Accepted in *NAVIGATION, Journal of the Institute of Navigation*

AGPC-SLAM: Absolute Ground Plane Constrained 3D LiDAR SLAM

Weisong Wen and Li-Ta Hsu*

***Abstract*— 3D LiDAR-based simultaneous localization and mapping (SLAM) is a well-recognized solution for mapping and localization applications. However, the typical 3D LiDAR sensor (e.g. Velodyne HDL-32E) only provides a very limited field of view vertically. As a result, the vertical accuracy of pose estimation is suffered. This paper aims to alleviate this problem by detecting the absolute ground plane to constrain the vertical pose estimation. Different from the conventional relative plane constraint, this paper employs the absolute plane distance to refine the position in the z-axis and the norm vector of the ground plane to constrain the attitude drift. Finally, relative positioning from LiDAR odometry, constraint from ground plane detection, and loop closure are integrated under a proposed factor graph-based 3D LiDAR SLAM framework (AGPC-SLAM). The effectiveness is verified using several datasets collected in scenes of Hong Kong.**

I. Introduction

Accurate mapping and localization (Dill & Uijt de Haag, 2016) are significant for autonomous systems, such as autonomous driving vehicles (ADV) (Huang et al., 2018), and indoor mobile robotics (Hess, Kohler, Rapp, & Andor, 2016). Great efforts have been devoted to achieving accurate simultaneous localization and mapping (SLAM) using 3D light detection and ranging (LiDAR) (Hess et al., 2016) sensor, due to its robustness compared with the vision-based SLAM methods (Qin, Li, & Shen, 2018). Vision-based SLAM based on the passive sensor, the camera, can be sensitive to the illumination and viewpoint change. Conversely, the active sensor, the 3D LiDAR, can provide distance measurements for surrounding environments which is invariant to the illumination. The outstanding robustness and precision make the 3D LiDAR an indispensable sensor for large-scale mapping and localization.

The 3D LiDAR-based SLAM (Koide, Miura, & Menegatti, 2018a; Lin & Zhang, 2020; Shan & Englot, 2018; Wen, Zhang, & Hsu, 2020; Zhang & Singh, 2014; Zhao, Fang, Li, & Scherer, 2019) has been extensively studied in the past decades. In general, the 3D LiDAR SLAM algorithm can be gracefully divided into two parts, the front-end (Grisetti, Kummerle, Stachniss, & Burgard, 2010) and the backend (Grisetti et al., 2010). The front-end focuses on the point cloud registration (Pang et al., 2019; Saarinen, Andreasson, Stoyanov, & Lilienthal, 2013). The backend integrates multiple constraints, such as positioning from the front end, positioning from the global navigation satellite system (GNSS) (Dow, Neilan, & Rizos, 2009; Shetty & Xingxin Gao, 2019), and loop closure (Magnusson, Andreasson, Nuchter, & Lilienthal, 2009b). The accuracy of point cloud registration, which is the major part of the front-end, dominates the performance of the 3D LiDAR SLAM. Therefore, numerous works are studied to improve the point cloud registration process, such as the iterative closest point (ICP) (Kuramachi, Ohsato, Sasaki, & Mizoguchi, 2015), the normal distribution transform (NDT) (Magnusson, Lilienthal, & Duckett, 2007), and the LiDAR odometry and mapping (LOAM) (Zhang & Singh, 2017). The LOAM algorithm obtains the top one accuracy in the KITTI dataset odometry benchmark (Geiger, Lenz, & Urtasun, 2012) until Dec 2020, due to its feature extraction and impressive data association structure. Unlike the point-wise registration methods (e.g. ICP and NDT), the LOAM extracts the meaningful edge and planar features from raw 3D point clouds, leading to lower computational load and decreased sensitivity to the local minimums. Instead of simply relying on finding the transformation between consecutive frames of point clouds (Low, 2004; Pang et al., 2019; Saarinen et al., 2013) via scan-to-scan manner, the LOAM decouples the registration problem into two parts, coarse odometry, and fine mapping. Firstly, the coarse odometry conducts the scan-to-scan matching of the edge and planar features respectively to estimate a coarse relative transformation. Secondly, the innovative fine-mapping process is conducted to map the current frame of point clouds to the continuously generated global map (scan-to-map) based on the initial guess derived by the coarse odometry. The mapping process can help to mitigate the accumulated relative drift from LiDAR odometry in the first step. In short, the LOAM obtains a better performance comparing with the listed ICP and NDT. However, due to the limited field of view (FOV, typically $+10^{\circ} \sim -30^{\circ}$ vertically, $0 \sim 360^{\circ}$ horizontally) for typical 3D LiDAR, the available features in the vertical direction are significantly fewer than the one in the horizontal direction. As a result, inevitably, the 3D LiDAR SLAM can drift in the vertical direction. To find out the major challenge of 3D LiDAR SLAM in urban canyons, we extensively evaluated the performance of 3D LiDAR SLAM in diverse urban canyons (Wen, Hsu, & Zhang, 2018). The results also showed that the vertical drift due to limited FOV of 3D LiDAR was one of the major problems to be solved for the application of 3D LiDAR SLAM in urban canyons.

According to our evaluations (Wen et al., 2018) in numerous urban scenarios, it is shown that the ground surface is usually available for ground vehicles installed with 3D LiDAR. Furthermore, the 3D LiDAR scanning can provide sufficient ground points even in dense traffic scenes, which makes ground surface detection possible. Inspired by this

Weisong Wen and Li-Ta Hsu are with the Hong Kong Polytechnic University, Hong Kong. (corresponding author to provide e-mail: lt.hsu@polyu.edu.hk).

remarkable feature, this paper proposes to detect and apply the absolute ground plane constraint (AGPC) to constrain the state of the vehicle to further mitigate the vertical drift. In other words, the AGPC is employed to improve the geometry of the constraint. To mitigate the overall drift over time, the loop closure detection method is applied to exploit the loop closure constraint. Finally, a factor graph-based 3D LiDAR SLAM framework (AGPC-SLAM) is proposed to fuse relative positioning from LiDAR odometry, constraints from AGPC, and loop closure.

The major contributions of this paper are listed as follows:

1) Proposing to exploit the AGPC to mitigate the vertical drift of 3D LiDAR SLAM.

2) Proposing a general 3D SLAM framework (AGPC-SLAM) is proposed to integrate the constraint of relative positioning from LiDAR odometry, the AGPC, and the constraint from the loop closure. The proposed AGPC-SLAM is a general framework that can easily fuse additional information from sensors such as the GNSS receiver and magnetometers.

3) The proposed method is validated using two challenging large-scale datasets collected in the typical scenes of Hong Kong. We believe that the proposed method can have a positive impact on both the academic and industrial fields.

The remainder of the paper is structured as follows. The related work is reviewed in Section II. Then the overview of the proposed method is introduced in Section III. The detail of the methodology is introduced in Section IV before the experimental evaluation being presented in Section V. The conclusions and future work are given in Section VI.

II. Related Work

To handle this problem, the LeGO-LOAM (Shan & Englot, 2018) proposes to optimize the z-axis related states based on planar features. The drift in the z-axis and pitch angle is slightly mitigated compared with the LOAM algorithm (Zhang & Singh, 2017). However, as the vertical states are estimated relatively concerning the planar points, the drifted error can still accumulate over time with the vertical positioning error reaching more than 10 meters (Shan & Englot, 2018) in the evaluated dataset. Recently, they extend their LeGO-LOAM to the LiDAR/inertial (Shan et al., 2020) integration to mitigate the overall drift where the high-frequency pose estimation from inertial measurement unit (IMU) pre-integration is used as the initial guess of the mapping process. Meanwhile, the motion distortion is handled with the help of the IMU pre-integration. However, the improvement relies on the cost of the IMU. In other words, the additional IMU cannot essentially solve the problem of weak constraint in the vertical direction.

The recent work of the team from the Hong Kong University of Science and Technology (HKUST) (Ye, Chen, & Liu, 2019) proposes to tightly integrate the LiDAR and IMU to mitigate the overall drift. The tight integration scheme can effectively improve the geometry of the constraints arising from the raw point clouds. With the help of inertial measurements, the overall accuracy is improved compared with the LiDAR standalone SLAM. According to their evaluation, the vertical drift is improved compared with the LOAM. However, the performance relies on the quality of the applied IMU, and the tightly coupled integration process introduces a significantly higher computational load. Meanwhile, the problem of vertical drift is still unsolved. Instead of simply based on relative positioning, the work (Zheng, Zhu, Xue, Liu, & Fan, 2019) utilizes the global positioning system (GPS) to mitigate the drift of LiDAR SLAM. Nevertheless, the performance of GPS solutions relies heavily on environmental conditions, and the high-rising buildings in urban can significantly degrade their performance, leading to large positioning errors (Wen, Bai, Kan, & Hsu, 2019; Wen, Zhang, & Hsu, 2019). Similar works were also done in (Chang, Niu, & Liu, 2020; He, Yuan, Zhuang, & Hu, 2020).

The work in (Lin & Zhang, 2020) included the loop closure into the LOAM to further reduce the drift. Unfortunately, the loop closure is hard to be detected due to the distinct vertical drift. Interestingly, the work in (Zuo, Geneva, Lee, Liu, & Huang, 2019) made use of both the camera and LiDAR to derive improved odometry accuracy. The camera can help the LiDAR standalone odometry to survive in a sparse area as the camera can capture texture information. Despite this, our previous work (Bai, Wen, & Hsu, 2020) shows that the visual-based positioning shares a similar drawback of being sensitive to the dynamic objects in urban canyons. In short, the existing work tends to employ additional sensors to improve the overall accuracy of the 3D LiDAR SLAM, which can only partially reduce the speed of the vertical drift (Shan et al., 2020; Ye et al., 2019) or rely on the GNSS positioning accuracy (Chang et al., 2020; He et al., 2020; Zheng et al., 2019).

III. Overview of the Proposed Method

A. Notations and Coordinates

Matrices are denoted as uppercase bold letters. Vectors are denoted as lowercase bold letters. Variable scalars are denoted as lowercase italic letters. Constant scalars are denoted as lowercase letters.

To make the proposed framework clearer, the following notations are defined and followed by the rest of this paper.

a) The local world frame (W$\{X^W, Y^W, Z^W\}$) is fixed to the world at the start point of the vehicle, which can also be fixed to the GNSS absolute frame when GNSS is available.

b) The local base frame (L$\{X^L, Y^L, Z^L\}$) is originated at the starting point of the local LiDAR odometry.

c) The vehicle body frame (B$\{X^B, Y^B, Z^B\}$) is fixed at the center of the 3D LiDAR sensor.

The considered coordinate system is shown in Fig. 1. The ($\mathbf{T}_B^L$) denotes the transformation from the body frame to the local base frame, encoding the information of the LiDAR odometry during the experiment. The transformation ($\mathbf{T}_L^W$) denotes the transformation between the local base frame and the local world frame, which is the drift compensation estimated by the additional constraints (e.g. the loop closure constraints and the AGPC constraint). In other words, the local-based frame is fixed on the local world frame if there are no ground and loop closure constraints. Note that the definition

of the coordinates is referred to in the work in (Mascaro, Teixeira, Hinzmann, Siegwart, & Chli, 2018) which is commonly used in the SLAM field with loop closure constraint.

Therefore, the state of the ego-vehicle at epoch k is encoded by the transformation from the body frame to the local world frame, $\mathbf{T}_{B,k}^{W}$ as follow:

$$\mathbf{T}_{B,k}^{W} = (\mathbf{p}_k^W, \mathbf{q}_k^W)^T \quad (1)$$
$$\text{With } \mathbf{T}_{B,k}^{W} = \boldsymbol{T}_{L,k}^{W}\boldsymbol{T}_{B,k}^{L}$$

where the $\mathbf{p}_k^W$ represents the position and the $\mathbf{q}_k^W$ represents the orientation in the local world frame with quaternion form.

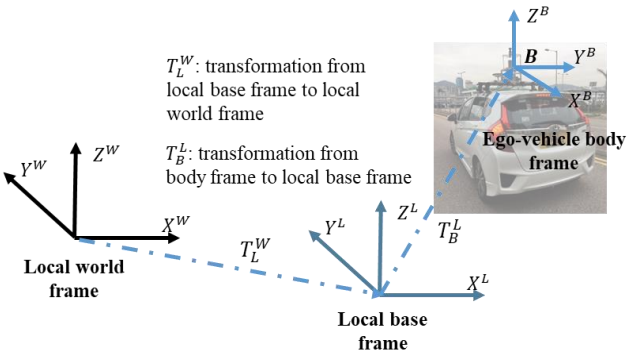

Fig. 1 Overview of the applied coordinate systems. The local world frame is fixed at the starting point of the vehicle with x-right, y-forward, and z-up. The local base frame is fixed at the start point at the beginning. The body frame is fixed at the center of the 3D LiDAR with x-right, y-forward, and z-up.

B. Overview of the Proposed AGPC-SLAM

The overview of the proposed AGPC-SLAM framework is shown in Fig. 2. The input of the AGPC-SLAM is the 3D LiDAR point cloud ($\mathbf{S}_k$, the raw point cloud at epoch k). The outputs of the proposed AGPC-SLAM are the vehicular trajectory and the consistent point cloud map.

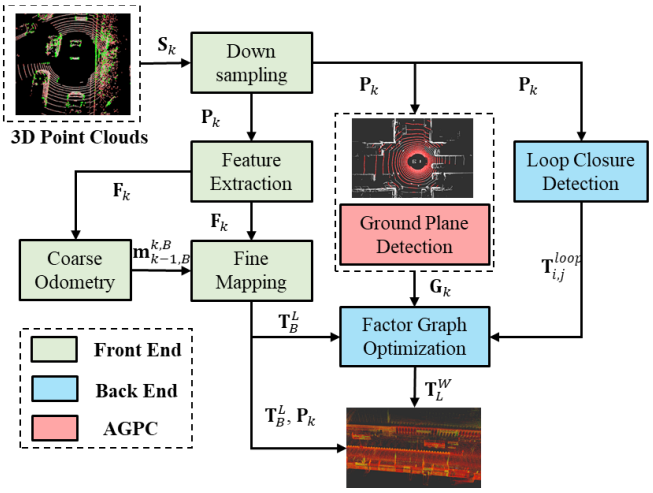

Fig. 2 Overview of the proposed AGPC-SLAM. The input is the 3D point cloud from 3D LiDAR. The outputs include the points map and pose estimation of vehicular trajectory.

The AGPC-SLAM framework can be divided into two major parts, the front-end (the light green shaded boxes in Fig. 2), and the back-end (the light blue shaded boxes in Fig. 2). The light red shaded box denotes the AGPC constraint proposed in this paper. Finally, the constraints from the front-end and the AGPC are integrated using factor graph optimization in the back-end. The details of the major parts are presented in the following Sections.

IV. Methodology

A. Front-End

The performance of the front-end is dominated by LiDAR odometry whose objective is to make the best use of the consecutive point clouds to estimate the relative motion. The LiDAR odometry is implemented based on the LOAM algorithm proposed in (Zhang & Singh, 2017) where three steps are included for the LiDAR odometry; features extraction, coarse odometry, and fine mapping. Although the LiDAR odometry itself is not the contribution of this paper, we still brief present the formulation for completeness.

1) Feature Extraction

Different from the point-wise registration method (e.g. NDT and ICP), the feature definition and extraction proposed in (Zhang & Singh, 2014) explores the representative primitives before the data association process. The input of the feature extraction is the $\mathbf{P}_k\{\mathbf{P}_{k,1}, \mathbf{P}_{k,2}, \ldots, \mathbf{P}_{k,i}, \mathbf{P}_{k,N}\}$. The variable N denotes the number of points inside a frame of point cloud. A point is classified as a planar or an edge point depending on the roughness of its local region. The roughness of the local region is determined as follow (Zhang & Singh, 2017):

$$c_{k,i} = \frac{1}{|\mathbf{S}|\cdot||\mathbf{P}_{k,i}||}\left|\left|\sum_{j\in\mathbf{S}, j\neq i}(\mathbf{P}_{k,j} - \mathbf{P}_{k,i})\right|\right| \quad (2)$$

where the $c_{k,i}$ represents the roughness of the point. The variable $\mathbf{S}$ denotes the small local region near the given point $\mathbf{P}_{k,i}$ and usually, 5 points are involved in the local region. the $\mathbf{P}_{k,j}$ indicates the point belonging to the local region $\mathbf{S}$. If the calculated roughness is larger than a pre-determined threshold (t_c), the point is classified as an edge point. The point with roughness being small than the threshold is classified as a planar point. The output of the feature extraction process is the feature set $\mathbf{F}_k\{\mathbf{F}_k^p, \mathbf{F}_k^E\}$, where the $\mathbf{F}_k^p$ and $\mathbf{F}_k^E$ represent the feature set containing all the planar and edge points set, respectively. Meanwhile, the author (Zhang & Singh, 2017) also proposed careful feature point selection strategies to avoid unreliable points. For example, points that are on the boundary of the occluded regions should not be selected as those points can be unobservable in the coming epochs. Meanwhile, the points that lie on local planar surfaces that are roughly parallel to the LiDAR beams should not be selected as well. We strictly follow these strategies to select feature points from the raw 3D point clouds.

2) Coarse Odometry

Based on the features extracted in the last section, coarse odometry is performed to efficiently estimate the relative motion between consecutive frames of point clouds. The relative motion is calculated by conducting point-to-edge and point-to-plane scan-matching. In short, the objective is to find the corresponding features for points in $\mathbf{F}_k\{\mathbf{F}_k^p, \mathbf{F}_k^E\}$ from the feature points set $\mathbf{F}_{k-1}\{\mathbf{F}_{k-1}^p, \mathbf{F}_{k-1}^E\}$. The detailed steps can be

found in (Zhang & Singh, 2017). We formulate the point cloud registration process as follows using

$$\mathbf{m}_{k-1,B}^{k,B} = PCR(\mathbf{F}_k\{\mathbf{F}_k^p, \mathbf{F}_k^E\}, \mathbf{F}_{k-1}\{\mathbf{F}_{k-1}^p, \mathbf{F}_{k-1}^E\}) \quad (3)$$

where the PCR denotes the point cloud registration function. The output of the point cloud registration process is the coarse relative motion, denoted by $\mathbf{m}_{k-1,B}^{k,B}$. Be noted that the $\mathbf{m}_{k-1,B}^{k,B}$ is the motion between the frame $k-1$ and k. Since the $\mathbf{m}_{k-1,B}^{k,B}$ is estimated based on scan-to-scan matching, the efficiency is guaranteed. Meanwhile, the coarse motion estimation is applied as the initial guess for the fine mapping in the next section.

3) Fine Mapping

To refine the relative motion estimation, the fine-mapping process is applied to refine the $\mathbf{m}_{k-1,B}^{k,B}$. The principle of the mapping process is that the extracted $\mathbf{F}_k\{\mathbf{F}_k^p, \mathbf{F}_k^E\}$ is mapped into the incrementally built map to refine the motion estimate, $\mathbf{m}_{k-1,B}^{k,B}$. Be noted that the map here is generated incrementally. This was one of the major contributions of the work (Zhang & Singh, 2017) that provide impressive performance locally. The output of the mapping process is the $\mathbf{T}_{k,B}^L$. However, the mapping process is conducted at a low frequency circle due to the computational cost. Therefore, the local transform integration is applied to integrate the high frequency but rough relative motion estimate ($\mathbf{m}_{k-1,B}^{k,B}$) and the low frequency but locally accurate motion estimation ($\mathbf{T}_{k,B}^L$). The detail of the mapping process can be found in (Zhang & Singh, 2014).

B. AGPC Detection

This section presents the detection of the AGPC based on the raw 3D point clouds from 3D LiDAR. For a given epoch k, the clouds $\mathbf{P}_k$ usually involves abundant points scanned from the ground surface for ground vehicles installed with 3D LiDAR. For a typical 3D LiDAR sensor installed on the roof of a vehicle, the sensor height relative to the ground surface is approximately known. Numerous works (Choi, Park, Byun, & Yu, 2014; Yang & Förstner, 2010) were done to extract the ground plane based on the raw 3D point clouds. The random sample consensus (RANSAC) is one of the most efficient algorithms for detecting the ground plane parameters from the raw 3D point clouds. In this paper, we define the model parameters of a ground plane as (a) the Euclidean distance from the center of 3D LiDAR to the detected ground surface, (b) and the norm vector of the surface as follow:

$$\mathbf{G}_k = [G_k^x \quad G_k^y \quad G_k^z \quad d_k]^T \quad (4)$$

where the (G_k^x, G_k^y, G_k^z) represents the norm vector and the d_k denotes the Euclidean distance. Therefore, the ground plane detection problem using the RANSAC can be defined as follows:

$$\mathbf{G}_k{}^* = \text{argmin} \sum_{\mathbf{P}_{k,i} \in \mathcal{G}_k} f(\mathbf{G}_k, \mathbf{P}_{k,i}) \quad (5)$$

where the $\mathbf{G}_k{}^*$ denotes the optimal parameters of the ground plane. The variable $\mathcal{G}_k$ denotes a set of the given points on the ground surface. The operator $f(*)$ is employed to evaluate the distance between a given point $\mathbf{P}_{k,i}$ and the plane $\mathbf{G}_k$. Given a point $\mathbf{P}_{k,i} = \{x_{k,i}, y_{k,i}, z_{k,i}\}$ and ground plane $\mathbf{G}_k$, the distance can be calculated as follows:

$$f(\mathbf{G}_k, \mathbf{P}_{k,i}) = \left\| G_k^x x_{k,i} + G_k^y y_{k,i} + G_k^z z_{k,i} - d_k \right\| \quad (6)$$

where the $f(\mathbf{G}_k, \mathbf{P}_{k,i})$ represents the Euclidean distance. The detail of the plane detection is given in the following Algorithm 1. Be noted that the $\mathbf{P}_k$ is filtered before being input the Algorithm by only keeping the points with z-axis values ranging from -2.5 to 2.5 meters. The parameters required in Algorithm 1 include the minimum number of points ($\mathbf{t}_{np}$) required to detect the ground plane, the maximum number of iterations ($\mathbf{t}_{iter}$) allowed in the RANSAC, a threshold value ($\mathbf{t}_{dis}$) of distance for judging whether a point fits the ground plane, a threshold of the number of points ($\mathbf{t}_a$) belonging to the close data indicating that the estimated ground plane model fits well to the points.

Algorithm 1: Ground plane detection using RANSAC

Inputs: Point clouds $\mathbf{P}_k$,
Outputs: ground plane parameters $\mathbf{G}_k{}^*$
Step 1: Initialize $t \leftarrow 0$, $\mathbf{G}_k \leftarrow [0,0,1,2]$ and $\mathbf{E}_t \leftarrow \infty$.
Step 2: while $t >$ $\mathbf{t}_{iter}$

- **Step 2-1**: Select $\mathbf{T}_{np}$ points randomly from $\mathbf{P}_k$ getting $\mathbf{Q}_k$. Fit the $\mathbf{G}_k$ using the selected points ($\mathbf{Q}_k$). Initialize an empty set $\mathbf{A}_k$.
- **Step 2-2**: For every point $\mathbf{R}_{k,i}$ inside a set $\mathbf{R}_k = \{\mathbf{P}_k - \mathbf{Q}_k\}$, evaluate the distance ($e_{k,i}$) between the $\mathbf{R}_{k,i}$ and the plane $\mathbf{G}_k$. If $e_{k,i} < \mathbf{t}_{dis}$, add $\mathbf{R}_{k,i}$ to $\mathbf{A}_k$.
- **Step 2-3**: If the number of points inside $\mathbf{A}_k$ is larger than $\mathbf{t}_a$. Fit the parameters $\mathbf{G}_k$ again based on the $\mathbf{Q}_k$ and $\mathbf{A}_k$. Calculate the fitting error $\mathbf{E}_t$ based on (6). If $\mathbf{E}_t < \mathbf{E}_{t-1}$, assign $\mathbf{G}_k{}^* = \mathbf{G}_k$.

Step 3: Finish the algorithm and the ground plane parameters are estimated as $\mathbf{G}_k{}^*$.

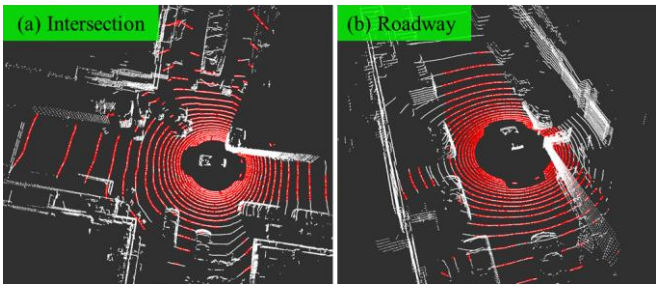

Fig. 3 Illustration of ground detection using Algorithm 1. The white points represent the raw point cloud from 3D LiDAR. The red points denote the detected ground point cloud.

The ground detection results based on Algorithm 1 are illustrated in Fig. 3 which shows the ground detection with data collected from typical intersections and roadways. Fig. 3-(a) shows ground detection results in an intersection case. The white points denote the non-ground points and the red points denote the ground points. For the roadway case in Fig. 3-(b) with the dynamic vehicles being located on both sides, the ground points in the front and back of the ego-vehicle are also effectively detected and annotated with red points.

C. Back-End: Loop Closure Detection

The loop closure (Magnusson, Andreasson, Nuchter, & Lilienthal, 2009a) is an effective approach for reducing the accumulated error over time. The principle of loop closure is to detect the frames of the point cloud with a large percentage of overlap. The accumulated driving distance is denoted as D_k from the first epoch to the current one. The loop closure detection is performed only when the D_k exceeds a threshold ($\mathbf{t}_D$) that tuned heuristically. Then, the loop closure is checked by matching the current keyframe and historical keyframes using the NDT in (Magnusson et al., 2009a). The loop closure is found when the fitness condition (Magnusson et al., 2009a) is smaller than a given threshold ($\mathbf{t}_F$). The output of the loop closure is the relative constraint between the current frame and the historical keyframe as follows.

$$\mathbf{T}_{i,j}^{loop} = (\mathbf{p}_{i,j}^{loop}, \mathbf{q}_{i,j}^{loop})^T \tag{7}$$

where the $\mathbf{T}_{i,j}^{loop}$ denotes the transformation between the two keyframes corresponding to the detected loop closure, frame i, and j. The $\mathbf{p}_{i,j}^{loop}$ denotes the translation and the $\mathbf{q}_{i,j}^{loop}$ represents the rotation in quaternion form.

D. Back-End: Factor Graph-based Optimization

This section presents the factor graph construction based on the previously derived constraints and optimization. To effectively integrate the measurements from the LiDAR odometry (Section IV-A), APGC detection (Section IV-B), and loop closure (Section IV-C), we make use of the state-of-the-art factor graph (Indelman, Williams, Kaess, & Dellaert, 2012) to formulate the sensor fusion as a non-linear optimization problem. The conventional filtering-based sensor fusion such as the extended Kalman filter and its variant (Li, Li, Ji, & Dai, 2015) which exploits the first-order Taylor expansion only once prone to get into sub-optimal. The major advantages of the factor graph are the re-linearization and iteration which can enable the optimized states to approach the optimal iteratively. The graph structure of the proposed AGPC-SLAM is shown in Fig. 4 which includes the listed three kinds of constraints and ego vehicle states. Be noted that the state of the ego-vehicle at epoch k is represented by the $\mathbf{T}_{B,k}^{W}$ based on (1), which encodes the transformation between the body frame to the local world frame.

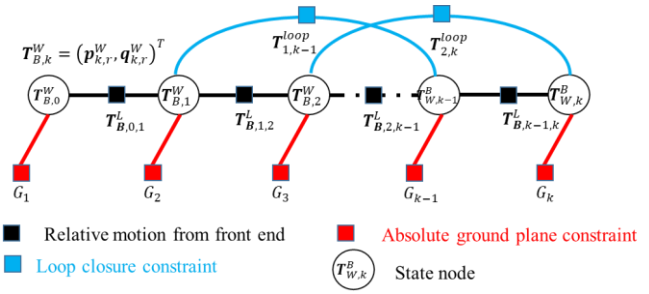

Fig. 4 Graph structure of the proposed AGPC-SLAM. The circle denotes the state of the ego vehicle. the black line represents the constraint derived from the LiDAR odometry. The red line represents the constraint derived from ground plane detection. The light blue line represents the constraint from loop closure.

Following (Indelman et al., 2012), the objective of the factor graph optimization is to minimize the error function between the observations and the states. The error function for the LiDAR odometry derived relative measurements are expressed as:

$$||\mathbf{e}_k^{LiDAR}||_{\mathbf{\Sigma}_k^{LiDAR}}^2 = \left\| (\mathbf{T}_{B,k-1}^{W\ -1}\mathbf{T}_{B,k}^{W}) \ominus (\mathbf{T}_{B,k-1}^{L\ -1}\mathbf{T}_{B,k}^{L}) \right\|_{\mathbf{\Sigma}_k^{LiDAR}}^2 \tag{8}$$

where the $\mathbf{e}_k^{LiDAR}$ represents the error function for the relative motion between node k and $k+1$ for the LiDAR odometry (front end). $\mathbf{\Sigma}_k^{LiDAR}$ denotes the information matrix of the error function which is tuned experimentally. $\mathbf{T}_{B,k}^{W}$ and $\mathbf{T}_{B,k-1}^{W}$ denote the states at epoch k and $k-1$. $\mathbf{T}_{B,k}^{L}$ represents the motion which is derived from LiDAR odometry in Section IV-A. The operator "$\ominus$" is the minus operation of the homogeneous matrix in $SE(3)$.

Similarly, the error function for the loop closure between node i and j is as follows:

$$||\mathbf{e}_{i,j}^{loop}||_{\mathbf{\Sigma}_{i,j}^{loop}}^2 = ||\left(\mathbf{T}_{B,i}^{W\ -1}\mathbf{T}_{B,j}^{W}\right) \ominus \mathbf{T}_{L,i}^{W}\mathbf{T}_{i,j}^{loop}||_{\mathbf{\Sigma}_{i,j}^{loop}}^2 \tag{9}$$

where the $\mathbf{\Sigma}_{i,j}^{loop}$ denotes the information matrix for the loop closure constraint $\mathbf{T}_{i,j}^{loop}$ between nodes i and j.

Regarding the constraint from the ground plane detection, the measurement contains the distance (d_k) from the center of LiDAR to the detected ground surface, and the norm vector (G_k^x, G_k^y, G_k^z) of the ground surface. The d_k does not suffer from drift over time as it is relative to the absolute ground plane. Be noted that this is satisfied under the assumption that the vehicle is driving on almost plane ground.

Given the pose estimation $\mathbf{T}_{B,k}^{W}$ at a given epoch k, the $\mathbf{G}_k$ can be transformed into the local world frame as $\mathbf{G}_k^W$:

$$\mathbf{G}_k^W = [G_k^{w,x'} \quad G_k^{w,y'} \quad G_k^{w,z'} \quad d_k{}']^T \tag{10}$$

where the followings are satisfied.

$$[G_k^{w,x'} \quad G_k^{w,y'} \quad G_k^{w,z'}]^T = \mathbf{q}_k^W [G_k^x \quad G_k^y \quad G_k^z]^T \tag{11}$$

$$d_k{}' = d_k - \mathbf{p}_k^{W\ T}[G_k^{w,x'} \quad G_k^{w,y'} \quad G_k^{w,z'}]^T \tag{12}$$

After transforming the $\mathbf{G}_k$ into the local world frame, this paper applies the minimum parameterization proposed in (Ma, Kerl, Stückler, & Cremers, 2016) $\tau(\mathbf{G}_k^W) = (\theta, \varphi, d)$, where θ, φ, d are the azimuth angle, the elevation angle and the distance with respect to the $\mathbf{G}_k^W$, respectively. The $\tau(\mathbf{G}_k^W)$ can be derived as follows (Koide, Miura, & Menegatti, 2018b; Ma et al., 2016):

$$\tau(\mathbf{G}_k^W) = [\arctan\left(\frac{G_k^{y'}}{G_k^{x'}}\right) \quad \arctan\left(\frac{G_k^{z'}}{\sqrt{(G_k^{x'2} + G_k^{y'2} + G_k^{z'2})}}\right) \quad d_k{}']^T \tag{13}$$

Therefore, the error function for the ground plane constraint can be expressed as follows:

$$||\mathbf{e}_k^{ground}||_{\mathbf{\Sigma}_k^{ground}}^2 = ||\tau(\mathbf{G}_k^W) - \tau(\mathbf{G}_0^W)||_{\mathbf{\Sigma}_k^{ground}}^2 \tag{14}$$

where the $\mathbf{\Sigma}_k^{ground}$ is the information matrix of the $\tau(\mathbf{G}_k^W)$ which is experimentally determined. $\mathbf{G}_0^W = [0 \quad 0 \quad 1 \quad 0]^T$ is the expected prior concerning the ground constraint.

In this case, we formulate three kinds of error functions for constraints. Therefore, the optimal state set $\mathbf{T}_B^W = \{\mathbf{T}_{B,0}^W, \mathbf{T}_{B,1}^W, \mathbf{T}_{B,2}^W, \dots, \mathbf{T}_{B,k}^W, \dots\}$ can be solved as follows:

$$\mathbf{T}_B^{W*} = \text{argmin} \sum_{k=0,\dots K}(||\mathbf{e}_k^{LiDAR}||_{\mathbf{\Sigma}_k^{LiDAR}}^2 + ||\mathbf{e}_{i,j}^{loop}||_{\mathbf{\Sigma}_{i,j}^{loop}}^2 + ||\mathbf{e}_k^{ground}||_{\mathbf{\Sigma}_k^{ground}}^2) \quad (15)$$

where the $\mathbf{T}_B^{W*}$ denotes the state set to be estimated. The variable K denotes the number of epochs involved in the optimization. The operation $\sum_{k=0,\dots K}*$ denotes the summation of all the error functions. The $\mathbf{\Sigma}_k^{LiDAR}$, $\mathbf{\Sigma}_{i,j}^{loop}$ and $\mathbf{\Sigma}_k^{ground}$ denote the information matrix of the listed three kinds of constraint.

E. Global Mapping and Global Transformation Integration

The point clouds map relative to the local world frame can be obtained by registering all the frames of point clouds based on the states set $\mathbf{T}_B^W$ optimized in Section IV-D.

To guarantee real-time performance, the factor graph optimization is conducted at a frequency of 1 Hz. Each time when the optimization is finished, the transformation between the local world frame and the local base frame can be derived as follows:

$$\mathbf{T}_{L,k}^W = \mathbf{T}_{B,k}^W(\mathbf{T}_{B,k}^L)^{-1} \quad (16)$$

Be noted that the $\mathbf{T}_{B,k}^L$ is obtained by tracking the relative motion from LiDAR odometry at a frequency of 10 Hz (the frequency of the raw 3D point clouds). Once the $\mathbf{T}_B^L$ is obtained at epoch l and $l > k$. The $\mathbf{T}_{B,l}^W$ is derived as follows:

$$\mathbf{T}_{B,l}^W = \mathbf{T}_{L,k}^W \mathbf{T}_{B,l}^L \quad (17)$$

therefore, the $\mathbf{T}_B^W$ can be obtained at a frequency of 10 Hz which is significant for the application with real-time performance requirements.

V. Experiment Results and Discussions

To validate the performance of the proposed AGPC-SLAM, we conduct two real experiments in scenes of Hong Kong. The first scene is Nathan Road, which is one of the typical scenes in Hong Kong. Meanwhile, the road is almost flat throughout the test and the driving distance is about 4.2 kilometers. We are fully aware that the proposed method relies on the assumption that the ground is almost flat during the operation. Therefore, we conduct the other experiment validation in the other scene (the driving distance is about 2.1 kilometers) with a partial ramp road to further evaluate how the proposed method can perform.

During the experiment, the 3D LiDAR (Velodyne 32) is used to collect 3D point clouds. Besides, the NovAtel SPAN-CPT, a GNSS (GPS, GLONASS, and Beidou) RTK/INS (fiber-optic gyroscopes, FOG) integrated navigation system was used to provide ground truth of positioning. The gyro bias in-run stability of the FOG is 1 degree per hour, and its random walk is 0.067 degrees per hour. The baseline between the rover and the GNSS base station is within 7 km. All the data were collected and synchronized using a robot operation system (ROS) (Quigley et al., 2009). All the data are post-processed using a desktop (Intel Core i7-9700K CPU, 3.60Ghz) computer. Be noted that the performance of the proposed method is evaluated by aligning the pose from SLAM and the NovAtel SPAN-CPT to the east, north, and up (ENU) coordinate. The extrinsic transformation matrix between the ENU frame and the local world frame is provided by the NovAtel SPAN-CPT. The parameters used in the proposed AGPC-SLAM are given in Table 1.

Table 1. Parameters used during the experimental validation

Para.	t_{np}	t_{dis}	t_a	t_{iter}	t_D
Values	800	0.25	500	1000	20.0
Para.	t_F	$\mathbf{\Sigma}_k^{LiDAR}$	$\mathbf{\Sigma}_{i,j}^{loop}$	$\mathbf{\Sigma}_k^{ground}$	t_c
Values	0.80	$10\,\mathbf{I}_{6\times6}$	$10\,\mathbf{I}_{6\times6}$	$4\,\mathbf{I}_{3\times3}$	0.1

A. Experimental Validation in Scene 1

The experimental setup is shown in Fig. 5. During the evaluation, we compare the proposed AGPC-SLAM framework with the state-of-the-art LeGO-LOAM (Shan & Englot, 2018).

1) Evaluation Metrics

As Fig. 5-(b) shows, the fixed solution is hard to obtain using NovAtel SPAN-CPT throughout the evaluated scene 1. This is caused by low satellite visibility due to the dense and tall buildings. However, we find that the positioning status of the NovAtel SPAN-CPT is healthy in several road intersections (see the A, B, C, D, E, and F in Fig. 5-(c)) where the satellite geometry is better. We carefully check the positioning results of the NovAtel SPAN-CPT in the selected six intersection points and at least the float solution is obtained. Therefore, we propose to evaluate the performance of the proposed method based on the following metrics.

(1) *Ground control point (GCP) error*: evaluating the accuracy at 6 selected GCPs (A~F in Fig. 5). Both absolute translation and rotation errors are evaluated.

(2) *Accumulated error*: evaluating the accumulated error after driving a loop with/without the loop closure constraints. Without loss of generality, the accumulated error metric is also adopted in the KITTI dataset (Geiger et al., 2012) for performance evaluation of the LiDAR odometry.

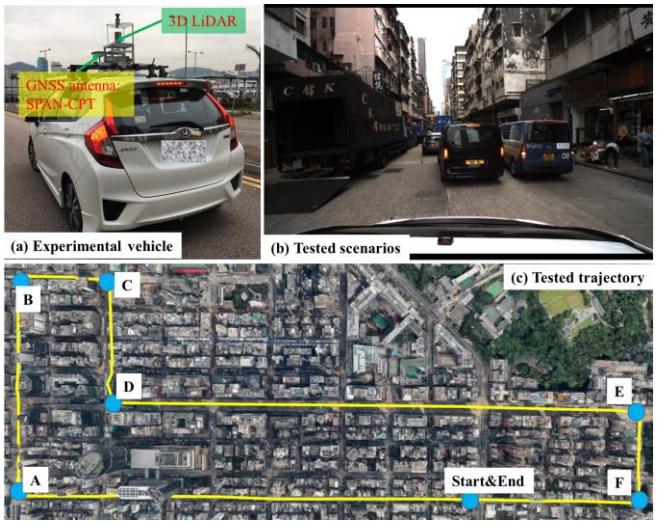

Fig. 5 (a) data collection vehicle with all the sensors installed in a compact sensor kit, (b) tested scenarios, and (c) trajectory with a driving distance of 4.2 km.

2) Performance Analysis

The mean errors of the rotation and translation of the six selected GCPs using are shown in Tables 2 and 3, respectively. Regarding the mean rotation error (Table 2), the LeGO-LOAM introduces a mean error in pitch angle (6.93°). The error in yaw and roll are 2.57° and 2.16°, respectively. With the help of the proposed AGPC, the error in pitch angle is decreased to 1.35° which shows that the proposed method can effectively mitigate the drift in pitch angle. The performance in the yaw and roll angle is also improved slightly.

Table 2. Performance of the rotation estimation in scene 1. %: [Total]/[total driving distance])

Method	Roll	Pitch	Yaw	Total	%
LeGO-LOAM (Shan & Englot, 2018)	2.57°	6.93°	2.16°	7.70°	0.18 %
AGPC-SLAM (proposed)	1.36°	1.35°	2.61°	3.24°	0.07 %
AGPC-SLAM (with loop closure)	1.29°	1.21°	1.53°	2.33°	0.056 %

Table 3. Performance of the translation estimation in scene 1. (AE: accumulated absolute error. %: [ENU]/[total driving distance])

Method	East	North	Up	ENU	AE	%
LeGO-LOAM (Shan & Englot, 2018)	3.89m	11.58 m	43.83m	45.49 m	34.80 m	1.08 %
AGPC-SLAM	1.39m	2.65m	0.41 m	3.02m	1.89m	0.04 %
AGPC-SLAM (loop closure)	1.32m	2.36m	0.40 m	2.73m	0.52m	0.06 5%

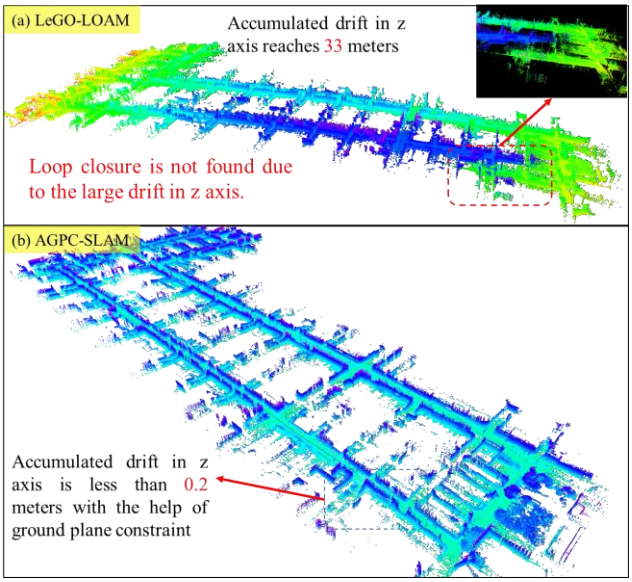

Fig. 6 Illustration of the map generated by the two methods. Illustration of the map generated using LeGO-LOAM (top panel) and the proposed AGPC-SLAM (bottom panel). The color of the map is annotated by the value of the z-axis (height) of each point.

Regarding the mean translation error, the errors of LeGO-LOAM in the east and north direction are 3.89 and 11.58 meters, respectively. Meanwhile, the error in the altitude (up) is about 43.83 meters. The total mean error in east, north, and up directions is 45.49 meters. The accumulated error (the error at the last epoch) reaches 34.80 meters. The last column represents the percentage of mean error concerning the total driving distance. With the help of the proposed AGPC-SLAM, the mean error is decreased to 3.02 meters and the accumulated error is only 1.89 meters. The significant improvement in the vertical direction shows that the proposed AGPC can effectively constrain the drift. Meanwhile, the detailed performances at six GCPs can be found in the Appendix of this paper (see Fig. 12 and 13).

After applying the loop closure constraint, the LeGO-LOAM cannot successfully detect the loop closure due to the large drift which can be seen in the top panel of Fig. 6. Therefore, the loop closure result for LeGO-LOAM is not provided in Tables 2 and 3. However, the rotation estimation for the proposed method is improved with the help of loop closure with a mean rotation error of 2.33° and the result of the map can be seen in the bottom panel Fig. 6. With the help of the loop closure, the mean error of translation decreases to 2.73 meters and the accumulated error is only 0.52 meters. The percentage of the error is only 0.065%. Although the loop closure detection is not the major contribution of this paper, the results show that the proposed AGPC could further enhance the detection of the loop closure.

Fig. 7 shows the details of the generated point map with the color being annotated by the reflectivity. The lane lines and building boundaries are clear which can be seen from Fig. 7-(a) and Fig. 7-(b). The detailed video of the proposed AGPC-SLAM in scene 1 can be found via the link

(https://www.youtube.com/watch?v=3nS895StJUo&feature=youtu.be).

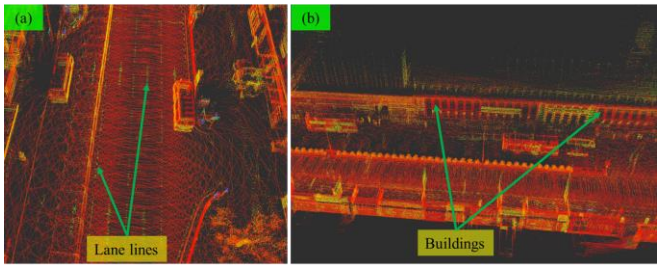

Fig. 7 Details of the generated map with reflectivity using the proposed AGPC-SLAM in scene 1.

B. Experimental Validation in Scene 2

1) Evaluation Metrics

To further evaluate how the proposed method can perform in the scenario with partial ramp road, we perform the other experiment in Tsim Sha Tsui of Hong Kong. The scene is shown in Fig. 8 where the red dots denote the evaluated trajectory. The ramp road starts with an altitude of 0 meters (the height is relative to the start point), increases to a maximum altitude of 2 meters, and decreases to 0 meters by the end of the ramp road. More importantly, the fixed solutions are available frequently during the test. Therefore, the pose from the NovAtel SPAN-CPT is directly used as the ground truth solution of scene 2.

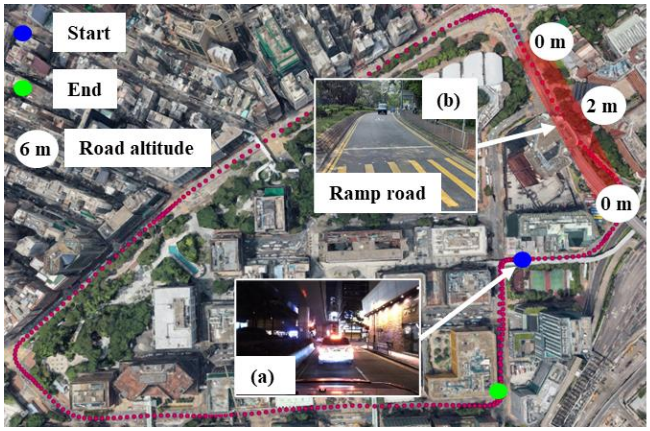

Fig. 8 (a) Environmental condition at the start point (blue shaded circle). (b) the scene with partial ramp road (red shaded area).

Since the LiDAR SLAM only provides the relative positioning concerning the starting point. Therefore, we apply the popular EVO (Grupp, 2017) toolkit to evaluate the relative error (RPE) for translation and rotation. Meanwhile, the evaluation metrics are as follows:

(1) *RMSE*: Root-mean-square error of the relative translation and rotation.

(2) *MEAN*: Mean error of the relative translation and rotation.

To further show the effectiveness of the proposed method in mitigating the vertical and overall drift, we add three additional evaluation metrics as follows:

(1) *Altitude*: the accumulated altitude drift by the end of the trajectory.

(2) *AE*: the accumulated error of translation (meters) and rotation (degrees) by the end of the trajectory.

(3) *%*: calculated by the [accumulated drift]/[total driving distance]. This is adopted to evaluate the overall drift of the SLAM algorithm.

2) Performance Analysis

Table 4 shows the performance of the translation and rotation estimation using LeGO-LOAM and the proposed AGPC-SALM, respectively. Be noted that the loop closure is not applied regarding the results in Table 4, therefore, to evaluate the contribution of the standalone AGPC. Regarding the translation, both the MEAN and RMSE are reduced with the help of the AGPC using the proposed method. Although the LeGO-LOAM proposed to estimate the motion of the z-axis and pitch angle using ground points, the final altitude drift still reaches 7.36 meters by the end of the test. Fortunately, the drift in altitude decreases to only 0.21 meters with the help of the AGPC which shows the effectiveness of the proposed AGPC. Meanwhile, the overall drift is also reduced using the proposed AGPC-SLAM (see "AE" and "%" in Table 4).

Table 4. Performance of the translation and rotation estimation in scene 2.

Method	MEAN	RMSE	Altitude	AE	%
LeGO-LOAM (Shan & Englot, 2018) (Translation)	0.33 m	0.46 m	7.36 m	8.92 m	0.42 %
AGPC-SLAM (Translation)	0.27 m	0.38 m	0.21 m	5.26 m	0.26 %
LeGO-LOAM (Shan & Englot, 2018) (Rotation)	0.69°	1.22°		3.72°	0.19 %
AGPC-SLAM (Rotation)	0.58°	1.07°		1.87°	0.09 %

The continuous 3D trajectories of LeGO-LOAM, AGPC-SLAM, and ground truth are shown in Fig. 9. Specifically, the altitude during the test is shown in Fig. 10. The black curve denotes the altitude provided by the NovAtel SPAN-CPT. We can see that the altitude estimation using the LeGO-LOAM deviates significantly from the ground truth during epoch 150~240 which is corresponding to the red shaded area in Fig. 9. During this period, one side of the road is filled with buildings surfaces which is a feature-insufficient scene. The other side is dense foliage with numerous similar tree leaves, increasing the difficulty of finding correct correspondence of LOAM. As a result, the LeGO-LOAM drift significantly. With the help of the AGPC, the altitude drift is mitigated accordingly.

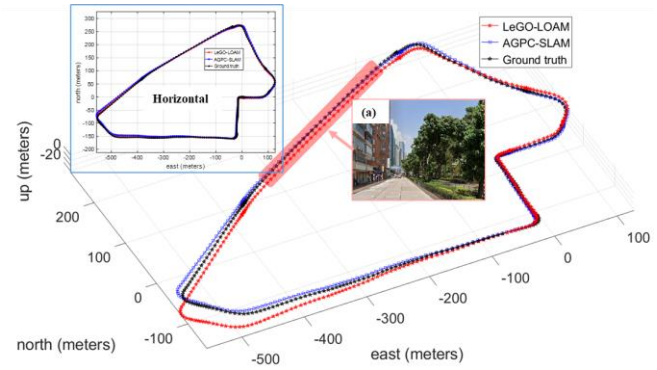

Fig. 9 Trajectories of the LeGO-LOAM (red curve), AGPC-SLAM (blue curve), and ground truth trajectory (black curve).

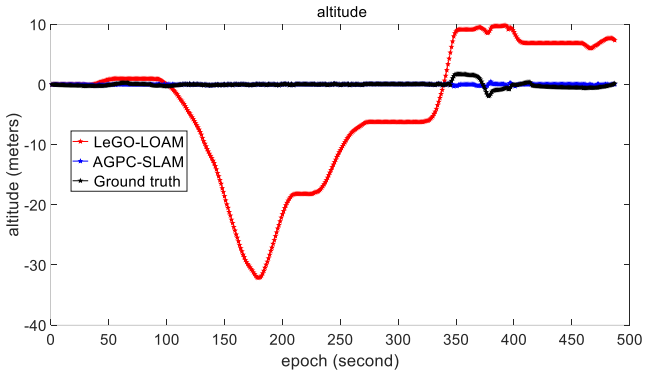

Fig. 10 Altitude of the LeGO-LOAM (red curve), AGPC-SLAM (blue curve), and ground truth trajectory (black curve) during the testing area with the ramp road.

The RPE of the translation and rotation estimations are provided in Figs. 11. The improvements can also be seen by the figure for both translation and rotation. The red and blue dash curves denote the translation errors of LeGO-LOAM and AGPC-SLAM, respectively. The black and cyan dash curves denote the rotation errors of LeGO-LOAM and AGPC-SLAM, respectively.

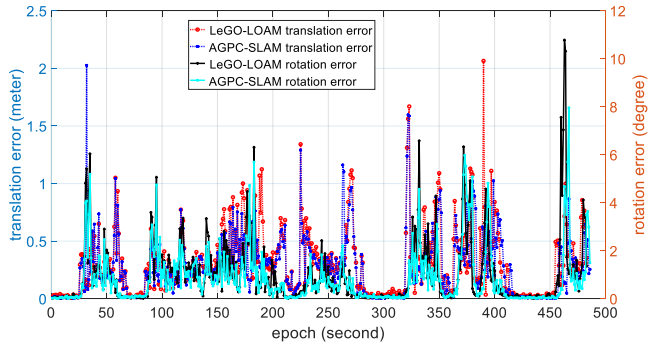

Fig. 11 Relative translation and rotation errors for LeGO-LOAM and the proposed AGPC-SLAM, respectively. The x-axis denotes the epoch and. The left y-axis denotes the translation error in the unit of meter. The right y-axis denotes the rotation error in the unit of degree.

As shown in Fig. 8, scene 2 involves a ramp road with the altitude deviating from 0 to 2 meters, which is corresponding to the annotated area by the red rectangle in Fig. 9. The estimated altitude using AGPC-SLAM deviates from the ground truth altitude (black curve). This is caused by the violation of the assumption that the ground plane is flat during the operation, which is required by the proposed method in this paper. In the future, we will study the ramp road surface identification and remove the AGPC when the slope surface is detected. The detailed video of the test in scene 2 can be found via the link (https://www.youtube.com/watch?v=mgLxxlhlscY&feature=youtu.be).

The performance of the translation and rotation angle estimation of AGPC-SLAM with loop closure is presented in Table 5. Although the loop closure is enabled in LeGO-LOAM, the loop is not successfully detected due to the large drift in the z-axis of LeGO-LOAM. However, the loop closure is detected using the proposed AGPC-SLAM as the vertical drift is improved. We believe that this is another contribution of the AGPC which improves the detection of the loop closure.

Table 5. Performance of translation and rotation estimation with loop closure

Method	Trans. RMSE	Rot. RMSE	Altitude
LeGO-LOAM (Shan & Englot, 2018) (Loop closure)	Loop Not Detected	Loop Not Detected	Loop Not Detected
AGPC-SLAM (Loop closure)	0.43 m	0.64°	0.13 m

VI. Conclusion and Future Work

To mitigate the drift of the LiDAR SLAM in the vertical direction due to the limited field of view of the LiDAR sensor, this paper proposes an AGPC-SLAM that achieved improved accuracy compared with the existing state-of-the-art LeGO-LOAM. This paper innovatively proposes to employ absolute ground plane detection to mitigate the drift in z-axis related states. Moreover, better-constrained positioning in the z-axis and pitch angle can also improve the positioning in x- and y-axes. We test the proposed method in two typical scenarios in Hong Kong. The results show that the proposed method can effectively mitigate the drift on the z-axis in both the evaluated scenarios.

One of the limitations of our work is that the proposed method can be effective only when the flat ground plane is available. In some extremities, the ground can be fully occluded by the surrounding dynamic vehicles which can lead to misidentification of the ground surface. Besides, the ground surface can be a slope that cannot be simply modeled using a plane function. In the future, we will study the scene with slope and derive the corresponding constraints to alleviate the drift of 3D LiDAR SLAM in urban canyons.

APPENDIX: PERFORMANCE OF AGPC-SLAM AT 6 GCPS

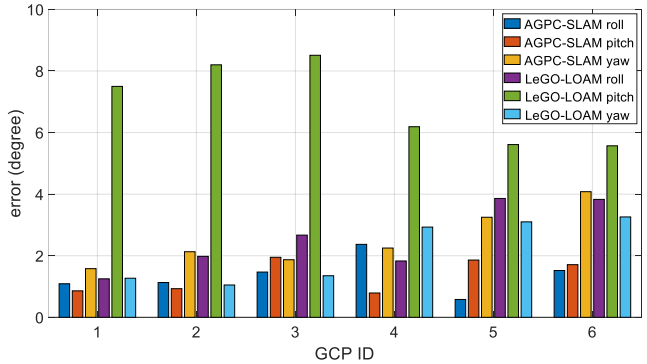

Fig. 12 Errors of the roll, pitch, and yaw angles for LeGO-LOAMand the proposed AGPC-SLAM. The x-axis denotes the ID of the GCPs from 1 to 6. The y-axis denotes the errors.

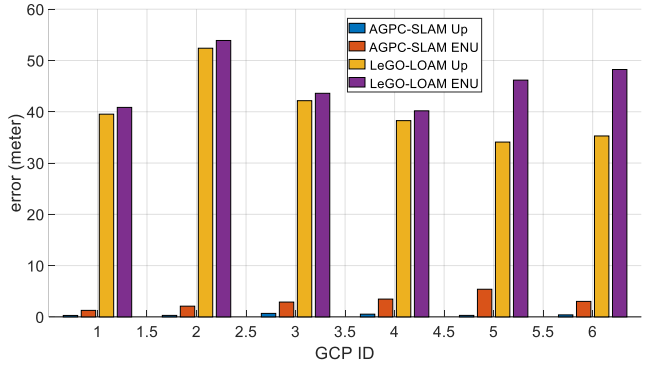

Fig. 13 Errors of the up (altitude), and ENU for LeGO-LOAM and the proposed AGPC-SLAM, respectively. The x-axis denotes the ID of the GCPs from 1 to 6. The y-axis denotes the errors.

REFERENCES

Bai, X., Wen, W., & Hsu, L.-T. (2020). Robust Visual-Inertial Integrated Navigation System Aided by Online Sensor Model Adaption for Autonomous Ground Vehicles in Urban Areas. *Remote Sensing, 12*(10), 1686. https://doi.org/10.3390/rs12101686

Chang, L., Niu, X., & Liu, T. (2020). GNSS/IMU/ODO/LiDAR-SLAM Integrated Navigation System Using IMU/ODO Pre-Integration. *Sensors, 20*(17), 4702. https://doi.org/10.3390/s20174702

Choi, S., Park, J., Byun, J., & Yu, W. (2014). *Robust ground plane detection from 3D point clouds.* Paper presented at the 2014 14th International Conference on Control, Automation and Systems (ICCAS 2014). 10.1109/ICCAS.2014.6987936

Dill, E. T., & Uijt de Haag, M. (2016). 3D Multi‐copter navigation and mapping using GPS, inertial, and LiDAR. *Navigation: Journal of The Institute of Navigation, 63*(2), 205-220. https://doi.org/10.1002/navi.134

Dow, J. M., Neilan, R. E., & Rizos, C. (2009). The international GNSS service in a changing landscape of global navigation satellite systems. *Journal of Geodesy, 83*(3), 191-198. http://dx.doi.org/10.1007/s00190-009-0315-4

Geiger, A., Lenz, P., & Urtasun, R. (2012). *Are we ready for autonomous driving? the kitti vision benchmark suite.* Paper presented at the 2012 IEEE conference on computer vision and pattern recognition. 10.1109/CVPR.2012.6248074

Grisetti, G., Kummerle, R., Stachniss, C., & Burgard, W. (2010). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine, 2*(4), 31-43. 10.1109/MITS.2010.939925

Grupp, M. (2017). evo: Python package for the evaluation of odometry and slam. *Note: https://github. com/MichaelGrupp/evo Cited by: Table, 7*. https://github. com/MichaelGrupp/evo

He, G., Yuan, X., Zhuang, Y., & Hu, H. (2020). An Integrated GNSS/LiDAR-SLAM Pose Estimation Framework for Large-Scale Map Building in Partially GNSS-Denied Environments. *IEEE Transactions on Instrumentation and Measurement, 70*, 1-9. 10.1109/TIM.2020.3024405

Hess, W., Kohler, D., Rapp, H., & Andor, D. (2016). *Real-time loop closure in 2D LIDAR SLAM.* Paper presented at the 2016 IEEE International Conference on Robotics and Automation (ICRA). 10.1109/ICRA.2016.7487258

Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., . . . Yang, R. (2018). *The apolloscape dataset for autonomous driving.* Paper presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 10.1109/TPAMI.2019.2926463

Indelman, V., Williams, S., Kaess, M., & Dellaert, F. (2012). *Factor graph based incremental smoothing in inertial navigation systems.* Paper presented at the Information Fusion (FUSION), 2012 15th International Conference on.

Koide, K., Miura, J., & Menegatti, E. (2018a). A Portable 3D LIDAR-based System for Long-term and Wide-area People Behavior Measurement. https://doi.org/10.1177/1729881419841532

Koide, K., Miura, J., & Menegatti, E. (2018b). A portable 3d lidar-based system for long-term and wide-area people behavior measurement. *IEEE Trans. Hum. Mach. Syst.*

Kuramachi, R., Ohsato, A., Sasaki, Y., & Mizoguchi, H. (2015). *G-ICP SLAM: An odometry-free 3D mapping system with robust 6DoF pose estimation.* Paper presented at the 2015 IEEE International Conference on Robotics and Biomimetics (ROBIO). 10.1109/ROBIO.2015.7418763

Li, Q., Li, R., Ji, K., & Dai, W. (2015). *Kalman filter and its application.* Paper presented at the 2015 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS). 10.1109/ICINIS.2015.35

Lin, J., & Zhang, F. (2020). *Loam livox: A fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV.* Paper presented at the 2020 IEEE International Conference on Robotics and Automation (ICRA). 10.1109/ICRA40945.2020.9197440

Low, K.-L. (2004). Linear least-squares optimization for point-to-plane icp surface registration. *Chapel Hill, University of North Carolina, 4*(10), 1-3.

Ma, L., Kerl, C., Stückler, J., & Cremers, D. (2016). *CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM.* Paper presented at the 2016 IEEE International Conference on Robotics and Automation (ICRA). 10.1109/ICRA.2016.7487260

Magnusson, M., Andreasson, H., Nuchter, A., & Lilienthal, A. J. (2009a). *Appearance-based loop detection from 3D laser data using the normal distributions transform.* Paper presented at the 2009 IEEE International Conference on Robotics and Automation. 10.1109/ROBOT.2009.5152712

Magnusson, M., Andreasson, H., Nuchter, A., & Lilienthal, A. J. (2009b). *Appearance-based loop detection from 3D laser data using the normal distributions transform.* Paper presented at the Robotics and Automation, 2009. ICRA'09. IEEE International Conference on. 10.1109/ROBOT.2009.5152712

Magnusson, M., Lilienthal, A., & Duckett, T. (2007). Scan registration for autonomous mining vehicles using 3D‐NDT. *Journal of Field Robotics, 24*(10), 803-827. https://doi.org/10.1002/rob.20204

Mascaro, R., Teixeira, L., Hinzmann, T., Siegwart, R., & Chli, M. (2018). *GOMSF: Graph-Optimization based Multi-Sensor Fusion for robust UAV pose estimation.* Paper presented at the 2018 IEEE International Conference on Robotics and Automation (ICRA). 10.1109/ICRA.2018.8460193

Pang, S., Kent, D., Cai, X., Al-Qassab, H., Morris, D., & Radha, H. (2019). *3D Scan Registration Based Localization for Autonomous Vehicles-A Comparison of NDT and ICP under Realistic Conditions.* Paper presented at the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall). 10.1109/VTCFall.2018.8690819

Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics, 34*(4), 1004-1020. 10.1109/TRO.2018.2853729

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., . . . Ng, A. Y. (2009). *ROS: an open-source Robot Operating System.* Paper presented at the ICRA workshop on open source software.

http://www.cim.mcgill.ca/~dudek/417/Papers/quigley-icra2009-ros.pdf

Saarinen, J., Andreasson, H., Stoyanov, T., & Lilienthal, A. J. (2013). *Normal distributions transform Monte-Carlo localization (NDT-MCL).* Paper presented at the Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. 10.1109/IROS.2013.6696380

Shan, T., & Englot, B. (2018). *LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain.* Paper presented at the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., & Rus, D. (2020). LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. *arXiv preprint arXiv:2007.00258*. 10.1109/IROS45743.2020.9341176

Shetty, A., & Xingxin Gao, G. (2019). Adaptive covariance estimation of LiDAR ‐ based positioning errors for UAVs. *Navigation, 66*(2), 463-476. https://doi.org/10.1002/navi.307

Wen, W., Bai, X., Kan, Y.-C., & Hsu, L.-T. (2019). Tightly Coupled GNSS/INS Integration Via Factor Graph and Aided by Fish-eye Camera. *Ieee Transactions on Vehicular Technology*. 10.1109/TVT.2019.2944680

Wen, W., Hsu, L.-T., & Zhang, G. (2018). Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong. *Sensors, 18*(11), 3928. https://doi.org/10.3390/s18113928

Wen, W., Zhang, G., & Hsu, L.-T. (2020). Object-Detection-Aided GNSS and Its Integration With Lidar in Highly Urbanized Areas. *IEEE Intelligent Transportation Systems Magazine, 12*(3), 53-69. 10.1109/MITS.2020.2994131

Wen, W., Zhang, G., & Hsu, L. T. (2019). Correcting NLOS by 3D LiDAR and building height to improve GNSS single point positioning. *Navigation, 66*(4), 705-718. https://doi.org/10.1002/navi.335

Yang, M. Y., & Förstner, W. (2010). *Plane detection in point cloud data.* Paper presented at the Proceedings of the 2nd int conf on machine control guidance, Bonn.

Ye, H., Chen, Y., & Liu, M. (2019). Tightly Coupled 3D Lidar Inertial Odometry and Mapping. *arXiv preprint arXiv:1904.06993*. 10.1109/ICRA.2019.8793511

Zhang, J., & Singh, S. (2014). *LOAM: Lidar Odometry and Mapping in Real-time.* Paper presented at the Robotics: Science and Systems. 10.15607/RSS.2014.X.007

Zhang, J., & Singh, S. (2017). Low-drift and real-time lidar odometry and mapping. *Autonomous robots, 41*(2), 401-416. 10.1007/s10514-016-9548-2

Zhao, S., Fang, Z., Li, H., & Scherer, S. (2019). *A robust laser-inertial odometry and mapping method for large-scale highway environments.* Paper presented at the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 10.1109/IROS40897.2019.8967880

Zheng, L., Zhu, Y., Xue, B., Liu, M., & Fan, R. (2019). *Low-cost GPS-aided lidar state estimation and map building.* Paper presented at the 2019 IEEE International Conference on Imaging Systems and Techniques (IST). 10.1109/IST48021.2019.9010530

Zuo, X., Geneva, P., Lee, W., Liu, Y., & Huang, G. (2019). LIC-Fusion: LiDAR-Inertial-Camera Odometry. *arXiv preprint arXiv:1909.04102*. 10.1109/IROS40897.2019.8967746