InTEn-LOAM: Intensity and Temporal Enhanced LiDAR Odometry and Mapping

Shuaixin Li\textsuperscript{a}, Bin Tian\textsuperscript{b,\textasteriskcentered}, Zhu Xiaozhou\textsuperscript{a}, Gui Jianjun\textsuperscript{a}, Yao Wen\textsuperscript{a}, Guangyun Li\textsuperscript{c,\textasteriskcentered\textasteriskcentered}

\textsuperscript{a}the National Innovation Institute of Defence Technology, PLA Academy of Military Sciences, Beijing 100079, China
\textsuperscript{b}the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Science, Beijing 100081, China
\textsuperscript{c}the Department of Geospatial Information, PLA Information Engineering University, Zhengzhou 450001, China

Abstract

Traditional LiDAR odometry (LO) systems mainly leverage geometric information obtained from the traversed surroundings to register laser scans and estimate LiDAR ego-motion, while it may be unreliable in dynamic or unstructured environments. This paper proposes InTEn-LOAM, a low-drift and robust LiDAR odometry and mapping method that fully exploits implicit information of laser sweeps (i.e., geometric, intensity, and temporal characteristics). Scanned points are projected to cylindrical images, which facilitate the efficient and adaptive extraction of various types of features, i.e., ground, beam, facade, and reflector. We propose a novel intensity-based points registration algorithm and incorporate it into the LiDAR odometry, enabling the LO system to jointly estimate the LiDAR ego-motion using both geometric and intensity feature points. To eliminate the interference of dynamic objects, we propose a temporal-based dynamic object removal approach to filter them out before map update. Moreover, the local map is organized and downsampled using a temporal-related voxel grid filter to maintain the similarity between the current scan and the static local map. Extensive experiments are conducted on both simulated and real-world datasets. The results show that the proposed method achieves similar or better accuracy w.r.t the state-of-the-arts in normal driving scenarios and outperforms geometric-based LO in unstructured environments.

Keywords: SLAM; LiDAR odometry; dynamic removal; point intensity; scan registration

1. INTRODUCTION

Autonomous robots and self-driving vehicles must have the ability to localize themselves and intelligently perceive the external surroundings. Simultaneous localization and mapping (SLAM) focuses on the issue of vehicle localization and navigation in unknown environments, which plays a major role in many autonomous driving and robotics-related applications, such as mobile mapping\cite{Li et al. (2020)}, space exploration\cite{Ebadi et al. (2020)}, robot localization\cite{Filipenko and Afanasyev (2018)}, and high-definition map production\cite{Yang et al. (2018)}. In accordance with the
on-board perceptional sensors, it can be roughly classified into two categories, i.e., camera-based and LiDAR-based SLAM. Compared with images, LiDAR (Light detection and ranging) point clouds are invariant to the changing illumination and sufficiently dense for 3D reconstruction tasks. Accordingly, LiDAR SLAM solutions have become a preferred choice for self-driving car manufacturers than vision-based solutions Milz et al. (2018); Campos et al. (2020); Qin et al. (2018). Note that methods with loop closure are often called ‘SLAM solutions’, while those without the module are called ‘odometry solutions’. However, both of them owns abilities of self-localization in unknown scenes and mapping the traversed environments. For instance, though LOAM Zhang and Singh (2017) and LeGO-LOAM Shan and Englot (2018) achieve low-drift and real-time pose estimation and mapping, only LeGO-LOAM can be referred as complete SLAM solution since it is a loop closure-enabled system.

![Figure 1: Overview of the proposed InTEn-LOAM system](image)

---

It has witnessed remarkable progress in LiDAR-based SLAM for the past decade Zhang and Singh (2017); Behley and Stachniss (2018); Shan and Englot (2018); Jiao et al. (2020); Zhou et al. (2021); Koide et al. (2019). The state-of-the-art solutions have shown remarkable performances, especially in structured urban and indoor scenes. Recent years have seen solutions for more intractable problems, e.g., fusion with multiple sensors Zhao et al. (2019); Palieri et al. (2020); Shan et al. (2020); Qin et al. (2020); Lin et al. (2021), adapting to cutting-edge solid-state LiDAR Li et al. (2021), global localization Dube et al. (2020), improving the efficiency of optimization back-end Droeschel et al. (2017); Ding et al. (2020), etc., yet many issues remain unsolved. Specifically, most conventional LO solutions currently ignore intensity information from the reflectance channel, though it reveals reflectivities of different objects in the real world.
An efficient incorporation approach of making use of point intensity information is still an open problem since the intensity value is not as straightforward as the range value. It is a value w.r.t many factors, including the material of target surface, the scanning distance, the laser incidence angle, as well as the transmitted energy. Besides, the laser sweep represents a snapshot of surroundings, and thus moving objects, such as pedestrians, vehicles, etc., may be scanned. These dynamic objects result in ‘ghosting points’ in the accumulated points map and may increase the probability of incorrect matching, which deteriorates the localization accuracy of LO. Moreover, improving the robustness of point registration in some geometric-degraded environments, e.g., long straight tunnel, is also a topic worthy of in-depth discussion. In this paper, we present InTEn-LOAM (as shown in Fig.1) to cope with the aforementioned challenges. The main contributions of our work are summarized as four-fold:

• We propose an efficient range-image-based feature extraction method that is able to adaptively extract features from the raw laser scan and categorize them into four different types in real-time.

• We propose a coarse-to-fine, model-free method for online dynamic object removal enabling the LO system to build a purely static map by removing all dynamic outliers in raw scans. Besides, we improved the voxel-based downsize filter, making use of the implicitly temporal information of consecutive laser sweeps to ensure the similarity between the current scan and the local map.

• We propose a novel intensity-based points registration algorithm that directly leverages reflectance measurements to align point clouds, and we introduce it into the LO framework to achieve jointly pose estimation utilizing both geometric and intensity information.

• Extensive experiments are conducted to evaluate the proposed system. Results show that InTEn-LOAM achieves similar or better accuracy in comparison with state-of-the-art LO systems and outperforms them in unstructured scenes with sparse geometric features.

2. RELATED WORK

2.1. Point cloud registration and LiDAR odometry

Point cloud registration is the most critical problem in LiDAR-based autonomous driving, which is centered on finding the best relative transformation of point clouds. Existing registration techniques can be either categorized into feature-based and scan-based methods Furukawa et al. (2015) in terms of the type of data, or local and global methods Zong et al. (2018) in terms of the registration reference. Though the local registration requires a good transformation initial, it has been widely used in LO solutions since sequentially LiDAR sweeps commonly share large overlap, and a coarse initial can be readily predicted.

For feature-based approaches, different types of encoded features, e.g., FPFH (fast point feature histogram) Rusu et al. (2009), CGF (compact geometric feature) Khoury et al. (2017), and arbitrary shapes are extracted to establish valid data associations. LOAM Zhang and Singh (2017) is one of the pioneering works of feature-based LO, which extracts plane and edge features based on the sorted smoothness of each point. Many follow-up works follow the proposed feature extraction scheme Shan et al. (2020); Li et al. (2021); Qin et al. (2020); Lin et al. (2021). For example, LeGO-LOAM Shan and Englot (2018) additionally segmented ground to bound the drift in the ground norm direction. MULLS (multi-metric linear least square) Pan et al.
explicitly classifies features into six types, (facade, ground, roof, beam, pillar, and encoded points) using the principle component analysis (PCA) algorithm and employs the least square algorithm to estimate the ego-motion, which remarkably improves the LO performance, especially in unstructured environments. Yin et al. (2020) proposes a convolutional auto-encoder (CAE) to encode feature points for conducting a more robust point association.

Scan-based local registration methods iteratively assign correspondences based on the closest-distance criterion. The iterative closest point (ICP) algorithm, introduced by Besl and McKay (1992), is the most popular scan registration method. Many variants of ICP have been derived for the past three decades, such as Generalized ICP (GICP) Segal et al. (2009) and improved GICP Yokozuka et al. (2021). Many LO solutions apply variants of ICP to align scans for its simplicity and low computational complexity. For example, Moosmann and Stiller (2011) employs standard ICP, while Palieri et al. (2020) and Behley and Stachniss (2018) employ GICP and normal ICP. The normal distributions transform (NDT) method, first introduced by Biber and Straßer (2003), is another popular scan-based approach, in which surface likelihoods of the reference scan are used for scan matching. Because of that, there is no need for computationally expensive nearest-neighbor searching in NDT, making it more suitable for LO with large-scale map points Zhou et al. (2021); Zhao et al. (2019); Koide et al. (2019).

2.2. Fusion with point intensity

Some works have attempted to introduce the intensity channel into scan registration. Inspired by GICP, Servos and Waslander (2017) proposes the multi-channel GICP (MCGICP), which integrates color and intensity information into the GICP framework by incorporating additional channel measurements into the covariances of points. In Khan et al. (2016), a data-driven intensity calibration approach is presented to acquire a pose-invariant measure of surface reflectivity. Based on that, Wang et al. (2021) establishes voxel-based intensity constraints to complement the geometric-only constraints in the mapping thread of LOAM. Pan et al. (2021) assigns higher weights for associations with similar intensities to suppress the effect of outliers adaptively. Besides, the end-to-end learning-based registration framework, named Deep VCP (virtual corresponding points) Lu et al. (2019), is proposed, achieving comparable accuracy to prior state-of-the-arts. The intensity channel is used to find stable and robust feature associations, which are helpful to avoid the interference of negative true matchings.

2.3. Dynamic object removal

A good amount of learning-based works related to dynamic removal have been reported in Guo et al. (2020). In general, the trained model is used to predict the probability score that a point originated from dynamic objects. The model-based approaches enable to filter out of the dynamics independently, but they also require laborious training tasks, and the segmentation performance is highly dependent on the training dataset.

Traditional model-free approaches rely on differences between the current laser scan and previous scans Yoon et al. (2019); Dewan et al. (2016); Kim and Kim (2020). Though it’s convenient and straightforward, only points that have fully moved outside their original position can be detected/removed.

3. METHODOLOGY

The proposed framework of InTEn-LOAM consist of 5 submodules, i.e., feature extraction filter (FEF), scan-to-scan registration (S2S), scan-to-map registration (S2M), temporal-based
Figure 2: Overall workflow of InTEn-LOAM.

voxel filter (TVF) and dynamic object removal (DOR) (see Fig 2). Following LOAM, the LiDAR odometry and mapping are executed on two parallel threads to improve the running efficiency.

3.1. Feature extraction filter

The workflow of FEF is summarized in Fig 3, which corresponds to the gray block in Fig 2. The FEF receives a raw scan frame and outputs four types of features, i.e., ground, facade, edge, and reflector, and two types of cylindrical images, i.e., range and label image.

3.1.1. Motion compensation

Given the point-wise timestamp of a scan \( P \), the reference pose for a point \( p_i \in P \) at timestamp \( \tau_i \) can be interpolated by the relative transformation \( T_{r,s} = [R_{r,s}, t_{r,s}] \) under the assumption of uniform motion:

\[
T_{s,t} = [\text{slerp}(R_{r,s}, s, s_i)^7, -s_i \cdot T_{r,s}^{-1} \cdot t_{r,s}],
\]

where slerp(\cdot) represents the spherical linear interpolation. The time ratio \( s_i \) is \( s_i = \frac{\tau_e - \tau_s}{\tau_e - \tau_s} \), where \( \tau_s, \tau_e \) stand for the start and end timestamps of the laser sweep, respectively. Then, the distorted points can be deskewed by transforming to the start timestamp \( T_{s,t} \cdot p_i \in P' \).

3.1.2. Scan preprocess

The undisordered points \( P' \) are first preprocessed. The main steps are as below:

I. Scan projection. \( P' \) is projected into a cylindrical plane to generate range and intensity images, i.e. \( D \) and \( I \) (see Fig 1(d) and (e)). A point with 3D coordinates \( p_i = [x, y, z]^T \) can be...
Figure 3: The workflow of FEF.

Projected as a cylindrical image pixel \([u, v]^\top\) by:

\[
\begin{pmatrix}
u \\
v
\end{pmatrix}
= \begin{pmatrix}
[1 - \arctan(y, x) \cdot \pi^{-1}] \cdot \frac{y}{w} \\
\left(\arcsin\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) + \theta_d\right) \cdot \frac{h}{\pi}
\end{pmatrix},
\]

where \(\theta = \theta_d + \theta_v\) is the vertical field-of-view of the LiDAR, and \(w, h\) are the width and height of the resulting image. In \(D\) and \(I\), each pixel contains the smallest range and the largest reflectance of scanning points falling into the pixel, respectively. In addition, \(P'\) is also preprocessed as segment image \(S\) (see Fig. 1(b)) according to azimuthal and radial directions of 3D points, and each pixel contains the lowest \(z\). The former converter is the same as \(u\) in Eq. (2), while the latter is equally spaced with the distance interval \(\Delta\rho\):

\[
\rho = \lfloor \sqrt{(x^2 + y^2 + z^2)} / \Delta\rho \rfloor,
\]

where \(\lfloor \cdot \rfloor\) indicates rounding down operator. Note that the size of \(S\) is not the same as \(D\).

II. Ground segmentation. The method from [Himmelsbach et al. 2010] is applied in this paper with the input of segment image \(S\). Each column of \(S\) is fitted as a ground line \(l_i = a_i \cdot \rho + b_i\). Then, residuals can be calculated, which represents the differences between the predicted and the observed \(z\):

\[
r(u, v) = l_i(D(u, v)) - D(u, v) \cdot \sin(\theta_v),
\]

where \(\theta_v\) indicates the vertical angle of the \(v\)th row in \(D\). Pixels with residuals smaller than the threshold \(T_{\text{th}}\) will be marked as ground pixels with label identity 1.
III. Object clustering. After the ground segmentation, the angle-based object clustering approach from [Bogoslavskyi and Stachniss (2017)] is conducted to group pixels into different clusters with identified labels and generate a label image \( I \) (see the label image in Fig. 2).

IV. Create feature images. We partition the intensity image \( I \) into \( M \times N \) blocks and establish intensity histograms for each block. The extraction threshold \( \Theta_{th} \) of each intensity block is adaptively determined by taking the median of the histogram. Besides, intensity difference image \( I_{\Delta} \), normal image \( N \), curvature image \( C \), are created by:

\[
I_{\Delta}(u, v) = I(u, v) - I(u, v + 1),
\]

\[
N(u, v) = (\Pi[D(u + 1, v)] - \Pi[D(u, v)])
\times (\Pi[D(u, v + 1)] - \Pi[D(u, v)]),
\]

\[
C(u, v) = \frac{1}{N \cdot D(u, v)} \sum_{i,j \in N} (D(u, v) - D(u + i, v + j))
\]

where \( \Pi[] : D \rightarrow \mathcal{P} \) denotes the mapping function from a range image pixel to a 3D point. \( N \) is the neighboring pixels count. Furthermore, pixels in the cluster with fewer than 15 points are marked as noises and blocked. All the valid-or-not flag is stored in a binary mask image \( B \).

3.1.3. Feature extraction

According to the above feature images, pixels of four categories of features can be extracted. Then 3D feature points, i.e., ground \( P^G \), facade \( P^F \), edge \( P^E \), and reflector \( P^R \), can be obtained per the pixel-to-point mapping relationship. Specifically,

- Points correspond to pixels that meet \( L(u, v) = 1 \) and \( B(u, v) \neq 0 \) are categorized as \( P^G \).
- Points correspond to pixels that meet \( C(u, v) > \Theta_{th} \) and \( B(u, v) \neq 0 \) are categorized as \( P^E \).
- Points correspond to pixels that meet \( C(u, v) < \Theta_{th} \) and \( B(u, v) \neq 0 \) are categorized as \( P^F \).
- Points correspond to pixels that meet \( I(u, v) > \Theta_{th} \) and \( B(u, v) \neq 0 \) are categorized as \( P^R \).

Besides, to keep the gradient of local intensities points in pixels that meet \( I(u, v) > \Theta_{th} \), as well as their neighbors are all included in \( P^R \).

To improve the efficiency of scan registration, the random downsample filter (RDF) is applied on \( P^G \) and \( P^R \) to obtain \( N^G \) downsampled edge features \( P^{G'} \) and \( N^R \) facade features \( P^{R'} \). To obtain \( N^E \) refined edge features \( P^{E'} \) and \( N^F \) refined facade features \( P^{F'} \), the non-maximum suppression (NMS) filter based on point curvatures is applied on \( P^E \) and \( P^F \).

3.2. Intensity-based scan registration

Similar to the geometric-based scan registration, given the initial guess of the transformation \( T_{t,s} \) from source points \( P^t \) to target points \( P^t \), we try to estimate the LiDAR motion \( T_{t,s} \) by matching the local intensities of the source and target. In the case of geometric features registration, the motion estimation is solved through nonlinear iterations by minimizing the sum of Euclidean distances from each source feature to their correspondence in the target scan. In the case of reflecting features registration, however, we minimize the sum of intensity differences instead. The fundamental idea of the intensity-based point cloud alignment method proposed
in this paper is to make use of the similarity of intensity gradients within the local region of laser scans to achieve scan matching. Because of the discreteness of the laser scan, sparse 3D points in a local area are not continuous, causing the intensity values of the laser sweep non-differentiable. To solve this issue, we introduce a continuous intensity surface model using the local support characteristic of the B-spline basis function. A simple intensity surface example is shown in Fig.4.

3.2.1. B-spline intensity surface model

The intensity surface model presented in this paper uses the uniformly distributed knots of the B-spline; thus, the B-spline is defined fully by its degree [Sommer et al., 2020]. Specifically, the intensity surface is a space spanned by three $d$-degree B-spline functions on the orthogonal axes, and each B-spline is controlled by $d+1$ knots on the axis. Mathematically, the B-spline intensity surface in local space is a scalar-valued function $\mu(\mathbf{p}) : \mathbb{R}^3 \rightarrow \mathbb{R}$, which builds the mapping relationship between a 3D point $\mathbf{p} = [x, y, z]^T$ and its intensity value. The mapping function is defined by the tensor product of three B-spline functions and control points $c_{i,j,k} \in C$ in the local space:

\[
\mu(\mathbf{p}) = \sum_{i=0}^{d+1} \sum_{j=0}^{d+1} \sum_{k=0}^{d+1} c_{i,j,k} b_i^d(x) b_j^d(y) b_k^d(z) = \text{vec}(b_i^d \otimes b_j^d \otimes b_k^d)^T \cdot \text{vec}(C)
\]

where $b^d$ is the $d$-degree B-spline function. We use the vectorization operator vec(·) and Kronecker product operator $\otimes$ to transform the above equation in the form of matrix multiplication. In this paper, the cubic ($d = 3$) B-spline function is employed.

3.2.2. Observation constraint

The intensity observation constraint is defined as the residual between the intensities of source points and their predicted intensities in the local intensity surface model. Fig.4 demonstrates how to predict the intensity on the surface patch for a reflector feature point. The selected point $\mathbf{p} \in \mathcal{P}_s$ with intensity measurement $\eta$ is transformed to the model frame by $\bar{\mathbf{q}} = T_{s,p} \cdot \mathbf{p}$. Then the nearest point $\mathbf{q} \in \mathcal{P}_t$ and its R-neighbor points $\mathbf{q}_n \in \mathcal{P}_t$, $n = 1 \cdots N$ can be searched. Given the uniform space of the B-spline function $\kappa$, the neighborhood points $\mathbf{q}_n$ can be voxelized with the center $\mathbf{q}$ and the resolution $\kappa \times \kappa \times \kappa$ to generate control knots $c_{\mathbf{q}}$ for the local intensity
surface. The control knot takes the value of the average intensities of all points in a voxel. To sum up, the residual is defined as:

\[ r_I(\tilde{T}_t, s) = [\phi(\bar{T}_t, s) \cdot p^\top \cdot c_q - \eta]. \] (7)

Stacking normalized residuals to obtain residual vector \( r_I(\tilde{T}_t, s) \), and computing the Jacobian matrix of \( r_I \) w.r.t. \( T_t, s \), denoted as \( J_I = \partial r_I / \partial T_t, s \). The constructed nonlinear optimization problem can be solved by minimizing \( r_I \) toward zero using L-M algorithm. Note that Lie group and Lie algebra are implemented for the 6-DoF transformation in this paper.

### 3.3. Dynamic object removal

The workflow of the proposed DOR is shown in Fig. 5, which corresponds to the pink block in Fig.2. Inputs of the DOR filter include the current laser points \( P_k \), the previous static laser points \( P_{s,k-1} \), the local map points \( M_k \), the current range image \( D_{P_k} \), the current label image \( L_{P_k} \), and the estimated LiDAR pose in the world frame \( \tilde{T}_{w,k} \). The filter divides \( P_k \) into two categories, i.e., the dynamic \( P_{d,k} \) and static \( P_{s,k} \). Only static points will be appended into the local map for map update. The DOR filter introduced in this paper exploits the similarity of point clouds in the adjacent time domain for dynamic points filtering and verifies dynamic objects based on the segmented label image.

#### 3.3.1. Rendering range image for the local map

Both downsampling with coarse resolution and uneven distribution of map points may result in pixel holes in the rendered range image. Considering the great similarity of successive laser sweeps in the time domain, we use both the local map points \( M_k \) and the previous static laser points \( P_{s,k-1} \) to generate the to-be-rendered map points \( E_k \):

\[ E_k = T_{w,k}^{-1} \cdot M_k \cup T_{k,k-1} \cdot P_{s,k-1}. \] (8)
The rendered image $D_{M_k}$ and the current scan image $D_{P_k}$ are shown in the second and third rows of Fig.5. A pedestrian can be clearly distinguished in $D_{P_k}$ but not in $D_{M_k}$.

3.3.2. Temporal-based dynamic points searching

Dynamic pixels in $D_{P_k}$ can be coarsely screened out in accordance with the depth differences between $D_{P_k}$ and $D_{M_k}$. In particular if the depth difference at $[u, v]^T$ is larger than the threshold $Th_{\Delta d}$, the pixel will be marked as dynamic. Consecutively, we can also generate a binary image $D_{B_k}$ indicating whether the pixel is dynamic or not:

$$D_{B_k}(u, v) = \begin{cases} 1, & \text{if } D_{M_k}(u, v) - D_{P_k}(u, v) > Th_{\Delta d} \\ 0, & \text{otherwise} \end{cases}$$

where $D_{M_k}(u, v) \neq 0$ and $D_{P_k}(u, v) \neq 0$. An example of $D_{B_k}$ is shown in the fourth row of Fig.5, in which red pixels represent the static and purple pixels represent the dynamic. To improve the robustness of the DOR filter to different point depths, we use the adaptive threshold $Th_{\Delta d} = s_d \cdot D_{P_k}(u, v)$, where $s_d$ is a constant coefficient.

3.3.3. Dynamic object validation

It can be seen from $D_{B_k}$ that it generates a large number of false positive (FP) dynamic pixels using the pixel-by-pixel depth comparison. To handle the above issue, we utilize the label image to validate dynamic according to the fact that points originating from the same object should have the same status label. We denote the pixel number of a segmented object and the dynamic pixel number as $N_i$ and $N_{dpi}$, which can be counted from $L_{P_k}$ and $D_{B_k}$, respectively. Two basic assumptions generally hold in terms of dynamic points, i.e., a. ground points cannot be dynamic; b. the percentage of FP dynamic pixels in a given object will not be significant. According to the above assumptions, we can validate dynamic pixels at the object level:

$$\frac{N_{dpi}}{N_i} \geq Th_N \& L_{P_k}(u, v) \neq 1 ?$$

$$D_{A_k}(u, v) = \begin{cases} 1, & \text{if } D_{A_k}(u, v) > Th_{\Delta d} \\ 0, & \text{otherwise} \end{cases}$$

The E.q.(10) indicates that only objects that is marked as the non-ground object or own the dynamic pixel ratio larger than the threshold will be recognized as dynamic. In $D_{A_k}$, pixels belonging to dynamic objects will retain the depth differences, while the others will be reset as 0. As the depth difference image shown in the sixth row of Fig.5 though many FP dynamic pixels are filtered out after the validation, the true positive (TP) dynamic pixels from the moving pedestrian on the right side are still remarkable. Then, the binary image $D_{B_k}$ is updated by substituting the refined $D_{A_k}$ into E.q.(9).

3.3.4. Points classification

According to $D_{B_k}$, dynamic 3D points in extracted features can be marked using the mapping function $\Pi[\cdot] : D \mapsto P$. Since the static feature set is the complement of the dynamic feature set w.r.t. the full set of extracted features, the static features can be filtered by $P_{x,k} = P_k - P_{d,k}$.

3.4. LiDAR odometry

Given the initial guess $T_{k-1}$, extracted features, i.e., downsampled ground and reflector features $P_{G'}$ and $P_{R'}$, as well as refined edge and facade features $P_{E'}$ and $P_{F'}$, are utilized to
estimate the optimal estimation of $T_{k,k-1}$, and then the LiDAR pose $T_{w,k}$ in the global frame is reckoned. The odometry thread corresponds to the green S2S block in Fig.2 and the pseudo code is shown in Algorithm 1. To improve the performance of geometric-only scan registration, the proposed LO incorporates reflector features and estimate relative motion by jointly solving the multi-metric nonlinear optimization (NLO).

3.4.1. Constraint model

As shown in Fig.6, constraints are modeled as the point-to-model intensity difference (for reflector feature) and the point-to-line (for edge feature)/point-to-plane (for ground and facade feature) distance, respectively.

I. Point-to-line constraint. Let $p_i \in P_{E,k}, i = 1 \cdots N_E$ be an edge feature point. The association of $p_i$ is the line connected by $q_j, q_m \in P_{E,k-1}$, which represent the closest point of $T_{k-1,k} \cdot p_i$ in $P_{E,k-1}$ and the closest neighbor in the preceding and following scan lines to the $q_j$ respectively. The constraint equation is formulated as the point-to-line distance:

$$r_{E,i} = \|v_j \times (T_{k-1,k} \cdot p_i)\|,$$

$$v_j = \frac{q_j - q_m}{\|q_j - q_m\|}.$$  \hspace{1cm} (11)

The $N_E \times 1$ edge feature error vector $r_E$ is constructed by stacking all normalized edge residuals (Line 11).

II. Point-to-plane constraint. Let $p_i \in P_{F,k}, (P_{G,k}), i = 1 \cdots N_F(N_G)$ be a facade or ground feature point. The association of $p_i$ is the plane constructed by $q_j, q_m, q_n$ in the last ground and facade feature points, which represent the closest point of $T_{k-1,k} \cdot p_i$, the closest neighbor in the preceding and following scan lines to the $q_j$ and the closest neighbor in the same scan line to $q_j$ respectively. The constraint equation is formulated as the point-to-plane distance:

$$r_{G,i} = r_{F,i} = n_j \cdot (T_{k-1,k} \cdot p_i),$$

$$n_j = \frac{(q_j - q_m) \times (q_j - q_n)}{\| (q_j - q_m) \times (q_j - q_n) \|}.$$  \hspace{1cm} (12)

The $N_F \times 1$ facade feature error vector $r_F$ and the $N_G \times 1$ ground feature error vector $r_G$ are constructed by stacking all normalized facade and ground residuals (Line 12).

III. Point-to-model intensity difference constraint. The constraint equation is formulated as E.q. (7). The $N_R \times 1$ intensity feature error vector $r_R$ is constructed by stacking all reflector features (Line 13).

Figure 6: Overview of four different types of feature associations. (a) Reflector; (b) Facade; (c) Edge; (d) Ground feature association.
3.4.2. Transformation estimation

According to constraint models introduced above, the nonlinear least square (LS) function can be established for the transformation estimation (Line 12):

$$\hat{T}_{k-1,k} = \arg\min_{T_{k-1,k}} \left( r_{G}^{T}r_{G} + r_{F}^{T}r_{F} + r_{E}^{T}r_{E} + r_{R}^{T}r_{R} \right).$$  \hspace{1cm} (13)

The special euclidean group exp($\xi_{k-1,k}$) = $T_{k-1,k}$ is implemented during the nonlinear optimization iteration. Then $T_{k-1,k}$ can be incrementally updated by:

$$\xi_{k-1,k} \leftarrow \xi_{k-1,k} + \delta \xi_{k-1,k}.$$  \hspace{1cm} (14)

where

$$\delta \xi_{k-1,k} = (J^{T}J)^{-1} J^{T}r,$$

$$J = \left[ J_{G}, \cdots, J_{F}, \cdots, J_{E}, \cdots, J_{R} \right],$$

$$r = \left[ r_{G}^{T}, \cdots, r_{F}^{T}, \cdots, r_{E}^{T}, \cdots, r_{R}^{T} \right].$$  \hspace{1cm} (15)
The Jacobian matrix of constraint equation w.r.t. $\xi_{k-1, k}$ is denoted as $J$. Matrix components are listed as follow.

\[
J_{G,i} = \frac{\partial r_{G,i}}{\partial \xi_{k-1, k}} = n_{jm,n}^T \frac{\partial (T_{1-1,k} p_i)}{\partial \xi_{k-1, k}}, \\
J_{F,i} = \frac{\partial r_{F,i}}{\partial \xi_{k-1, k}} = n_{jm,n}^T \frac{\partial (T_{1-1,k} p_i)}{\partial \xi_{k-1, k}}, \\
J_{E,i} = \frac{\partial r_{E,i}}{\partial \xi_{k-1, k}} = (v_{jm}^T (T_{k-1,k} p_i))^T \frac{\partial (T_{1-1,k} p_i)}{\partial \xi_{k-1, k}}, \\
J_{R,i} = \frac{\partial r_{R,i}}{\partial \xi_{k-1, k}} = \frac{\partial (T_{k-1,k} p_i)^T}{\partial (T_{1-1,k} p_i)} \frac{\partial (T_{1-1,k} p_i)}{\partial \xi_{k-1, k}} \cdot c_i. \tag{16}
\]

3.5. LiDAR mapping

There is always an inevitable error accumulation in the LiDAR odometry, resulting in a discrepancy $\Delta T_k$ between the estimated and actual pose. In other words, the estimated transform from the LiDAR odometry thread is not the exact transform from the LiDAR frame $\{L\}$ to the world frame $\{W\}$ but from $\{L\}$ to the drifted world frame $\{W'\}$:

\[
T_{w,k} = \Delta T_k T_{w',k}. \tag{17}
\]

One of the main tasks of LiDAR mapping thread is optimizing the estimated pose from the LO thread by the scan-to-map registration (green S2M block in Fig.2). The other is managing the local static map (brown TVF and pink DOR blocks in Fig.2). The pseudo code is shown in Algorithm 2.

3.5.1. Local feature map construction

In this paper, the pose-based local feature map construction scheme is applied. In particular, the pose prediction $\hat{T}_{w,k}$ is calculated by Eq.(17) under the assumption that the drift between $\Delta T_k$ and $\Delta T_{k-1}$ is tiny (Line 2). Feature points scanned in the vicinity of $\hat{T}_{w,k}$ are merged (Line 4) and filtered (Line 5) to construct the local map $M_k$. Let $\Gamma(\cdot)$ denotes the filter, and $n \in N$ denotes timestamps of surrounding scans. The local map is built by:

\[
M_k = \Gamma \left( \sum_{m \in N} T_{w,n} \cdot P_{r,n} \right). \tag{18}
\]

The conventional voxel-based downsample filter voxelizes the point cloud and retains one point for each voxel. The coordinate of retained point is averaged by all points in the same voxel. However, for the point intensity, averaging may cause the loss of similarity between consecutive scans. To maintain the local characteristic of the point intensity, we utilize the temporal information to improve the voxel-based downsample filter. In the TVF, a temporal window is set for the intensity average. Specifically, the coordinate of the downsampled point is still the mean of all points in the voxel, but the intensity is the mean of points in the temporal window, i.e., $|t_k - t_n| < Th$, where $t_k$ and $t_n$ represent timestamps of the current scan and selected point, respectively.
3.5.2. Mapping update

The categorized features are jointly registered with feature maps in the same way as in the LiDAR odometry module. The low-drift pose transform $T_{w,k}$ can be estimated by scan-to-map alignment (Line 7). Since the distribution of feature points in the local map is disordered, point neighbors cannot be directly indexed through the scan line number. Accordingly, the K-D tree is utilized for nearest points searching, and the PCA algorithm calculates norms andprimary directions of neighbouring points.

Finally, the obtained $T_{w,k}$ is fed to the DOR filter to filter out dynamic points in the current scan. Only static points $P_{s,k}$ are retained in the local feature map list (Line 20). Moreover, the odometry reference drift is also updated by Eq. (17), i.e. $\Delta T_k = T_{w,k}T_{w',k}^{-1}$ (Line 18).

Algorithm 2: LiDAR Mapping

**Input:** Extracted feature points for registration $P_{g,k}$, $P_{f,k}$, $P_{e,k}$, $P_{r,k}$, feature points for mapping, $P_{g,k}$, $P_{f,k}$, $P_{e,k}$, $P_{r,k}$, estimated transform from LiDAR odometry $T_{v,k}$, scan depth image $D_k$, and labeled image $L_k$

**Output:** refined pose $T_{w,k}$, static scan points $P_{s,k}$

1. the transform drift $\Delta T_k$ and scan keyframes can be loaded from the buffer;

   // Roughly transform the reckoned pose to the world frame

2. $T_{w,k} \leftarrow \Delta T_k \cdot T_{w',k}$;

   // Main

3. if skip a number of frames to keep system efficiency then

   // construct local points map

4. $\mathcal{M}_k \leftarrow \text{searchSurroundKF}(T_{w,k})$;

5. temporalVoxelFilter($\mathcal{M}_k$);

6. for a number of iterations do

   // Find feature associations by parallel threads

7. $r_g, J_g \leftarrow \text{GroundAssoc}(T_{w,k}, P_{g,k}, \mathcal{M}_{g,k})$;

8. $r_f, J_f \leftarrow \text{FacadeAssoc}(T_{w,k}, P_{f,k}, \mathcal{M}_{f,k})$;

9. $r_e, J_e \leftarrow \text{EdgeAssoc}(T_{w,k}, P_{e,k}, \mathcal{M}_{e,k})$;

10. $r_r, J_r \leftarrow \text{ReflectAssoc}(T_{w,k}, P_{r,k}, \mathcal{M}_{r,k})$;

   // Update the estimated pose by the nonlinear optimization

11. $\tilde{T}_{w,k} \leftarrow \text{MultiMetricNLO}(J, r)$;

   // Convergency

12. if convergency $\leftarrow \text{ConvergCond}(\tilde{T}_{w,k}, T_{w,k})$;

   if convergency then

      | break; |

end

end

// Update parameters

17. $T_{w,k} \leftarrow \tilde{T}_{w,k}$;

18. $w' \cdot T_{w,k} = T_{w,k} \cdot T_{w',k}$;

   // Update the local feature map

19. DownsizeFilter($P_{g,k}$, $P_{f,k}$, $P_{e,k}$, $P_{r,k}$);

20. $P_{s,k} \leftarrow \text{DORFilter}(P_{s}, \mathcal{M}_k, D_k, L_k, T_{w,k})$;

21. InsertAsKF($P_{s,k}, T_{w,k}$);

end
4. Experiments

In this section, the proposed InTEn-LOAM is evaluated qualitatively and quantitatively on both simulated and real-world datasets, covering various outdoor scenes. We first test the feasibility of each functional module, including the feature extraction module, intensity-based scan registration, and dynamic points removal. Then we conduct a comprehensive evaluation for InTEn-LOAM in terms of positioning accuracy and constructed map quality. During experiments, the system processing LiDAR scans runs on a laptop computer with 1.8GHz quad cores and 4Gib memory, on top of the robot operating system (ROS) in Linux.

The simulated test environment was built based on the challenging scene provided by the DARPA Subterranean (SubT) Challenge\textsuperscript{1}. We simulated a 1000m long straight mine tunnel (see Fig.7(b)) with smooth walls and reflective signs that are alternatively posted on both sides of the tunnel at 30m intervals. Physical parameters of the simulated car, such as ground friction, sensor temperature and humidity are consistent with reality to the greatest extent. A 16-scanline LiDAR is on the top of the car. Transform groundtruths were exported at 100Hz. The real-world dataset was collected by an autonomous driving car with a 32-scanline LiDAR (see Fig.7(a)) in the autonomous driving test field, where a 150m long straight tunnel is existed. Moreover, the KITTI odometry benchmark\textsuperscript{2} was also utilized to compare with other state-of-the-art LO solutions.

\begin{figure}[h]
\centering
\subfloat[]{
\includegraphics[width=0.4\textwidth]{a}
}
\hfill
\subfloat[]{
\includegraphics[width=0.4\textwidth]{b}
}
\caption{Dataset sampling platform. (a) Autonomous driving car; (b) Simulated mine car and scan example. Magenta laser points are reflected from brown signs in the simulated environments.}
\end{figure}

4.1. Functional module test
4.1.1. Feature extraction module

We validated the feature extraction module on the real-world dataset. In the test, we set the edge feature extraction threshold as $Th_E = 0.3$, the facade feature extraction threshold as $Th_F = 0.1$, and the intensity difference threshold as $Th_M = 80$, and partitioned the intensity image into $16 \times 4$ blocks.

\textsuperscript{1}https://github.com/osrf/subt
\textsuperscript{2}http://www.cvlibs.net/datasets/kitti/eval_odometry.php
Fig. 8 shows feature extraction results. It can be seen that edges, planes, and reflectors can be correctly extracted in various road conditions. With the effect of ground segmentation, breakpoints on the ground (see orange box region in Fig. 8(a)) are correctly marked as plane, avoiding the issue that breakpoints are wrongly marked as edge features due to their large roughness values. In the urban city scene, conspicuous intensity features can be easily found, such as landmarks and traffic lights (see Fig. 8(b)). Though there are many plane features in the tunnel, few valid edge features can be extracted (see Fig. 8(c)). In addition, sparse and scattered plant points with large roughness values (see orange box region in Fig. 8(d)) are filtered as outliers with the help of the object clustering.

According to the above results, some conclusions can be drawn: (1) The number of plane features is always much greater than that of edge features, especially in open areas, which may cause the issue of constraint-unbalance during the multi-metric nonlinear optimization. (2) Static reflector features widely exist in real-world environments, which are useful for the feature-based scan alignment, and should not be ignored. (3) The adapative intensity feature extraction approach makes it possible to manually add reflective targets in feature-degraded environments.

4.1.2. Intensity-based scan registration

We validated the intensity-based scan registration method on the simulated dataset. To highlight the role of intensity-based scan registration, we quantitatively evaluated the relative accuracy of the proposed method and compared the result with prevalent geometric-based scan registration methods, i.e., edge and surface feature registration of LOAM [Zhang and Singh (2017)], multi-metric registration of MULLS [Fan et al. (2021)] and NDT of HDL-Graph-SLAM [Koide et al., 2019]. The evaluation used the simulated tunnel dataset, which is a typical geometric-degraded environment. The measure used to evaluate the accuracy of scan registration is the relative transformation error. In particular, differences between the groundtruth \( T_{k+1}^{GT} \) and the estimated relative transformation \( T_{k+1,k} \) are calculated and represented as an error vector, i.e., \( r_k = \text{vec}(T_{k+1,k}^GT - T_{k+1,k}) \). The norms of translational and rotational parts of \( r_k \) are illustrated in Fig. 9. Note that the result of intensity-based registration only utilizes measurements from the intensity channel of laser scan, instead of all information including range, bearing and intensity of laser scan. The figures show that all four rotation errors of different approaches are less than 0.01°, while errors of InTEn-LOAM and MULLS are less than 0.001°. It demonstrates that laser points from the tunnel wall and ground enable to provide sufficient geometric constraints for the accuracy of relative attitude estimation. However, there are significant differences in relative translation errors (RTE). The intensity-based scan registration achieves the best RTE (less than 0.02m), which is much better than the feature-based of LOAM and NDT of HDL-Graph-SLAM (0.4m and 0.1m), and better than the intensity-based weighting of MULLS(0.05). The result proves the correctness and feasibility of the proposed intensity-based approach under the premise of sufficient intensity features. It also reflects the necessity of fusing reflectance information of points in poorly structured environments.

4.1.3. Dynamic object removal

We validated the DOR module on Seq.07 and 10 of the KITTI odometry dataset. The test result was evaluated by qualitative evaluation method, i.e., marking dynamic points for each scan frame and qualitatively judging the accuracy of the dynamic object segmentation according to the actual targets in the real world the dynamic points correspond to.

Fig. 10 exhibits DOR examples for a single frame of laser scan at typical urban driving scenes. It can be seen that dynamic objects, such as vehicles crossing the intersection, vehicles and
pedestrians traveling in front of/behind the data collection car, can be correctly segmented by the proposed DOR approach no matter the sampling vehicle is stationary or in motion. Fig.11 shows constructed maps at two representative areas, i.e., intersection and busy road. Maps were incrementally built by LOAM (without DOR) and InTEn-LOAM (with DOR) method. We can figure out from the figure that the map built by InTEn-LOAM is better since the DOR module effectively filters out dynamic points to help to accumulate a purely static points map. In contrast, the map constructed by LOAM owns a large amount of ‘ghost points’ increasing the possibility of erroneous point matching.

In general, the above results prove that the DOR method proposed in this paper owns the ability to segment dynamic objects for a scan frame correctly. However, it also has some shortcomings. For instance, (1) The proposed comparison-based DOR filter is sensitive to the quality of laser scan and the density of the local points map, causing the omission or mis-marking of some dynamic points (see the green circle box in the top of Fig.10(a) and the red rectangle box in the bottom of Fig.10(b)); (2) Dynamic points in the first frame of scan cannot be marked using the proposed approach (see the red rectangle box in the top of Fig.10(b)).

4.2. Pose transform estimation accuracy

Table 1: Quantitative evaluation and comparison on KITTI dataset.

| Method      | #00U | #01H | #02C | #03C | #04C | #05C | #06U | #07U | #08U | #09C | #10C | Avg. | time [ms/frame] |
|-------------|------|------|------|------|------|------|------|------|------|------|------|------|-----------------|
| LOAM        | 0.78 | -1.43 | -0.92 | -0.86 | -0.71 | -0.57 | -0.65 | -0.63 | -1.12 | -0.77 | -0.79 | -0.84 | 0.10            |
| IMLS-SLAM   | 0.50 | -0.82 | -0.53 | -0.68 | -0.33 | -0.32 | -0.33 | -0.80 | -0.55 | -0.53 | -0.53 | -0.57 | 1.25            |
| MC2SLAM     | 0.51 | -0.79 | -0.54 | -0.65 | -0.44 | -0.27 | -0.31 | -0.34 | -0.84 | -0.46 | -0.46 | -0.56 | 0.10            |
| SuMa        | 0.69 | 0.30 | 1.70 | 0.50 | 1.10 | 0.40 | 0.70 | 0.50 | 0.40 | 0.20 | 0.50 | 0.30 | 0.07            |
| LO-Net      | 0.78 | 0.42 | 1.42 | 0.40 | 1.01 | 0.45 | 0.73 | 0.59 | 0.56 | 0.54 | 0.62 | 0.35 | 0.10            |
| MULLS-LO    | 0.51 | 0.18 | 0.62 | 0.09 | 0.55 | 0.17 | 0.61 | 0.22 | 0.35 | 0.08 | 0.28 | 0.17 | 0.08            |
| InTEn-LOAM  | 0.52 | 0.21 | 0.64 | 0.35 | 0.54 | 0.28 | 0.37 | 0.31 | 0.24 | 0.11 | 0.34 | 0.31 | 0.09            |

4.2.1. KITTI dataset

The quantitative evaluations were conducted on the KITTI odometry dataset, which is composed of 11 sequences of laser scans captured by a Velodyne HDL-64E LiDAR with GPS/INS groundtruth poses. We followed the odometry evaluation criterion from Geiger et al. (2012) and used the average relative translation and rotation errors (RTE and RRE) within a certain distance range for the accuracy evaluation. The performance of the proposed InTEn-LOAM, and other six state-of-the-art LiDAR odometry solutions whose results are taken from their original papers, are reported in Table[1]. Plots of average RTE/RRE over fixed lengths are exhibited in Fig.12. Note that all comparison methods did not incorporate the loop closure module for more objective accuracy comparison. Moreover, an intrinsic angle correction of 0.2° is applied to KITTI raw scan data for better performance[Pan et al.](2021).

Fig.12 demonstrates that accuracies in different length ranges are stable, and the maximums of average RTE and RRE are less than 0.32° and 0.22°/100m. It also can be seen from the table that the average RTE and RRE of InTEn-LOAM are 0.54° and 0.26°/100m, which outperforms the LOAM accuracy of 0.84%. The comprehensive comparison shows that InTEn-LOAM is superior or equal to the current state-of-the-art LO methods. Although the result of MULLS slightly better than that of InTEn-LOAM, the contribution of InTEn-LOAM is significant considering its excellent performance in long straight tunnel with reflective markers. InTEn-LOAM costs around 90ms per frame of scan with about 3k and 30k feature points in the current scan points
and local map points, respectively. Accordingly, the proposed LO method is able to operate faster than 10Hz on average for all KITTI odometry sequences and achieve real-time performance.

For in-depth analysis, three representative sequences, i.e., Seq.00, 01, and 05, were selected. Seq.00 is a urban road dataset with the longest traveling distance, in which big and small loop closures are included, while geometric features are extremely rich. Consequently, the sequence is suitable for visualizing the trajectory drift of InTEn-LOAM. Seq.01 is a highway dataset with the fastest driving speed. Due to the lack of geometric features in the highway neighborhood, it is the most challenging sequence in the KITTI odometry dataset. Seq.05 is a country road sequence with great variation in elevation and rich structured features.

For Seq.01, it can be seen from Fig.13(c) that areas with landmarks are circled by blue bounding boxes, while magenta boxes highlight road signs on the roadside. The drift of the estimated trajectory of Seq.01 by InTEn-LOAM is quite small (see Fig.13(d)), which reflects that the roadside guideposts can be utilized as reflector features since their high-reflective surfaces, and are conducive to improving the LO performance in such geometric-sparse highway environments. The result also proves that the proposed InTEn-LOAM is capable of adaptively mining and fully exploiting the geometric and intensity features in surrounding environments, which ensures the LO system can accurately and robustly estimate the vehicle pose even in some challenging scenarios. In terms of Seq.00 and 05, both two point cloud maps show excellent consistency in the small loop closure areas (see blue bound regions in Fig.13(a) and (e)), which indicates that InTEn-LOAM owns good local consistency. However, in large-scale loop closure areas, such as the endpoint, the global trajectory drifts incur a stratification issue in point cloud maps (see red bound regions in Fig.13(a) and (e)), which are especially significant in the vertical direction. (see plane trajectory plots in Fig.13(b) and (f)). This phenomenon is because constraints in the z-direction are insufficient in comparison with other directions in the state space since only ground features provide constraints for the z-direction during the point cloud alignment.

4.2.2. Autonomous driving dataset

The other quantitative evaluation test was conducted on the autonomous driving field dataset, the groundtruth of which is referred to the trajectory output of the onboard positioning and orientation system (POS). There is a 150m long tunnel in the data acquisition environment, which is extremely challenging for most LO systems. The root means square errors (RMSE) of horizontal position and yaw angle were used as indicators for the absolute state accuracy. LOAM, MULLS, and HDL-Graph-SLAM were utilized as control groups, whose results are listed in Table.2.

| method           | Positioning error [m] | Heading error [◦] |
|------------------|-----------------------|-------------------|
|                  | x                     | y                 | yaw               |
| LOAM             | 29.478                | 19.220            | 35.828            | 1.586 |
| HDL-Graph-SLAM   | 119.756               | 75.368            | 141.498           | 2.408 |
| MULLS-LO         | 4.133                 | 3.900             | 7.043             | 1.403 |
| InTEn-LOAM       | 1.851                 | 1.337             | 2.664             | 0.476 |

Both LOAM and HDL-Graph-SLAM failed to localize the vehicle with 34.654m and 141.498m positional errors, respectively. MULLS and the proposed InTEn-LOAM are still able to function properly with 2.664m and 7.043m of positioning error and 0.476◦ and 1.403◦ of heading error within the path range of 1.5km. To further investigate the causes of this result, we plotted the cumulative distribution of absolute errors and horizontal trajectories of three LO systems, as shown in Fig.14.
From the trajectory plot, we can see that the overall trajectory drift of InTEn-LOAM and MULLS are relatively small, indicating that these two approaches can accurately localize the vehicle in this challenging scene by incorporating intensity features into the point cloud registration, and using intensity information for the feature weighting. The estimated position of LO inevitably suffers from error accumulation which is the culprit causing trajectory drift. It can be seen from the cumulative distribution of absolute errors that the absolute positioning error of InTEn-LOAM is no more than $10$ m, and the attitude error is no more than $1.5^\circ$. The overall trends of rotational errors of the other three systems are consistent with that of InTEn-LOAM. Results in Table 2 also verify that their rotation errors are similar. The cumulative distribution curves of absolute positioning errors of LOAM and HDL-Graph-SLAM do not exhibit smooth growth but a steep increase in some intervals. The phenomenon reflects the existence of anomalous registration in these regions, which is consistent with the fact that the scan registration-based motion estimation in the tunnel is degraded. MULLS, which incorporates intensity measures by feature constraints weighting, present a smooth curve as similar as InTEn-LOAM. However, the absolute errors of positioning (no more than $19$ m) and heading (no more than $2.5^\circ$) are both large than those of our proposed LO system. We also plotted the RTE and RRE of all four approaches (see Fig 15). It can bee seen that the differences of the RRE between four systems are small, representing that the heading estimations of all theseLO systems are not deteriorated in the geometric-degraded long straight tunnel. In contrast, the RPEs are quite different. Both LOAM and HDL-Graph-SLAM suffer from serious scan registration drifts, while MULLS and InTEn-LOAM are able to positioning normally, and achieve very close relative accuracy.

4.3. Point cloud map quality

4.3.1. Large-scale urban scenario

The qualitative evaluations were conducted by intuitively comparing the constructed map by InTEn-LOAM with the reference map. The reference map is built by merging each frame of laser scan using their groundtruth poses. Maps of Seq.06 and 10 are displayed in Fig 16 and Fig 17, which are the urban scenario with trajectory loops and the country road scenario without loop, respectively.

Although the groundtruth is the post-processing result of POS and its absolute accuracy reaches centimeter-level, the directly merged points map is blurred in the local view. By contrast, maps built by InTEn-LOAM own better local consistency, and various small targets, such as trees, vehicles, and fences, etc., can be clearly distinguished from the points map. The above results prove that the relative accuracy of InTEn-LOAM outperforms that of the GPS/INS post-processing solution, which is very critical for the mapping tasks.

4.3.2. Long straight tunnel scenario

The second qualitative evaluation test was conducted on the autonomous driving field dataset. There is a $150$m long straight tunnel in which we alternately posted some reflective signs on sidewalls to manually add some intensity features in such registration-degraded scenario. Maps of InTEn-LOAM, MULLS, LOAM, and HDL-Graph-SLAM are shown in Fig 18. It intuitively shows that both LOAM and HDL-Graph-SLAM present different degrees of scan registration degradation, while the proposed InTEn-LOAM achieves correct motion estimation by jointly utilizing both sparse geometric and intensity features, as shown in Fig 18. Although MULLS is also able to build a correct map since it utilizes intensity information to re-weight geometric feature constraints during the registration iteration, its accuracies of both pose estimation and mapping are inferior to the proposed LO system.
In addition, we constructed the complete point cloud map for the test field using InTEn-LOAM and compared the result with the local remote sensing image, as shown in Fig.19. It can be seen that the consistency between the constructed point cloud map and regional remote sensing image is good, qualitatively reflecting that the proposed InTEn-LOAM has excellent localization and mapping capability without error accumulation in the around 2km long exploration journey.

5. Conclusions

In this work, we present a LiDAR-only odometry and mapping solution named InTEn-LOAM to cope with some challenging issues, i.e., dynamic environments, intensity channel incorporation. A temporal-based dynamic removal method and a novel intensity-based scan registration approach are proposed, and both of them are utilized to improve the performance of LOAM. The proposed system is evaluated on both simulated and real-world datasets. Results show that InTEn-LOAM achieves similar or better accuracy in comparison with the state-of-the-art LO solutions in normal environments, and outperforms them in challenging scenarios, such as long straight tunnel. Since the LiDAR-only method cannot adapt to aggressive motion, our future work involves developing a IMU/LiDAR tightly coupled method to escalate the robustness of motion estimation.

6. Acknowledgement

7. References

References

Behley, J., Stachniss, C., 2018. Efficient surfel-based slam using 3d laser range data in urban environments., in: Robotics: Science and Systems.
Besl, P.J., McKay, N.D., 1992. Method for registration of 3-d shapes, in: Sensor fusion IV: control paradigms and data structures, International Society for Optics and Photonics. pp. 586–606.
Biber, P., Stradal, W., 2003. The normal distributions transform: A new approach to laser scan matching, in: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453), IEEE. pp. 2743–2748.
Bogoslavskyi, I., Stachniss, C., 2017. Efficient online segmentation for sparse 3d laser scans. PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science 85, 41–52.
Campos, C., Elvira, R., Rodriguez, J.J.G., Montiel, J.M., Tardós, J.D., 2020. Orb-slam3: An accurate open-source library for visual, visual-inertial and multi-map slam. arXiv preprint arXiv:2007.11189 .
Dewan, A., Caselitz, T., Tipaldi, G.D., Burgard, W., 2016. Motion-based detection and tracking in 3d lidar scans, in: 2016 IEEE international conference on robotics and automation (ICRA), IEEE. pp. 4508–4513.
Ding, W., Hou, S., Gao, H., Wan, G., Song, S., 2020. Lidar inertial odometry aided robust lidar localization system in changing city scenes, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 4322–4328.
Droeschel, D., Schwarz, M., Behnke, S., 2017. Continuous mapping and localization for autonomous navigation in rough terrain using a 3d laser scanner. Robotics and Autonomous Systems 88, 104–115.
Dubé, R., Cramarise, A., Dugas, D., Sommer, H., Dymczyk, M., Nieto, J., Siegwart, R., Cadena, C., 2020. Segmap: Segment-based mapping and localization using data-driven descriptors. The International Journal of Robotics Research 39, 339–355.
Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E., Thakur, A., Funahiki, N., Morrell, B., Wood, S., et al., 2020. Lamp: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 80–86.
Filipenko, M., Afanasyev, I., 2018. Comparison of various slam systems for mobile robot in an indoor environment, in: 2018 International Conference on Intelligent Systems (IS), IEEE. pp. 400–407.
leveraging fully unsupervised convolutional auto-encoder for interest point detection and feature description. arXiv preprint arXiv:2001.01354.
Yokozuka, M., Koide, K., Oishi, S., Banno, A., 2021. Litamin2: Ultra light lidar-based slam using geometric approximation applied with kl-divergence. arXiv preprint arXiv:2103.00784.
Yoon, D., Tang, T., Barfoot, T., 2019. Mapless online detection of dynamic objects in 3d lidar, in: 2019 16th Conference on Computer and Robot Vision (CRV), IEEE. pp. 113–120.
Zhang, J., Singh, S., 2017. Low-drift and real-time lidar odometry and mapping. Autonomous Robots 41, 401–416.
Zhao, S., Fang, Z., Li, H., Scherer, S., 2019. A robust laser-inertial odometry and mapping method for large-scale highway environments, in: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE. pp. 1285–1292.
Zhou, B., He, Y., Qian, K., Ma, X., Li, X., 2021. S4-slam: A real-time 3d lidar slam system for ground/watersurface multi-scene outdoor applications. Autonomous Robots 45, 77–98.
Zong, W., Li, G., Li, M., Wang, L., Li, S., 2018. A survey of laser scan matching methods. Chinese Optics 11, 914–930.
Figure 8: Feature extraction results in different scenes. (a) Open road; (b) City avenue; (c) Long straight tunnel; (d) Roadside green belt. (plane, reflector, edge and raw scan points). Objects in the real-world scenes and their counterparts in laser scans are circled by boxes (reflector features, edge features, some special areas).
Figure 9: Relative error plots. (a) Relative translation error curves; (b) Relative rotation error curves. (NDT of HDL-Graph-SLAM, feature-based registration approach of LOAM, the proposed intensity-based registration approach).
Figure 10: DOR examples for a single frame of laser scan. (a) Seq.07. Vehicles crossing the intersection when the data collection vehicle stops and waits for the traffic light (top); The cyclist traveling in the opposite direction when the data collection vehicle driving along the road (bottom). (b) Seq.10. Followers behind the data collection vehicle as it travels down the highway at high speed (top); Vehicles driving in the opposite direction and in front of the data collection vehicle when it slows down (bottom). (Facade, ground, edge and dynamics for points true positive, false positive and true negative for dynamic segmentation boxes.)
Figure 11: The comparison between local maps of LOAM and InTEn-LOAM. (a) Map at the intersection (b) Map at the busy road. In each subfigure, the top represents the map of LOAM w/o DOR, while the bottom represents the map of InTEn-LOAM w/ DOR.
Figure 12: The average RTE and RRE of InTEn-LOAM over fixed lengths. (a) RTE; (b) RRE.
Figure 13: Constructed points maps with details and estimated trajectories. (a), (c), (e) maps of Seq.00, 01, and 05; (b), (d), (f) trajectories of Seq.00, 01, and 05 (groundtruths and InTEn-LOAM)
Figure 14: Cumulative distributions of absolute state errors and estimated trajectories. (a) Cumulative distributions of the absolute positioning errors; (b) Cumulative distributions of the absolute rotational errors; (c) Estimated trajectories. (InTEn-LOAM, LOAM, HDL-Graph-SLAM, groundtruth)
Figure 15: The average RTE and RRE of LO systems over fixed lengths. (a) RTE; (b) RRE.
Figure 16: InTEn-LOAM’s map result on urban scenario (KITTI seq.06): (a) overview, (b) map in detail of circled areas, (c) reference map comparison.

Figure 17: InTEn-LOAM’s map result on country scenario (KITTI seq.10): (a) overview, (b) map in detail of circled areas, (c) reference map comparison.
Figure 18: LO systems’ map results on autonomous driving field dataset in the tunnel region. (a) InTEn-LOAM, (b) LOAM, (c) HDL-Graph-SLAM, (d) MULLS.
Figure 19: InTEn-LOAM’s map result on autonomous driving field dataset. (a) the constructed point cloud map, (b) local remote sensing image and estimated trajectory.