The ISC is a 2D matrix calculated by equally dividing polar coordinates in azimuthal and radial directions into N_s sectors and N_r rings. Each subspace is represented by the max intensity of the points within the area. Given a key frame, we can extract the ISC descriptor Ω from both intensity and geometry information. To compare a query ISC descriptor Ω^q with a candidate ISC descriptor Ω^c, let $\mathbf{v}_i^q$ and $\mathbf{v}_i^c$ be the ith column of Ω^q and Ω^c. The similarity score $\varphi(\Omega^q, \Omega^c)$ is found by taking the mean cosine distance of each sector:

$$\varphi(\Omega^q, \Omega^c) = \frac{1}{N_s} \sum_{i=0}^{N_s-1} \left(\frac{\mathbf{v}_i^{qT} \cdot \mathbf{v}_i^c}{\|\mathbf{v}_i^q\| \cdot \|\mathbf{v}_i^c\|} \right).$$
(14)

The loop closure can be determined by setting a threshold on $\varphi(\Omega^q, \Omega^c)$.

Loop closure detection is able to efficiently identify loop pairs. However, false detection can lead to failure of pose graph optimization. In order to prevent false positives, geometry consistency verification is used to check the similarity of candidate frames. For the candidate loop frame, we search for the nearby line and plane information from the global map. And for current frame, edge and planar features are extracted and matched to the corresponding global lines and planes by minimizing the point-to-edge and point-to-plane distances. When two frames are unrelated, the sum of distances is often high. Hence the false positive can be filtered out by setting a threshold on it. With the revisited place identified, we can add the edge between two frames into the pose graph and global optimization can be applied to correct the drifts [34].

IV. EXPERIMENT EVALUATION

A. Experiment Setup

In this section we present comprehensive experiments of the proposed Intensity-SLAM, including the evaluation on public dataset and practical experiments on a real robot platform. For public dataset test, the proposed method is evaluated in KITTI [15] that is commonly used for SLAM evaluation. The algorithm is implemented in C++ on Robotics Operating System (ROS) Melodic and Linux Ubuntu 18.04. In order to measure the computational cost of different approaches, all algorithms are tested on a desktop computer equipped with an Intel 6-core i7-8700 processor CPU. The instruction set of Eigen is SSE2. In the practical test, the proposed method is integrated into an Automated Guided Vehicle (AGV) in a warehouse environment. In the AGV test, our algorithm is running in an Intel NUC mini PC with an Intel i7-10710 U processor which is often used on mobile robots.

B. Evaluation Metric

To evaluate the accuracy of localization, We use Average Translational Error (ATE) and Average Rotational Error (ARE) [15]:

$$E_{rot}(\mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{i,j \in \mathcal{F}} \angle[\hat{\mathbf{T}}_j \hat{\mathbf{T}}_i^{-1} \mathbf{T}_i \mathbf{T}_j^{-1}]$$

$$E_{trans}(\mathcal{F}) = \frac{1}{|\mathcal{F}|} \sum_{i,j \in \mathcal{F}} \|\hat{\mathbf{T}}_j \hat{\mathbf{T}}_i^{-1} \mathbf{T}_i \mathbf{T}_j^{-1}\|_2, \quad (15)$$

where $\mathcal{F}$ is a set of frames (i, j), $\mathbf{T}$ and $\hat{\mathbf{T}}$ are estimated and true LiDAR poses respectively, and $\angle[\cdot]$ is the rotation angle. The computing cost is measured by the average processing time of each frame.

C. Evaluation on Public Dataset

The KITTI dataset is collected from an autonomous driving car running in a large scale urban area. It is challenging due to the complex environment with both static and dynamic objects. The proposed method is firstly evaluated on the KITTI dataset with various scenarios such as urban, highway and country. In total we tested over more than 10 000 frames with more than 10 km travelling distance. Some of the experimental results are shown in Fig. 1, in which Sequence 00, 02, 05, 07, and 09 are selected for demonstration. The trajectory generated from the
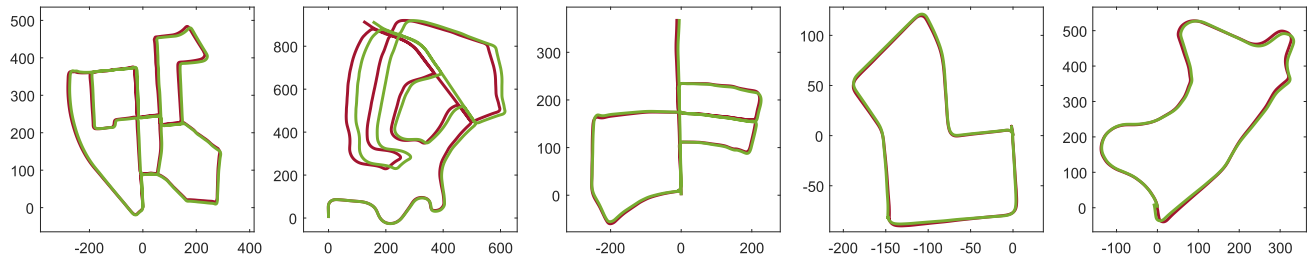
Fig. 4. Result of the proposed method on KITTI dataset. From left to right are Sequence 00, 02, 05, 07, and 09. The result of intensity-SLAM is plotted in green and the ground truth is plotted in red.
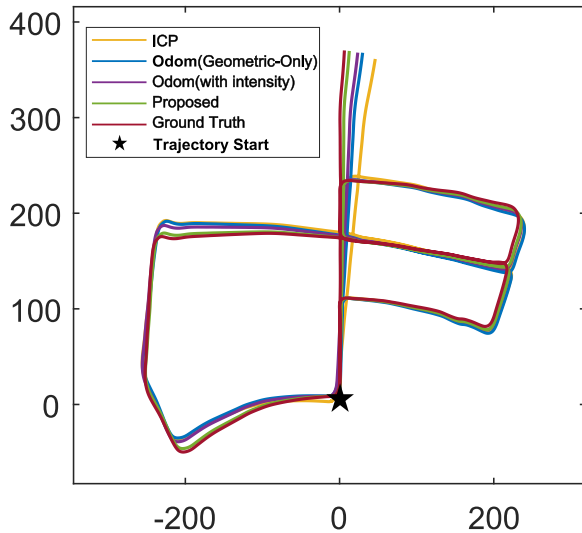

Fig. 5. Performance comparison on KITTI Sequence 05.

TABLE I
RESULT ANALYSIS OF INTENSITY-AIDED METHOD AND NON-INTENSITY-AIDED
METHOD ON KITTI THE DATASET

Method	Time (ms)	ATE (%)	ARE (deg/m)
Iterative Closest Point	142	1.031	0.0092
Odometry(geometric-only)	**92**	0.691	0.0057
Odometry(with intensity)	113	0.627	0.0053
Intensity-SLAM	115	**0.529**	**0.0044**

proposed method is in green and the ground truth is in red. It can be seen that our method is able to track the robot's pose accurately.

D. Ablation Study of Intensity Information

We also demonstrate the effect of intensity information in localization. To be consistent, the loop closure detection is removed and only front-end laser odometry is compared. In the experiment, the "geometric-only laser odometry" method only includes point-to-edge and point-to-plane residual, while "odometry with intensity" method integrates intensity residual into the geometric-only laser odometry. All other parameters are kept the same. Moreover, we also include the result of the Iterative Closest Point method for comparison. The results are shown in Fig. 5 and Table I. In particular, we use KITTI

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT APPROACHES ON
KITTI SEQUENCE 05

Methods	Time (ms)	ATE (%)	ARE (deg/m)
Intensity-SLAM	115	**0.529**	**0.0044**
LeGO-LOAM	**84**	0.724	0.0072
HDL-Graph-SLAM	346	1.127	0.0098
A-LOAM	182	0.702	0.0058

Sequence 05 for illustration. It can be seen that by adding the intensity residual, the overall ATE is reduced from 0.69% to 0.62% and the overall ARE is reduced from $5.7 \times 10^{-5} \deg/m$ to $5.3 \times 10^{-5} \deg/m$. The intensity information improves the translational and rotational accuracy by around 10%, while it does not increase the computing time too much.

E. Comparison With Existing Methods

We also compare our approach with state-of-the-art methods including LeGO-LOAM [13], A-LOAM, and HDL-GRAPH-SLAM [8]. All approaches are tested using the same settings and the results are shown in Table II. It can be seen from the table that our method is competitive in terms of computational cost. And it is worthy noting that our method also achieves the best accuracy and outperforms existing geometric-only methods.

F. Validation on Practical Robots

In this section, we test the performance of our method on a practical robot in a warehouse environment. To this end, we implement our algorithm on an industrial AGV that is used for grabbing and transporting materials. As shown in Fig. 6(a), the AGV is equipped with an Intel NUC mini computer and a Velodyne VLP-16 LiDAR for perception and localization. In this experiment, the robot uses Intensity-SLAM for localization and is supposed to fetch items from the target storage shelf and return. The localization and mapping result is shown in Fig. 6(b), with the robot's trajectory plotted in red. It can be seen that our method is able to provide reliable localization and generate a dense 3D map for a complex indoor environment. In particular, the robot is manually controlled by a joystick to return to the same location and we record the final localization drift. The final localization drift is about 5 cm, which is accurate enough for general indoor robotic applications.
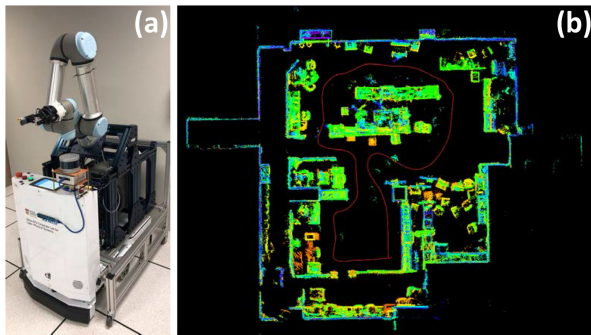
Fig. 6. Localization and mapping result in warehouse environment. (a) AGV platform for warehouse manipulation. (b) Localization and mapping result of the proposed method. The robot's trajectory is plotted in red color.

V. CONCLUSION

In this letter, we present a novel intensity-assisted full SLAM framework for LiDAR-based localization system. Existing methods mainly leverage geometric-only features to estimate location and ignore intensity information. We begin with the physical model of intensity measurement and illustrate the use of intensity information. We introduce the intensity map and intensity residual for pose estimation to improve the localization accuracy. Moreover, we propose a full SLAM structure that consists of both front-end and back-end systems, namely Intensity-SLAM. Thorough experiments are performed in different scenarios, including indoor AGV mapping in warehouse and outdoor autonomous driving in urban areas. The experimental results show that by integrating intensity information, the localization accuracy is significantly improved. The proposed method also achieves much lower translational error and rotational error compared to the state-of-the-art geometric-only methods. Our method is publicly available.

REFERENCES

[1] D. Sun, F. Geiger, and B. Nebel, "Towards effective localization in dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 4517–4523.

[2] E. Stenborg, C. Toft, and L. Hammarstrand, "Long-term visual localization using semantically segmented images," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6484–6490.

[3] R. Liu, C. Yuen, T.-N. Do, D. Jiao, X. Liu, and U.-X. Tan, "Cooperative relative positioning of mobile users by fusing IMU inertial and UWB ranging information," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 5623–5629.

[4] M. Sweatt, A. Ayoade, Q. Han, J. Steele, K. Al-Wahedi, and H. Karki, "WiFi based communication and localization of an autonomous mobile robot for refinery inspection," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 4490–4495.

[5] Q. Li, J. P. Queralta, T. N. Gia, Z. Zou, and T. Westerlund, "Multi-sensor fusion for navigation and mapping in autonomous vehicles: Accurate localization in urban environments," *Unmanned Syst.*, vol. 8, no. 03, pp. 229–237, 2020.

[6] C. Debeunne and D. Vivet, "A review of visual-lidar fusion based simultaneous localization and mapping," *Sensors*, vol. 20, no. 7, 2020, Art. no. 2068.

[7] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robot.: Sci. Syst.*, vol. 2, p. 9, 2014.

[8] 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.

[9] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek, "The trimmed iterative closest point algorithm," in *Object Recognit. Supported User interact. Service Robots*, vol. 3, 2002, pp. 545–548, doi: 10.1109/ICPR.2002.1047997.

[10] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, 2019, Art. no. 1729881419841532.

[11] D. Rozenberszki and A. Majdik, "LOL: Lidar-only odometry and localization in 3D point cloud maps," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4379–4385.

[12] 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.

[13] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "DeepICP: An end-to-end deep neural network for 3D point cloud registration," *CoRR*, vol. abs/1905.04153, 2019, *arXiv:1905.04153*.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[15] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-net: Towards learning based LiDAR localization for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6389–6398.

[16] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robot.: Sci. Syst.*, vol. 2, no. 4, 2009, Art. no. 435.

[17] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D ndt representations," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1377–1393, 2012.

[18] X. Chen *et al.*, "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proc. Robot.: Sci. Syst.*, Corvalis, Oregon, USA, Jul. 2020, doi: 10.15607/RSS.2020.XVI.009.

[19] Y. Tian, X. Liu, L. Li, and W. Wang, "Intensity-assisted ICP for fast registration of 2D-LIDAR," *Sensors*, vol. 19, no. 9, 2019, Art. no. 2124.

[20] R. A. Hewitt and J. A. Marshall, "Towards intensity-augmented slam with LiDAR and ToF sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 1956–1961.

[21] D. P. Frost, O. Kähler, and D. W. Murray, "Object-aware bundle adjustment for correcting monocular scale drift," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4770–4776.

[22] R. Hewitt, "Intense navigation: Using active sensor intensity observations to improve localization and mapping," Ph.D. dissertation, Queen's University, 2018.

[23] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. Seventh IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157, doi: 10.1109/ICCV.1999.790410.

[24] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2548–2555.

[25] S. Khan, D. Wollherr, and M. Buss, "Modeling laser intensities for simultaneous localization and mapping," *IEEE Robot. Automat. Lett.*, vol. 1, no. 2, pp. 692–699, Jul. 2016.

[26] S. Kohlbrecher, J. Meyer, O. von Stryk, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *Proc. IEEE Int. Symp. Safety, Secur. Rescue Robot.*, Nov. 2011, pp. 155–160.

[27] 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.

[28] S. Khan, D. Wollherr, and M. Buss, "Adaptive rectangular cuboids for 3D mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 2132–2139.

[29] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[30] J. J. Moré, "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Numerical Analysis*, Berlin, Heidelberg: Springer, 1978, pp. 105–116.

[31] D. L. Rizzini, "Place recognition of 3D landmarks based on geometric relations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2017, pp. 648–654.

[32] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point cloud descriptors for place recognition using sparse visual information," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 4830–4836.

[33] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4802–4809.

[34] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, winter 2010.