2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Tightly-Coupled Multi-Sensor Fusion for Localization with LiDAR Feature Maps

Liangliang Pan, Kaijin Ji, and Ji Zhao

Abstract— **Robust and accurate pose estimation in long-term localization is crucial to autonomous driving. In this paper, we dealt with absolute localization with a LiDAR feature map and multi-sensor measurements. We proposed a tightly-coupled fusion method with fixed-lag smoothing. A sliding window of recently maintained states is estimated by minimizing a joint cost function. This cost function includes residuals of global LiDAR registration and relative kinematic constraints from an IMU and wheel encoders. In addition, we enhance the robustness of our method by improving LiDAR registration. To achieve this goal, LiDAR feature maps with a hybrid of geometric and normal distribution features are constructed and exploited. The effectiveness of the proposed method is verified in several challenging test sequences over 200km. The experimental results demonstrate that the proposed method achieves accurate localization and high robustness in challenging scenarios even when the LiDAR observation is degraded.**

I. INTRODUCTION

Autonomous driving has attracted much attention in recent years, and high-precision localization is a crucial component for it. At present, there are mainly vision-based solutions [1], [2], [3] and LiDAR-based solutions [4], [5], [6], [7], [8], [9]. Visual information contains rich semantic informations, but absolute localization based on vision is still fragile in many scenes. In contrast, LiDARs can obtain accurate 3D information of environments directly, and it does not has much dependence on scenes. As a result, LiDAR-based localization is popular in current autonomous driving. This paper also takes a LiDAR as the main sensor.

Most localization systems in autonomous driving rely on Global Navigation Satellite System (GNSS), which provides global positioning through triangulating multi-satellite signals. However, GNSS measurements might be noisy and cannot be reliable all the time. IMU and wheel encoder are also widely used for vehicles to enhance localization performance. An IMU can provide accurate motion prediction in a short period. Wheel encoders, when combined with vehicle kinematic constraints, also provide accurate motion prediction in a short period. The motion predictions provided by an IMU and wheel encoders can not only provide an initial value for point cloud registration, but can also be used to smooth localization results. Thus it is common to develop localization systems based on multi-sensor fusion. A tightly-coupled IMU and wheel encoder method was introduced in [10], which effectively suppresses the problem of unobservability of partial state variables. Tightly-coupled frameworks for LiDAR-IMU odometry (LIO) were proposed in [11], [12]. In [13], a loosely-coupled fusion framework is proposed for absolute localization. In this paper, we proposed a tightly-coupled framework for absolute localization with LiDAR feature maps. A typical result in a challenging scenario is shown in Fig. 1.

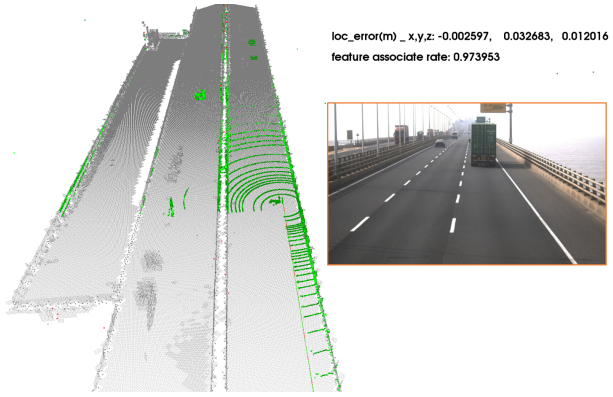

Fig. 1. Online performance of the proposed method on a sea-crossing bridge. **Left**: The gray point cloud represents feature points of a pre-built feature map; the green points are points of the current LiDAR frame. **Right**: The image captured by a forward-facing camera visualizes the current working environment. Though the LiDAR features constrain weakly in forward direction, the statistic above the image from offline evaluation indicates the system works well.

LiDAR-based localization methods mainly include intensity-based methods [4], [5], [14], distribution-based methods [6], [7], and geometric feature-based methods [8], [9]. Generally speaking, geometric feature-based methods are accurate and efficient, but they are prone to degenerate when geometric features are rare in unstructured environments. Distribution-based methods have less dependence on environments but they are less efficient to obtain high accuracy. In this paper, we constructed and exploited feature maps with a hybrid of geometric-based feature and distribution-based feature. As a result, our method obtains the advantages of these two complementary features in terms of accuracy and robustness.

The main contributions of this paper are summarized as follows: (1) We proposed a tightly-coupled localization fusion framework with LiDAR feature maps, which considers global feature map constraints and local motion constraints jointly. (2) We exploit both geometric and normal distribution features in LiDAR feature maps to enhance accuracy and robustness. The effectiveness of the proposed method is verified using many real-world challenging sequences and road tests.

L. Pan, K. Ji and J. Zhao are with TuSimple, Beijing, China. Emails: {liangliang.pan, kaijin.ji}@tusimple.ai, zhaoji84@gmail.com. J. Zhao is the corresponding author.

II. RELATED WORK

A. Vision-based Localization

The vision-based localization scheme [1], [2], [3] is widely used in structured scenes with rich semantic constraints. It mainly aligns the semantic information (such as road lane, traffic light, street signs, etc) in an image with the HD map for vehicle's pose estimation. Ma. et al proposed a vision-based high-definition semantic localization method [3] by fusing measurements from GPS, IMU, wheel encoders and cameras.

B. LiDAR-based Localization

LiDAR-based localization methods associate features of current LiDAR frame with a global map, and optimize poses using feature correspondences. There are mainly three categories.

Intensity-based methods: Levinson et al. first proposed an intensity-based localization approaches [4], which integrates measurements from GPS, IMU, and wheel encoder. It was later improved by [5], in which the map is upgraded from reflectivities of 2D grids to Gaussian distributions of reflectivities. Wolcott and Eustice proposed a method which represents the map as a mixed Gaussian model containing both the intensity and altitude of a point cloud [14]. A recent work [13] separated the map representation of the mixed Gaussian distribution into two single Gaussian distributions to represent intensity and altitude attributes respectively.

Distribution-based methods: Normal distribution transform (NDT) is the representative method and has been widely used for point cloud registration and vehicle localization [6], [7], [15], [16]. It first divided point cloud into regular voxels, and points in each voxel are approximated by a 3D Gaussian distribution. Then the pose between two point clouds is optimized by maximizing the joint Gaussian probability density of all voxels.

Geometric features based methods: Zhang and Singh proposed a method called LOAM [8], [9] for LiDAR odometry. First it extracts distinctive geometric features, and then aligns the current geometry features with a LiDAR feature map using Iterative Closest Point (ICP) [17]. The features in LOAM include edge and surface features, and they are extracted by calculating curvature of the original point cloud. A low-level semantic segmentation-based LiDAR localization method was proposed in [18]. It relies on features including ground, road curb, edge, and surface, and it matches these features with a pre-build feature map. Despite the success of these methods, all the aforementioned methods rely on the stable association of salient geometric features and are prone to degradation in scenes with sparse structures.

C. Sensor Fusion Localization Frameworks

Sensor fusion methods mainly include filtering-based and optimization-based methods. Popular filters include extended Kalman filter (EKF) and particle filter (PF). EKF-based localization methods are widely used in robotics [16][18][19], while the linearization in its prediction stage would lose certain accuracy. PF-based localization methods [4], [13], [15] are more robust than EKF-based methods. While the random sampling in a PF is time-consuming and a PF usually maintains three-dimensional states only (horizontal position and yaw). The batch optimization approach [20] maintains a set of historical observations and performs a nonlinear optimization over the past several states to produce more accurate results. In [11], it integrates local LiDAR constraints and IMU for vehicle localization. However, this method lacks absolute observation constraints and suffers from cumulative errors, making it unsuitable for long-term vehicle localization. In [21], a loosely-coupled method was proposed, in which the relative pose was estimated by an LIO module independently of the global observations and was aligned to the global frame via pose-graph optimization.

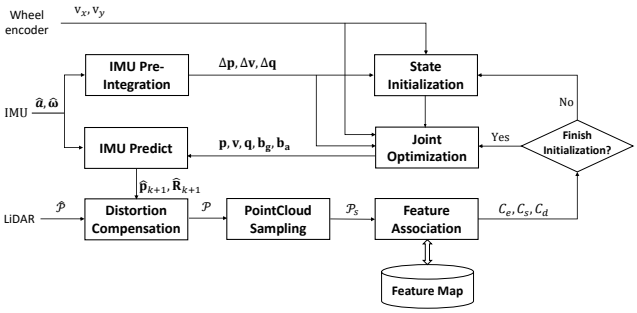

Fig. 2. Tightly-coupled multi-sensor fusion localization framework.

III. TIGHTLY-COUPLED SENSOR FUSION FRAMEWORK

A. System Review

The flowchart of the proposed method is shown in Fig. 2. We assume all intrinsic parameters of sensors are known, and the extrinsic parameters between an LiDAR, an IMU, and wheel encoders have been calibrated in advance. One exception is that extrinsic parameter of LiDAR and IMU are refined online. All sensors are triggered by hardware for clock synchronization. We also assume the IMU frame coincides with the body frame for convenience. Thus all state variables are expressed and optimized in the IMU frame.

The notations in this paper are summarized in Table I. We use a fixed-lag smooth optimizer. The maintained state variables in a sliding window are summarized as

$$\begin{cases} \mathbf{X}_{k-m+1:k} = [\mathbf{X}_{k-m+1}, \mathbf{X}_{k-m+2}, \ldots, \mathbf{X}_k], \\ \mathbf{X}_k = \begin{bmatrix} \mathbf{p}_k^T & \mathbf{v}_k^T & \mathbf{q}_k^T & \mathbf{b}_{a_k}^T & \mathbf{b}_{g_k}^T & \mathbf{p}_{lb}^T & \mathbf{q}_{lb}^T \end{bmatrix}^T, \end{cases} \tag{1}$$

where $\mathbf{p}_k$ and $\mathbf{v}_k$ are IMU position and velocity relative to world frame $\mathcal{F}_W$. $\mathbf{q}_k$ is a unit quataternion [22] associate with rotation matrix $\mathbf{R}_k$ which rotates vector from body frame $\mathcal{F}_B$ to global frame $\mathcal{F}_W$. $\mathbf{b}_{a_k}$ and $\mathbf{b}_{g_k}$ represent the accelerometer and gyroscope bias at time k. In addition, we estimate the extrinsic parameters (rotation $\mathbf{q}_{lb}$ and translation $\mathbf{p}_{lb}$) from $\mathcal{F}_B$ to $\mathcal{F}_L$ online. An illustration of the proposed sliding window method is shown in Fig. 3.

TABLE I

NOTATIONS AND EXPLANATION

Notation	Explanation
$\mathcal{F}$	$\mathcal{F}_B$, $\mathcal{F}_W$ and $\mathcal{F}_L$ represent body frame of IMU, world frame, and LiDAR frame, respectively.
$\mathbf{T}$	$\mathbf{T}_b^a \in \mathrm{SE}(3)$ represents transformation from $\mathcal{F}_b$ to $\mathcal{F}_a$. $\mathbf{T}_B^L = [\mathbf{p}_B^L, \mathbf{q}_B^L]$ is the extrinsic parameter From $\mathcal{F}_B$ to $\mathcal{F}_L$.
$\mathbf{R}, \mathbf{q}$	$\mathbf{R}_b^a \in \mathrm{SO}(3)$ represents the rotation matrix from $\mathcal{F}_b$ to $\mathcal{F}_a$. $\mathbf{q}_b^a$ is the Hamilton unit quaternion of $\mathbf{R}_b^a$.
$\mathbf{p}$	$\mathbf{p}_b^a \in \mathbb{R}^3$ represents the translation from $\mathcal{F}_b$ to $\mathcal{F}_a$.
$\mathbf{v}$	$\mathbf{v} \in \mathbb{R}^3$, linear velocity.
$\hat{\mathbf{v}}$	velocity observed by wheel encoders.
$\mathbf{b}$	$\mathbf{b} \in \mathbb{R}^3$, $\mathbf{b}_{a_k}$ and $\mathbf{b}_{g_k}$ is the bias of accelerometer and gyroscope at time k, respectively.
$\mathbf{n}$	$\mathbf{n} \in \mathbb{R}^3$, $\mathbf{n}_{a_k}$ and $\mathbf{n}_{g_k}$ is the noise of accelerometer and gyroscope at time k, respectively.
$\mathbf{X}$	$\mathbf{X}_k$ is the state vector of a vehicle at time k.
$\mathcal{U}$	$\mathcal{U} = \{\mathbf{u}_i\}$, $i \in \{1, 2, \cdots, m\}$ is a series of sensor input.
$\mathcal{Z}$	$\mathcal{Z} = \{\mathbf{z}_k\}_k$, $k \in \{1, 2, \cdots, n\}$ is a series of observations from IMU, LiDAR, and wheel encoder.
$\hat{\mathbf{a}}$, $\hat{\boldsymbol{\omega}}$	raw IMU measurements, which correspond to acceleration and angular velocity in $\mathbb{R}^3$. $\mathbf{a}$ and $\boldsymbol{\omega}$ correspond to acceleration and angular velocity without bias.
$\mathbf{g}$	gravitational acceleration in $\mathcal{F}_W$.
Σ	information matrix.
$\widehat{\mathcal{P}}$	point cloud. $\mathcal{P}$ is point cloud without motion distortion. $\mathcal{P}_\mathcal{S}$ represent the sampled LiDAR features.
$\mathcal{C}$	$\mathcal{C}_e$, $\mathcal{C}_s$ and $\mathcal{C}_d$ represent global LiDAR constraints of edge, surface, and normal distribution features, respectively.
$\mathcal{B}$	constraints of IMU factor.
$\mathcal{V}$	constraints of velocity factor introduced by wheel encoders.

B. LiDAR Feature Map Generation

To perform online localization, a scalable high-precision map should be built in advance. Our map contains three kinds of features: edge features, surface features, and 3D distributions. Compared with the sparse 3D point cloud or NDT map, our maps contain both geometric features (edge and surface) and 3D normal distribution features, enabling our method to obtain robust and accurate feature association. The map generation contains the following three steps:

LiDAR pose generation: We use an offline SLAM framework based on graph optimization to fully fuse observations from various sensors, including high-precision GNSS data with post-processing, an IMU, wheel encoders, and two LiDARs, to generate accurate LiDAR poses. Discrete laser scans are aligned with LiDAR poses to generate point cloud maps with global consistency. Since the generation of LiDAR poses is beyond the scope of this work, we would ignore specific details.

Undistortion of point cloud: Inspired by related work [23], under the assumption of uniform motion between consecutive timestamps, we use the motion prediction provided by an IMU and wheel encoders as a motion model to undistort the point cloud.

Parametrization of a feature map: After a globally consistent point cloud map is obtained, we extract salient LiDAR features from the dense point cloud map. First we divide the point cloud map into voxels, and calculate the mean and covariance matrix for each voxel. Then we classify a voxel as an edge, a surface or a normal distribution feature by investigating eigenvalues of the covariance matrix.

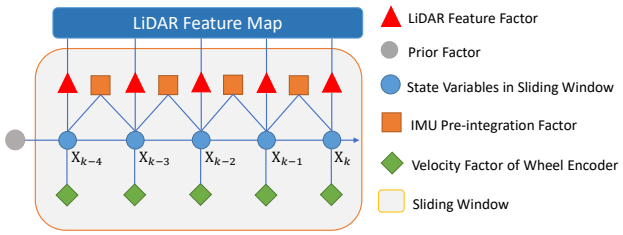

Fig. 3. Overview of our proposed tightly-coupled multi-sensor fusion. There are four types of factors: global LiDAR factor (red triangle), pre-integration factor (orange square), velocity factor (green prism), and prior factor (grey circle). The state variables to be optimized are depicted with blue circle.

Specifically, having two small and one large eigenvalues indicates an edge feature; having one small and two large eigenvalues indicates a surface feature; otherwise it indicates a normal distribution feature. Finally, the equations of line features, normal vectors of surface features, the mean and covariance matrix of normal distribution features are calculated to parameterize the corresponding voxels, respectively.

C. IMU Pre-integration Factor

Once the current LiDAR frame and raw IMU data are obtained, a set of raw IMU data between two consecutive LiDAR scans are propagated using Euler integration. The raw IMU observation contains the acceleration and angular velocity with bias. The observation model of an IMU at timestamp k is

$$\begin{cases} \widehat{\boldsymbol{\omega}}_k = \boldsymbol{\omega}_k + \mathbf{b}_{g_k} + \mathbf{n}_{g_k}, \\ \widehat{\mathbf{a}}_k = \mathbf{a}_k - \mathbf{R}_k^{-1}\mathbf{g} + \mathbf{b}_{a_k} + \mathbf{n}_{a_k}. \end{cases} \tag{2}$$

And the predicted state at timestamp $k+1$ is

$$\begin{cases} \widehat{\mathbf{R}}_{k+1} = \mathbf{R}_k \exp((\widehat{\boldsymbol{\omega}}_k - \bar{\mathbf{b}}_g)\Delta t), \\ \widehat{\mathbf{p}}_{k+1} = \mathbf{p}_k + \mathbf{v}_k \Delta t + \frac{1}{2}(\mathbf{R}_k(\widehat{\mathbf{a}}_k - \bar{\mathbf{b}}_a) + \mathbf{g})\Delta t^2, \end{cases} \tag{3}$$

where $\mathbf{R}_k$ represents the rotation matrix from $\mathcal{F}_B$ to $\mathcal{F}_W$ at time k; Δt is the time interval between timestamp k and $k+1$; $\exp(\cdot)$ is the exponential map from $\mathfrak{so}(3) \to \mathrm{SO}(3)$; $\bar{\mathbf{b}}_a$ and $\bar{\mathbf{b}}_g$ are the gyroscope and accelerometer bias of the last timestamp, respectively.

Moreover, the pre-integration method [24] is used to obtain the pre-integrated variables $\Delta\mathbf{p}$, $\Delta\mathbf{v}$, and $\Delta\mathbf{q}$, which represent increments of displacement, velocity, and rotation, respectively. The residual $\mathbf{r}_\mathcal{B}$ of IMU pre-integration [24] is defined as

$$\mathbf{r}_\mathcal{B}(\mathbf{X}; \mathcal{Z}) = \begin{bmatrix} \mathbf{R}_k^T(\mathbf{p}_{k+1} - \mathbf{p}_k - \mathbf{v}_k\Delta t - \frac{1}{2}\mathbf{g}\Delta t^2) \\ \mathbf{R}_k^T(\mathbf{v}_{k+1} - \mathbf{g}\Delta t) - \Delta\mathbf{v} \\ 2[\Delta\mathbf{q}^{-1} \otimes \mathbf{q}_k^{-1} \otimes \mathbf{q}_{k+1}]_{xyz} \\ \mathbf{b}_{a_{k+1}} - \mathbf{b}_{a_k} \\ \mathbf{b}_{g_{k+1}} - \mathbf{b}_{g_k} \end{bmatrix} \tag{4}$$

where "$\otimes$" represents quaternion multiplication, and $[\cdot]_{xyz}$ stands for the imaginary part of a quaternion.

The IMU prediction module provides an initial guess for feature association and joint optimization, which is crucial to improve the robustness and accuracy of our localization method.

D. LiDAR Feature Factor

Once raw point cloud $\widehat{\mathcal{P}}$ is received, first we compensate the distortion caused by rapid movement of the vehicle. Then we obtain a set of undistorted point clouds $\mathcal{P}$ in the LiDAR frame relative to the beginning of the laser scan. Traditional geometric feature extraction methods [8][9][11] calculate curvature of hundreds of thousands of points in original point cloud, and select the edge and surface features of the current frame by curvature. In contrast, we uniformly sample limited points $\mathcal{P}_{\mathcal{S}}$ from a dense point cloud as candidate feature points, and then calculate the smoothness for each sampled points [9] which is used for subsequent geometric verification of feature association. Our sampling strategy can save much processing time while the accuracy and robustness of feature association remain unaffected.

After sampling a set of candidate feature points $\mathcal{P}_{\mathcal{S}}$, first we transform them into global frame by an initial guess from motion prediction. Then we associate those points with the pre-build map. The association will be fixed in the subsequent optimization. If the transformed query point is located at a voxel of a specific feature (edge, surface, or normal distribution) and its curvature meets the threshold range of the corresponding feature, it passes the geometric verification. For points that passed the geometric verification, the corresponding observation model is used to construct observation constraints. The residuals of three types of LiDAR features are formulated as below

$$\begin{cases} \mathbf{r}_{\mathcal{C}_e}(\mathbf{X}_k; \mathbf{p}_i^{pt}) = (\mathbf{p}_2^e - \mathbf{p}_1^e) \times (\mathbf{R}_k \mathbf{p}_i^{pt} + \mathbf{p}_k - \mathbf{p}_1^e), \\ \mathbf{r}_{\mathcal{C}_s}(\mathbf{X}_k; \mathbf{p}_i^{pt}) = \boldsymbol{n}^T (\mathbf{R}_k \mathbf{p}_i^{pt} + \mathbf{p}_k), \\ \mathbf{r}_{\mathcal{C}_d}(\mathbf{X}_k; \mathbf{p}_i^{pt}) = (\mathbf{R}_k \mathbf{p}_i^{pt} + \mathbf{p}_k - \mathbf{p}^d) \Sigma^{-1} (\mathbf{R}_k \mathbf{p}_i^{pt} + \mathbf{p}_k - \mathbf{p}^d), \end{cases} \tag{5}$$

where $\mathbf{r}_{\mathcal{C}_e}$, $\mathbf{r}_{\mathcal{C}_s}$, and $\mathbf{r}_{\mathcal{C}_d}$ represent the geometric residuals of point to edge, point to surface, and point to normal distribution features, respectively. $\mathbf{R}_k$ and $\mathbf{p}_k$ represent the rotation matrix and translation to be optimized. $\mathbf{p}_i^{pt}$ represents the 3D coordinates of a feature point in current LiDAR frame. $\mathbf{p}_1^e$ and $\mathbf{p}_2^e$ represent arbitrary two points on the corresponding edge feature. $\boldsymbol{n}$ is the normal vector of the corresponding surface feature in the map. $\mathbf{p}^d$ and Σ^{-1} represent the mean and covariance of the points in a normal distribution feature.

Given a set of sampled feature points $\mathcal{P}_{\mathcal{S}}$, the loss of LiDAR factor is

$$E_{\mathcal{C}}(\mathbf{X}_k; \mathcal{Z}) = \sum_{i \in \mathcal{C}_e} \| \mathbf{r}_{\mathcal{C}_e}(\mathbf{X}_k; \mathcal{Z}_{ki}^e) \|^2_{\Sigma^e_{k,map}} + \sum_{i \in \mathcal{C}_s} \| \mathbf{r}_{\mathcal{C}_s}(\mathbf{X}_k; \mathcal{Z}_{ki}^s) \|^2_{\Sigma^s_{k,map}} + \sum_{i \in \mathcal{C}_d} \| \mathbf{r}_{\mathcal{C}_d}(\mathbf{X}_k; \mathcal{Z}_{ki}^d) \|^2_{\Sigma^d_{k,map}}, \tag{6}$$

where $\mathbf{r}_{\mathcal{C}_e}(\mathbf{X}_k; \mathcal{Z}_{ki}^e)$, $\mathbf{r}_{\mathcal{C}_s}(\mathbf{X}_k; \mathcal{Z}_{ki}^s)$ and $\mathbf{r}_{\mathcal{C}_d}(\mathbf{X}_k; \mathcal{Z}_{ki}^d)$ represent residuals associate edge feature $\mathcal{Z}_{ki}^e$, surface feature $\mathcal{Z}_{ki}^s$ and normal distribution feature $\mathcal{Z}_{ki}^d$ with corresponding map voxels, respectively. See Eq. (5) for details. $\Sigma^e_{k,map}$, $\Sigma^s_{k,map}$, $\Sigma^d_{k,map}$ represent the information matrices of edge, surface and normal distribution features in k-th frame, respectively. The diagonal elements of these matrices reflect the registration accuracy with the map.

E. Velocity Factor of Wheel Encoders

An observation of a wheel encoder includes velocity $[v_x, v_y]$ on horizontal plane and yaw rate $\dot{\theta}$ in body frame, which is obtained by fusing the raw pulse data of the wheel encoder with the Ackermann motion model. Since wheel encoders are relatively independent of LiDAR and IMU, tightly coupling the observations of wheel encoders into localization is crucial to improve robustness. Wheel encoders are especially useful for scenarios when other sensors are degraded, such as the LiDAR is largely occluded by dynamic objects or the IMU is under degradation caused by uniform motion [25]. The velocity constraint of wheel encoders can effectively bound the velocity and the bias of gyroscope and accelerometer, which can effectively improve the convergence of the optimization problem, and enhance the localization accuracy and robustness.

The residual $\mathbf{r}_{\mathcal{V}}$ of the wheel encoder about partial velocity variables is defined as

$$\mathbf{r}_{\mathcal{V}}(\mathbf{R}_k; \hat{\mathbf{v}}_k) = \mathbf{v}_k - \mathbf{R}_k \hat{\mathbf{v}}_k, \tag{7}$$

where $\hat{\mathbf{v}}_k = [v_x, v_y, 0]^T$ is the observation of wheel encoder in body frame $\mathcal{F}_B$.

F. State Initialization

The state variables that need to be initialized include the IMU velocity $\mathbf{V}_{1:m}$, the gravity vector $\mathbf{g}$, gyroscope bias $\mathbf{b}_g$, and acceleration bias $\mathbf{b}_a$. We perform joint optimization for LiDAR and wheel encoder observations in a sliding window. First, we obtain motion prediction from an IMU for rotation and wheel encoders. Then the LiDAR features (including edge, surface, normal distribution features) of each LiDAR frame in the sliding window are associated with a feature map by motion prediction mentioned in Section III-C. Finally, we formulate the constraints of LiDAR features and motion constraints between frames into a nonlinear optimization problem and solve the LiDAR pose of each frame by

$$\mathbf{T}_k^w = \underset{\mathbf{T}_k^w}{\arg\min} E_{\mathcal{C}}(\mathbf{X}_k; \mathcal{Z}) + \| \mathbf{r}_{odo}(\mathbf{X}_k; \hat{\mathbf{v}}_k) \|^2_{\Sigma^v_{k,k+1}}. \tag{8}$$

The loss $E_{\mathcal{C}}(\mathbf{X}_k; \mathcal{Z})$ introduced by the LiDAR factor is defined in Eq. (6). The residual for motion constraint of wheel encoders is defined as

$$\mathbf{r}_{odo}(\mathbf{X}_k; \hat{\mathbf{v}}_k) = \mathbf{p}_{k+1} - \mathbf{R}_{k+1} \mathbf{R}_k^T \mathbf{p}_k - \hat{\mathbf{v}}_k \Delta t. \tag{9}$$

Once the LiDAR poses in a sliding window are obtained, we calculate the gyroscope bias $\mathbf{b}_g$, gravity vector $\mathbf{g}$, and velocity $\mathbf{v}$ of each frame in the sliding window by following the initialization strategy of [26].

G. Tightly-Coupled Multi-Sensor Fusion

We propose a joint optimization framework based on a fixed-lag smoother, as shown in Fig. 3. The fixed-lag smoother is an iterative algorithm, which alternately updates state variables and marginalizes old constrains in a sliding window [20]. It recursively maintains an estimate of the total probability density of the state variables of last m timestamps, which is an incremental state estimation method based on a graph optimization [27]. The joint optimization can be formulated as a maximum a posterior (MAP) problem

$$\mathbf{P}(\mathbf{X}|\mathcal{U},\mathcal{Z}) \propto \mathbf{P}\left(\mathbf{X}_0\right)\prod_{k=1}^{m}\mathbf{P}\left(\mathbf{X}_k|\mathbf{X}_{k-1},\mathcal{U}_k\right)\prod_{k=1}^{m}\prod_{i=1}^{n}\mathbf{P}\left(\mathcal{Z}_{ki}|\mathbf{X}_k\right), \quad (10)$$

where k indexs over timestamps, and i over observations. $\mathbf{P}(\mathbf{X}_0)$ is the prior of initial state; $\mathbf{P}\left(\mathbf{X}_k|\mathbf{X}_{k-1},\mathcal{U}_k\right)$ is the motion model constraint; $\mathbf{P}\left(\mathcal{Z}_{ki}|\mathbf{X}_k\right)$ represents the observation model from the i-th sensor.

Under the assumption of zero-mean Gaussian noise for each model, we can convert the MAP estimation problem into a nonlinear least-squares problem

$$\begin{aligned}\mathbf{X}^* = &\arg\min_{\mathbf{X}} \left\|\mathbf{r}_{prior}(\mathbf{X})\right\|^2 + \sum_{k=1}^{m}\sum_{i\in\mathcal{C}} E_{\mathcal{C}}\left(\mathbf{X}_k;\mathcal{Z}\right) \\ &+ \sum_{k=1}^{m}\sum_{i\in\mathcal{B}} \left\|\mathbf{r}_{\mathcal{B}}\left(\mathbf{X}_k;\mathcal{Z}\right)\right\|^2_{\Sigma^B_{k,k+1}} \\ &+ \sum_{k=1}^{m}\sum_{i\in\mathcal{B}} \left\|\mathbf{r}_{\mathcal{V}}\left(\mathbf{X}_k;\mathcal{Z}\right)\right\|^2_{\Sigma^V_{k,k+1}}, \end{aligned} \quad (11)$$

where $\mathbf{r}_{prior}$, $\mathbf{r}_{\mathcal{C}}$, $\mathbf{r}_{\mathcal{B}}$, and $\mathbf{r}_{\mathcal{V}}$ corresponds to prior factor, LiDAR factor, IMU pre-integration factor, and velocity factor of wheel encoders, respectively. $\Sigma^B_{k,k+1}$ in Eq. (11) and $\Sigma^V_{k,k+1}$ are information matrices for the IMU factor and velocity factor, respectively.

The LiDAR factor, IMU factor, and velocity factor of wheel encoders has been introduced in previous subsections. The initial values of the state variables are obtained by the state initialization method described in section III-F. The joint optimization problem mainly converts the constraints from different factors and priors into a joint optimization problem.

IV. Experiments

One of our test vehicles is shown in Fig. 4. The vehicle is equipped with a NovAtel GNSS RTK receiver integrated with a high-precision commercial IMU and dual GNSS antenna, two 40-beam LiDARs, two wheel encoders, and multiple cameras. In our experiments, the ground truth of localization is estimated by a pose graph optimization method, which fuses NovAtel post-processing data, LiDAR scans, wheel encoder reading, and GNSS loop closure constraints. The ground truth can be considered sufficiently accurate. Our method uses a single LiDAR sensor, an IMU, and two wheel encoders. Note that NovAtel is only used to generate the ground truth.

Fig. 4. Our test vehicle is equipped with two 40-beam LiDARs, a NovAtel GNSS RTK receiver integrated with an EG320N IMU and dual GNSS antenna, two wheel encoders, and multiple cameras.

We would like to verify the effectiveness of the proposed localization method in different real scenarios. However, there is no publicly available outdoor datasets for localization evaluation, which contains a set of multi-sensor raw observation and centimeter-level ground-truth. So we collected 5 scenes from our road test data over 200km, which cover scenes in different counties. As shown in Fig. 5, the scenes include highway, rainy highway, sea-crossing bridge, urban local road sections, and tunnel scenes.

In the following experiments, we fix all the parameters in our method. In each LiDAR frame, 1500 candidate feature points are sampled. The resolution of the feature map is 0.5m, i.e., the voxel size is 0.5m×0.5m×0.5m. The size of the sliding window is 3. Our method is implemented in C++. All optimization problems in our method is solved by Ceres solver [28].

A. Localization Performance

To evaluate the localization performance comprehensively, we adopted multiple evaluation metrics. The average longitudinal error, lateral error, Euclidean 3D error, and heading error are used, which reflect the localization accuracy. The lateral error is especially important for autonomous driving safety. In addition, the stability of the localization result is important for perception and planner decision. The oscillation of localization will bring potential safety hazards to autonomous driving. So we also used the smoothness of localization introduced by [3], which is defined as the difference between the temporal gradient of the ground truth and that of the estimated poses.

We use high-precision RTK (real-time kinematic) results as a baseline for localization accuracy verification. The performance comparison is shown in Table II. Bold font numbers represent better performance.

Accuracy: It can be seen that our method achieves smaller localization errors than the baseline across all metrics in most of the test sequences. It is notable that our method achieves an average lateral error of 1.6cm, average longitudinal error of 6.0cm, and average heading error of 0.027°.

Robustness: Our method greatly improves the performance over the more challenging scenario (rainy scenes

Fig. 5. Sample images from 5 test sequences. From left to right: highway, rainy local scene, rainy bridge, urban local, and tunnel.

TABLE II
LOCALIZATION PERFORMANCE ON REAL-WORLD TEST SEQUENCES

Scenes	dist (km)	method	Lat.(m)		Long.(m)		Euclidean 3D (m)		Heading (deg)		Lat.	Long.	Lat.	Long.
			mean	max	mean	max	mean	max	mean	max	<0.1m(%)	<0.1m(%)	Smooth	Smooth
Highway	83.95	ours	**0.007**	**0.065**	**0.016**	**0.141**	**0.020**	**0.141**	**0.021**	**0.141**	**100.00**	**99.983**	**0.0094**	**0.0107**
		RTK	0.047	0.192	0.364	1.505	0.460	1.542	0.388	0.550	95.674	28.878	0.4214	0.3340
Rain_local	7.39	ours	**0.008**	**0.104**	**0.033**	0.991	**0.039**	0.992	**0.039**	0.993	99.983	**94.918**	**0.0179**	**0.0216**
		RTK	0.021	0.340	0.230	**0.477**	0.460	**0.640**	0.105	**0.451**	**100.00**	4.500	0.0420	0.0300
Rain_bridge	22.75	ours	**0.007**	**0.187**	**0.027**	**0.846**	**0.045**	**0.863**	**0.045**	0.663	**99.771**	**97.788**	**0.0099**	**0.0176**
		RTK	0.021	0.240	0.392	3.530	0.524	3.54	0.123	**0.451**	99.335	3.426	0.0808	0.1684
Urban_local	9.08	ours	**0.017**	**0.095**	**0.017**	**0.104**	**0.028**	**0.120**	**0.029**	**0.190**	**100.00**	**99.909**	**0.0091**	**0.0094**
		RTK	0.051	0.138	0.418	4.658	0.447	4.659	0.291	0.494	94.879	19.784	0.3744	0.4704
Tunnel	1.04	ours	**0.008**	**0.033**	**0.044**	**0.515**	**0.049**	**0.516**	**0.049**	**0.516**	**100.00**	**88.672**	**0.0252**	**0.0212**
		RTK	0.111	1.156	0.257	2.247	0.369	2.650	0.226	0.682	84.246	30.797	0.1022	0.1115

TABLE III
EUCLIDEAN 3D ERROR FOR ABLATION STUDY. (UNIT: METER)

sequence	liw_full		S1		S2		S3	
	mean	max	mean	max	mean	max	mean	max
Highway	**0.020**	**0.141**	0.022	0.178	Failed		0.021	0.211
Rain_local	**0.039**	0.992	0.042	**0.944**	0.126	2.208	Failed	
Rain_bridge	0.045	**0.863**	0.048	1.744	0.116	4.992	**0.0404**	1.689
Urban_local	**0.028**	**0.120**	0.032	0.158	0.039	0.145	0.037	0.191
Tunnel	**0.049**	0.516	0.058	**0.399**	Failed		0.067	0.691

TABLE IV
RUNTIME OF MODULES IN OUR METHOD

Total	Distortion Compensation	Feature Sampling	Data Association	Joint Optimization
69.4 ms	8.4 ms	2.1 ms	3.1 ms	55.6 ms

and tunnels) in terms of longitudinal/altitude error percentage, heading error percentage, and smoothness. Specifically, frames with a lateral error of less than 10cm account for more than 99%, and frames with a longitudinal error less than 20cm account for $92\% \sim 97\%$ in all scenes. The smoothness of the localization results is in the order of 0.01. Putting aside the difference of scenes, we have an order of magnitude higher smoothness than the results in [3].

B. Ablation Study

To verify the effect of different factors on localization results, we reduce the motion/observation constraints of different sensors to verify the localization accuracy and robustness of the method. Different settings are define below. **liw_ful** uses all constraints; **S1** uses LiDAR's geometric features (edge, surface), IMU factor and velocity factor; **S2** uses LiDAR normal distribution features, IMU factor and velocity factor; **S3** uses all LiDAR features and IMU factor of LiDAR (without velocity factor). The results of ablation study are shown in Table III. It can be seen that **liw_ful** and **S1** are robust to run all the test sequences. **liw_ful** setting has the smallest overall localization error. In contrast, **S2** and **S3** have failure cases. The results demonstrate the effectiveness of the multi-sensor fusion in our method.

C. Runtime and Memory Usage

Using default parameters, our method can achieve a real-time 10Hz frame rate on our computing platform (a desktop with Core i7-7700K CPU@4.2GHz and 16GB RAM). Table IV lists the runtime of main modules in our method. By adjusting the size of the sliding window and the sampling points of each LiDAR frame, our method is flexible to achieve trade-off between localization performance and efficiency.

V. CONCLUSIONS

This paper proposed a tightly-coupled localization framework with a LiDAR feature map, which fuses multi-sensor observations and motion constraints. A hybrid of LiDAR geometric features and normal distribution features are considered, which increases the stability of feature association and can improve the accuracy and robustness of localization in challenging scenarios. Our method simplifies the LiDAR preprocessing and feature association module, which enables our system achieving a real-time frame rate (10Hz) in real applications. The proposed method is a general optimization framework that can easily integrate other absolute and relative observations.

REFERENCES

[1] Markus Schreiber, Carsten Knöppel, and Uwe Franke, "LaneLoc: Lane marking based localization using highly accurate maps," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 449–454.

[2] Ryan W Wolcott and Ryan M Eustice, "Visual localization within LIDAR maps for automated urban driving," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014, pp. 176–183.

[3] Weichiu Ma, Ignacio Tartavull, Ioan Andrei Bârsan, Shenlong Wang, Min Bai, Gellert Mattyus, Namdar Homayounfar, Shrinidhi Kowshika Lakshmikanth, Andrei Pokrovsky, and Raquel Urtasun, "Exploiting sparse semantic HD maps for self-driving vehicle localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IR0S)*, 2019, pp. 5304–5311.

[4] Jesse Levinson, Michael Montemerlo, and Sebastian Thrun, "Map-based precision vehicle localization in urban environments.," in *Robotics: science and systems (RSS)*, 2007, vol. 4, p. 1.

[5] Jesse Levinson and Sebastian Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 4372–4378.

[6] Peter Biber and W Strasser, "The normal distributions transform: A new approach to laser scan matching," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003, vol. 3, pp. 2743–2748.

[7] Martin Magnusson, Henrik Andreasson, Andreas Nüchter, and Achim J. Lilienthal, "Automatic appearance-based loop detection from three-dimensional laser data using the normal distributions transform," *Journal of Field Robotics (JFR)*, vol. 26, no. 11-12, pp. 892–914, 2009.

[8] Ji Zhang and Sanjiv Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems Conference (RSS)*, 2014, pp. 109–111.

[9] Ji Zhang and Sanjiv Singh, "Laser–visual–inertial odometry and mapping with high robustness and low drift," *Journal of Field Robotics (JFR)*, vol. 35, no. 8, pp. 1242–1264, 2018.

[10] Kejian J Wu, Chao X Guo, Georgios Georgiou, and Stergios I Roumeliotis, "VINS on wheels," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5155–5162.

[11] Haoyang Ye, Yuying Chen, and Ming Liu, "Tightly coupled 3D LiDAR inertial odometry and mapping," in *International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3144–3150.

[12] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," *arXiv preprint arXiv:2007.00258*, 2020.

[13] Guowei Wan, Xiaolong Yang, Renlan Cai, Hao Li, Yao Zhou, Hao Wang, and Shiyu Song, "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 4670–4677.

[14] Ryan W Wolcott and Ryan M Eustice, "Robust LIDAR localization using multiresolution gaussian mixture maps for autonomous driving," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 3, pp. 292–319, 2017.

[15] Syed Zeeshan Ahmed, Vincensius Billy Saputra, Saurab Verma, Kun Zhang, and Albertus Hendrawan Adiwahono, "Sparse-3D lidar outdoor map-based autonomous vehicle localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, 2019, pp. 1614–1619.

[16] Shibo Zhao, Zheng Fang, Haolai Li, and Sebastian Scherer, "A robust laser-inertial odometry and mapping method for large-scale highway environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS)*, 2019, pp. 1285–1292.

[17] Paul J Besl and Neil D McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 14, pp. 239–256, 1992.

[18] Hang Liu, Qin Ye, Hairui Wang, Liang Chen, and Jian Yang, "A precise and robust segmentation-based lidar localization system for automated urban driving," *Remote Sensing*, vol. 11, no. 11, pp. 1348, 2019.

[19] Xingxing Zuo, Wenlong Ye, Yulin Yang, Renjie Zheng, Teresa Vidalcalleja, Guoquan Huang, and Yong Liu, "Multimodal localization: Stereo over LiDAR map," *Journal of Field Robotics (JFR)*, vol. 37, no. 6, pp. 1003–1026, 2020.

[20] Tue Cuong Dong-Si and Anastasios I. Mourikis, "Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 5655–5662.

[21] Wendong Ding, Shenhua Hou, Hang Gao, Guowei Wan, and Shiyu Song, "LiDAR inertial odometry aided robust LiDAR localization system in changing city scenes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[22] Joan Sola, "Quaternion kinematics for the error-state Kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[23] Chi Hay Tong, Sean Anderson, Hang Dong, and Timothy D. Barfoot, "Pose interpolation for laser-based visual odometry," *Journal of Field Robotics (JFR)*, vol. 31, no. 5, pp. 731–757, 2014.

[24] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics (TRO)*, vol. 33, no. 1, pp. 1–21, 2016.

[25] Kejian J Wu and Stergios I Roumeliotis, "Unobservable directions of VINS under special motions," Tech. Rep., University of of Minnesota, 2016.

[26] Tong Qin, Peiliang Li, and Shaojie Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics (TRO)*, vol. 34, no. 4, pp. 1004–1020, 2018.

[27] Frank Dellaert, Michael Kaess, et al., "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[28] Sameer Agarwal, Keir Mierle, and Others, "Ceres solver," `http://ceres-solver.org`.