Dynamic Object-aware LiDAR Odometry Aided by Joint Weightings Estimation in Urban Areas

Feng Huang, Weisong Wen, Jiachen Zhang*, Chaoqun Wang and Li-Ta Hsu

Abstract—**Dynamic object detection from point clouds has been widely studied in recent years to achieve accurate and robust LiDAR odometry for autonomous driving. Satisfactory accuracy can be achieved by dynamic object detection from point clouds has been widely studied in recent years to achieve accurate and robust LiDAR odometry for autonomous driving. Satisfactory accuracy can be achieved by detecting and removing the object points in the urban environment. However, it is still not clear how dynamic objects numerically affect the performance of LiDAR odometry. In addition, the existing solutions tended to directly remove the LiDAR features belonging to the dynamic object, which can lead to the degradation of the geometry constraints of the surrounding features. This paper aims to give answers to these problems by evaluating the effects of dynamic objects as well as reweighting both dynamic objects and static objects. Three factors affecting the performance of LiDAR odometry in highly dynamic scenarios, including the *number*, *geometry distribution*, and *velocity of the dynamic objects*, are first extensively studied using generated scenarios by leveraging real data. Instead of brutely removing the dynamic features, this paper proposes to adaptively assign weightings to the dynamic features. Then both the dynamic and static features are employed to estimate the LiDAR odometry. The effectiveness of the proposed method is verified using *UrbanNav* and *nuScenes* datasets that include numerous dynamic and static objects. To benefit the community, the implementation of the dynamic vehicle simulator and the code for the proposed method are both open-sourced[1].**

***Index Terms*—3D LiDAR; Weighting estimation; Dynamic objects; Urban areas; Autonomous driving**

I. Introduction

LiDAR odometry is challenged by dynamic objects in urban areas: Simultaneous localization and mapping (SLAM) [1-3] is fundamental for most autonomous systems. The three-dimensional (3D) light detection and ranging (LiDAR) provides accurate and reliable 3D point clouds of the surroundings. Thus it has recently been widely utilized to provide position and map solutions for autonomous systems [4-7]. However, most existing LiDAR-based odometry [8, 9] methods rely on the assumption that the environment is static, where they perform well with rich geometric information provided by static measurements. Unfortunately, in urban canyons, the odometry estimation suffers from significant degradation [10, 11] due to excessively dynamic objects. The trailing smears, produced by the accumulation of point clouds from the dynamic objects, pollute the static point cloud and lead to incorrect data associations [12, 13]. Such degradation of several popular LiDAR odometry methods resulting from dynamic objects is detailly reviewed in our previous work [14]. In summary, resistance against dynamic objects is essential for accurate and reliable LiDAR odometry in urban canyons.

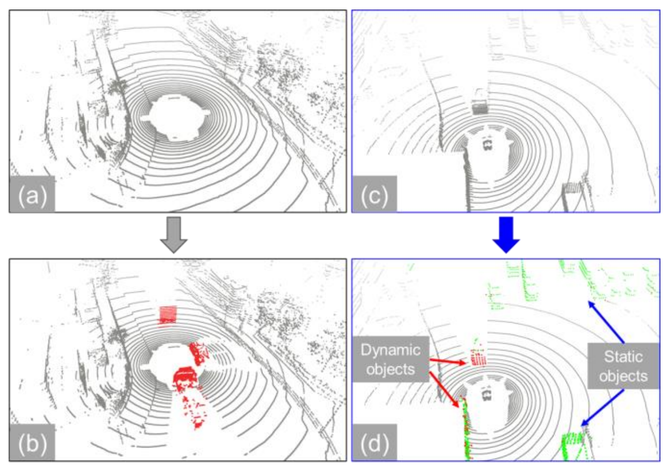

Fig. 1. Illustration of the contributions of this work. LiDAR points in a 3D static environment are marked in grey. (a) represent the scanned cloud in the static environment; (b) The red cloud represents the dynamic vehicles produced by the proposed vehicle simulator. It is integrated with (a) for investigating the impacts of dynamic objects on the degeneration of LiDAR odometry; (c) the grey cloud represents the simulated point cloud with dynamic objects (d) The proposed joint weighting optimization for LiDAR odometry with dynamic object awareness. There are both healthy features and ill features on static or dynamic objects simulated based on (b). Healthy ones assigned with high weight are marked in green. The unhealthy ones assigned with low weights are marked in red.

Direct removal of dynamic objects can lead to LiDAR feature geometry degradation: Lots of effort has been made to improve the LiDAR odometry confronting dynamic objects [15-17]. The most common way is that only static features are remained and used for the odometry estimation, with dynamic features detected and removed. Those existing work has two key drawbacks: (1) In highly dynamic scenarios with numerous

Feng Huang, Weisong Wen, Jiachen Zhang and L.T. Hsu are with the Hong Kong Polytechnic University Shenzhen Research Institute, China, and also with the Hong Kong Polytechnic University, Hong Kong (correspondence e-mail: zjczjc@tju.edu.cn).

Chaoqun Wang is with Shandong University, Shandong, China.

[1] https://github.com/DarrenWong/code_for_dynaLO

moving objects, the remained static features after the removal of excessive dynamic features provide insufficient constraints for the odometry. The odometry degeneration is consequently caused as shown in our previous work [18]. (2) The deep neural network (DNN) [19] or conventional parameter-based methods [20] capable of dynamic object detection is required to remove them so that the odometry is free from dynamic measurements. Therefore, the performance improvement of the odometry suffering from dynamic objects is dependent on accurate detection methods. Unfortunately, there is no effective remedy for false detections which make the odometry even worse. To fill these two gaps, this paper proposes a dynamic object-aware LiDAR odometry method that effectively utilizes features from both dynamic and static objects, providing sufficient constraints even in highly dynamic scenes. Fig. 1 demonstrates the contributions of our proposed pipeline, which are summarized as follows:

(1) *Numerical analysis of the impacts of dynamic objects on LiDAR odometry*: This paper investigates the impacts of dynamic objects on the degeneration of LiDAR odometry. Three key factors are taken into consideration including the number, the geometric distribution, and the velocity of the dynamic objects. For extensiveness and flexibility, the investigation is performed by adding simulated dynamic objects into real data, which could be custom-tailored by the researchers. The implementation kit[1] is open-sourced to benefit the research society.

(2) *Adaptive weighting estimation of LiDAR features from dynamic objects*: This paper proposes an adaptive weighting-based LiDAR odometry method. Contributions from available features are evaluated comprehensively so that both static and dynamic features can be reasonably utilized. The unhealthy features are automatically assigned with low weights while healthy ones with high weights. The false exclusion of feature subsets caused by the false detection of dynamic objects is thus avoided.

The rest of the paper is structured as follows. The related works are presented in Section II. An overview of the proposed method is followed in Section III. Section IV elucidates the degeneracy factors induced by dynamic objects, as well as their corresponding scene generation by adding simulated objects. The proposed LiDAR odometry with adaptive weighting is presented in Section V. Numerous experiments were conducted to verify the effectiveness of the proposed method in Section VI. Finally, the conclusions and future perspectives are presented in Section VII.

II. Related Work

A. Degeneracy Analysis Caused by Dynamic Objects

To evaluate the influence of dynamic objects on the localization and mapping problems, numerous literatures [21, 22] were presented for visual-inertial odometry (VIO). In [23], it is proved that the VIO can be degenerated by a large number of dynamic objects in urban scenarios. The work [21] investigated the degeneration of the VIO referring to the pixel-based percentage of dynamic objects in highly dynamic environments using a simulated dataset. Another team [24] explored the performance degeneration of RGB-D SLAM in dynamic environments and proposed motion removal to improve VIO. Such evaluation works for LiDAR-based odometry is less in comparison. A recent study [25] reviewed several LiDAR inertial fusion algorithms and evaluated their performance in highly dynamic environments. The work argues that integration with inertial units can alleviate the degeneration of LiDAR odometry arising from dynamic objects. Our previous research [14] showed that dynamic objects degenerated the performance of several popular LiDAR odometry. However, no further investigation is performed on the relationship between the degeneration and the detailed status (such as the density, the velocity, etc.) of the dynamic objects.

In short, these methods mentioned above proved the presence of dynamic objects has impacts on localization and mapping problems. However, there are still limitations on the quantitative information of how the dynamic objects affect the state estimation in the real world.

B. LiDAR Odometry with Resistance to Dynamic Objects

To mitigate the degeneration caused by dynamic objects in LiDAR odometry, a random sampling consensus (RANSAC) based method [20] was proposed to exclude dynamic objects that are assumed to provide false feature associations. However, the RANSAC is essentially a maximum consensus method that rejects the minority. In highly dynamic scenarios, RANSAC tends to fail due to the dynamic measurements account for a large proportion. The Euclidean clustering method [26] was adopted to classify and exclude the dynamic vehicles. However, manual adjustment of the artificial threshold for classification is required. Moreover, dynamic objects beyond the few predefined categories couldn't be recognized.

Change detection [27] was proposed to detect the dynamic objects in point clouds leveraging the inconsistency check of multi-epoch LiDAR data. Yoon *et al.* [28] exploited the discrepancies between the query scan and the reference scan to identify the potential dynamic points. In other words, the method aimed to identify the inconsistency between the point clouds from the consecutive frames. Another work in [17] proposed to exclude dynamic objects from the point cloud based on the voxel ray casting-based method. However, it is a time-consuming task as it needs to traverse the voxel grid. To alleviate the computational load, a range image-based visibility check was proposed in [29] to remove the dynamic points on the map directly. However, this method requires a good initial guess of the pose to be estimated which is not readily available. In recent years, a large body of deep learning-based algorithms are incorporated into LiDAR odometry methods to detect dynamic objects, the removal of which is performed before the odometry estimation [30]. LO-Net [31] deployed a mask prediction network to minimize the loss of LiDAR points that contain dynamic objects. Recent work [15] formulates the LiDAR measurements into range images. Point-wise semantic labels indicating dynamic objects are then produced based on

RangeNet++ [32]. Our previous work [33] showed an improvement in accuracy by proposing a coarse-to-fine LiDAR odometry with the PV-RCNN [34]. An unsupervised dynamic awareness LiDAR odometry method was proposed in [16] where manual labeling and pre-training are avoided. The dynamic objects were labeled automatically by an occupancy grid-based method. Then the features located on the dynamic objects are excluded from the odometry estimation. However, in highly dynamic urban areas, excessive exclusion of the LiDAR features arising from the dynamic objects can lead to the geometry degradation of LiDAR constraints. *Is it possible to also make use of the detected dynamic features instead of direct exclusion*? Instead of directly excluding the features from dynamic objects, the outlier mitigation methods are investigated [35], aiming at reducing the large impact of outliers in the optimization problem. Recent work in [36] combines the graduated-non-convexity (GNC) and robust estimation to resist outliers. The non-convexity induced by the robust kernel is tackled through GNC. However, these methods relied on the selection of the kernel parameters of those robust estimation. Meanwhile, a good initial guess is required. In short, dynamic features awareness for static feature selection is essential for accurate and robust LiDAR odometry. It is still an open question in urban canyons to reliably model and mitigate the impacts of dynamic features.

III. Overview of the Proposed Method

Before diving into the details of the proposed method, notations in this paper are introduced first. Matrices are defined in bold uppercase. Vectors are defined in bold lowercase. Variable scalars are defined as lowercase italic. Constant scalars are defined as lowercase letters. The following symbols are defined and followed by the rest of the paper,

a) The LiDAR body frame at timestamp k is represented as $\{.\}^{L_k}$, fixed at the center of the onboard LiDAR sensor.
b) The world frame is represented as $\{.\}^{W}$, which is aligned with the initial position of the vehicle. For example, $\mathbf{T}_{L_k}^{W} \in SE(3)$ is denoted as the LiDAR pose at timestamp k in the world frame.
c) The pose of the i-th dynamic object and static object is denoted as $\mathbf{T}_{d,i}^{\{.\}}$ and $\mathbf{T}_{s,i}^{\{.\}}$. The translation part of $\mathbf{T}_{d,i}^{\{.\}}$ and $\mathbf{T}_{s,i}^{\{.\}}$ is defined as $\mathbf{p}_{d,i}^{\{.\}}$ and $\mathbf{p}_{s,i}^{\{.\}}$, respectively. Dynamic object body frame is defined as $\{.\}^{c}$, which is fixed at the axis origin.
d) The LiDAR point cloud collected at timestamp k is represented as $\boldsymbol{\mathcal{P}}^{L_k}$, which is the point set of $\{\mathbf{p}_1^{L_k}, \mathbf{p}_2^{L_k} \dots \mathbf{p}_n^{L_k}\}$. Similarly, the dynamic objects, static objects and aggregated cloud in the LiDAR frame at timestamp k are represented as $\boldsymbol{\mathcal{P}}_d^{L_k}$, $\boldsymbol{\mathcal{P}}_s^{L_k}$ and $\boldsymbol{\mathcal{P}}_a^{L_k}$, respectively.

An overview of the proposed method is illustrated in Fig. 2. The input of the system is the LiDAR point cloud, the object database, and the object pose information. The output of the system is the pose estimation of the LiDAR odometry. The pipeline mainly contains two parts: (1) the scene composition, which aggerates $\boldsymbol{\mathcal{P}}_d^{L_k}$ and $\boldsymbol{\mathcal{P}}_s^{L_k}$ with $\boldsymbol{\mathcal{P}}^{L_k}$ to produce $\boldsymbol{\mathcal{P}}_a^{L_k}$. It serves to investigate the impacts of dynamic objects on the degeneration of the LiDAR odometry. Specifically, the dynamic object from the object database could be aggregated with $\boldsymbol{\mathcal{P}}^{L_k}$ arbitrarily, as its pose and velocity in the LiDAR body frame are customized by the user. The static object from the object database is anchored in $\{.\}^{W}$ by the user. It could be aggregated with $\boldsymbol{\mathcal{P}}^{L_k}$ via transformation from $\{.\}^{W}$ to $\{.\}^{L_k}$. The transformation matrix $\mathbf{T}_{W}^{L_k}$ is provided by NovAtel SPAN-CPT. The (2) part is dynamic object-aware LiDAR odometry (LO) aided by weighting optimization. Planar features are extracted from $\boldsymbol{\mathcal{P}}_a^{L_k}$ and denoted as $\boldsymbol{\mathcal{F}}_a^{L_k}$. A local map is constructed by planar features from several historical frames [37]. Each feature in the current scan is associated with a planar patch in the map. The total association cost is optimized to produce the pose estimation. The conventional LO method is implemented in this way [37]. Note that the weightings of the associations provided by features on dynamic objects are jointly optimized in our proposed method. The large residuals induced by ill-conditioned associations on dynamic objects will be down-weighted and minimized iteratively. The superiority of the proposed LO enhanced by the weighting optimization scheme in comparison with conventional LO is evaluated using the composed scene elaborated in the last paragraph.

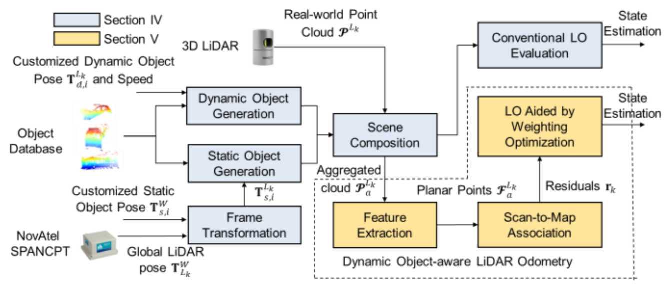

Fig. 2. Overview of the proposed method.

IV. LiDAR Odometry Degeneracy Induced By Dynamic Objects

This section defines the three characters to represent the impacts of the dynamic objects on the performance of LiDAR odometry, namely (1) the number of dynamic points that have a direct impact on the data association of LiDAR odometry. In particular, the denser number of dynamic objects is expected to have stronger impacts on the LO. (2) the geometric distribution of dynamic points, which affects the constraint of the state optimization. In particular, it is expected that the uniformly distributed dynamic objects would lead to less impact on the LO. (3) the relative velocity of the dynamic objects, which explores moving dynamic points related to the accuracy of LiDAR odometry. In particular, the larger velocity of the dynamic objects tends to lead to larger inconsistency in the feature association, thus degrading the performance of LO.

The listed three factors dominate the degree of degeneracy of the LiDAR odometry induced by the dynamic objects. The generation of the high dynamic scenario for further investigation of the LiDAR odometry performance is introduced in the following section.

A. Number of Dynamic Objects

The number of dynamic objects is represented by the percentage of the dynamic features among the features utilized for the LiDAR odometry estimation as follows,

$$D_1 = \frac{N_{d,k}}{N_k} \tag{1}$$

in which N_k denotes the total number of features utilized to estimate $\mathbf{T}_{L_k}^W$. $N_{d,k}$ denotes the number of features located on dynamic objects among N_{L_k}. Intuitively, the larger number of dynamic objects induces more incorrect data associations thus heavier odometry degeneration.

B. Geometric Distribution of Dynamic Objects

In the global navigation satellite system (GNSS), the dilution of precision (DOP) is proposed to quantify the error in the receiver positional measurement propagated from the satellite geometry description [38]. Inspired by DOP, the covariance of the error in translation estimation induced by the geometric distribution of the dynamic object is represented as follows,

$$\mathbf{Q}_d = \left(\mathbf{J}_{\mathbf{d,t}}^T \mathbf{J}_{\mathbf{d,t}}\right)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_y\sigma_x & \sigma_z\sigma_x \\ \sigma_y\sigma_x & \sigma_y^2 & \sigma_z\sigma_y \\ \sigma_z\sigma_x & \sigma_z\sigma_y & \sigma_z^2 \end{bmatrix} \tag{2}$$

in which $\sigma_{(\cdot)}\sigma_{(\cdot)}$, with $(\cdot)$ representing x, y, or z, denotes the covariance on the specific dimensions of the translation estimation. $\mathbf{J}_{d,t}$ denotes the Jacobian matrix of the association costs produced by the dynamic planar features referring to translation part $\mathbf{t}$ of the estimated pose**,**

$$\mathbf{J}_{\mathbf{d,t}} = [\mathbf{J}_{d1,\mathbf{t}} \quad \mathbf{J}_{d2,\mathbf{t}} \quad \cdots \quad \mathbf{J}_{dm,\mathbf{t}}]^T \in \mathbb{R}^{m\times 3} \tag{3}$$

For i-th planar point in the dynamic features, the Jacobian matrix of residuals can be computed by [37],

$$\mathbf{J}_{di,\mathbf{t}} = \frac{\partial \mathbf{r}_{di}}{\partial \mathbf{t}^T} = \mathbf{n}_i^T \in \mathbb{R}^{1\times 3} \tag{4}$$

where $\mathbf{r}_{di}$ is the association residual. $\mathbf{n}_i$ represent the normal vector [37] of the planar patch corresponding to the i-th planar feature. As clock bias [38] is not involved in LiDAR odometry, the geometric distribution of the dynamic objects is inherited from the position DOP (PDOP) [38] and represented as follows,

$$D_2 = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2} \tag{5}$$

Specifically, well-distributed dynamic features, such as those evenly distributed, provide well-conditioned constraints for pose estimation in comparison to those badly distributed ones. In this case, the covariance of the translation error is supposed to be small thus producing small D_2. In other words, smaller D_2 reflects the better geometric distribution of dynamics, causing slighter LiDAR odometry degeneracy.

C. Relative Velocity of Dynamic Objects

The velocity of the dynamic objects relative to the LiDAR is defined as follows,

$$D_3 = \frac{\left\|\mathbf{p}_d^{L_k} - \mathbf{p}_d^{L_{k+1}}\right\|}{\Delta t} \tag{6}$$

where the $\mathbf{p}_d^{L_k} \in \mathbb{R}^3$ and $\mathbf{p}_d^{L_{k+1}} \in \mathbb{R}^3$ represent the position of dynamic features in the L_k frame and the L_{k+1} frame respectively. $\|\cdot\|$ denotes the Euclidean norm of the vector. Δt denotes the time between the two frames. Within a moderate region, the larger relative velocity of the dynamic objects degrades the LiDAR odometry heavier as much false association would be introduced. However, beyond this region, when the relative velocity is significantly large, the dynamic object is lagging far behind or moving far over the LiDAR respectively. In this case, the dynamic objects couldn't be observed in two consecutive frames. Therefore, no degeneracy is induced as no association is provided for the LiDAR odometry by such dynamic objects.

D. Dynamic Scene Generation

To generate a realistic dynamic scene containing numerous vehicles, we initially delve into the LiDAR optical model. The raw reflection data from LiDAR can be categorized into several types [39], including solid targets (such as cars and pedestrians), soft targets (rain, fog, dust, or snow) and noise-related measurements. Our previous work [40] has assessed the performance of LiDAR positioning under adverse weather conditions. In this paper, our objective is to investigate the impact of dynamic objects on the degradation of LiDAR odometry. In comparison to adverse weather and noise factors, the influence of dynamic vehicles on LiDAR primarily manifests as a backscattering effect. Fig. 3 illustrates the backscattering effect on the transmission of LiDAR signals hitting vehicles and buildings.

The pulse propagation in the presence of scatter particles is discussed in [39]. The impulse response of the LiDAR estimates the transformation of the range and intensity of the original point cloud and the modified cloud that contains dynamic objects. In clear weather conditions, the impulse response function [39] of LiDAR data can be derived as follows,

$$H(R) = \frac{\tau(R)}{R^2} \cdot \rho_0 \delta(R - R_0) \tag{7}$$

R is the range of the reflection point while $\tau(R)$ is the ratio of the area illuminated by the transmitter and the area observed by the receiver ($\tau(R)$ normally equals 1 for larger than 2-meter observation). ρ_0 denotes the reflectivity of the target. For the metal component of the vehicle, ρ_0 can be larger than $10/\pi$. $\delta(*)$ is the Dirac delta function and R_0 is the range of the hard target. $\delta(*)$ equals zero everywhere except at $R = R_0$.

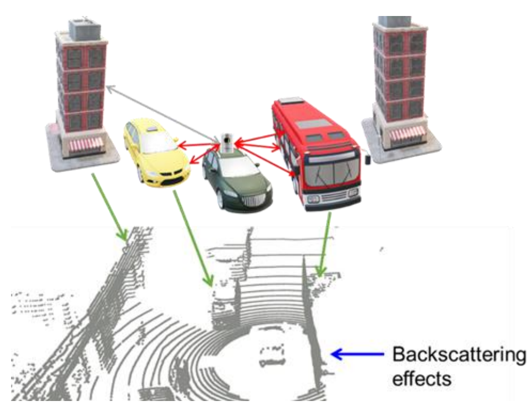

Fig. 3. Illustration of the hard target (vehicles, buildings) effect on LiDAR in a challenging dynamic area (top: the graph view; bottom: the corresponding real point cloud). The double arrow marked in red and grey indicates the backscattering of dynamic vehicles and buildings on LiDAR, respectively. The arrow marked in green illustrates the LiDAR points from the vehicles and buildings while the arrow marked in blue illustrates the backscattering effects on the bus. Most of the LiDAR points are hit on the dynamic object which introduces a large number of outliers.

To verify the impact of the listed factors on the performance of LiDAR odometry in dynamic scenarios, we leverage real-world dynamic objects and static environments to construct realistic scenes on top of the LiDAR model in Eq. (7). In particular, we build a collection of dynamic objects using data collected from the vehicle and exploit the ground truth information to generate dynamic objects as well as static objects.

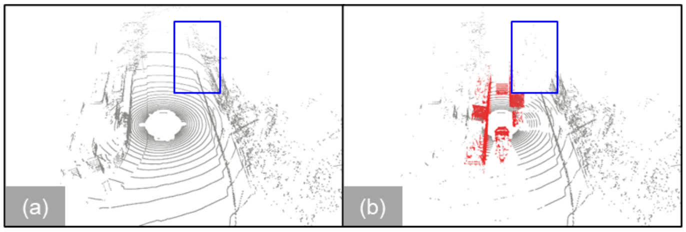

Fig. 4. Demonstration of the simulated LiDAR point cloud. LiDAR points in a 3D static environment are marked in grey, while dynamic object points are labeled in red. (a) The point cloud in the static urban environment. (b) Highly dynamic urban area with various real dynamic objects. The blue areas in (a) and (b) represent the points blocked by the dynamic vehicle that is not scanable in (b) using the proposed simulator.

We argue that we can create more realistic dynamic scenarios by exploiting the dynamic objects in actual data. To generate such a simulation, numerous dynamic objects, including cars, trucks, and double-decker buses, have been collected in different urban areas using 3D LiDAR. Then, highly dynamic scenarios can be constructed by transforming the $i-th$ dynamic object $\boldsymbol{\mathcal{P}}_{d,i}^{c}$ to the specific locations $\mathbf{T}_{c,i}^{L_k}$ relating to the ego vehicle.

$$\boldsymbol{\mathcal{P}}_{d,i}^{L_k} = \mathbf{T}_{c,i}^{L_k}\boldsymbol{\mathcal{P}}_{d,i}^{c} \tag{8}$$

$\boldsymbol{\mathcal{P}}_{d,i}^{L_k}$ indicates the i-th dynamic object in the LiDAR frame. Fig. 4 presents the composition of the dynamic objects and the static environment (shown in Fig. 4 (a)). Note that the points blocked by the objects in Fig. 4 (b) are missing as the vehicle object is a hard target [39] for the LiDAR scanning system in which the LiDAR plus is reflected once it hits the object. Besides, a point cloud containing both dynamic and static objects is required to evaluate the rejecting scheme of dynamic-aware LiDAR odometry. The ground truth information provided by our vehicle platform is exploited to transform the pre-defined static object $\boldsymbol{\mathcal{P}}_{s,i}^{w}$ to the LiDAR local frame $\boldsymbol{\mathcal{P}}_{s,i}^{L_k}$,

$$\boldsymbol{\mathcal{P}}_{s,i}^{L_k} = (\mathbf{T}_{L_k}^{W})^{-1}\boldsymbol{\mathcal{P}}_{s,i}^{w} \tag{9}$$

$\mathbf{T}_{L_k}^{W}$ represents the ego vehicle pose at timestamp k in the world frame. The detail of the dynamic scenario generation is shown in Algorithm 1.

The generated $\boldsymbol{\mathcal{P}}_{a}^{L_k}$, which is to be used for LiDAR odometry with dynamic object reweighting in the next section. Fig. 5 showcased the real-world point cloud and the simulated point cloud generated using the proposed LiDAR simulator in two different datasets. The simulated point cloud that contained objects, matched closely with the real data.

Algorithm 1: Dynamic scenario generation

Inputs: Original LiDAR point cloud at timestamp k as $\boldsymbol{\mathcal{P}}^{L_k}$, objects and their pose parameters

Outputs: The simulated point cloud $\boldsymbol{\mathcal{P}}_{a}^{L_k}$

Step 1: Initialize $\boldsymbol{\mathcal{P}}_{a}^{L_k} \leftarrow$ empty.

Step 2: Object creation

- **Step 2-1**: Transform the dynamic object points $\boldsymbol{\mathcal{P}}_{d,i}^{c}=\{\mathbf{p}_{d,1}^{c}, \mathbf{p}_{d,2}^{c}, \dots, \mathbf{p}_{d,n}^{c}\}$ which docked at the origin into the LiDAR local frame $\{\mathbf{p}_{d,1}^{L_k}, \mathbf{p}_{obj,2}^{L_k}, \dots, \mathbf{p}_{obj,n}^{L_k}\}$ using the object transformation via (7).
- **Step 2-2**: Transform the static object points $\{\mathbf{p}_{s,1}^{W}, \mathbf{p}_{s,2}^{W}, \dots, \mathbf{p}_{s,n}^{W}\}$ into LiDAR local frame $\{\mathbf{p}_{s,1}^{L_k}, \mathbf{p}_{s,2}^{L_k}, \dots, \mathbf{p}_{s,n}^{L_k}\}$ using the ego-vehicle ground truth via Eq. (9).
- **Step 2-3**: Filtering the points in the $\boldsymbol{\mathcal{P}}^{L_k}$ as $\widetilde{\boldsymbol{\mathcal{P}}^{L_k}}$ if the line segments between LiDAR origin and the points intersect with any objects based on LiDAR ray casting [39, 41].
- **Step 2-4**: Composing the dynamic object points $\boldsymbol{\mathcal{P}}_{d}^{L_k}$ and static points $\boldsymbol{\mathcal{P}}_{s}^{L_k}$ over the $\widetilde{\boldsymbol{\mathcal{P}}^{L_k}}$

$$\boldsymbol{\mathcal{P}}_{a}^{L_k} = \widetilde{\boldsymbol{\mathcal{P}}^{L_k}} + \boldsymbol{\mathcal{P}}_{d}^{L_k} + \boldsymbol{\mathcal{P}}_{s}^{L_k}$$

Step 3: Finish the algorithm and output the aggregated point cloud as $\boldsymbol{\mathcal{P}}_{a}^{L_k}$.

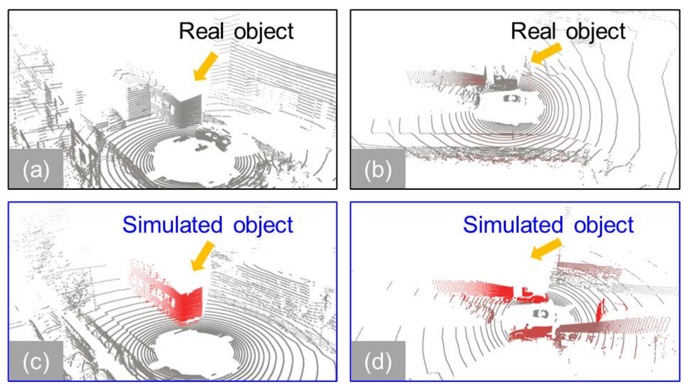

Fig. 5. Illustration of the simulated point cloud (c) and (d) that contained objects that matched the real data in (a) and (b).

V. LIDAR ODOMETRY AIDED BY WEIGHTING OPTIMIZATION

The conventional feature-wise LiDAR odometry [42] can be expressed as,

$$\mathbf{T}_{L_k}^{W}=\underset{\mathbf{T}_{L_k}^{W}}{argmin}\frac{1}{2}\{\sum_{i=1}^{m}\left\|\mathbf{r}_{i}^{L_k}(\mathbf{T}_{L_k}^{W})\right\|^2\} \tag{10}$$

m indicates the number of associated features while $\mathbf{r}_{i}^{L_k}$ represent the residuals from edges and planar features. This method works well in a constrained environment without many dynamic objects. Unfortunately, the conventional formulation suffers from the incorrect data association produced by excessive dynamic features. Thus the pose estimation will be significantly degraded [16]. To address these problems, this paper proposes a novel term $\rho(*)$ inspired by the switchable constraints [43],

$$\rho\left(\omega_i, \mathbf{r}_{ds,i}^{L_k}\right) = {\omega_i}^2\left\|\mathbf{r}_{ds,i}^{L_k}\right\|^2 + k^2(1-\omega_i)^2 \quad (11)$$

where $\mathbf{r}_{ds,i}^{L_k}$ represents the residual from both dynamic and static LiDAR points. $\omega_i \in [0,1]$ denotes the switchable variables associated with the dynamic residuals while k stands for the const parameter for the switch prior constraints term. The switchable constraints can include or exclude the LiDAR features through the associated weighting factor ω_i. The switchable prior constraints are required to anchor switchable variables at their initialization. All the initial values of switchable variables are set to 1, which means that all the residuals are accepted as static features in the beginning. Then, the LiDAR point is determined as a static point if ω_i close to 1 or classified as a dynamic point if ω_i near 0. Recall (9)-(10), the adaptively weighting LiDAR odometry can be written as,

$$\mathbf{T}_{L_k}^{W} = \underset{\mathbf{T}_{L_k}^{W}, \mathcal{W}}{argmin}\, \frac{1}{2}\{\textstyle\sum_{i=1}^{m}\left\|\mathbf{r}_{ev,i}^{L_k}(\mathbf{T}_{L_k}^{W})\right\|^2 + \sum_{i=1}^{N_{ds,k}}\rho\left(\omega_i, \mathbf{r}_{ds,i}^{L_k}\right)\} \quad (12)$$

where $\mathcal{W}$ denotes the sets of weighting ω_i of each residual. $\mathbf{r}_{ev,i}^{L_k}$ indicates the residual from environmental points $\boldsymbol{\mathcal{P}}_{ev,i}^{L_k}$. $N_{ds,k}$ indicates the number of associated LiDAR points from the simulated objects. The process of the joint weightings estimation is shown in Algorithm 2. To minimize the cost function (11), the first-order gradient of $\rho(*)$ relative to ω_i is defined as,

$$\frac{\partial\rho}{\partial\omega_i} = 2\omega_i\left\|\mathbf{r}_{ds,i}^{L_k}\right\|^2 - 2k^2(1-\omega_i) \quad (13)$$

The optimal ω_i can be achieved where the gradient is zero,

$$\omega_i = \frac{k^2}{\left\|\mathbf{r}_{ds,i}^{L_k}\right\|^2 + k^2} \quad (14)$$

The optimal $\hat{\rho}$ can be computed by substituting (13) into (10),

$$\hat{\rho}\left(\omega_i, \mathbf{r}_{ds,i}^{L_k}\right) = \frac{k^2\left\|\mathbf{r}_{ds,i}^{L_k}\right\|^2}{k^2 + \left\|\mathbf{r}_{ds,i}^{L_k}\right\|^2} \quad (15)$$

Algorithm 2: LiDAR odometry aided by Joint Weightings Estimation

<u>Inputs</u>: Aggregated point cloud $\boldsymbol{\mathcal{P}}_{a}^{L_k}$, environment point cloud $\boldsymbol{\mathcal{P}}_{ev}^{L_k}$ and object point cloud $\boldsymbol{\mathcal{P}}_{ds}^{L_k}$ which aggregated point cloud $\boldsymbol{\mathcal{P}}_{a}^{L_k} = \{\boldsymbol{\mathcal{P}}_{ev}^{L_k}, \boldsymbol{\mathcal{P}}_{ds}^{L_k}\}$

<u>Outputs</u>: The optimal pose estimation $\mathbf{T}_{L_k}^{W}$

<u>Step 1</u>: Extract features $\boldsymbol{\mathcal{F}}_{a}^{L_k}$ from $\boldsymbol{\mathcal{P}}_{a}^{L_k}$.

<u>Step 2</u>: Obtain the residuals $\mathbf{r}_{ev}^{L_k}$ and $\mathbf{r}_{ds}^{L_k}$ from environment features and object features, respectively.

<u>Step 3</u>: **For each $\mathbf{r}_{ev,i}^{L_k} \in \mathbf{r}_{ev}^{L_k}$**

A squared residual function is applied via conventional state estimation Eq. (10): $\left\|\mathbf{r}_{ds,i}^{L_k}\right\|^2$

End for

<u>Step 4</u>: **For each $\mathbf{r}_{ds,i}^{L_k} \in \mathbf{r}_{ds}^{L_k}$**

A switchable constraint [43] is applied based on Eq. (11): $\rho\left(\omega_i, \mathbf{r}_{ds,i}^{L_k}\right) = {\omega_i}^2\left\|\mathbf{r}_{ds,i}^{L_k}\right\|^2 + k^2(1-\omega_i)^2$

End for

<u>Step 5</u>: Minimize the cost function to obtain optimal $\mathbf{T}_{L_k}^{W}$

$$\mathbf{T}_{L_k}^{W} \leftarrow \underset{\mathbf{T}_{L_k}^{W}, \mathcal{W}}{argmin}\, \frac{1}{2}\{\textstyle\sum_{i=1}^{m}\left\|\mathbf{r}_{ev,i}^{L_k}(\mathbf{T}_{L_k}^{W})\right\|^2 + \sum_{i=1}^{N_{ds,k}}\rho\left(\omega_i, \mathbf{r}_{ds,i}^{L_k}\right)\}$$

Figs. 6 and 7 show the adaptive loss corresponding to residual $\mathbf{r}_{ds,i}^{L_k}$ from [-1.0, 1.0] with various ω_i and k. As shown in Fig. 6, the loss is a convex function except the $\omega_i = 0$ when the gradient is zero. Therefore, the weighting of the dynamic object which is associated with a large residual can be optimized to near 0. Fig. 7 represents that the increase of k will increase the loss, which means a large impact on Eq. (12) compared to the residual from static environments $\mathbf{r}_{i}^{L_k}$. Determining a proper value for the prior k is important for the reweighting scheme and we will explain it in the experimental validation in detail.

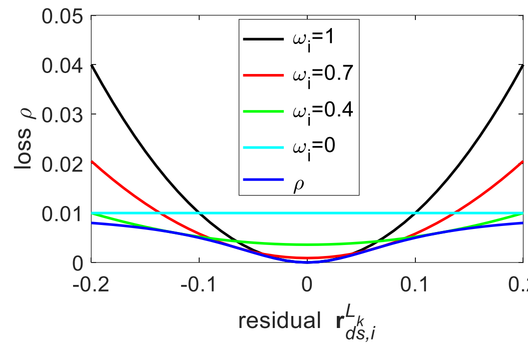

Fig. 6. $\rho(*)$ with respect to different ω_i if $k = 0.1$. $\hat{\rho}$ represents the optimal loss.

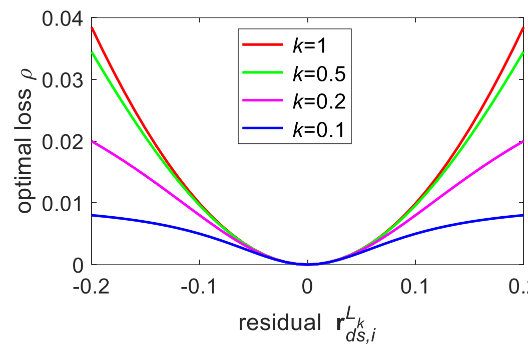

Fig. 7. The optimal $\hat{\rho}$ with respect to different k.

VI. Experiment Results

A. Experiment Setup

Sensor setups: The performances of the proposed method are evaluated using *UrbanNav* [44] and the *nuScenes* [45] datasets.

Our recently open-sourced *UrbanNav* [44] dataset contains the data from the Global navigation satellite system (GNSS), inertial navigation system (INS), cameras, and LiDARs equipped on the vehicle platform, as shown in Fig. 8 (a). In addition, NovAtel SPAN-CPT [46] integrates a fiber optics gyroscope (FOG) and GNSS-RTK to provide the ground truth (GT) positioning. Furthermore, the measurements from SPAN-CPT are further tightly coupled using the NovAtel Inertial Explorer [46] software to provide centimeter-level positioning accuracy. In this experiment, the point cloud is captured by the Velodyne HDL-32E [47] at a frequency of 10 Hz. The ground truth of the LiDAR pose is obtained from the NovAtel SPAN-CPT [46] at a frequency of 100 Hz. The extrinsic spatial transformation matrix between the two sensors has been calibrated in advance. The timestamps of the two-kind sensor measurements are synchronized with GPS time using pulse per second (PPS) signals [48].

To increase the diversity of the evaluation, the *nuScenes* [45] are collected in Boston and Singapore under various weather conditions. The sensor setup for the *nuScenes* collection is shown in Fig. 9 (a). The 32-line LiDAR is utilized to provide a point cloud at a frequency of 20 Hz. The ground truth, with an error of less than 10 centimeters, is obtained through sensor fusion and HD map. Besides, it provides 3D bounding boxes and semantic-level annotation in keyframes (image, LiDAR and Radar) for all objects.

Experimental scenes: To verify the effectiveness of the proposed method, we conducted numerous experiments in typical urban areas in Hong Kong. The urban environments are collected in urban areas without dynamic objects, as shown in Fig. 8 (b). We first conducted the experiment in which the static scenes were composed with different percentages of dynamic objects (shown in Fig. 10). Then, we performed another experiment in which the same amount of dynamic objects was integrated into different directions under the LiDAR body frame (shown in Fig. 11). Besides, we performed the third experiment investigating the degeneracy caused by the speed of dynamic objects (shown in Fig. 12). In the end, we evaluated the performance of the proposed method using the integrated scenes of *UrbanNav* and *nuScenes*, including both dynamic and static objects (shown in Figs. 13-14).

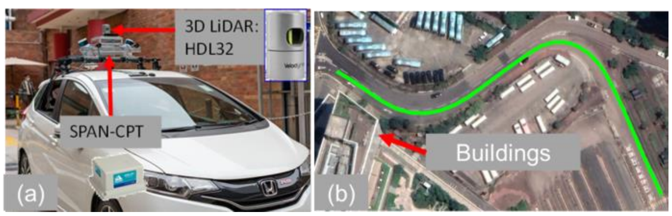

Fig. 8. (a) Setup for the sensor platform in *UrbanNav*; (b) The evaluated static scene and its ground truth trajectory.

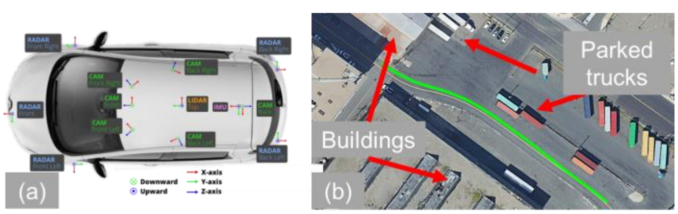

Fig. 9. Sensor setup for the *nuScenes* dataset collection; (b) Bird view of a typical scene in the *nuScenes* dataset. Parked trucks and buildings are pointed out. The ground truth of the data collection vehicle is illustrated as a green curve.

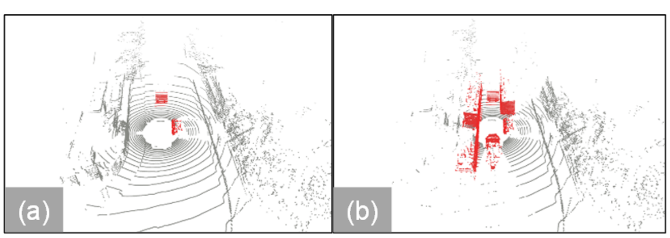

Fig. 10. Illustration of the simulated point cloud with different percentages of dynamic objects: (a) 10% dynamic objects; (b) 60% dynamic objects.

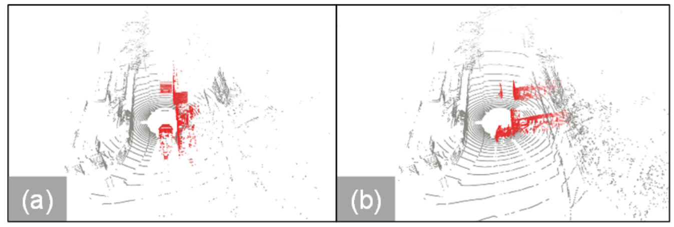

Fig. 11. Illustration of the simulated point cloud (a) and (b) with vertical and horizontal dynamic points distribution.

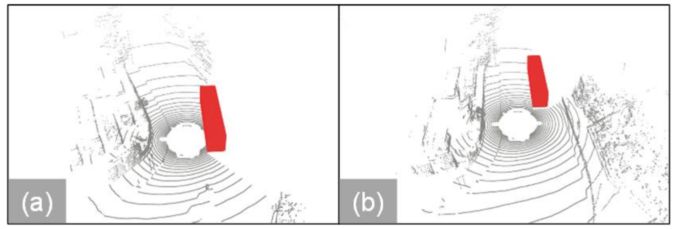

Fig. 12. Illustration of the simulated point cloud (a) and (b) with dynamic objects at a relative speed of 5m/s.

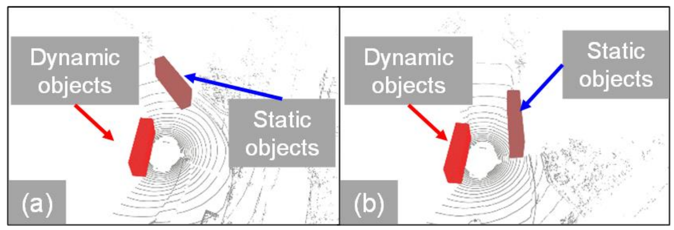

Fig. 13. Illustration of the integrated point cloud with dynamic objects for evaluation in *UrbanNav*. (a) the scanned point cloud at timestamp t; (b) the scanned point cloud at timestamp $t+1$.

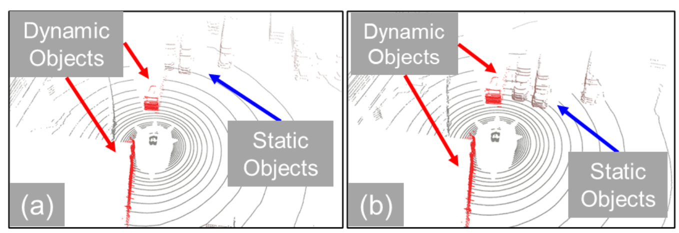

Fig. 14. Illustration of the integrated point cloud with dynamic objects for evaluation in *nuScenes* (the data contains parked bus). (a) the scanned point cloud at timestamp t; (b) the scanned point cloud at timestamp $t+1$.

Evaluation metrics: We first analyzed the performance of LiDAR odometry with different levels of dynamic objects. Second, the performance of the proposed method was evaluated via the relative pose error (RPE) to investigate the local consistency of the trajectory with standard practice [49].

To verify the effectiveness of the proposed method in this paper, we evaluate the following method,

(1) **LO** [42]: The conventional LiDAR Odometry.
(2) **T-LOAM** [50]: The state-of-the-art LiDAR odometry with outlier resistance. The truncated least squares method is adopted to mitigate the effect of dynamic objects.
(3) **LO-R** [33]: The LiDAR Odometry aided dynamic object removal in our previous work.
(4) **LO-RW**: The proposed LiDAR Odometry with dynamic

object reweighting. We use $k = 0.1$ for the penalty const parameter in Eq. (11) experimentally.

B. Verification of LiDAR Odometry Degeneracy Induced by Dynamic Objects

1) Verification of the Number of Dynamic Objects

Table 1 presents the performance evaluation of LiDAR odometry caused by the number of dynamic objects. The first column in Table 1 indicates the different percentages of the dynamic object based on Eq. (1). The first row in Table 1 shows the positioning error of the static environments (0% dynamic objects). An RMSE of 0.170 meters was obtained with a standard deviation of 0.085 meters. Interestingly, dynamic objects with no more than 40% have fewer effects on the state estimation, according to our experiment results. The positioning error increased significantly to 1.330 meters with a standard deviation of 1.170 meters after the number of dynamic objects reached 50%. It can be shown that the number of dynamic objects affects the performance of LiDAR odometry gradually. The performance of 80% of dynamic objects is much worse than that of the other methods, up to 6.484 meters.

Table 1. Performance of LiDAR odometry in terms of numbers of dynamic objects.

Number of Dynamic Objects	RMSE (m)	MEAN (m)	STD (m)	Max (m)
0%	0.170	0.147	0.085	0.569
10%	0.171	0.148	0.086	0.573
20%	0.181	0.158	0.088	0.572
30%	0.177	0.154	0.087	0.572
40%	0.191	0.167	0.091	0.566
50%	1.330	0.632	1.170	4.318
60%	2.998	1.682	**2.481**	7.146
70%	6.417	6.394	0.545	7.457
80%	**6.484**	**6.459**	0.566	**7.477**

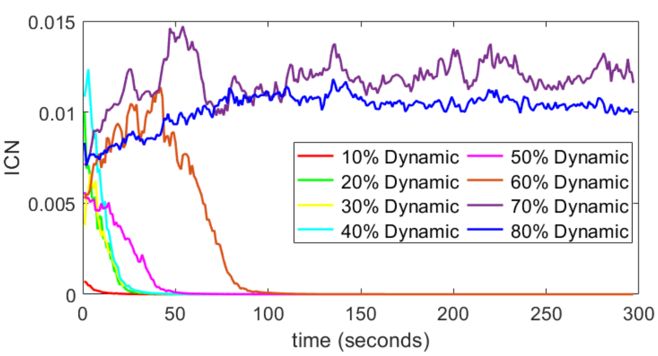

Fig. 15. Illustration of the inverse condition number (ICN) from $\mathbf{J}_\mathbf{d}^T\mathbf{J}_\mathbf{d}$ with different levels of dynamic points.

To evaluate the impacts of the dynamic points on the state estimation, the inverse condition number (ICN) was proposed in [51]. ICN is computed as the inverse of the ratio between maximum eigenvalue and minimum eigenvalue of $\mathbf{J}_\mathbf{d}^T\mathbf{J}_\mathbf{d}$, where $\mathbf{J}_\mathbf{d}$ indicates the Jacobian matrix of the dynamic points referring to the estimated state in this paper. A large ICN indicates dynamic points affect the optimization heavily. Fig. 15 presents the ICN of $\mathbf{J}_\mathbf{d}^T\mathbf{J}_\mathbf{d}$ from different levels of dynamic objects. The constraint from dynamic objects increased if the dynamic objects reached 40%, a similar phenomenon to the positioning error. The dynamic objects (up to 60%) have fewer constraints after 100 seconds because there are some health measurements from several high buildings (shown in Fig. 8 (b)). However, a large ICN can be seen if the dynamic objects are 70% and 80%, which means that excessive outliers play a crucial impact in the final optimization.

2) Verification of Geometric Distribution of dynamic objects

Table 2 presents the performance of the LiDAR odometry with the different geometric distributions of dynamic objects. Based on the results presented in Table 1, where a significant positioning error was observed, we have chosen the 50% dynamic objects as our baseline verification. We then proceeded to relocate these dynamic objects to create varied distributions (shown in Fig. 11). The second column in Table 2 shows the *D2* values that belong to the distributions of dynamic objects using Eq. (5). An RMSE of 1.330 meters was obtained with a standard deviation of 1.170 meters from the baseline, which *D2* equals 0.00357 (50% dynamic objects). With increased *D2*, the positioning error decreased significantly from 1.330 to 0.839 meters with a standard deviation of 0.736 meters. The accuracy is further decreased to 0.357 meters by an increased value of *D2* in Sequence 3. It can be concluded that the more evenly the geometric distribution of dynamic objects, the less accuracy of the LiDAR odometry.

Fig. 16 presents the ICN of $\mathbf{J}_\mathbf{d}^T\mathbf{J}_\mathbf{d}$ from dynamic objects with different geometric distributions. Lower ICN can be observed from the dynamic objects once relaxing the geometric distributions.

Table 2. Performance of LiDAR odometry in terms of different geometric distribution D2 of dynamic objects at 50% dynamic objects.Sequence 1: the baseline scenarios with 50% dynamic objects; Sequence 2: rotate the dynamic objects 90 degrees around the z-axis of the object to generate different distribution; Sequence 3: relocate the objects 2 meters away around the ego-vehicle.

Scenario	D2	RMSE (m)	MEAN (m)	STD (m)
Sequence 1	0.00357	1.330	0.632	1.170
Sequence 2	0.00630	0.839	0.403	0.736
Sequence 3	0.00846	0.357	0.237	0.266

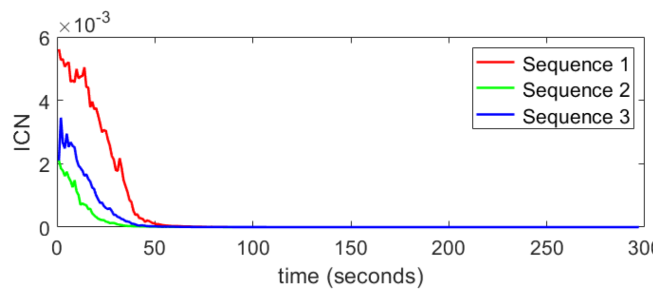

Fig. 16. Illustration of the ICN from $\mathbf{J}_\mathbf{d}^T\mathbf{J}_\mathbf{d}$ with different geometric distributions of dynamic points.

3) Verification of Relative Velocity of Dynamic Objects

Table 3 depicts the accuracy of the LiDAR odometry under different conditions (shown in Fig. 12) of relative velocity. The second column shows the different speeds of the dynamic vehicle, while the third column demonstrates the positioning results of LiDAR odometry. The results of Sequences 1-3 shown in Table 3 indicate that only one moving vehicle has less impact on the state estimation. The positioning error increased to 4.375 meters, with a stand deviation of 3.699 meters after placing one more vehicle on the left with a speed of 5m/s. The error further increased to 8.192 meters in Sequence 5 compared with Sequence 4 with increased speed. Interestingly, the positioning error is extremely reduced to 0.408 meters, with a standard deviation of 0.271 meters. The possible reason for this is that outliers with large movements will be considered as corresponding features by the data association using a k-d tree.

In short, the positioning results are affected by the relative velocity factor, but faster speed might not lead to worse results.

Table 3. Performance of LiDAR odometry in terms of the relative velocity of dynamic objects. D_3indicates the relative speed of the dynamic objects, while right and both represent the placement of the dynamic objects on the right side and both sides.

Scenario	D_3(m/s)	RMSE (m)	MEAN (m)	STD (m)
Sequence 1	5, right	0.198	0.167	0.106
Sequence 2	10, right	0.181	0.153	0.097
Sequence 3	20, right	0.178	0.148	0.098
Sequence 4	5, both	4.375	2.336	3.699
Sequence 5	10, both	8.192	4.688	6.717
Sequence 6	20, both	0.408	0.305	0.271

4) *Analysis of Residuals*

To show further details of three factors denominating the performance of LiDAR odometry, we analyzed the residuals from evaluated scenarios, as shown in Fig. 17. The x-axis denotes the value of residuals while the y-axis denotes the volume associated with residuals in the histogram.

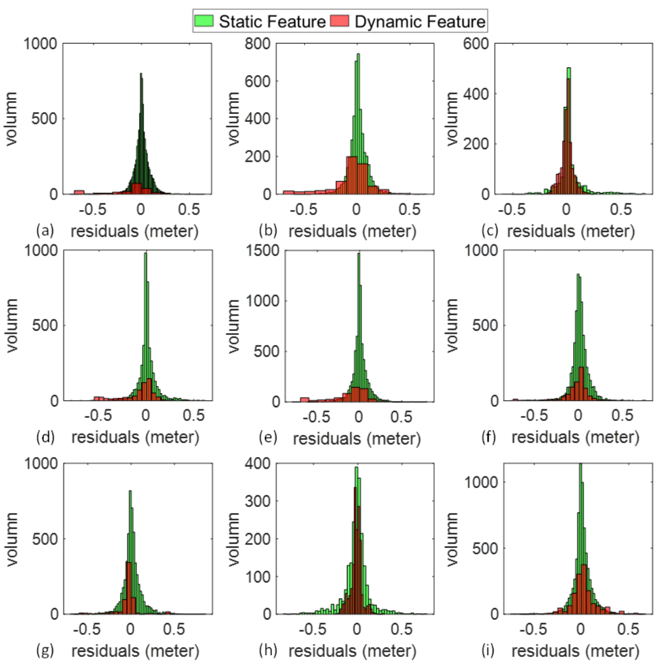

Fig. 17. Illustration of the residual distributions of the compared scenarios on a single epoch. The x-axis denotes the value of residuals. The y-axis represents the volume associated with each bin of the residuals in the histogram. The residual of the static feature is marked in green, while the residual of the dynamic feature is marked in brown. The residuals are presented under scenarios (a) 20% of dynamic objects; (b) 50% of dynamic objects; (c) 80% of dynamic objects; (d) *D2* value equals 0.00357; (e) *D2* value equals 0.00630; (f) *D2* value equals to 0.00846; (g) right site dynamic objects with relative speed 5m/s; (h) both side dynamic objects with relative speed 10m/s; (i) both side dynamic objects with relative speed 20m/s.

Firstly, Fig. 17 (a) shows the residuals from 20% of dynamic objects. Dynamic objects can be classified more easily as they have large residuals. As a result, it has few effects on the accuracy of the LiDAR odometry, which is shown in Table 1. With the increasing dynamic objects, the outlier residual is hard to identify as it also has a lower value as static features (Fig. 17 (b)-(c)), resulting in a large positioning error.

Secondly, Fig. 17 (d)-(f) shows the residuals related to different *D2* values. It can be seen in Fig.17-(e) that the residuals from the dynamic objects become larger as the increase of *D2*. Therefore, the outliers can be mitigated if the distribution of the dynamic objects is less evenly at the same number of dynamic objects.

Thirdly, Fig. 17 (g)-(i) presents the residuals in various dynamic objects with relative speed. Interestingly, dynamic objects with large relative speeds have fewer residuals than dynamic objects. We can conclude that the outliers with rapid movements might not be associated with the corresponding features between the concessive frames.

C. *Verification of Reweighting LiDAR Odometry*

1) *Verification of Proposed Method in the Mid-Dynamic Scene*

To validate the performance of the proposed reweighting method, experiments were conducted under the *mid-dynamic* scene (24% dynamic objects, 30% static objects). Table 4 demonstrates the positioning results using the listed four methods, while Fig. 18 and Fig. 19 represent the positioning error and trajectories of the listed methods. An RMSE of 0.206 meters was obtained using the LO method, with a standard deviation of 0.147 meters. With the help of truncated least squares, the RMSE error of T-LOAM was reduced to 0.096 meters. T-LOAM demonstrated the most accurate among the four methods. The RMSE decreased from 0.206 to 0.147 meters after removing the object points. With the help of the proposed method, both RMSE and standard deviation were reduced to 0.118 meters and 0.033 using the LO-RW method. The proposed LO-RW method shows a better performance, especially under the turning area marked in Fig. 19. We can confirm the effectiveness of the proposed method in the *mid-dynamic* area. To further validate our proposed method, another challenging experiment is evaluated in the next section.

Table 4. Positioning performance of the proposed method in the mid-dynamic scene (24% dynamic points).

Results	LO	T-LOAM	LO-R	LO-RW
RMSE(m)	0.206	0.096	0.147	0.118
MEAN(m)	0.144	0.081	0.142	0.113
STD(m)	0.147	0.053	0.038	0.033

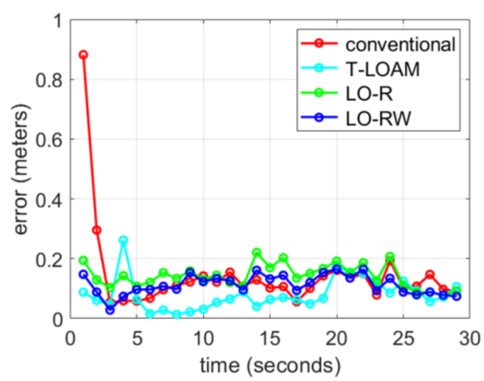

Fig. 18. Positioning error of the four methods. The x-axis and y-axis denote the epoch and error, respectively.

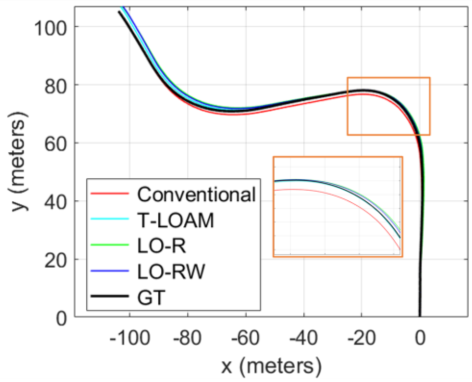

Fig. 19. 2D positioning trajectories of the four methods. The x-axis and y-axis denote the x and y directions, respectively. The black curve represents the ground truth.

2) *Verification of Proposed Method in the High-Dynamic Scene*

To challenge the performance of the proposed method, another experiment was conducted in a harsh scene, which is a highly dynamic environment with 40% moving objects. The results are presented in Table 5 to show the effectiveness of the proposed LiDAR odometry. The T-LOAM fails to optimize the trajectory, as presented in Fig. 21, owing to the incorrect outlier rejection under a large number of dynamic objects. An RMSE of 1.349 meters was obtained using the conventional LO method with a standard deviation of 1.104 meters. The performance was worse than the performance in the *mid-dynamic* scene due to the excessive dynamic objects in this scenario. The error decreased to 0.146 meters after excluding all the objects using LO-R. With the help of object weighting optimization, the error was reduced to 0.110 meters using LO-RW. The improvement in the results shows the effectiveness of the proposed method for LiDAR odometry in highly dynamic environments.

Fig. 20 and Fig. 21 show the positioning error and trajectories, respectively. The more accurate trajectory is estimated using LO-RW with the help of dynamic object reweighting. After applying the object reweighting, the positioning performance of LiDAR odometry is improved significantly in the dynamic scene.

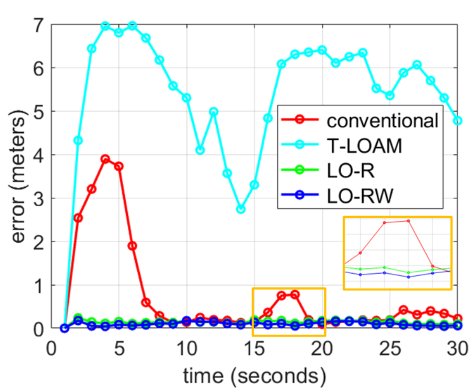

Fig. 20. Positioning error of the four methods in the *high-dynamic* scene. The x-axis and y-axis denote the epoch and error, respectively. The orange box is the area with dynamic and static objects.

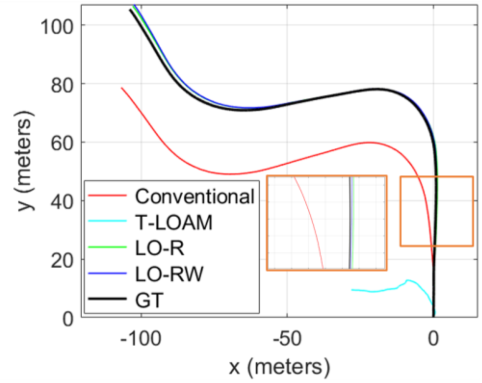

Fig. 21. Positioning trajectories of the four methods in the *high-dynamic* scene. The x-axis and y-axis denote the x and y direction, respectively. The black curve represents the ground truth. The orange box is the area with dynamic and static objects.

Table 5. Performance of the proposed method in the high-dynamic scene (40% dynamic points).

Results	LO	T-LOAM	LO-R	LO-RW
RMSE (m)	1.340	5.660	0.146	0.110
MEAN(m)	0.759	5.555	0.141	0.102
STD (m)	1.104	1.084	0.035	0.041

3) *Verification of Proposed Method using nuScenes Dataset*

To evaluate the performance of the proposed method against adverse conditions, the third experiment is conducted combined with simulated dynamic objects from *nuScenes* [45] Dataset. Table 6 presents the positioning results. An RMSE of 2.592 meters was obtained using the conventional LO method with a standard deviation of 1.755 meters. Similar to the high-dynamic scene, the T-LOAM failed to optimize the trajectory thus causing a large error. LO-R performance is worse than LO due to the insufficient constraint after removing all the vehicle points， as shown in Fig. 22 (b). LO-RW achieves the best accuracy with an RMSE of 0.729 m. The superiority is benefited from the reweighting strategy. It’s concluded that the effectiveness of the proposed method is verified in extensive datasets. Fig. 1-(d) represents the final weighting for all the objects in the *nuScenes*. Interestingly, the majority of points from the front side of the dynamic vehicle were assigned to low weight due to inconsistency, while the points from both sites of the dynamic object were assigned to higher weight as the residual of point to surface remained small, although the vehicle is moving forward. In addition, the static features were assigned high weighting, which can positively contribute to the final optimization.

Table 6. Performance of the proposed method in the *nuScenes*.

Results	LO	T-LOAM	LO-R	LO-RW
RMSE (m)	2.592	6.043	5.069	0.729
MEAN(m)	1.907	5.729	4.354	0.446
STD (m)	1.755	1.922	2.595	0.576

The positioning error and trajectories are illustrated in Fig. 23 and Fig. 24, respectively. After applying the proposed object reweighting, the positioning performance of LiDAR odometry is improved significantly in this challenging scene. However, we observed an unexpected error denoted by the orange box in

Fig. 23. It's because the tunnel-like area provides insufficient vertical constraint even without dynamic object removal. Therefore, additional sensors are required to achieve robust state estimation.

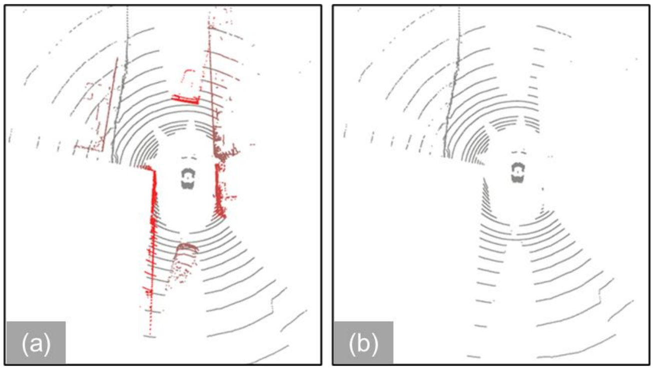

Fig. 22. The example cloud in *nuScenes* (a) with dense objects marked in red; (b) after the objects are removable.

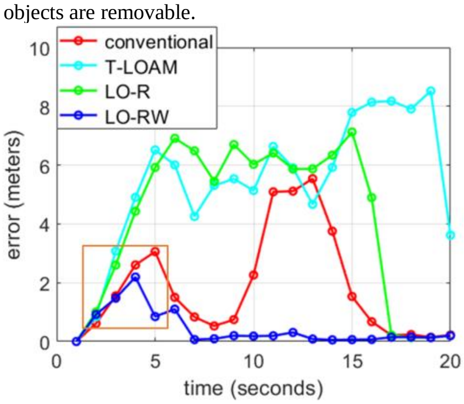

Fig. 23. 3D Positioning error of the four methods in *nuScenes*. The x-axis and y-axis denote the epoch and error, respectively. The orange box is the tunnel-like area (see Fig. 22-a) with dynamic and static objects. It has a negative impact on state estimation.

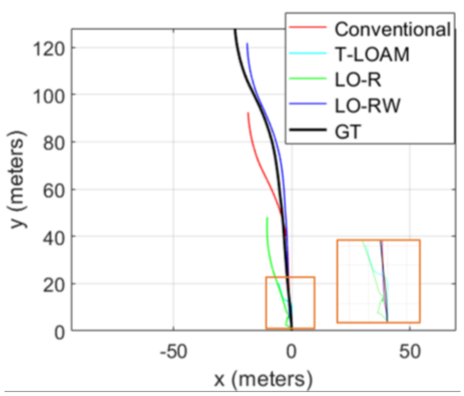

Fig. 24. Positioning trajectories of the four methods in *nuScenes*. The x-axis and y-axis denote the x and y direction, respectively. The black curve represents the ground truth. The orange box is the area with a tunnel-like scene (see Fig. 22-a). It has a negative impact on state estimation.

4) Analysis of Weighting and Residual

As mentioned in Section VI-(A), the const penalty parameter for adaptive weighting is set to 0.1 experimentally. Fig. 25 demonstrates the weighting with different penalty parameters in one single epoch. It can be observed from Fig. 25-(a) that all the weighting for dynamic objects is near 1 if the penalty parameter is set to 1. The final residual of Eq. (11) is highly affected by the penalty term due to the large residual raised by the penalty parameter. With the decrease of the penalty parameter, the inconsistency (car front and rear) of the dynamic objects between concessive epochs are assigned with lower weighting, as expected, as shown in Fig. 25-(b). However, Fig. 25-(c) shows that the weighting for both dynamic objects and static objects is close to 0 if we further reduce the penalty parameter to 0.01. All the objects have less constraint on the state estimation. In other words, the performance of the proposed method is similar to the LO-R method with dynamic object removal.

To examine the residual of the optimization after applying the dynamic object reweighting method, Table 7 demonstrates the residuals from different iterations with the estimated state in terms of the static points and the object points. It can be observed that the residuals of static points and total cost are decreased gradually by increasing the number of iterations. The reweighting scheme can provide a better initial guess for the static feature association. Therefore, the performance of the proposed method can be further improved by coarse-to-fine optimization.

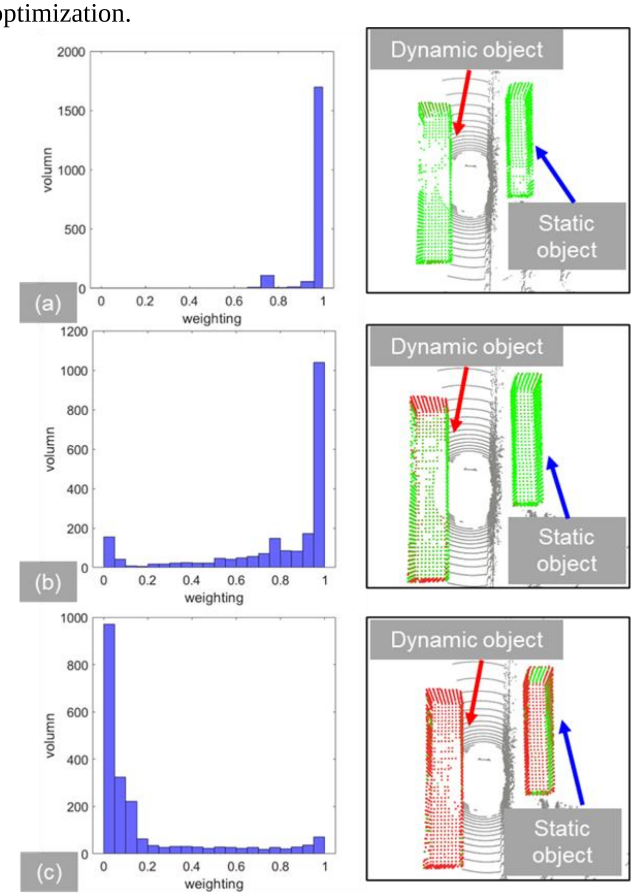

Fig. 25. Illustration of weighting by different penalty parameters of state optimization. (a) penalty parameter $k = 1$; (b) penalty parameter $k = 0.1$; (c) penalty parameter $k = 0.01$; The lower weighting is indicated by red while higher weighting is marked by green.

5) Analysis of Convexity

To avoid our solution being trapped in a local minimum within the possible solution space, we investigate the convexity of the state estimation using the proposed method. We added a 10/100/1000 meters translation offset to the initial guess of the state after feature association. Table 8 shows the residuals are examined after each optimization using Ceres Solver. It can be observed that the cost can be minimized after server iterations even if the initial guess offset is 1000 meters. We can conclude that our problem is convex within less than 1000 meters.

Table 7. Residual of the proposed method in terms of iteration in one single epoch. Each iteration will use the initial guess from the previous estimated state.

Iteration	Static Residual (m)	Dynamic Residual (m)	Total Residual (m)
1	9.588	0.805	10.393
2	9.216	0.780	9.995
3	8.952	0.809	9.760
4	8.810	0.827	9.637
5	8.729	0.829	9.558
6	8.780	0.835	9.615
7	8.775	0.834	9.609
8	8.775	0.833	9.608

Table 8. Results of cost based on different initial guesses in single epoch.

Offset (m)	Iteration	Cost	Cost Change
10	1	2.715e+05	/
	2	42.508	2.71e+05
	3	22.886	19.622
	4	22.842	0.042
100	1	2.701e+07	/
	2	86.257	2.70e+07
	3	23.790	0.963
	4	22.826	0.029
1000	1	2.684e+09	/
	2	6.574e+05	2.68e+09
	3	259.113	6.57e+05
	4	24.997	234.116
	5	22.679	2.318
	6	22.659	0.020

6) *Analysis of Computational Time Cost*

To investigate the computational efficiency of the proposed method, the computation time cost of the evaluated method is presented in Table 9. We compare the feature extraction, optimization processes and total odometry using our evaluated dataset. In the optimization phase, our proposed LO-RW required an additional 40 milliseconds compared to the LO mainly due to the estimation of more parameters. Overall, the performance of the proposed method can be real-time.

Table 9. The computation time cost of the evaluated method. MD is short for the mid-dynamic scene in Section VI-C (1) while HD is short for the high-dynamic scene in Section VI-C (2). ms short for millisecond. The time of odometry equals feature extraction plus the optimization time.

Results		LO	T-LOAM	LO-R	LO-RW
MD Feature Extraction	MEAN (ms)	14.5	24.4	14.8	15.0
	MAX(ms)	20.6	48.0	22.4	22.1
	STD (ms)	2.0	4.1	2.1	2.1
MD Optimization	MEAN (ms)	72.1	36.6	71.5	111.7
	MAX(ms)	96.9	67.0	100.0	175.2
	STD (ms)	13.3	5.2	13.9	20.1
MD Odometry	MEAN (ms)	86.6	61.0	86.3	126.7
HD Feature Extraction	MEAN (ms)	15.4	24.6	15.5	15.4
	MAX(ms)	22.0	43.0	21.2	21.0
	STD (ms)	1.8	3.7	2.0	2.03
HD Optimization	MEAN (ms)	82.9	48.0	71.8	116.3
	MAX(ms)	111.8	73.0	108.8	196.3
	STD (ms)	17.2	12.0	15.8	26.9
HD Odometry	MEAN (ms)	98.3	72.6	87.3	131.7

VII. Conclusions and Future Work

This paper presents a comprehensive evaluation of how the dynamic objects degenerated the performance of LiDAR odometry. Different from our previous work [33], this study reweights the dynamic object, which improves state estimation both in highly dynamic and adverse scenes. The experimental results show the effectiveness of the proposed method, which overcomes the traditional method and the method with dynamic object removal [33].

In the future, generative artificial intelligence [52] for LiDAR scene generation is an interesting topic for the evaluation of the adverse conditions against LiDAR. Besides, we will investigate the complementary of LiDAR and GNSS [53, 54] in order to reduce the global positioning error for autonomous driving. With the increasing deployment of roadside infrastructure, utilizing the intelligent infrastructure [55] to enhance the performance of LiDAR odometry is a future direction toward fully autonomous driving.

Acknowledgment

This work was supported by the Guangdong Basic and Applied Basic Research Foundation (2021A1515110771) and University Grants Committee of Hong Kong under the scheme Research Impact Fund (R5009- 21). This research was also supported by the Faculty of Engineering, The Hong Kong Polytechnic University under the project “Perception-based GNSS PPP-RTK/LVINS integrated navigation system for unmanned autonomous systems operating in urban canyons.”

References

[1] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving," *IEEE Transactions on Intelligent Vehicles,* vol. 2, no. 3, pp. 194-220, 2017, doi: 10.1109/TIV.2017.2749181.

[2] C. Shu and Y. Luo, "Multi-Modal Feature Constraint Based Tightly Coupled Monocular Visual-LiDAR Odometry and Mapping," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 5, pp. 3384-3393, 2023, doi: 10.1109/TIV.2022.3215141.

[3] A. Chalvatzaras, I. Pratikakis, and A. A. Amanatiadis, "A Survey on Map-Based Localization Techniques for Autonomous Vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1574-1596, 2023, doi: 10.1109/TIV.2022.3192102.

[4] X. Li, S. Teng, B. Liu, X. Dai, X. Na, and F. Y. Wang, "Advanced Scenario Generation for Calibration and Verification of Autonomous Vehicles," *IEEE Transactions on Intelligent Vehicles,* vol. 8, no. 5, pp. 3211-3216, 2023, doi: 10.1109/TIV.2023.3269428.

[5] L. Zhang, W. Wen, L.-T. Hsu, and T. Zhang, "An Improved Inertial Preintegration Model in Factor Graph Optimization for High Accuracy Positioning of Intelligent Vehicles," *IEEE Transactions on Intelligent Vehicles*, 2023.

[6] P. Yan, W. Wen, and L.-T. Hsu, "Integration of Vehicle Dynamic Model and System Identification Model for Extending the Navigation Service Under

Sensor Failures," *IEEE Transactions on Intelligent Vehicles*, 2023.

[7] Y. Wang, C. Zhao, J. Liang, M. Wen, Y. Yue, and D. Wang, "Integrated Localization and Planning for Cruise Control of UGV Platoons in Infrastructure-Free Environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, pp. 10804-10817, 2023, doi: 10.1109/TITS.2023.3283513.

[8] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, 1992, doi: 10.1109/34.121791.

[9] P. Biber and W. Straßer, "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)*, 2003, vol. 3: IEEE, pp. 2743-2748.

[10] W. Wen, L.-T. Hsu, and G. Zhang, "Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong," *Sensors*, vol. 18, no. 11, p. 3928, 2018.

[11] Y. Zhong, F. Huang, J. Zhang, W. Wen, and L.-T. Hsu, "Low-cost solid-state LiDAR/inertial-based localization with prior map for autonomous systems in urban scenarios," *IET Intelligent Transport Systems*, vol. 17, no. 3, pp. 474-486, 2023, doi: https://doi.org/10.1049/itr2.12273.

[12] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett, "Recurrent-OctoMap: Learning state-based map refinement for long-term semantic mapping with 3-D-Lidar data," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3749-3756, 2018.

[13] Z. Zhao, W. Zhang, J. Gu, J. Yang, and K. Huang, "Lidar Mapping Optimization Based on Lightweight Semantic Segmentation," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 3, pp. 353-362, 2019, doi: 10.1109/TIV.2019.2919432.

[14] F. Huang, W. Wen, J. Zhang, and L. T. Hsu, "Point Wise or Feature Wise? A Benchmark Comparison of Publicly Available Lidar Odometry Algorithms in Urban Canyons," *IEEE Intelligent Transportation Systems Magazine*, pp. 2-20, 2022, doi: 10.1109/MITS.2021.3092731.

[15] D. Chen and G. X. J. N. Gao, "Probabilistic graphical fusion of LiDAR, GPS, and 3D building maps for urban UAV navigation," vol. 66, no. 1, pp. 151-168, 2019.

[16] P. Pfreundschuh, H. F. C. Hendrikx, V. Reijgwart, R. Dubé, R. Siegwart, and A. Cramariuc, "Dynamic Object Aware LiDAR SLAM based on Automatic Generation of Training Data," presented at the IEEE International Conference on Robotics and Automation (ICRA), 2021.

[17] J. Schauer and A. Nüchter, "The peopleremover—removing dynamic objects from 3-d point cloud data by traversing a voxel occupancy grid," *IEEE robotics and automation letters*, vol. 3, no. 3, pp. 1679-1686, 2018.

[18] X. Bai, W. Wen, and L.-T. Hsu, "Degeneration-Aware Outlier Mitigation for Visual Inertial Integrated Navigation System in Urban Canyons," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-15, 2021.

[19] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338-4364, 2021, doi: 10.1109/TPAMI.2020.3005434.

[20] W. Tan, H. Liu, Z. Dong, G. Zhang, and H. Bao, "Robust monocular SLAM in dynamic environments," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013: IEEE, pp. 209-218.

[21] K. Minoda, F. Schilling, V. Wüest, D. Floreano, and T. Yairi, "VIODE: A Simulated Dataset to Address the Challenges of Visual-Inertial Odometry in Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1343-1350, 2021, doi: 10.1109/LRA.2021.3058073.

[22] W. Wang *et al.*, "TartanAir: A Dataset to Push the Limits of Visual SLAM," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 24 Oct.-24 Jan. 2021 2020, pp. 4909-4916, doi: 10.1109/IROS45743.2020.9341801.

[23] X. Bai, W. Wen, and L.-T. Hsu, *Performance Analysis of Visual/Inertial Integrated Positioning in Typical Urban Scenarios of Hong Kong*. 2020.

[24] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110-122, 2017.

[25] X. Xu *et al.*, "A review of multi-sensor fusion slam systems based on 3D LIDAR," *Remote Sensing*, vol. 14, no. 12, p. 2835, 2022.

[26] W. Wen, G. Zhang, and L.-T. Hsu, "GNSS NLOS Exclusion Based on Dynamic Object Detection Using LiDAR Point Cloud," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[27] R. Qin, J. Tian, and P. Reinartz, "3D change detection–approaches and applications," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 122, pp. 41-56, 2016.

[28] D. Yoon, T. Tang, and T. Barfoot, "Mapless online detection of dynamic objects in 3d lidar," in *2019 16th Conference on Computer and Robot Vision (CRV)*, 2019: IEEE, pp. 113-120.

[29] G. Kim and A. Kim, "Remove, then Revert: Static Point cloud Map Construction using Multiresolution Range Images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020: IEEE/RSJ.

[30] C. Xu *et al.*, "Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation," in *European Conference on Computer Vision*, 2020: Springer, pp. 1-19.

[31] Q. Li *et al.*, "LO-Net: Deep Real-Time Lidar Odometry," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,

15-20 June 2019, pp. 8465-8474, doi: 10.1109/CVPR.2019.00867.

[32] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019: IEEE, pp. 4213-4220.

[33] F. Huang, D. Shen, W. Wen, J. Zhang, and L.-T. Hsu, "A Coarse-to-Fine LiDAR-Based SLAM with Dynamic Object Removal in Dense Urban Areas," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 3162-3172.

[34] S. Shi *et al.*, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13-19 June 2020 2020, pp. 10526-10535, doi: 10.1109/CVPR42600.2020.01054.

[35] T. Pfeifer, S. Lange, and P. Protzel, "Advancing Mixture Models for Least Squares Optimization," *IEEE Robotics and Automation Letters,* vol. 6, no. 2, pp. 3941-3948, 2021, doi: 10.1109/LRA.2021.3067307.

[36] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection," *IEEE Robotics and Automation Letters,* vol. 5, no. 2, pp. 1127-1134, 2020, doi: 10.1109/LRA.2020.2965893.

[37] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021: IEEE, pp. 4390-4396.

[38] R. B. Langley, "Dilution of precision," *GPS world,* vol. 10, no. 5, pp. 52-59, 1999.

[39] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Advances in Radio Science,* vol. 9, no. B. 2, pp. 49-60, 2011.

[40] J. Chang, R. Hu, F. Huang, D. Xu, and L.-T. Hsu, "LiDAR-based NDT Matching Performance Evaluation for Positioning in Adverse Weather Conditions," *IEEE Sensors Journal,* 2023.

[41] A. Williams, S. Barrus, R. K. Morley, and P. Shirley, "An efficient and robust ray-box intersection algorithm," in *ACM SIGGRAPH 2005 Courses*, 2005, pp. 9-es.

[42] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots,* vol. 41, no. 2, pp. 401-416, 2017.

[43] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph SLAM," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 7-12 Oct. 2012 2012, pp. 1879-1884, doi: 10.1109/IROS.2012.6385590.

[44] L.-T. Hsu *et al.*, "Hong Kong UrbanNav: An Open-Source Multisensory Dataset for Benchmarking Urban Navigation Algorithms," *NAVIGATION: Journal of the Institute of Navigation,* vol. 70, no. 4, p. navi.602, 2023, doi: 10.33012/navi.602.

[45] H. Caesar *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11621-11631.

[46] N. Inc. "Waypoint Inertial Explorer 8.80 Post Processing Software." https://novatel.com/products/waypoint-post-processing-software (accessed Oct 23, 2023).

[47] M. Veľas, M. Španěl, Z. Materna, and A. Herout, "Calibration of rgb camera with velodyne lidar," 2014.

[48] X. Niu, K. Yan, T. Zhang, Q. Zhang, H. Zhang, and J. Liu, "Quality evaluation of the pulse per second (PPS) signals from commercial GNSS receivers," *GPS solutions,* vol. 19, pp. 141-150, 2015.

[49] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012: IEEE, pp. 573-580.

[50] P. Zhou, X. Guo, X. Pei, and C. Chen, "T-loam: truncated least squares lidar-only odometry and mapping in real time," *IEEE Transactions on Geoscience and Remote Sensing,* vol. 60, pp. 1-13, 2021.

[51] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 16-21 May 2016 2016, pp. 809-816, doi: 10.1109/ICRA.2016.7487211.

[52] C. Li *et al.*, "Generative AI meets 3D: A Survey on Text-to-3D in AIGC Era," *arXiv preprint arXiv:2305.06131,* 2023.

[53] J. Zhang, W. Wen, F. Huang, Y. Wang, X. Chen, and L.-T. Hsu, "GNSS-RTK Adaptively Integrated with LiDAR/IMU Odometry for Continuously Global Positioning in Urban Canyons," *Applied Sciences,* vol. 12, no. 10, p. 5193, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/10/5193.

[54] X. Liu, W. Wen, and L. T. Hsu, "GLIO: Tightly-Coupled GNSS/LiDAR/IMU Integration for Continuous and Drift-free State Estimation of Intelligent Vehicles in Urban Areas," *IEEE Transactions on Intelligent Vehicles,* pp. 1-12, 2023, doi: 10.1109/TIV.2023.3323648.

[55] F. Huang, H. Chen, A. Urtay, D. Su, W. Wen, and L.-T. Hsu, "Roadside Infrastructure assisted LiDAR/Inertial-based Mapping for Intelligent Vehicles in Urban Areas," in *2023 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, 24-28 Sept. 2023 2023 IEEE.

Feng Huang received his bachelor's degree from Shenzhen University in Automation in 2014 and his MSc in Electronic Engineering at Hong Kong University of Science and Technology in 2016. He is a Ph.D. student in the Department of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University. His research interests including localization and sensor fusion for autonomous driving.

Weisong Wen (Member, IEEE) received a BEng degree in Mechanical Engineering from Beijing Information Science and Technology University (BISTU), Beijing, China, in 2015, and an MEng degree in Mechanical Engineering from the China Agricultural University, in 2017. After that, he received a PhD degree in Mechanical Engineering from The Hong Kong Polytechnic University (PolyU), in 2020. He was also a visiting PhD student with the Faculty of Engineering, University of California, Berkeley (UC Berkeley) in 2018. Before joining PolyU as an Assistant Professor in 2023, he was a Research Assistant Professor at AAE of PolyU since 2021. He has published 30 SCI papers and 40 conference papers in the field of GNSS (ION GNSS+) and navigation for Robotic systems (IEEE ICRA, IEEE ITSC), such as autonomous driving vehicles. He won the innovation award from TechConnect 2021, the Best Presentation Award from the Institute of Navigation (ION) in 2020, and the First Prize in Hong Kong Section in Qianhai-Guangdong-Macao Youth Innovation and Entrepreneurship Competition in 2019 based on his research achievements in 3D LiDAR aided GNSS positioning for robotics navigation in urban canyons. The developed 3D LiDAR-aided GNSS positioning method has been reported by top magazines such as Inside GNSS and has attracted industry recognition with remarkable knowledge transfer.

Jiachen Zhang is currently a postdoc fellow at the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University. She is working in the IPN Lab with Dr.Li-Ta Hsu and Dr.Weisong Wen. She obtained her bachelor's degree and Ph. D degree in 2016 and 2022 respectively from Tianjin University, China. Her research topics focus on LiDAR SLAM in challenging urban canyons, and safety-certifiable LiDAR localization for life-critical unmanned applications.

Chaoqun Wang received the B.E. degree in automation from Shandong University, Jinan, China, in 2014, and the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, in 2019. During his Ph.D. study, he spent six months at The University of British Columbia, Vancouver, BC, Canada, as a Visiting Scholar. He was a PostDoctoral Fellow with the Department of Electronic Engineering, The Chinese University of Hong Kong, from 2019 to 2020. He is currently a Professor with the Robot Institute, College of Control Science and Engineering, Shandong University. His current research interests include autonomous vehicles, embodied exploration, and path planning.

Li-Ta Hsu received B.S. and Ph.D. degrees in aeronautics and astronautics from National Cheng Kung University, Taiwan, in 2007 and 2013, respectively. He is currently an associate professor with the Department of Aeronautical and Aviation Engineering. The Hong Kong Polytechnic University, before he served as a post-doctoral researcher at the Institute of Industrial Science at the University of Tokyo, Japan. In 2012, he was a visiting scholar at University College London, the U.K. His research interests include GNSS positioning in challenging environments and localization for pedestrians, autonomous driving vehicle, and unmanned aerial vehicle.