Low-Cost GPS-Aided LiDAR State Estimation and Map Building

Preprint · October 2019

CITATIONS
0

READS
64

5 authors, including:

Yilong Zhu
The Hong Kong University of Science and Technology
15 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Bohuan Xue
The Hong Kong University of Science and Technology
7 PUBLICATIONS **0** CITATIONS
SEE PROFILE

Ming Liu
The Hong Kong University of Science and Technology
253 PUBLICATIONS **2,613** CITATIONS
SEE PROFILE

Rui Fan
University of California, San Diego
48 PUBLICATIONS **267** CITATIONS
SEE PROFILE

Low-Cost GPS-Aided LiDAR State Estimation and Map Building

Linwei Zheng[1,2], Yilong Zhu[1,2], Bohuan Xue[1,2], Ming Liu[1], Rui Fan[2,3]
[1]Shenzhen Unity-Drive Innovation Technology Co. Ltd., Shenzhen, China.
[2]Robotics Institute, Hong Kong University of Science and Technology, Hong Kong SAR, China.
[3]Hangzhou ATG Intelligent Technology Co. Ltd., Hangzhou, China.
Emails: zhenglinwei@unity-drive.com, {yzhubr, bxueaa, eelium, eeruifan}@ust.hk

***Abstract*—Using different sensors in an autonomous vehicle (AV) can provide multiple constraints to optimize AV location estimation. In this paper, we present a low-cost GPS-assisted LiDAR state estimation system for AVs. Firstly, we utilize LiDAR to obtain highly precise 3D geometry data. Next, we use an inertial measurement unit (IMU) to correct point cloud misalignment caused by incorrect place recognition. The estimated LiDAR odometry and IMU measurement are then jointly optimized. We use a lost-cost GPS instead of a real-time kinematic (RTK) module to refine the estimated LiDAR-inertial odometry. Our low-cost GPS and LiDAR complement each other, and can provide highly accurate vehicle location information. Moreover, a low-cost GPS is much cheaper than an RTK module, which reduces the overall AV sensor cost. Our experimental results demonstrate that our proposed GPS-aided LiDAR-inertial odometry system performs very accurately. The accuracy achieved when processing a dataset collected in an industrial zone is approximately 0.14 m.**

I. Introduction

Simultaneous localization and mapping (SLAM) is a very important function in autonomous vehicles (AVs) [1]. Using AV sensor data and perception output, a SLAM module can not only estimate the location of an AV, but also build and update a 3D world map [2]. A highly accurate global navigation satellite system (GNSS), e.g., a real-time kinematic (RTK) module or a differential Global Positioning System (GPS), can provide centimeter-level AV localization accuracy [3], [4]. RTK modules utilize carrier-phase positioning data provided by base stations to compensate for the estimation error during transmission. However, the base station coverage can limit the available area of the RTK GPS receiver [5]. Furthermore, RTK modules cannot solve the signal multipath problem and GPS signal shielding problem, which can occasionally make the GPS-based localization approach infeasible [6].

Although many researchers have managed to deal with the aforementioned problems, the solutions are very limited [6]. On the other hand, LiDAR can provide accurate depth information, and therefore, is generally used for highly precise AV localization [2], [7], [8]. In practice, LiDAR-based localization methods perform very well in a static environment containing good geometry features. However, these methods can easily fail when there exists a dynamic object, e.g., a truck or a pedestrian. In an open environment, it is very difficult to find geometrical features when using only LiDAR; however, GPS can work consistently. Conversely, for an environment with many buildings, LiDAR can find useful features for point cloud, but GPS will lose signal [2]. In addition to GPS and LiDAR, inertial measurement units (IMUs) are generally used for ego-motion estimation [2], yet it can suffer greatly from dual integration error accumulation. The fusion of data collected by GPS, LiDAR and IMU will compensate the defects of each other. Therefore, multi-sensor systems are commonly used in autonomous cars.

A. System Overview

In this paper, we present a novel and robust LiDAR, IMU, and GPS state estimator for an AV. The system pipeline is shown in Fig. 1. The input of the system is the raw point cloud data collected using LiDAR. The point cloud is then processed to extract useful features. Next, the extracted features are used to register two different point clouds. A high-frequency IMU is used to give an initial guess for LiDAR registration. The preintegration of the IMU is considered as an edge constraint and is optimized together with the LiDAR odometry constraint. The LiDAR-inertial module can produce an optimized odometry at 10 Hz. The odometry is further optimized by integrating the GPS data at about 1 Hz. Finally, AV locations can be accurately estimated. The environment map is also built with undistorted point clouds and merged into the previous map based on loop closure detection.

In the optimization process, we consider both the sensor measurement as well as the extrinsic parameters. The proposed pose graph is shown in Fig. 2. Between each pair of LiDAR frames, the IMU output is used to obtain an initial guess for point cloud registration. The measurement is pre-integrated and added to a pose graph [9], and then non-linear graph optimization is applied to refine the vehicle state. The GPS measurement is also added to the pose graph to establish an additional constraint. Based on accurate state estimation, we can build a precise environment map. To deal with the LiDAR distortion caused by movement, we introduce a continuous-time spline to de-distort each frame of the point cloud from the LiDAR. [10]–[12] give us the ability to interpolate the quaternion, and the continuous-time odometry spline technique is given in [13] and [14]. For the reusability of the 3D map, we also merge the map on the pose graph with the method introduced in [15]. The odometry is mapped to the local east, north, up (ENU) coordinate of the start point. We tested our system in a variety of environments. We use Ceres Solver [16] to solve the non-linear least square problems in our program.

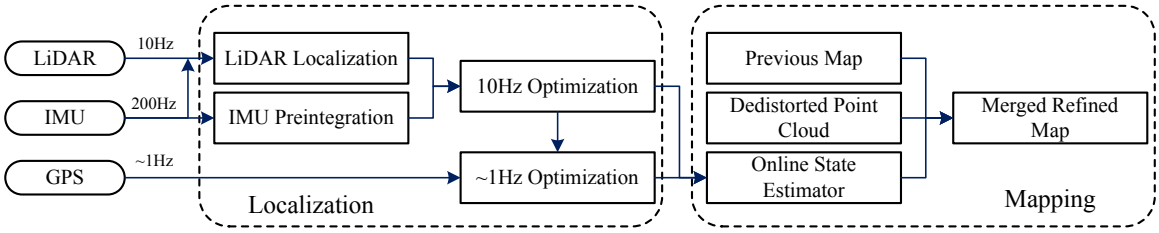


Fig. 1. System pepeline

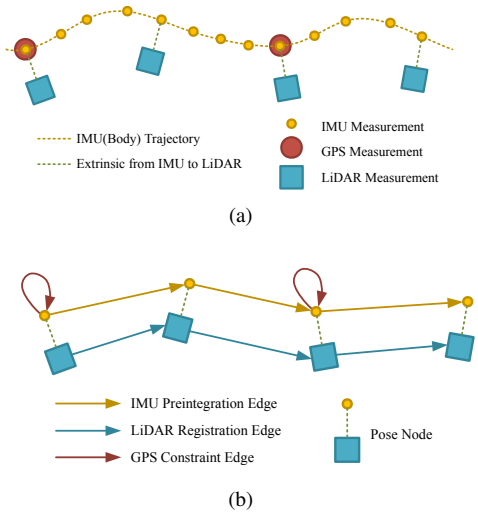


Fig. 2. Pose graph illustration.

B. Contributions

The contributions of this work include:

- A low-cost odometry and mapping system where the GPS constraint edge is integrated into the pose graph to optimize the LiDAR-inertial odometry.
- A practical mapping method that can merge a pre-built map and newly added map, and refine each frame of the point cloud to provide a more accurate reusable map.
- Two datasets, one collected in an industrial zone and the other a harbor.

C. Paper Structure

The remainder of this paper is structured as follows: Section II discusses the related work. Section III presents our LiDAR-inertial odometry method. Section IV introduces our proposed GPS optimization approach. Section V presents our 3D continuous mapping algorithm. The experimental results are illustrated and the system performance is discussed in Section VI. Finally, Section VII concludes the paper and provides recommendations for future work.

II. Related Work

Over the past two decades, many researchers have utilized GPS to achieve accurate odometry results [3], [4]. However, as discussed in Section I, the GPS signal is vulnerable to environment changes. Therefore, researchers have been working on environment perception and AV state estimation with the use of multiple sensors [7], [8], [17]–[19]. For example, passive sensors such as cameras can be used to capture 2D/3D visual data of the environment [20], [21]. In [17], a vision-based odometry system named ORB-SLAM is presented, where the image feature points are used to register different input frames. However, the robustness of such a system can be easily effected by various environment conditions, e.g., light changes [18]. In addition to cameras, LiDAR, which measures the distance to an object by detecting and analyzing the reflected light [2], is also generally used for AV location estimation and 3D world map creation. Compared to cameras, LiDAR is less effected by light changes, and therefore is more feasible for all-weather use [2]. For instance, [7] and [8] designed a LiDAR-based odometry system, which uses edge and surface information to register point clouds. To better constrain the matching relation of the point cloud, we typically segment the ground planes in advance. For example, [19] and [22]–[25] present a variety of methods for point cloud segmentation. Although such methods have achieved some impressive experimental results in most situations, there still exist some drawbacks. For example, the features in the point cloud are not as rich as those extracted from an image [23]. Furthermore, especially in a wide feature-less area, a LiDAR-based system can easily fail [2].

To better estimate the sensor state, many researchers have turned their focus towards multi-sensor data fusion. EKF and UKF are two popular traditional fusion techniques demonstrated by [3], [4], [26]. Recently, several non-linear optimization methods have been proposed to fuse different types of data. For example, Kümmerle et al. [9] proposed a graph-based optimization approach, which treats the robot state as a node and the constraint gained from measurement as an edge. Wang et al. [27] proposed a method to fuse the data collected by a multiple-sensor system that consists of RTK modules, a LiDAR sensor, and IMUs. However, the performance of such a system is subject to GPS single point positioning. Koide et al. [28] used an offline SLAM method based on both LiDAR and GPS. However, the system cannot generate odometry in real time, limiting its practical application. Moreover, the LiDAR component is based on a normal distributions transform (NDT), which is not elegant. In this paper, we use the aforementioned pose graph approach but simplify the non-linear optimization problem. Furthermore, we improve the method by utilizing a low-cost GPS instead of an RTK module, and apply feature-based point cloud registration instead of NDT. To build a clear and reusable map, we deal with the distortion of the point cloud from multi-beam LiDAR caused by the AV

movement. Although [7] has already solved this problem, the solution is based on basic linear interpolation between Euler angles. Due to the gimbal lock, the interpolation cannot always work well. Our method applies B-spline interpolation in a $\mathbb{SE}3$ space and achieves a better de-distortion result.

III. LiDAR-Inertial Localization

A. IMU prediction

We briefly denote frame as follows. $(\cdot)^W$ denotes world frame W, $(\cdot)^I$ denotes IMU frame I, and $\boldsymbol{T}_I^W$ is the IMU pose in the world frame. In our system, the IMU state is treated as the vehicle body state. The raw measurement data includes acceleration $\boldsymbol{a}^I(t)$ and angular velocity $\boldsymbol{\omega}^I(t)$ in IMU frame I. $\boldsymbol{g}^W$ represents a gravity constant in the world frame W. Considering the measurement noise $\boldsymbol{n}$ and bias $\boldsymbol{b}$, the IMU state is as follows:

$$\begin{aligned}\hat{\boldsymbol{\omega}}^W &= \boldsymbol{\omega}^I - \boldsymbol{b_\omega}^I - \boldsymbol{n_\omega}^I,\\ \hat{\boldsymbol{a}}^W &= \boldsymbol{R}_I^W(\boldsymbol{a}^I - \boldsymbol{b_a}^I - \boldsymbol{n_a}^I) + \boldsymbol{g}^W,\end{aligned} \tag{1}$$

where $\boldsymbol{R}_I^W$ is the rotation matrix from the IMU frame to the world coordinate system. We also define the translation from the IMU frame to the world coordinate system as t_I^W. Then, the transformation matrix is written as $\boldsymbol{T}_I^W = [\boldsymbol{R}_I^W|\boldsymbol{t}_I^W]$. Let the IMU frame I represent the state of the vehicle. We can use $\boldsymbol{T}_I^W$ to denote the pose of the vehicle in the world frame.

To predict the initial guess of the next incoming LiDAR frame, we need to integrate the IMU state during the interval between i and j:

$$\begin{aligned}&\boldsymbol{v}_j - \boldsymbol{v}_i = \boldsymbol{R}_{I\ i}^W(\boldsymbol{a}^I - \boldsymbol{b_a}^I - \boldsymbol{n_a}^I)\Delta t + \boldsymbol{g}^W\Delta t,\\ &\boldsymbol{t}_j - \boldsymbol{t}_i = \boldsymbol{v}_i\Delta t + \frac{1}{2}\boldsymbol{R}_{I\ i}^W(\boldsymbol{a}^I - \boldsymbol{b_a}^I - \boldsymbol{n_a}^I)\Delta t^2 + \frac{1}{2}\boldsymbol{g}^W\Delta t^2,\\ &\boldsymbol{R}_{I\ i}^{W\ T}\boldsymbol{R}_{I\ j}^W = \exp(\boldsymbol{\omega}^I - \boldsymbol{b_\omega}^I - \boldsymbol{n_\omega}^I\Delta t).\end{aligned} \tag{2}$$

B. IMU Preintegration

To avoid repeatedly integrating the measurement when solving the non-linear optimization problem, we adopt the IMU preintegration method stated in [29] and [30] into our system. The key idea is to define a relative motion of the IMU regardless of the pose and velocity at a specific time. The preintegration model between state i and state j is as follows:

$$\begin{aligned}\Delta\boldsymbol{R}_{ij} &= \boldsymbol{R}_i^T\boldsymbol{R}_j,\\ \Delta\boldsymbol{v}_{ij} &= \boldsymbol{R}_i^T(\boldsymbol{v}_j - \boldsymbol{v}_i - \boldsymbol{g}\Delta t_{ij}),\\ \Delta\boldsymbol{t}_{ij} &= \boldsymbol{R}_i^T(\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{v}_i\Delta t_{ij} - \frac{1}{2}\boldsymbol{g}\Delta t_{ij}^2).\end{aligned} \tag{3}$$

$\Delta\boldsymbol{R}_{ij}, \Delta\boldsymbol{v}_{ij}$, and $\Delta\boldsymbol{t}_{ij}$ are defined as the preintegrated measurement of the rotation, velocity, and translation, respectively. All symbols here are denoted in the frame W, so we drop superscript in the equation for brevity. Noting that the measurements are independent to state i and state j, we do not need to recalculate the preintegrated measurement when optimizing the state. The preintegrated measurement is the constraint edge in the pose graph. We also estimate the bias from the optimization and update the bias to get a more accurate result.

C. LiDAR Localization

We use multi-beam LiDAR to obtain a 3D environment point cloud. There are many algorithms dealing with point cloud registration; iterative closest point (ICP) and NDT are the famous algorithms among them. However, these algorithms cannot perform in real time. To fully exploit the ground vehicle characteristics, we first divide the point cloud according to whether or not the point belongs to the ground plane. The road plane is segmented using the approach for road curb detection [22]. Second, we apply the edge-surface feature-based registration [7]. The local convex is defined to calculate the local smoothness on each beam of the LiDAR points. A larger value means that the local surface is sharper. The edge and surface are selected by the criterion of the local convex. The point cloud matching is a process of minimizing the point-line distance $\boldsymbol{d}_\mathcal{L}$ and point-plane distance $\boldsymbol{d}_\mathcal{P}$. The problem is stated as follows:

$$\boldsymbol{T} = \arg\min_{\boldsymbol{T}} \sum_{i\in\mathscr{E}} \boldsymbol{d}_{\mathcal{L}i} + \sum_{i\in\mathscr{S}} \boldsymbol{d}_{\mathcal{P}i} + \sum_{i\in\mathscr{G}} \boldsymbol{d}_{\mathcal{P}i}, \tag{4}$$

where $\mathscr{E}, \mathscr{S}, \mathscr{G}$ represent the sets of edge points, surface points, and ground points, respectively, and T is the pose of the vehicle, also known as the transformation from the body frame $\boldsymbol{B}$ to world frame $\boldsymbol{W}$. The pose $\boldsymbol{T}$ can be calculated by solving the non-linear least square problem above. The LiDAR constraint T and pre-integrated measurement $\Delta\boldsymbol{R}_{ij}, \Delta\boldsymbol{v}_{ij}, \Delta\boldsymbol{t}_{ij}$ are coupled to get the optimized odometry of the LiDAR-inertial system output.

IV. GPS Optimization

A. GPS Conversion

The obtained GPS data is latitude, longitude, and altitude (LLA). The measurement needs to be converted to the local coordinate before optimization. The conversion consists of two steps: from LLA to Earth-centered Earth-fixed (ECEF), and from ECEF to local ENU. The conversion from LLA to ECEF is as follows:

$$\begin{aligned}\boldsymbol{x} &= (\boldsymbol{R} + \boldsymbol{h})\cos\boldsymbol{\phi}\cos\boldsymbol{\lambda}\\ \boldsymbol{y} &= (\boldsymbol{R} + \boldsymbol{h}\cos\boldsymbol{\phi}\sin\boldsymbol{\lambda}\\ \boldsymbol{z} &= (\frac{\boldsymbol{b}^2}{\boldsymbol{a}^2}\boldsymbol{R} + \boldsymbol{h})\sin\boldsymbol{\phi},\end{aligned} \tag{5}$$

where $\boldsymbol{\phi}$ denotes latitude, $\boldsymbol{\lambda}$ represents longitude, $\boldsymbol{h}$ denotes altitude, and R is the Earth's radius defined as:

$$\boldsymbol{R} = \frac{\boldsymbol{a}^2}{\sqrt{(\boldsymbol{a}\cos\boldsymbol{\phi})^2 + (\boldsymbol{b}\sin\boldsymbol{\phi})^2}}, \tag{6}$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ is the equatorial radius and polar radius, respectively.

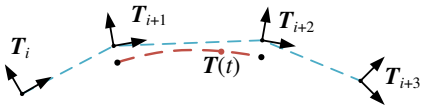


Fig. 3. Cubic B-Spline in $\mathbb{SE}3$

The origin of the local ENU coordinate is the position of the start point. The conversion from ECEF to ENU is as follows:

$$\begin{bmatrix} \boldsymbol{e} \\ \boldsymbol{n} \\ \boldsymbol{u} \end{bmatrix} = \begin{bmatrix} -\sin\boldsymbol{\lambda} & \cos\boldsymbol{\lambda} & 0 \\ -\cos\boldsymbol{\lambda}\sin\boldsymbol{\phi} & -\sin\boldsymbol{\lambda}\sin\boldsymbol{\phi} & \cos\boldsymbol{\lambda} \\ \cos\boldsymbol{\lambda}\cos\boldsymbol{\phi} & \sin\boldsymbol{\lambda}\cos\boldsymbol{\phi} & \sin\boldsymbol{\phi} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} - \boldsymbol{x}_0 \\ \boldsymbol{y} - \boldsymbol{y}_0 \\ \boldsymbol{z} - \boldsymbol{z}_0 \end{bmatrix}. \tag{7}$$

B. GPS Constraint

Since low-cost GPS does not provide a heading message, we only have knowledge of the position of the GPS frame G. To eliminate the influence of the obscure rotation, we install the GPS antenna on the top of the IMU. $\mathscr{P}$ represents the set of indexes of the synchronized pairs of the GPS position and IMU pose, $\boldsymbol{t}$ denotes the translation component in $[\boldsymbol{R}|\boldsymbol{t}]$, and $\boldsymbol{p}$ represents the position in the local ENU coordinate. We aim at minimizing the following residual function:

$$\sum_{i\in\mathscr{P}} ||\boldsymbol{t}_i - \boldsymbol{p}_{\boldsymbol{i}}||^2. \tag{8}$$

As shown in Fig. 2(b), the GPS constraint is an unary hyper-edge, which is connected to the IMU *pose* node. Here, we keep a sliding window of recent *poses* to get a refined result. The optimization considers the constraint of the preintegrated IMU measurement.

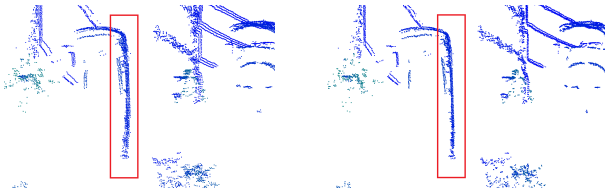
V. 3D Continuous Mapping

From the previous section, we can obtain a refined odometry, which is constrained by the LiDAR, an IMU, and a GPS. In this section, we undistort the point cloud and build a high-definition reusable map. Please note that the multi-beam LiDAR sensor is working continuously, even though we usually obtain discrete separated frames of the point cloud from the LiDAR driver. In other words, the points with different azimuths have different time stamps. Thus, we need to undistort each frame of the point cloud to get the original scan result. For instance, the timestamp of the beginning point and ending point in a single LiDAR scan have a difference of the period Δt.

To undistort each point cloud frame, we need to spline between the discrete odometry. We apply the uniform cubic B-Splines in $\mathbb{SE}3$ according to [13] and [14]. As can be seen in Fig. 3, we need four transformation matrices $\boldsymbol{T}_i, \boldsymbol{T}_{i+1}, \boldsymbol{T}_{i+2}$, and $\boldsymbol{T}_{i+3}$ to interpolate a smooth state $\boldsymbol{T}(t)$. We have a basis matrix $\tilde{\mathbf{M}}$:

$$\tilde{\mathbf{M}} = \begin{bmatrix} 6 & 5 & 1 & 0 \\ 0 & 3 & 3 & 0 \\ 0 & -3 & 3 & 0 \\ 0 & 1 & -2 & 1 \end{bmatrix}, \tag{9}$$

and the cumulative basis function in matrix form is



(a) (b)

Fig. 4. A comparison (a) a distorted point cloud and (b) undistorted point cloud.

Table I
Experimental results of the two datasets.

Dataset	max (m)	min (m)	mean (m)	SD (m)
industrial zone	0.9462	0.0061	0.1421	0.0808
Harbor	8.7316	0.0027	2.8068	2.1302

$$\begin{aligned} \tilde{\mathbf{B}} &= \frac{1}{3!}\begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix}\tilde{\mathbf{M}} \\ &= \frac{1}{3!}\begin{bmatrix} 6 \\ 5+3u-3u^2+u^3 \\ 1+3u+3u^2-2u^3 \\ u^3 \end{bmatrix}^{\mathrm{T}}, \\ u &= \frac{t-t_{i+1}}{\Delta t}, u \in [0,1), \end{aligned} \tag{10}$$

where Δt is the time interval between the adjacent transformations, and t_{i+1} is the timestamp of the $(i+1)$-th transformation. We assume that the angular velocity stays constant and the acceleration is zero during the interval. Then, we can define the following equation:

$$\begin{aligned} \boldsymbol{T}(t) =& \exp(\tilde{B}_{0,4}\log(\boldsymbol{T}_i))\prod_{k=1}^{3}\exp(\tilde{B}_{k,4}\log({\boldsymbol{T}_{i+k-1}}^{-1}\boldsymbol{T}_{i+k})) \\ =& \exp(\log(\boldsymbol{T}_i))\exp\left(\frac{1}{6}(t+3u-3u^2+u^3)\log(\boldsymbol{T}_{ii+1})\right) \\ & \exp\left(\frac{1}{6}(1+3u+3u^2-2u^3)\log(\boldsymbol{T}_{i+1i+2})\right) \\ & \exp\left(\frac{1}{6}u^3\log(\boldsymbol{T}_{i+2i+3})\right). \end{aligned} \tag{11}$$

The spline $\boldsymbol{T}(t)$ is used to undistort the points in each frame. As shown in Fig. 4, the original point cloud shown in Fig. 4(a) is refined after the undistortion and it becomes the point cloud shown in Fig. 4(b). Then, the undistorted point cloud is superimposed onto the map. The map point cloud and the origin of the map coordinate are saved for the purpose of reuse. After vehicle reboot, the system will load the pose graph and point cloud map. Along with the running of the vehicle, the new *nodes* and *edge* will be added to the pose graph. The joint optimization will merge the new and old maps.

VI. Experimental Results

Our vehicle was equipped with a Velodyne VLP-16 LiDAR sensor, a Ublox GYGPSV1 NEO-M8N GPS module, and an LPMS-ME1 IMU module. The LiDAR sensor has 16 channels of laser scanners and can provide a 3D point cloud at 10 Hz.

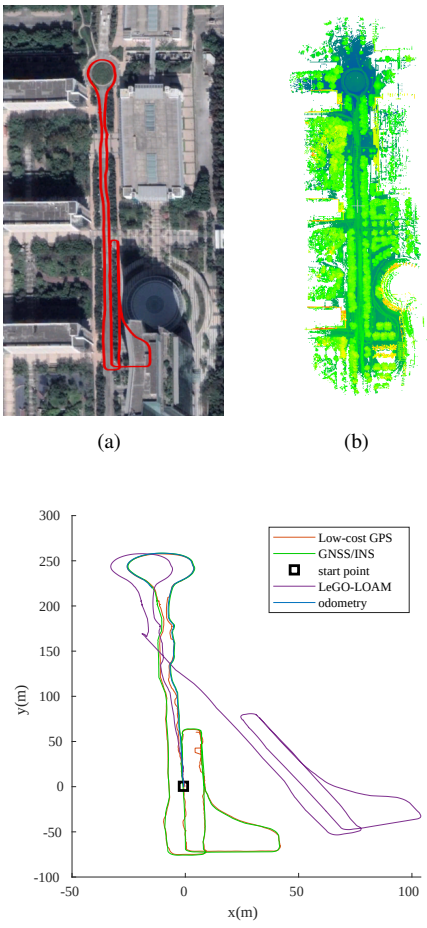


Fig. 5. Experimental results of the industrial zone dataset: (a) satellite view; (b) created map; (c) trajectory comparison.

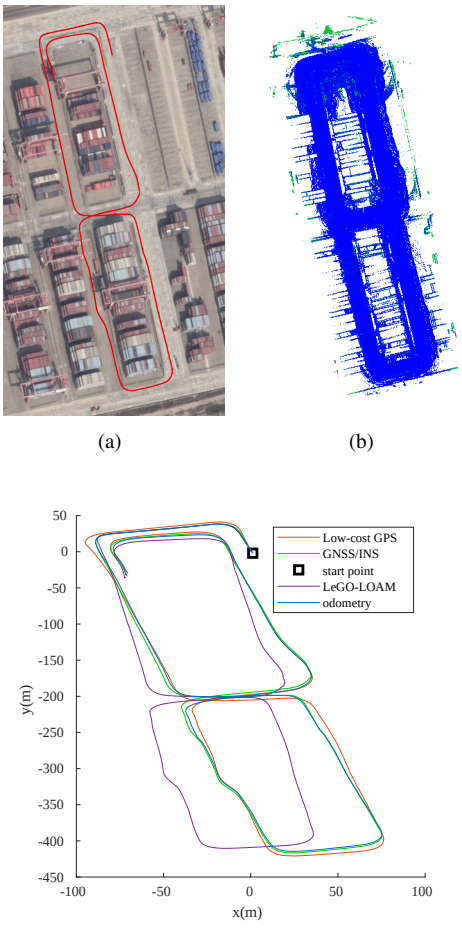


Fig. 6. Experimental results of the harbor dataset: (a) satellite view; (b) created map; (c) trajectory comparison.

The GPS module can provide data at 1 Hz, and the IMU can provide data at up to 400 Hz. We also utilized a highly precise GNSS (NovAtel 718d) to provide the ground truth with our system. The computing platform was designed based on an NUC (i7-7567U@3.50GHz, 16GB RAM) and it was assembled within the car. We implemented our algorithm in C++ under the framework of Robot Operating System (ROS). Our testing sites include an industrial zone and a wide harbor. Our experimental results are shown in Table I, where SD refers to standard deviation. The next subsections detail the experimental results of the industrial zone dataset and harbor dataset, respectively.

A. *Industrial Zone Test*

Firstly, we tested our algorithm in an industrial zone, where the surroundings are full of trees and buildings in which GPS cannot perform well. We use a GNSS/INS to provide precise ground truth with centimeter accuracy. The total driving distance is about 1.1 km.

The experimental results are shown in Fig. 5. The red line represents the AV trajectory provided by our low-cost GPS, and the green line represents the trajectory ground truth. Fig. 5(c) shows that our method (blue) outperforms LeGO-LOAM (purple). It is worthy noting that there is a huge deviation in the LeGO-LOAM trajectory at around (-20, 150). The reason is that a large truck is passing nearby at that time and LeGO-LOAM misaligns the point cloud due to the dynamic object (truck). In contrast, our method uses GPS to constrain the final odometry and provide a better trajectory, which looks very similar to the ground-truth trajectory. The mean error is only about 0.14 m.

B. *Harbor Test*

In addition, we also tested our algorithm in a wide harbor, where there are few obstructions in the area and our GPS works very well. In such a testing site, our LiDAR can barely find useful geometry features, such as edges and surfaces. As discussed in the previous subsection, LeGO-LOAM jumps suddenly because of the point cloud misalignment, while our method is able to trace the ground truth. This is because the GPS data plays an important role in the open area.

VII. Conclusion and Future Work

In this paper, we presented a robust and novel state estimate system for AVs. Our approach can provide reliable real-time

odometry with the support of LiDAR, an IMU, and a low-cost GPS. The proposed system also has features such as initialization, relocalization, map storage and merging, failure detection and recovery, and pose graph optimization. Our system achieved very good performance on public datasets, which were collected in a harbor and a industrial zone. Additionally, the reusable map can save many resources for online localization. The most practical contribution of the work is the use of a low-cost GPS instead of RTKs, which can also decrease the cost of an AV.

Although our system can be used in both broad and narrow environments with the use of both GPS and LiDAR, it needs to be improved to adapt to more complex environments. Furthermore, point cloud feature extraction also needs to be refined to gain a better feature association. While the system reduces the AV sensor cost without reducing the performance, the system robustness needs further improvement. Future work will include designing a GPS-aided stereo odometry system based on our previously published disparity estimation algorithm [31].

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China, under grant No. U1713211, the Research Grant Council of Hong Kong SAR Government, China, under Project No. 11210017, No. 21202816, and the Shenzhen Science, Technology and Innovation Commission (SZSTI) under grant JCYJ20160428154842603, awarded to Prof. Ming Liu.

REFERENCES

[1] R. Fan, "Real-time computer stereo vision for automotive applications," Ph.D. dissertation, University of Bristol, 2018.

[2] R. Fan, J. Jiao, H. Ye, Y. Yu, I. Pitas, and M. Liu, "Key ingredients of self-driving cars," *arXiv:1906.02939*, 2019.

[3] K. Ohno, T. Tsubouchi, B. Shigematsu, and S. Yuta, "Differential gps and odometry-based outdoor navigation of a mobile robot," *Advanced Robotics*, vol. 18, no. 6, pp. 611–635, 2004.

[4] F. Capezio, A. Sgorbissa, and R. Zaccaria, "Gps-based localization for a surveillance ugv in outdoor areas," in *Proceedings of the Fifth International Workshop on Robot Motion and Control, 2005. RoMoCo'05.* IEEE, 2005, pp. 157–162.

[5] R. B. Langley, "Rtk gps," *GPS World*, vol. 9, no. 9, pp. 70–76, 1998.

[6] N. Sünderhauf, M. Obst, G. Wanielik, and P. Protzel, "Multipath mitigation in gnss-based localization using robust optimization," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 784–789.

[7] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, 2014, p. 9.

[8] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[9] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3607–3613.

[10] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines*. springer-verlag New York, 1978, vol. 27.

[11] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *SIGGRAPH*, vol. 95, 1995, pp. 369–376.

[12] K. Qin, "General matrix representations for b-splines," *The Visual Computer*, vol. 16, no. 3, pp. 177–186, 2000.

[13] S. Lovegrove, A. Patron-Perez, and G. Sibley, "Spline fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras." in *BMVC*, vol. 2, no. 5, 2013, p. 8.

[14] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1425–1440, 2018.

[15] T. M. Bonanni, B. Della Corte, and G. Grisetti, "3-d map merging on pose graphs," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1031–1038, 2017.

[16] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[18] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[19] F. Moosmann, O. Pink, and C. Stiller, "Segmentation of 3d lidar data in non-flat urban environments using a local convexity criterion," in *2009 IEEE Intelligent Vehicles Symposium*. IEEE, 2009, pp. 215–220.

[20] R. Fan and M. Liu, "Road damage detection based on unsupervised disparity map segmentation," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[21] R. Fan, U. Ozgunalp, B. Hosking, M. Liu, and I. Pitas, "Pothole detection based on disparity transformation and road surface modeling," *IEEE Transactions on Image Processing*, vol. 29, pp. 897–908, 2020.

[22] W. Zhang, "Lidar-based road and road-edge detection," in *2010 IEEE Intelligent Vehicles Symposium*. IEEE, 2010, pp. 845–848.

[23] P. Chu, S. Cho, S. Sim, K. Kwak, and K. Cho, "A fast ground segmentation method for 3d point cloud." *Journal of information processing systems*, vol. 13, no. 3, 2017.

[24] M. Himmelsbach, F. v. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," in *2010 IEEE Intelligent Vehicles Symposium*. La Jolla, CA, USA: IEEE, Jun. 2010, pp. 560–565. [Online]. Available: http://ieeexplore.ieee.org/document/5548059/

[25] G. A. Marcon dos Santos, V. Terra Ferro, C. D. Noronha Vinhal, and G. d. Cruz Junior, "Fast algorithm for real-time ground extraction from unorganized stereo point clouds," *Pattern Recognition Letters*, vol. 84, pp. 192–198, Dec. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167865516302677

[26] J. L. Crassidis, "Sigma-point kalman filtering for integrated gps and inertial navigation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 2, pp. 750–756, 2006.

[27] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4670–4677.

[28] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional lidar-based system for long-term and wide-area people behavior measurement," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, p. 1729881419841532, 2019.

[29] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation." Georgia Institute of Technology, 2015.

[30] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, Feb. 2017.

[31] R. Fan, X. Ai, and N. Dahnoun, "Road surface 3d reconstruction based on dense subpixel disparity map estimation," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3025–3035, 2018.