

# LRMO: A Lightweight and Redundant Multi-Modal Odometry Framework for Robust Intelligent Vehicle Localization

Xinye Dai<sup>1</sup>, Liang Chen<sup>2</sup>, Zhiyong Tu<sup>1</sup>, Shiqi Zheng<sup>1</sup>, Shujie Zhou<sup>1</sup>, Fenfen Lin<sup>1</sup> and Weiwei Song<sup>1</sup>

<sup>1</sup> GNSS Research Center, Wuhan University, Wuhan China 430079, (xinyed, 2012301650022, shiqizheng, sizhou2020, mulin33, sww)@whu.edu.cn

<sup>2</sup> Beijing Institute of Tracking and Telecommunication Technology, Beijing, China, 100094, chenliang@beidou.gov.cn

**Keywords:** Intelligent vehicles, localization, multisensory integration.

## Abstract:

Reliable and robust self-localization is the essential component of intelligent vehicles (IV). Many scholarly works have been focused on developing accurate multi-modal integrated pose estimation schemes. Such single estimation engine design lacks consideration of potential individual sensor failures. In this paper, we present a resilient framework that exploits the redundancy of different sensors using a stack of odometry algorithms. The multiple pose estimation algorithms run in parallel with a general adaptivity and lightweight design. Specifically, we integrate the vehicle wheel encoder and the vehicle dynamics data to the filter-based LiDAR-inertial odometry. In contrast to most of the odometry algorithms which may fail entirely against temporary failures, the redundant system enables self-recovery of individual odometry through reinitialization. The most promising odometry is selected at each timestamp through weighting metric evaluation. In this way, our method can exploit the robustness and advantages of individual estimating engines. We evaluate our method on both research purpose IVs and mass-produced IVs. The experimental results suggest that our approach is resilient to various failure cases and achieves better performance than individual methods.

## 1. INTRODUCTION

Companies involved in the automotive industry and research laboratories have been aiming at developing more and more intelligent vehicles (IV). The ultimate goal is to enable fully autonomous vehicles driving worldwide. One of the essential functions for this dream is the accurate and robust self-localization. The most commonly known and adopted method is based on the global navigation satellite (GNSS). However, this system is not bulletproof, and suffers from multipath effect, electromagnetic interference, and signal blockages. Although such drawbacks can be partially solved through integration with inertial measurement unit (IMU) or wheel encoder, it still suffers from a significant drift in long-during operation. Therefore, intelligent vehicles nowadays are equipped with multiple sensors, such as radars, cameras, or light detection and ranging (LiDAR).

In the last decade, the visual or LiDAR based pose estimation systems have achieved remarkable results in different datasets [1], [2]. However, such frameworks may not operate stably under all conditions and application domains. The visual cameras are widely used for vehicle localization due to its low expenditure [3]–[5]. However, this sensor is limited to poorly illuminated conditions. To localize in such environments, LiDAR-based odometry [2], [6] is a promising solution due to its high-fidelity range measurements. In spite of this, these approaches are prone to fail in structure-less and repetitive environments such as planar long tunnels. Besides, the adverse weather conditions (fog, smoke, or dust) may hinder reliable LiDAR detections. Since these sensors are environment dependent, it is challenging to achieve robust performance if they are used as primary sensors to build a localization system. Therefore, a multi-modal fusion framework should be explored, which exploits the individual strengths of each sensor, and compensates for the drawbacks of the other.

In most cases, scholarly works focus on achieving better pose estimation results through designing a complex multi-sensor fusion framework. Such integration can be generally classified into either loosely coupled or tightly coupled methods. Loosely coupled odometry [7], [8] have been preferred more because of their simplicity and extendibility, where the pose estimations from individual measurements are fused separately. In contrast, tightly coupled frameworks [9], [10] directly fuse isolated sensor measurements in a joint manner and are usually difficult to extend to other sensors. Although having advantageous in accuracy, they may be vulnerable to cope with potential sensor failures since no redundancy method is considered. Besides, no recovery process is included in most schemes. On the other hand, loosely coupled methods distribute these risks into several subsystems and should have higher robustness.

This tricky conundrum about accuracy and robustness have been addressed by few studies in the literature [11]–[13]. The redundancy is realized with parallel running odometry methods, and the final state estimation can be selected from the average of parallel odometry [11] or the best results from sanity check [13]. However, these solutions are only designed for robots or typical datasets, the challenges regarding mass-produced IVs are not taken into consideration.

Limited computation resource: most multi-modal schemes are evaluated on laptops or high-performance workstations (e.g. Reinke et al. [13] used a 16 cores @ 3.6 GHz, i9-9900k CPU). In comparison, most IVs are equipped with automotive grade platforms (eg. 4 cores up to 2.1 GHz Qualcomm Snapdragon SA820A). Since the CPU is also in charge of complexed tasks such as controller area network (CAN) bus input/output, human interaction and display. The remained computation resource is highly limited for state estimation. Therefore, the commonly used iterative computation [2] is not suitable here.

A great variety of sensor setups: many multi-modal systems are designed for typical sensors, and difficult to extend to other

setups. Besides, as an indispensable sensor for most works, IMU is not included in many IVs. They utilize the wheel speed information from anti-lock braking system (ABS) and yaw rate data from electronic stability program (ESP) instead. In this phase, most widely applied algorithms cannot work for IVs.

Addressing the aforementioned problems, we seek to design a lightweight odometry framework for general IVs in this paper, that is, the IVs for research purpose and for mass-production. Instead of sticking to a specific method, our approach is built with redundancy, where various lightweight algorithms run in parallel and the most promising pose estimation is selected. Our proposed system presents the following contributions:

- 1) We propose to use redundant odometry to provide better pose estimation results instead of a single estimator. This scheme is lightweight and can reach real-time performance for IVs with limited computation resource. Besides, the robustness also benefits from the redundancy design.
- 2) We integrate the wheel odometer information to the iterated Kalman filter-based LiDAR-inertial system when IMU is available. On the other hand, the vehicle dynamics model is introduced when IMU is not available.
- 3) We add the self-recovery characteristic to both the LiDAR-inertial and Visual-inertial system. Unlike most of the algorithms which may fail entirely against temporary errors, our redundancy design can detect such errors and recover the individual systems.
- 4) We open source the datasets to benefit the community.

The rest of the paper is organized as follows: Section II illustrates the related works and comparisons. Section III presents the system overview along with some preliminary works, followed by detailed lightweight and general framework in Section IV. Then Section V shows experimental results on different platforms as well as some applications. Finally, Section VI concludes this paper and demonstrates some future research directions.

## 2. RELATED WORK

Vehicle pose estimation is a widely investigated topic in previous works. A great diversity of sensors and digital data have been utilized in this field, including inertial, wheel encoder [14], radars [15], GNSS [16], ultrasonic [17], LiDARs [6], and cameras [18]. However, most of them are merely focused on multi-modal fusion instead of a robust and redundant system to deal with sensor failures. In this section, we briefly review works on multi-modal SLAM and redundant odometry.

### 2.1 Multi-modal SLAM

The multi-modal SLAM can be generally classified into either loosely coupled or tightly coupled schemes. There are numerous works on loosely coupled odometry in the literature [2], [19], [20], where the pose estimations from individual measurements are fused separately. Since this scheme only performs frame-to-frame motion estimation, the global consistency is not guaranteed. In many of the recent works [1], [14], [21], [22], tight integration of multi-modal sensing

capabilities are explored for both accuracy and robustness improvement. The tight coupled manner can be classified into filter based or optimization-based frameworks. The filter based approaches employ a Kalman filter for joint state estimation, such as the multi-state constraint Kalman filter (MSCKF) applied in [26]. Although operating with high efficiency, the filter-based methods are usually less extendible to other sensors and may be vulnerable to potential sensor failures. Since the iterated Kalman filter based LiDAR-inertial scheme [21] only includes LiDAR and IMU for state estimation, we hereby extend this scheme with vehicle wheel odometer and vehicle dynamics for IV application. On the contrary, the optimization based approaches [10], [25] have proved advantageous for their expandability, where each sensor input can be viewed as a factor in the graph. Therefore, we construct a factor graph at the back-end of [21], where extra global measurements, such as the GNSS, can be added conveniently.

The aforementioned algorithms all have a single estimator design. Although achieving promising accuracy, they are prone to fail against either single sensor failures or estimation failures. The sensor failures include transitory loss of data stream, large outliers, or environment related problems (low illumination). The common estimation errors comprise of non-convergence, degeneracy, and high dynamic scenes [26]. Besides, most of them cannot recover from the temporary failures. Once certain failures are met, e.g., LiDAR degeneracy, they will fail entirely and generate meaningless pose estimation results. In this paper, we seek to add the self-recovery characteristic to the multi-modal SLAM algorithms.

### 2.2 Redundant Odometry

Previous works show that the multi vehicle design can cope with the single failures and improve the system robustness [27], [28]. Xu et al. [29] show that the cooperation of redundant multiple autonomous underwater vehicles can detect abnormal acoustic distance and retain the accurate information, ensuring a stable operation of the entire system. Besides, they also point out that the redundant system can estimate unknown parameters and reduce the influence of outliers to the estimation results [30]. Shan et al. [10] propose to use a health monitor system for each odometry, and they can easily switch to each other. The consequence of redundant odometry has also been emphasized by the Mars Exploration Rovers (MER) [31]. The MER system has successfully exhibited a switching behavior between the wheel, inertial, and visual odometry. The former two can compensate for the visual tracking failures, whereas the latter one can detect the unforeseen slips of the former two.

Our work follows the idea of [12] and [13]. They all share a same redundancy design, where various odometries are running in parallel, with the best estimation used as the output of the system. Specifically, the former work is targeted for the unmanned aerospace vehicles (UAVs), and the heterogeneous redundant odometry evaluates a stack of estimation algorithms through resiliency logic. It switches to an alternative method when the main estimation engine fails, along waiting for the main engine re-initialization. With a wide adaptability principle, this framework can be easily extended to a great

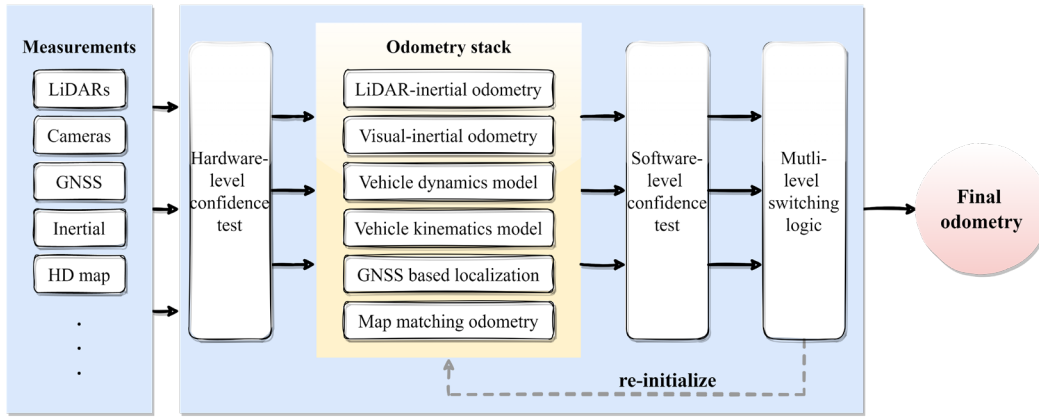


Fig. 1. The proposed system architecture.

variety of sensors, such as sonar, radar, LiDAR, camera, IMU and thermal. Since it is typically designed for UAVs, the vehicle dynamics and kinematics are not considered. On the contrary, the latter work focuses on the IV application, where multiple visual-LiDAR odometry run in parallel. The vehicle dynamics and kinematics is utilized to filter out failures, and a point cloud Chamfer distance-based criterion is used to find the most promising result for each timestamp. This scheme requires at least one LiDAR on the IV for computing weighting metric, which is not realistic for many mass-produced vehicles. Besides, it requires a high-performance laptop (i9-9900k CPU) for real-time operation, lacking efficiency for general IVs.

In this paper, we integrate the advantages of the above two schemes, with a general sensor adaptivity to both research purpose and mass-produced IVs. Besides, the parallel odometry all have a lightweight design for the resource limited IVs.

### 3. PRELIMINARY

The system diagram is visualized in Fig. 1. Since we seek to develop a generalized odometry scheme for IVs, the input information includes heterogeneous sensors and the high-definition (HD) maps. Such measurements have a diversity of physical phenomena to cope with the various possible failure cases. These inputs first undergo a hardware-level data test, including data stream existence, frequency, and individual verification. They are sent to parallel sub modules for individual pose estimation. The software-level test is then performed for separate results to remove clearly wrong results. Finally, the remaining odometry are evaluated with the best chosen as the output.

We first define the coordinates and notations used throughout this paper in Fig. 2 and TABLE I. In addition, we denote  $(\cdot)^B_L$  as the transformation from LiDAR frame to IMU frame.

We employ two mass produced IVs and two research purpose IVs throughout this paper. The general configurations of the IV (weight, CAD design model) are provided by the manufacturer. We use a calibration room to re-compute all the sensor extrinsic and the camera intrinsic. The GNSS antenna is treated as the primary sensor, which is set as the reference for all the sensor extrinsic. When GNSS is available at vehicle start point, all the results will be transferred into the global frame configured by GNSS. Once GNSS is temporarily unavailable, the pose estimation is only performed in the local coordinates, and wait

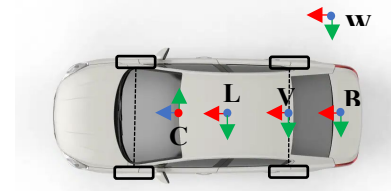


Fig. 2. Visualization of frame definitions. The red-green-blue color indicates the x-y-z coordinates, respectively. The original V frame is mounted at the vehicle mass center, and we shift it to the center of the rear axles. The B frame shares the same definition of IMU frame.

TABLE I  
 NOTATIONS THROUGHOUT THE PAPER

| Notations                | Explanations                                                                                                          |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Coordinates              |                                                                                                                       |
| $(\cdot)^W$              | The global frame used for vehicle navigation.                                                                         |
| $(\cdot)^V$              | The vehicle frame used for vehicle dynamics and kinematics.                                                           |
| $(\cdot)^B$              | The body frame which is also the IMU frame.                                                                           |
| $(\cdot)^O$              | The odometer frame, which is often expressed in B frame                                                               |
| $(\cdot)^L$              | The LiDAR frame, defined by the manufacturer.                                                                         |
| $(\cdot)^C$              | The camera frame, also defined by different manufacturer.                                                             |
| Expression               |                                                                                                                       |
| $(\cdot)$                | Noisy measurement or estimation of $(\cdot)$ .                                                                        |
| $\otimes$                | Multiplication between two quaternions.                                                                               |
| $\mathbf{p}$             | The position or translation vector.                                                                                   |
| $\mathbf{R}, \mathbf{q}$ | Two forms of rotation expression, $\mathbf{R} \in SO(3)$ is the rotation vector, $\mathbf{q}$ represents quaternions. |
| $\mathbf{p}$             | The linear velocity vector.                                                                                           |
| $\mathbf{x}$             | The vehicle full state vector.                                                                                        |
| $\mathbf{Z}$             | The full set of measurements.                                                                                         |
| $\eta$                   | The Gaussian noise.                                                                                                   |

for the GNSS initialization. We utilize either the network protocol or CAN bus to synchronize all the measurements.

### 4. MULTI-MODAL FRONT-END AND CANDIDATE ODOMETRY SELECTION

In this section, we employ six odometry methods to exploit the redundancy of multiple odometries for improving overall robustness and performance.

#### 4.1 Vehicle Kinematics

The raw accelerometer and gyroscope measurements,  $\hat{\mathbf{a}}$  and  $\hat{\boldsymbol{\omega}}$ , are given by:

$$\begin{aligned}\hat{\mathbf{a}}_k &= \mathbf{a}_k + \mathbf{R}_W^{B_k} \mathbf{g}^W + \mathbf{b}_{a_k} + \boldsymbol{\eta}_a, \\ \hat{\boldsymbol{\omega}}_k &= \boldsymbol{\omega}_k + \mathbf{b}_{\omega_k} + \boldsymbol{\eta}_\omega,\end{aligned}\quad (1)$$

where  $\boldsymbol{\eta}_a$  and  $\boldsymbol{\eta}_\omega$  are the zero-mean white Gaussian noise, with  $\boldsymbol{\eta}_a \sim \mathcal{N}(\mathbf{0}, \sigma_a^2)$ ,  $\boldsymbol{\eta}_\omega \sim \mathcal{N}(\mathbf{0}, \sigma_\omega^2)$ . The gravity vector in the world frame is denoted as  $\boldsymbol{g}^W = [0, 0, g]^T$ . Besides, the model of wheel odometer sensor is given by:

$$\boldsymbol{c}^{O_k} \hat{\mathbf{v}}^O = \mathbf{v}^O + \boldsymbol{\eta}_{s^O}, \quad (2)$$

where  $\boldsymbol{c}^{O_k}$  denotes the scale factor of the odometer modeled as random walk, with  $\boldsymbol{\eta}_{s^O} \sim \mathcal{N}(\mathbf{0}, \sigma_{s^O}^2)$ .

When an IMU is included in the vehicle, the vehicle state vector can then be denoted as:

$$\mathbf{x}_i = [\mathbf{q}_{B_i}^W \quad \mathbf{p}_{B_i}^W \quad \mathbf{v}_{B_i}^W \quad \mathbf{b}_a \quad \mathbf{b}_g \quad \mathbf{c}^O]^T \quad (3)$$

where  $\mathbf{p}_{B_i}^W \in \mathbb{R}^3$ ,  $\mathbf{v}_{B_i}^W \in \mathbb{R}^3$ , and  $\mathbf{q}_{B_i}^W \in SO(3)$  are the position, linear velocity, and orientation vector, respectively.  $\mathbf{b}_a$  and  $\mathbf{b}_g$  are the IMU gyroscope and accelerometer biases. Finally,  $\mathbf{c}^O$  is the scale factor of the odometer. According to the preintegration principle [1] and our previous work [6], the IMU and odometer increment between frame  $k$  and  $k+1$  as:

$$\begin{aligned} \boldsymbol{\alpha}_{B_{k+1}}^{B_k} &= \iint_{t=k}^{k+1} \mathbf{R}_{B_t}^{B_k} (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \boldsymbol{\eta}_a) dt^2, \\ \boldsymbol{\beta}_{B_{k+1}}^{B_k} &= \int_{t=k}^{k+1} \mathbf{R}_{B_t}^{B_k} (\hat{\mathbf{a}}_t - \mathbf{b}_{a_t} - \boldsymbol{\eta}_a) dt, \\ \boldsymbol{\gamma}_{B_{k+1}}^{B_k} &= \int_{t=k}^{k+1} \frac{1}{2} \Omega(\hat{\boldsymbol{\omega}}_t - \mathbf{b}_{\omega_t} - \boldsymbol{\eta}_\omega) \boldsymbol{\gamma}_{B_t}^{B_k} dt, \\ \boldsymbol{\alpha}_{O_{k+1}}^{O_k} &= \int_{t=k}^{k+1} \mathbf{R}_{O_t}^{O_k} (\boldsymbol{c}^{O_k} \hat{\mathbf{v}}^O - \boldsymbol{\eta}_{s^O}) dt, \end{aligned} \quad (4)$$

where,

$$\begin{aligned} \Omega(\boldsymbol{\omega}) &= \begin{bmatrix} -[\boldsymbol{\omega}]_X & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^T & 0 \end{bmatrix}, \\ [\boldsymbol{\omega}]_X &= \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \end{aligned} \quad (5)$$

Using the calibration parameter, we can also transform  $\boldsymbol{\alpha}_{O_{k+1}}^{O_k}$  into IMU frame  $\boldsymbol{\phi}_{B_{k+1}}^{B_k}$  with:

$$\boldsymbol{\phi}_{B_{k+1}}^{B_k} = \int_{t=k}^{k+1} \mathbf{R}_{B_t}^{B_k} \mathbf{R}_{O_t}^{B_k} (\boldsymbol{c}^{O_k} \hat{\mathbf{v}}^O - \boldsymbol{\eta}_{s^O}) dt \quad (6)$$

Thus, the discrete form of preintegrated IMU/odometer measurements  $[\hat{\boldsymbol{\alpha}}_{B_{i+1}}^{B_k}, \hat{\boldsymbol{\beta}}_{B_{i+1}}^{B_k}, \hat{\boldsymbol{\gamma}}_{B_{i+1}}^{B_k}, \hat{\boldsymbol{\phi}}_{B_{i+1}}^{B_k}]$  can be given by:

$$\begin{aligned} \hat{\boldsymbol{\alpha}}_{B_{i+1}}^{B_k} &= \hat{\boldsymbol{\alpha}}_{B_i}^{B_k} + \hat{\boldsymbol{\beta}}_{B_i}^{B_k} \delta t + \frac{1}{2} \mathbf{R}(\hat{\boldsymbol{\gamma}}_{B_i}^{B_k}) (\hat{\mathbf{a}}_i - \hat{\mathbf{b}}_{a_i}) \delta t^2, \\ \hat{\boldsymbol{\beta}}_{B_{i+1}}^{B_k} &= \hat{\boldsymbol{\beta}}_{B_i}^{B_k} + \mathbf{R}(\hat{\boldsymbol{\gamma}}_{B_i}^{B_k}) (\hat{\mathbf{a}}_i - \hat{\mathbf{b}}_{a_i}) \delta t, \\ \hat{\boldsymbol{\gamma}}_{B_{i+1}}^{B_k} &= \hat{\boldsymbol{\gamma}}_{B_i}^{B_k} \otimes \left[ \frac{1}{2} (\hat{\boldsymbol{\omega}}_i - \hat{\mathbf{b}}_{\omega_i}) \delta t \right], \end{aligned}$$

$$\hat{\boldsymbol{\phi}}_{B_{i+1}}^{B_k} = \hat{\boldsymbol{\phi}}_{B_i}^{B_k} + \mathbf{R}(\hat{\boldsymbol{\gamma}}_{B_i}^{B_k}) \hat{\mathbf{R}}_{O_i}^{B_i} \hat{\boldsymbol{c}}^{O_i} \hat{\mathbf{v}}^{O_i} \delta t, \quad (7)$$

Finally, the residual of preintegrated IMU/odometer measurements can be expressed as:

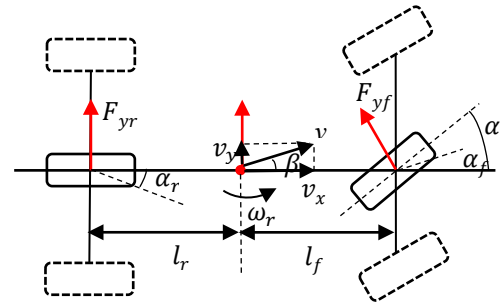
$$\begin{aligned} \mathbf{r}_j(\hat{\mathbf{z}}_{B_{k+1}}^{B_k}, \mathbf{x}) &= [\delta \boldsymbol{\alpha}_{B_{k+1}}^{B_k} \quad \delta \boldsymbol{\beta}_{B_{k+1}}^{B_k} \quad \delta \boldsymbol{\theta}_{B_{k+1}}^{B_k} \quad \delta \mathbf{b}_a \quad \delta \mathbf{b}_g \quad \delta \boldsymbol{\phi}_{B_{k+1}}^{B_k} \quad \delta \mathbf{c}^O]^T \\ &= \begin{bmatrix} \mathbf{R}_W^{B_k} (\mathbf{p}_{B_{k+1}}^W - \mathbf{p}_{B_k}^W + \frac{1}{2} \boldsymbol{g}^W \Delta t_k^2 - \mathbf{v}_{B_k}^W \Delta t_k) - \hat{\boldsymbol{\alpha}}_{B_{k+1}}^{B_k} \\ \mathbf{R}_W^{B_k} (\mathbf{v}_{B_{k+1}}^W + \boldsymbol{g}^W \Delta t_k - \mathbf{v}_{B_k}^W) - \hat{\boldsymbol{\beta}}_{B_{k+1}}^{B_k} \\ 2 \left[ (\mathbf{q}_{B_k}^W)^{-1} \otimes (\mathbf{q}_{B_{k+1}}^W) \otimes (\hat{\boldsymbol{\gamma}}_{B_{k+1}}^{B_k})^{-1} \right]_{2:4} \\ \mathbf{b}_{a_{k+1}} - \mathbf{b}_{a_k} \\ \mathbf{b}_{g_{k+1}} - \mathbf{b}_{g_k} \\ \mathbf{R}_W^{B_k} (\mathbf{p}_{B_{k+1}}^W - \mathbf{p}_{B_k}^W + \mathbf{R}_{B_{k+1}}^{B_k} \mathbf{p}_{O_{k+1}}^{B_{k+1}}) - \hat{\boldsymbol{\phi}}_{B_{k+1}}^{B_k} \\ \mathbf{c}^{O_{k+1}} - \mathbf{c}^{O_k} \end{bmatrix} \end{aligned} \quad (8)$$

We use  $\delta \boldsymbol{\theta}_{B_{k+1}}^{B_k}$  to represent the error state of a quaternion, and  $[\cdot]_{2:4}$  to take out the last three elements from a quaternion.

We use the factor graph [32] to solve this problem, where the optimal vehicle state estimation is a least square minimization problem using vehicle kinematics and the GNSS positioning from Section IV-E. Besides, we establish sliding windows to ensure the real-time performance of the optimization scheme. For a sliding window of  $N_K$  keyframes, the optimal states are obtained through minimizing:

$$\min_{\mathbf{x}} \{ \|\mathbf{r}_p\|^2 + \sum_{i=1}^{N_{JK}} \|\mathbf{r}_{j_i}\|^2 + \sum_{i=1}^{N_{GK}} \mathbf{r}_{g_i} \} \quad (9)$$

where  $\mathbf{r}_p$  is the prior factor marginalized by Schur-complement [1],  $\mathbf{r}_{j_i}$  is the residual of vehicle kinematics preintegration result, and the residual of global positioning system is  $\mathbf{r}_{g_i}$ .  $N_{JK}$  and  $N_{GK}$  denotes the number of preintegration and GNSS factors, respectively.



**Fig. 3.** The 2-DoF vehicle dynamics model.  $l_f$  and  $l_r$  are the distances from the center of mass to the vehicle front and rear axles.  $\alpha_f$  and  $\alpha_r$  indicate the front and rear wheel tire slip angles.  $\alpha$  denote the front wheel angle and  $\beta$  is the slip angle.  $v$  is the vehicle velocity at the center of mass, with  $v_x$  and  $v_y$  representing the lateral and longitudinal part.  $\omega_r$  denotes the vehicle yaw velocity. Finally,  $F_{yf}$  and  $F_{yr}$  describes the lateral force on the front and rear axles.

## 4.2 Vehicle Dynamics

The kinematics model assumes that no tire slip exists between wheels and ground which is only reasonable for low-speed and

small steering motion. In comparison, this assumption breaks down for higher speeds or with sharp turnings. Therefore, if no IMU is included, the dynamic vehicle model should be taken into consideration in this situation.

The vehicle dynamics model can be simplified as a two degree-of-freedom (2-DoF) bicycle model [33], describing the lateral and yaw motions as visualized in Fig. 3. Summing the forces and moments about the vehicle center of mass, the lateral dynamics of the bicycle model can be expressed as [34]:

$$\begin{aligned}\sum F_y &= F_{yf} + F_{yr} = m(\dot{v}_y + v_x \omega_r), \\ \sum M_z &= l_f F_{yf} - l_r F_{yr} = I_z \dot{\omega}_r,\end{aligned}\quad (10)$$

where  $m$  describes the vehicle mass and  $I_z$  is the yaw moment of inertia. Assume that the lateral force acting on tire is proportional to the tire slip angle, we can get the basic equation of motion for the bicycle model as:

$$\begin{aligned}(C_f + C_r)\beta + \frac{1}{v_x}(l_f C_f - l_r C_r)\omega_r - C_f \alpha &= m(\dot{v}_y + v_x \omega_r), \\ (l_f C_f - l_r C_r)\beta + \frac{1}{v_x}(l_f^2 C_f + l_r^2 C_r)\omega_r - l_f C_f \alpha &= I_z \dot{\omega}_r,\end{aligned}\quad (11)$$

where  $C_f$  and  $C_r$  are the lateral stiffness of front and rear wheels. When the vehicle is mainly moving straight,  $\dot{v}_y$  and  $\dot{\omega}_r$  can be assumed to be zero, then for a given timestamp  $k$ ,  $\beta$  and  $\omega_r$  can be computed through:

$$\begin{aligned}\beta_k &= \frac{1 + \frac{m}{2l} \frac{l_f}{l_r C_r} \hat{v}_k^2}{1 - C \hat{v}_k^2} \frac{l_r}{l} \alpha_k, \\ \hat{\omega}_k &= \frac{1}{1 - C \hat{v}_k^2} \frac{\hat{v}_k}{l} \alpha_k,\end{aligned}\quad (12)$$

with:

$$C = \frac{m(l_f C_f - l_r C_r)}{l^2 C_f C_r}, \quad l = l_f + l_r \quad (13)$$

Here  $\hat{v}_k$  and  $\alpha_k$  are the linear velocity measurements from vehicle chassis and steering angle transmitted to front wheel. Then the measurements of vehicle dynamics model  $\hat{\mathbf{v}}_k$  and  $\hat{\boldsymbol{\omega}}_k$  are expressed as:

$$\begin{aligned}\hat{\mathbf{v}}_k &= [\hat{v}_k \cos \beta_k \quad \hat{v}_k \sin \beta_k \quad 0]^T, \\ \hat{\boldsymbol{\omega}}_k &= [0 \quad 0 \quad \hat{\omega}_k]^T.\end{aligned}\quad (14)$$

Then the pre-integrated kinematics within the vehicle frame can be expressed as:

$$\begin{aligned}\boldsymbol{\alpha}_{V_{k+1}}^V &= \int_{t=k}^{k+1} \mathbf{R}_{V_t}^V (\hat{\mathbf{v}}_t - \boldsymbol{\eta}_v) dt, \\ \mathbf{r}_{V_{k+1}}^V &= \int_{t=k}^{k+1} \frac{1}{2} \Omega(\hat{\boldsymbol{\omega}}_t - \boldsymbol{\eta}_\omega) \mathbf{r}_{V_t}^V dt,\end{aligned}\quad (15)$$

and the related residual is:

$$\mathbf{r}_D(\hat{\mathbf{z}}_{B_{k+1}}^B, \mathbf{x}) = [\delta \boldsymbol{\alpha}_{B_{k+1}}^B \quad \delta \boldsymbol{\theta}_{B_{k+1}}^B]^T$$

$$= \begin{bmatrix} \mathbf{R}_W^{B_k} (\mathbf{p}_{B_{k+1}}^W - \mathbf{p}_{B_k}^W) - \mathbf{p}_V^B + \mathbf{R}_W^{B_k} \mathbf{R}_{B_{k+1}}^W \mathbf{p}_V^B - \hat{\boldsymbol{\alpha}}_{B_{k+1}}^{B_k} \\ 2 \left[ (\mathbf{q}_{V_k}^W)^{-1} \otimes (\mathbf{q}_{V_{k+1}}^W) \otimes (\hat{\mathbf{r}}_{V_{k+1}}^V)^{-1} \right]_{2:4} \end{bmatrix}. \quad (16)$$

The vehicle dynamics and GNSS information are also jointly optimized atop a factor graph, with:

$$\min_{\mathbf{x}} \{ \|\mathbf{r}_p\|^2 + \sum_{i=1}^{N_{D_K}} \|\mathbf{r}_{D_i}\|^2 + \sum_{i=1}^{N_{G_K}} \mathbf{r}_{G_i} \}, \quad (17)$$

where  $\mathbf{r}_{D_i}$  is the residual of vehicle dynamics preintegration result. Note that for IVs without an IMU, only the vehicle dynamics is utilized, whereas both states are optimized when IMU is available.

### 4.3 LiDAR-inertial Odometry

Nowadays, many IVs are equipped with LiDARs, and some of them even have a multiple-LiDAR setup. Considering the diversity of both LiDAR brands and placements on IVs, we seek to design a general LiDAR-inertial odometry.

The preprocessing process include outlier and distortion removal, as well as multiple scan synchronization. The noisy and outlier points are first removed with statistical outlier removal filter. Then we apply the vehicle dynamics increment model to de-skew the point cloud with linear interpolation. For the multi-LiDAR setup as visualized in Fig. 4, we leverage the procedure in our previous work [6], [35] to synchronize them in the algorithm.

We choose an efficient filter-based method Fast-lio2 [21] as our backbone. When the IMU is included, we also add wheel odometer information into the iterated Kalman filter. Following the  $\boxplus$  definition in [21], the continuous kinematic model at the IMU sampling period  $\Delta t$  can be discretized as:

$$\mathbf{x}_{i+1} = \mathbf{x}_i \boxplus (\Delta t \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, \mathbf{w}_i)) \quad (18)$$

where the state  $\mathbf{x}_i$ , input  $\mathbf{u}_i$ , process noise  $\mathbf{w}_i$  and the function  $\mathbf{f}$  are defined as:

$$\begin{aligned}\mathbf{x}_i &= [\mathbf{q}_{B_i}^W \quad \mathbf{p}_{B_i}^W \quad \mathbf{v}_{B_i}^W \quad \mathbf{b}_g \quad \mathbf{b}_a \quad \boldsymbol{\mathcal{G}}^W \quad \mathbf{q}_{L_i}^B \quad \mathbf{p}_{L_i}^B \quad \mathbf{p}_{O_i}^B \quad \mathbf{c}^O]^T, \\ \mathbf{u}_i &= [\hat{\boldsymbol{\omega}}_i \quad \hat{\mathbf{a}}_i \quad \hat{\mathbf{v}}_i^O]^T, \\ \mathbf{w}_i &= [\boldsymbol{\eta}_{\omega_i} \quad \boldsymbol{\eta}_{a_i} \quad \boldsymbol{\eta}_{s^O_i} \quad \mathbf{b}_{\omega_i} \quad \mathbf{b}_{a_i}]^T, \\ \mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{w}) &= \begin{bmatrix} \hat{\boldsymbol{\omega}} - \mathbf{b}_\omega - \boldsymbol{\eta}_\omega \\ \mathbf{v}_B^W + \frac{1}{2} (\mathbf{R}_B^W (\hat{\mathbf{a}} - \mathbf{b}_a - \boldsymbol{\eta}_a) + \boldsymbol{\mathcal{G}}^W) \Delta t \\ \mathbf{R}_B^W (\hat{\mathbf{a}} - \mathbf{b}_a - \boldsymbol{\eta}_a) + \boldsymbol{\mathcal{G}}^W \\ \mathbf{b}_\omega \\ \mathbf{b}_a \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{R}_O^B (\mathbf{c}^O \hat{\mathbf{v}}^O - \boldsymbol{\eta}_{s^O}) \\ \mathbf{c}^O \end{bmatrix}. \quad (19)\end{aligned}$$

We also implement the measurement model defined in [21], and the forward propagation is performed upon received IMU

input. Based on the  $\boxminus$  operator in [22] The covariance is propagated using the error state dynamic model as:

$$\begin{aligned}\tilde{\mathbf{x}}_{i+1} &= \mathbf{x}_{i+1} \boxminus \hat{\mathbf{x}}_{i+1} \\ &= \mathbf{F}_{\tilde{\mathbf{x}}} \tilde{\mathbf{x}}_i + \mathbf{F}_{\mathbf{w}} \mathbf{w}_i.\end{aligned}\quad (20)$$

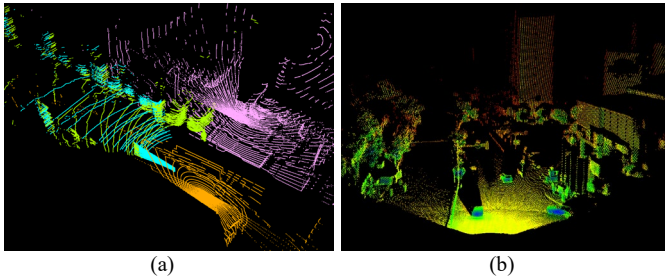
Here the term  $\tilde{\mathbf{x}}_{i+1}$  expresses the error in the estimation of a quantity, such that,  $\tilde{\mathbf{x}}_{i+1} = \mathbf{x}_{i+1} - \hat{\mathbf{x}}_{i+1}$ . The matrix  $\mathbf{F}_{\tilde{\mathbf{x}}}$  and  $\mathbf{F}_{\mathbf{w}}$  is computed following (21). The  $\text{Exp}(\mathbf{r})$  denotes the exponential map in [36] and  $\mathbf{A}(\mathbf{u})^{-1}$  follows the definition in [37]:

$$\begin{aligned}\text{Exp}(\mathbf{r}) &= \mathbf{I} + \frac{\mathbf{r}}{\|\mathbf{r}\|} \sin(\|\mathbf{r}\|) + \frac{\mathbf{r}^2}{\|\mathbf{r}\|^2}, \\ \mathbf{A}(\mathbf{u})^{-1} &= \mathbf{I} - \frac{1}{2} [\mathbf{u}]_{\wedge} + (1 - \alpha(\|\mathbf{u}\|)) \frac{[\mathbf{u}]_{\wedge}^2}{\|\mathbf{u}\|^2}, \\ \alpha(u) &= \frac{u \cos(u/2)}{2 \sin(u/2)}.\end{aligned}\quad (22)$$

The notation  $[\mathbf{u}]_{\wedge}$  indicates the skew-symmetric matrix of a vector  $\mathbf{u} \in \mathbb{R}^3$  that maps the cross product operation. Then the propagated covariance  $\hat{\mathbf{P}}_i$  can be calculated following:

$$\hat{\mathbf{P}}_{i+1} = \mathbf{F}_{\tilde{\mathbf{x}}} \hat{\mathbf{P}}_i \mathbf{F}_{\tilde{\mathbf{x}}}^T + \mathbf{F}_{\mathbf{w}} \mathbf{Q}_i \mathbf{F}_{\mathbf{w}}^T; \hat{\mathbf{P}}_0 = \bar{\mathbf{P}}_{k-1}, \quad (23)$$

where  $\mathbf{Q}_i$  is the covariance of the noise  $\mathbf{w}_i$ . Note that (24) is an iterative process, and the residual  $\mathbf{z}_i^k$  at the  $k$ -th iterate update is defined following [21] as:



**Fig. 4.** Examples of some LiDAR setups on IV. (a) is a single scan from our four-LiDAR setup on HQ-1 IV, the green and cyan indicate two front view LiDARs, the orange and purple denote two side blind spot LiDAR. (b) is a scan from our 300-channel LiDAR setup on HQ-2 IV.

$$\mathbf{z}_i^k = \mathbf{u}_i^T (\hat{\mathbf{T}}_{B_i}^{W_k} \hat{\mathbf{T}}_{L_i}^{W_k} \mathbf{p}_{L_i} - \mathbf{q}_i^W), \quad (24)$$

where  $\mathbf{p}_{L_i}$  is the measured LiDAR point in its own coordinate, and  $\mathbf{q}_i^W$  is the centroid of the fitted local small plane patch defined in [21]. Then (20) can be computed as:

$$\mathbf{x}_k \boxminus \hat{\mathbf{x}}_k = (\hat{\mathbf{x}}_k^k \boxminus \tilde{\mathbf{x}}_k^k) \boxminus \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^k \boxminus \hat{\mathbf{x}}_k + \mathbf{J}^k \tilde{\mathbf{x}}_k^k \quad (25)$$

where  $\mathbf{J}^k$  is the partial differentiation of  $(\hat{\mathbf{x}}_k^k \boxminus \tilde{\mathbf{x}}_k^k) \boxminus \hat{\mathbf{x}}_k$  w.r.t.  $\tilde{\mathbf{x}}_k^k$  evaluated at zero:

$$\mathbf{F}_{\tilde{\mathbf{x}}} = \begin{bmatrix} \text{Exp}(-\hat{\mathbf{w}}_i \Delta t) & \mathbf{0} & \mathbf{0} & -\mathbf{A}(\hat{\mathbf{w}}_i \Delta t)^T \Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{I} \Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{R}_{B_i}^W [\hat{\mathbf{a}}_i]_{\wedge} \Delta t & \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{R}_{B_i}^W \Delta t & \mathbf{I} \Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{F}_{\mathbf{w}} = \begin{bmatrix} -\mathbf{A}(\hat{\mathbf{w}}_i \Delta t)^T \Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\mathbf{R}_{B_i}^W \Delta t & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \Delta t & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \Delta t & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I} \Delta t \end{bmatrix} \quad (21)$$

$$\mathbf{J}^k = \begin{bmatrix} \mathbf{A}(\hat{\mathbf{R}}_{B_k}^{W_k} \boxminus \hat{\mathbf{R}}_{B_k}^{W_k})^{-T} & \mathbf{0}_{3 \times 15} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 9} \\ \mathbf{0}_{15 \times 3} & \mathbf{I}_{15 \times 15} & \mathbf{0}_{15 \times 3} & \mathbf{0}_{15 \times 9} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 15} & \mathbf{A}(\hat{\mathbf{R}}_{L_k}^{B_k} \boxminus \hat{\mathbf{R}}_{L_k}^{B_k})^{-T} & \mathbf{0}_{3 \times 9} \\ \mathbf{0}_{9 \times 3} & \mathbf{0}_{9 \times 15} & \mathbf{0}_{9 \times 3} & \mathbf{I}_{9 \times 9} \end{bmatrix} \quad (26)$$

here  $\hat{\mathbf{R}}_{B_k}^{W_k} \boxminus \hat{\mathbf{R}}_{B_k}^{W_k}$  and  $\hat{\mathbf{R}}_{L_k}^{B_k} \boxminus \hat{\mathbf{R}}_{L_k}^{B_k}$  is the error states of IMU's attitude and rotational extrinsic.

Then the combination of (24) and (25) yields the posteriori distribution of the state  $\mathbf{x}_k$ , which is a maximum-a-posteriori (MAP) problem, and can be solved by the iterated Kalman filter:

$$\min_{\tilde{\mathbf{x}}_k^k} \left( \|\mathbf{x}_k \boxminus \hat{\mathbf{x}}_k\|_{\hat{\mathbf{P}}_k}^2 + \sum_{i=1}^m \|\mathbf{z}_i^k + \mathcal{H}_i^k \tilde{\mathbf{x}}_k^k\|_{\mathcal{R}_i}^2 \right) \quad (27)$$

where  $\mathcal{H}_i^k$  is the Jacobin matrix and  $\mathcal{R}_i$  is the measurement noise defined in [22]. The iteration process is solved following [18], where for the first iteration,  $\hat{\mathbf{x}}_k^k = \hat{\mathbf{x}}_k$  and  $\mathbf{J}^k = \mathbf{I}$ . We also use the ikd-tree [38] for efficient point cloud organization.

Similarly, when the IMU is not available for some IVs, the system is changed to:

$$\mathbf{x}_i = [\mathbf{q}_{B_i}^W \quad \mathbf{p}_{B_i}^W \quad \mathbf{v}_{B_i}^W \quad \mathbf{q}_{L_i}^B \quad \mathbf{p}_{L_i}^B \quad \mathbf{q}_{V_i}^B \quad \mathbf{p}_{V_i}^B]^T. \quad (28)$$

Besides, the  $\mathbf{J}^k$  also changes accordingly,

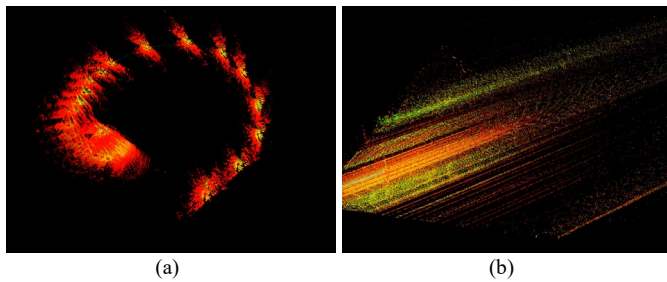
$$\mathbf{J}^k = \begin{bmatrix} \mathbf{A}(\delta \theta_{B_k}^W)^{-T} & \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{6 \times 3} & \mathbf{I}_{6 \times 6} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}(\delta \theta_{L_k}^B)^{-T} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{A}(\delta \theta_{V_k}^B)^{-T} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (29)$$

where  $\delta \theta_{B_k}^W$ ,  $\delta \theta_{L_k}^B$  and  $\delta \theta_{V_k}^B$  is the expression of  $\hat{\mathbf{R}}_{B_k}^{W_k} \boxminus \hat{\mathbf{R}}_{B_k}^{W_k}$ ,  $\hat{\mathbf{R}}_{L_k}^{B_k} \boxminus \hat{\mathbf{R}}_{L_k}^{B_k}$  and  $\hat{\mathbf{R}}_{V_k}^{B_k} \boxminus \hat{\mathbf{R}}_{V_k}^{B_k}$ .

When GNSS information is available, we also use the factor graph at the back end to obtain a GNSS constrained pose estimation result,

$$\min_{\mathbf{x}} \{ \|\mathbf{r}_p\|^2 + \sum_{i=1}^{N_{\mathcal{F}_K}} \|\mathbf{r}_{\mathcal{F}_i}\|^2 + \sum_{i=1}^{N_{\mathcal{G}_K}} \mathbf{r}_{\mathcal{G}_i} \}, \quad (30)$$

where  $\mathbf{r}_{\mathcal{F}_i}$  is the residual of the iterated Kalman filter and  $N_{\mathcal{F}_K}$  indicates the number of this item.



**Fig. 5.** Visualization of the failures caused by degeneracy. (a) is the irregular trajectory. (b) shows the ‘stopped’ trajectory.

Since our LiDAR-inertial system is designed with no degeneracy-aware module, it may fail at feature-poor areas, such as the planar tunnels, structure repeated noise barriers, or the featureless bridge. Once failure is encountered, many state-of-the-art (SOTA) LiDAR SLAM will die completely and are always non-recoverable. Therefore, we seek to develop a self-recoverable LiDAR SLAM with redundant odometries.

As visualized in Fig. 5, the pose estimation in degenerated areas have two common failures: one is the spinning trajectory, the other is the ‘stopped’ or slow-moving trajectory. Both of them will generate large errors compared with the absolute GNSS information. We hereby set a failure detection thread to compare the pose estimation result of LiDAR-inertial system and GNSS every three minutes. Once the difference is above a given threshold, the LiDAR-inertial system will reinitialize using the current GNSS information, and the Kalman filter will also restart. Note that if the GNSS measurement is currently unavailable or has large outliers, the system will turn to next period for comparison automatically.

#### 4.4 Visual-inertial Odometry

SLAM algorithms have a high CPU dependence, but most of the IVs are merely equipped with powerful GPU. Therefore, we seek to leverage the GPU parallelization to improve the Visual-inertial efficiency.

Our Visual-inertial odometry follows the lightweight design of Vins-mono [1] and Vins-Fusion<sup>1</sup>. In comparison, we adopt a GPU-accelerated feature detector [39] at the visual front-end. Besides, we also add the self-recovery characteristic as shown in Section IV-C, when the long-during failure is detected, the Visual-inertial system will re-initialize and restart.

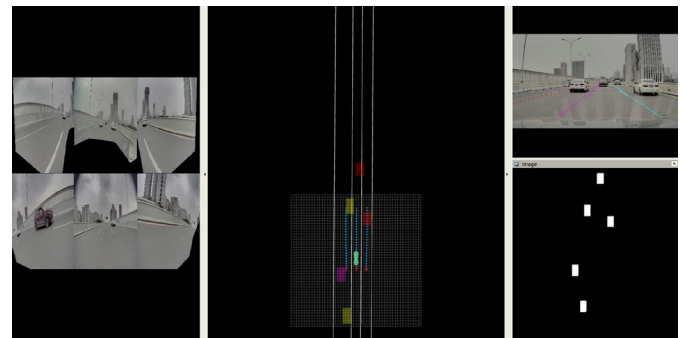
#### 4.5 GNSS and Map-matching Odometry

Many research IVs integrate the real-time kinematic (RTK) measurement into pose estimation to ensure high precision localization. However, the RTK information is not available for the massive produced IVs, and we merely use single point positioning (SPP) for all the IVs instead. Since the factor graph based GNSS optimization can simultaneously explore the time-correlation among historical measurements and effectively explore the time-correlation of pseudorange, carrier-phase, as well as doppler measurements. We leverage an open source GraphGNSSLib proposed in [40].

When the IVs enter districts with prior HD maps, the map-matching subsystem is awakened to match the real-time scan to the previous constructed map. To enable the real-time

performance, we employ a multi-threaded normal distribution transform (NDT) method [41] for map-based localization. NDT divides the 3D space into small cells, and calculate the local probability density function (PDF) in each cell. Then the point-to-distribution correspondences are computed within a scan pair to find the optimal transformation.

In addition, our system is also capable of matching Bird-eye-view (BEV) image’s semantic segmentation with the vector map constructed in Section V-F when the LiDAR sensor is not installed on the IV. As visualized in Fig. 6, we utilize the surround view and front view fisheye cameras to generate a BEV image through inverse perspective mapping (IPM). We train a convolutional network [42] that segments the BEV images into various road markings, e.g., road lines, text, zebra



**Fig. 6.** The registration of BEV semantic segmentation with vector map. The left six insets are the recertified surround view images. The middle inset is the real-time BEV semantic road lines (blue dotted lines) matched with the vector map (white lines), the small red, yellow, and purple rectangles are the detected vehicles projected onto BEV image. The right above inset shows the real-time road lines and markings segmentation result. The right bottom inset visualizes the surround vehicles (small white rectangles) in BEV.

stripes, and dynamic objects, e.g., vehicles, pedestrians. By doing so, the semantic features detected on the BEV image are lifted into the vehicle coordinate. Then the current pose of the IV is estimated by matching current feature points with the vector map using the iterative closest point (ICP) [43].

#### 4.6 Hardware and Software Level Verification

The hardware-level verification is conducted at the data preprocessing stage, including data stream existence, frequency, and individual verification. The data stream existence test aims to find out whether the required data input exist or not. Since the vehicle dynamics, kinematics, and Visual-inertial odometry have a factor graph design, they can still work for a short period when one of the input stream is lost temporarily. However, the LiDAR-inertial odometry has a filter-based structure, and it will fail immediately when no input is from either IMU or LiDAR [44]. Therefore, our LiDAR-inertial odometry will reinitialize and restart following Section IV-C when either IMU or LiDAR stream is lost for one second. For the other three odometries, this period is extended to thirty seconds. The data frequency test also follows this idea. The system set the stream with the lowest frequency as the primary input, and monitor the counts of other data within two consecutive frames continuously, e.g., the LiDAR is set as the primary input (10 Hz), and approximately ten frames of IMU input (100 Hz) should be found within two successive LiDAR

<sup>1</sup> <https://github.com/HKUST-Aerial-Robotics/VINS-Fusion>

scans [45]. Once this criterion is not hold for ten minutes, the system will send a warning to the user interface for a manual check, e.g., a yellow warning sign on the central control screen.

The individual verification is mainly for the perception sensors. Since the LiDAR sensors are often equipped at a low height of the vehicle, once the IV is in a busy district with vehicles all around, the LiDAR-inertial odometry may fail to the dynamic objects. Therefore, we monitor the Euclidian distance of the point clouds within each scan, if 70 % of the points are below ten meters to the LiDAR, the current frame is discarded for pose estimation. Since the Visual-inertial odometry may fail against the sudden illuminance variations, we first transform the gamma-compressed RGB values to linear RGB, and compute the relative luminance of each image. The image with either too high or too low value is discarded for pose estimation due to insufficient contrast.

The software-level test is performed for parallel pose estimation modules to remove clearly wrong results. We set the maximum speed of the vehicle as 250 km/h, and verify whether the displacement of each odometry is beyond this limit or not, e.g., once the displacements of two successive vehicle dynamics odometry (100 Hz) is beyond 0.7 m, it will be discarded in the next section, since it is clearly wrong pose estimation results. Similarly, we use the steering angle information to monitor the individual yaw estimation results.

#### 4.7 Switching Logic and Candidate Selection

After the software-level verification, the remaining odometry candidates will pass through a proposal evaluation process to select the best odometry candidate.



Fig. 7. The two employed IVs. (a) is the Voyah Free without any adaptations. (b) is the Hongqi H9 with deep modifications for autonomous driving purpose.

We denote the  $\mathbf{p}_{\mathcal{J}_k}^{\mathcal{J}_{k+1}}$ ,  $\mathbf{p}_{\mathcal{D}_k}^{\mathcal{D}_{k+1}}$ ,  $\mathbf{p}_{\mathcal{L}_k}^{\mathcal{L}_{k+1}}$ ,  $\mathbf{p}_{\mathcal{C}_k}^{\mathcal{C}_{k+1}}$ ,  $\mathbf{p}_{\mathcal{G}_k}^{\mathcal{G}_{k+1}}$ , and  $\mathbf{p}_{\mathcal{M}_k}^{\mathcal{M}_{k+1}}$  as the respective position variations of different odometries (kinematics, dynamics, LiDAR-inertial, Visual-inertial, GNSS and map matching) within two frames  $k$  and  $k+1$ . The best odometry is selected through weight comparison as described in (32). Since the vehicle dynamics odometry is the most robust (robust to wheel slip, degeneracy, bad illumination condition and satellite signal quality), we set the  $\mathbf{p}_{\mathcal{D}_k}^{\mathcal{D}_{k+1}}$  as the short-term reference to compute the weighting metric of others.

Given that the vehicle kinematics odometry accuracy may be higher when moving in straight line with low acceleration, we define  $\mathcal{M}_k^{\mathcal{J}}$  and  $\mathcal{M}_k^{\mathcal{Y}}$  in (32) to describe the acceleration and angular variations. Specifically, when the wheel slip happens at turnings ( $\|\mathbf{p}_{\mathcal{J}_k}^{\mathcal{J}_{k+1}}\| < \|\mathbf{p}_{\mathcal{D}_k}^{\mathcal{D}_{k+1}}\|$ ), the squaring position factor

$\mathcal{M}_k^{\mathcal{V}}$ , acceleration factor  $\mathcal{M}_k^{\mathcal{J}}$  and angular factor  $\mathcal{M}_k^{\mathcal{Y}}$  will amplify this influence. Then the kinematics weight  $\mathcal{W}_k^{\mathcal{J}}$  is small and vice versa.

As mentioned in Section IV-C, the pose estimation results of LiDAR-inertial will be either too small or too big at the degraded districts, and the weighting metric  $\mathcal{W}_k^{\mathcal{L}}$  is also small.

This also works for Visual-inertial system with bad lighting conditions or high dynamics (many vehicles around), GNSS and map matching odometry with large outliers.

$$\left\{ \begin{array}{l} \mathcal{M}_k^{\mathcal{V}} = \left( \frac{\|\mathbf{p}_{\mathcal{J}_k}^{\mathcal{J}_{k+1}}\|}{\|\mathbf{p}_{\mathcal{D}_k}^{\mathcal{D}_{k+1}}\|} \right)^2, 0 \leq \mathcal{M}_k^{\mathcal{V}} \leq 1 \\ \mathcal{M}_k^{\mathcal{J}} = \frac{\|\mathbf{v}_{\mathcal{J}_k}^{\mathcal{J}_{k+1}}\|}{\mathbf{v}_{emp}}, 0 \leq \mathcal{M}_k^{\mathcal{J}} \leq 1 \\ \mathcal{M}_k^{\mathcal{Y}} = \frac{\|\boldsymbol{\theta}_{\mathcal{Y}_k}^{\mathcal{Y}_{k+1}}\|}{\boldsymbol{\theta}_{emp}}, 0 \leq \mathcal{M}_k^{\mathcal{Y}} \leq 1 \\ \mathcal{M}_k^{\mathcal{R}} = \left( \frac{\|\mathbf{p}_{\mathcal{R}_k}^{\mathcal{R}_{k+1}}\| - \|\mathbf{p}_{\mathcal{D}_k}^{\mathcal{D}_{k+1}}\|}{\|\mathbf{p}_{\mathcal{D}_k}^{\mathcal{D}_{k+1}}\|} \right)^2, 0 \leq \mathcal{M}_k^{\mathcal{R}} \leq 1 \\ \mathcal{W}_k^{\mathcal{J}} = \mathcal{M}_k^{\mathcal{V}}(1 - \mathcal{M}_k^{\mathcal{J}} - \mathcal{M}_k^{\mathcal{Y}}), 0 \leq \mathcal{W}_k^{\mathcal{J}} \leq 1 \\ \mathcal{W}_k^{\mathcal{D}} = (\mathcal{M}_k^{\mathcal{J}} + \mathcal{M}_k^{\mathcal{Y}})(1 - \mathcal{M}_k^{\mathcal{V}}), 0 \leq \mathcal{W}_k^{\mathcal{D}} \leq 1 \\ \mathcal{W}_k^{\mathcal{R}} = 1 - \mathcal{M}_k^{\mathcal{R}}, 0 \leq \mathcal{W}_k^{\mathcal{R}} \leq 1 \end{array} \right. \quad (31)$$

where  $\mathbf{v}_{emp}$  and  $\boldsymbol{\theta}_{emp}$  denote the empirical threshold of the velocity and angle variation. The  $\mathbf{v}_{emp}$  is set as the product of the maximum acceleration and time duration within the  $k$  and  $k+1$ , e.g., the maximum acceleration of an IV is 0.55 g, and the time duration is 0.1 s, then  $\mathbf{v}_{emp}$  should be  $0.055 \text{ m/s}^2$ . Besides,  $\boldsymbol{\theta}_{emp}$  is set as the product of 10 degree and the given period. The symbol  $\mathcal{R}$  in  $\mathcal{M}_k^{\mathcal{R}}$  and  $\mathcal{W}_k^{\mathcal{R}}$  can be  $\mathcal{L}$ ,  $\mathcal{C}$ ,  $\mathcal{G}$ , and  $\mathcal{M}$ , representing the odometry factor and weighting metrics of diverse odometries. The odometry with the highest weighting metric will be selected as the final pose output.

#### 5. EXPERIMENTS

This section first presents odometry performance on various IVs, then shows the great adaptivity of our system to other functionalities, such as controlling and mapping.

##### 5.1 IV Setups

We employ two massive produced IVs, Voyah Free, a high configuration one with Snapdragon SA8155P processor and a low configuration one with NXP-IMA8 processor. Besides, two research IVs are also utilized, the detailed sensor setups are visualized in Fig. 7 and listed in TABLE II. Note that unlike many other research purpose IVs having a backpack-like sensor platform on the roof, the *HQ-1* and *HQ-2* are all designed with mass production purpose. Therefore, the LiDARs and cameras are all pre-installed in the vehicle (e.g. two Leishen CH32 are installed below the head light, two RS Bpearl are installed on each side of the front door).

The algorithms are implemented in C++ and perform under Linux. Since the CPU and GPU usage is strictly limited for each

vehicle (e.g. 20% computation resource of a single core is left for *Free-2* to perform our algorithm), our algorithm is modified for each CPU and GPU setups. Our algorithm benefit from the GPU acceleration, where many CPU heavy computation is accelerated by GPU parallelization. Such modification includes many code refactoring, cross compiling, and logic simplifying. For instance, we use libtbb for parallel computation instead of the omp in Fast-lio2 and NDT-omp. Besides, the voxelization is also accelerated by GPU.

## 5.2 Odometry Performance

This section aims to verify the accuracy of the proposed system. The ground truth is kept by the post processing result of a near navigation grade IMU and a Trimble GNSS receiver, with the RTK corrections sent from Qianxun SI. The post processing software jointly optimizes the IMU and GNSS data in a tightly coupled manner. The output positioning result can

TABLE II  
 DETAILED SENSOR SETUPS OF DIFFERENT IVs

|           | Mass production IVs |               | Research purpose IVs               |                   |
|-----------|---------------------|---------------|------------------------------------|-------------------|
| Name      | <i>Free-1</i>       | <i>Free-2</i> | <i>HQ-1</i>                        | <i>HQ-2</i>       |
| Model     | Voyah Free          |               | Hongqi H9                          |                   |
| Power     | Range extended      | Electrical    | Gas                                |                   |
| Processor | I.MA8               | SA8155P       | TITAN4C                            |                   |
| CPU       | Cortex-A72          | Kryo 485      | NXP MPC77XX                        |                   |
| GPU       | Included            | Adreno 640    | 2 x NVIDIA AGX Xavier              |                   |
| Cameras   | 4 surround, 3 front |               | 6 surround, 2 front (stereo)       |                   |
| LiDARs    | 0                   |               | 4<br>2 RS Bpearl<br>2 Leishen CH32 | 1<br>Innovusion   |
| IMU       | No                  |               | CHCNAV<br>CGI-220                  | ASENSING<br>P-Box |
| GNSS      | Included            |               |                                    |                   |
| Encoder   | 2                   | 4             | 2                                  |                   |

reach centimeter-level accuracy. For each experiment, we set the start point sharing the same coordinates of the post processed results. Besides, we assign each LiDAR/camera frame with GPS timestamp, allowing frame-to-frame pose evaluation with post processed results.

We employ the pure odometry from the IMU/yaw rate and wheel encoder (odom-pure), as well as our self-modified ORB-SLAM2 [46], Vins-mono [1], and Fast-lio2 [15] with vehicle dynamics information for comparison. Note that for Vins-mono, our modification also supports stereo camera. In addition, the accuracy of in-vehicle GNSS is also compared. This GNSS information is also added to the four odometry when available using the GTSAM [32]. Besides, the scan context [47] based loop detection module is also added to the back end of Fast-lio2. For the multi-LiDAR setup on *HQ-1*, the input is the merged scan of four LiDARs.

We choose 15 sequences for evaluation, covering highly dynamic, feature-poor, degenerated, fast changing illumination, and bad weathers. Two criteria, the maximum error (MAX) and root mean square error (RMSE), are computed and reported in TABLE III. It is clear that our method is superior to all other approaches in 10 out of 15 sequences and a comparable result with the best odometry for the remaining sequences. The most notable characteristics of the redundant odometry is the effect suppression of maximum errors. It is seen that our method always maintains a high precision accuracy no matter how bad

the worst individual odometry candidate is. Besides, the maximum errors of our system are always among the lowest ones for all sequences. This support the claim that the redundant odometry system yields better overall results.

For the individual odometry, we can infer that the vehicle dynamics aided LiDAR-inertial odometry has the best accuracy. This is mainly due to the accurate and direct range measurement. However, since the LiDARs are all installed at a low height, the odometry is more prone to fail at textureless scenarios, such as a long bridge in *City-day-busy1* and a tunnel in *Tunnel1*. For the visual approaches, it is clear the stereo camera outperforms the monocular camera with less scale drift and tracking loss. With the increased travelling distance, the monocular camera based method is merely dependent on GNSS for scale correction, and the *Free-1* and *Free-2* all have a similar accuracy with in-vehicle GNSS. For the vehicle dynamics and kinematics, they work perfectly at constant velocity and less turning scenarios, such as the two sequence on the highways, *Highway1* and *Highway2*. On the contrary, they are less reliable for the parking sequences, where turnings and wheel slipping are inevitable.

To further visualize how our system picks different odometry, we demonstrate when and which method have been selected on *City-night-busy* since all solutions are available along the path.

To further visualize how our system picks different odometry, we demonstrate when and which method have been selected on *City-night-busy* since all solutions are available along the path.

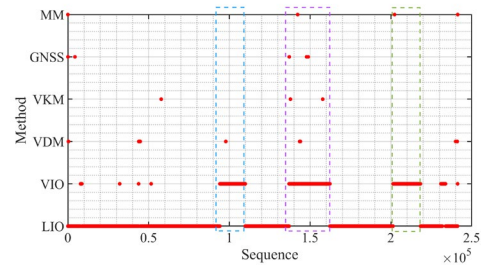


Fig. 8. Visualization of picked odometry at different time sequences for *City-night-busy*. The three insets below show the visual view of LIO failure scenes. The first is a noise barrier, where the laser scans cannot pass through, leading to degeneracy. The second is on a long bridge, which is a feature-less scene. The third is in a narrow lane, where many pedestrians block the LiDAR views.

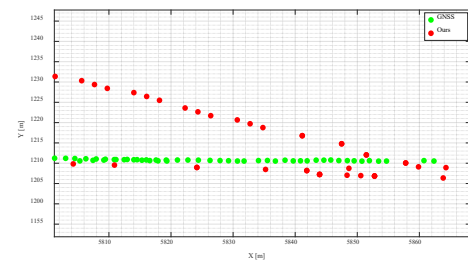


Fig. 9. The robustness evaluation against GNSS outliers. The green and red dots are the position output from in-vehicle GNSS and our system, respectively.

TABLE III  
 SHORT DESCRIPTION AND ACCURACY EVALUATION FOR ALL THE SEQUENCES

| RMSE [m] / MAX [m], with – and bold number indicates meaningless and best result, respectively. |                               |               |                  |                  |             |                  |                  |                   |
|-------------------------------------------------------------------------------------------------|-------------------------------|---------------|------------------|------------------|-------------|------------------|------------------|-------------------|
|                                                                                                 | Length [km]<br>/ duration [s] | IV            | Odom-pure        | ORB-SLAM2        | Vins-mono   | FAST-LIO2        | GNSS             | Ours              |
| <i>Pure-campus1</i>                                                                             | 1.1/198                       | <i>HQ-1</i>   | 1.64/3.16        | 0.94/1.99        | 1.18/2.85   | 0.54/1.45        | 6.09/18.51       | <b>0.37/0.72</b>  |
| <i>Pure-campus2</i>                                                                             | 1.2/196                       | <i>Free-1</i> | 4.18/22.36       | 6.08/15.86       | 4.21/17.43  | - / -            | 8.23/19.60       | <b>3.94/12.17</b> |
| <i>Campus-to-city1</i>                                                                          | 1.5/215                       | <i>HQ-2</i>   | 1.67/2.88        | 1.89/3.92        | 1.07/2.10   | 1.17/2.26        | 6.11/13.42       | <b>0.94/1.73</b>  |
| <i>Campus-to-city2</i>                                                                          | 3.6/589                       | <i>Free-2</i> | 2.95/1.92        | 4.87/7.56        | 5.69/10.24  | - / -            | <b>4.48/8.84</b> | 4.78/9.73         |
| <i>Ground-parking</i>                                                                           | 0.8/187                       | <i>HQ-2</i>   | 3.74/13.97       | 1.25/2.68        | 1.56/1.22   | <b>0.14/0.38</b> | 3.66/6.57        | 0.16/0.32         |
| <i>U-parking1</i>                                                                               | 0.6/143                       | <i>Free-2</i> | 9.79/36.12       | <b>0.56/0.87</b> | 0.83/1.28   | - / -            | - / -            | 0.65/1.01         |
| <i>U-parking2</i>                                                                               | 0.9/194                       | <i>HQ-1</i>   | 12.62/47.35      | - / -            | - / -       | 1.75/5.38        | - / -            | <b>1.19/3.35</b>  |
| <i>City-day-busy1</i>                                                                           | 23.7/2142                     | <i>HQ-1</i>   | 2.75/16.93       | 3.12/10.53       | 1.88/9.62   | - / -            | 5.29/32.38       | <b>1.42/4.73</b>  |
| <i>City-day-busy2</i>                                                                           | 19.6/1985                     | <i>Free-2</i> | 4.28/11.62       | 5.33/11.28       | 5.59/12.54  | - / -            | 7.94/71.12       | <b>4.21/9.58</b>  |
| <i>City-night-busy</i>                                                                          | 25.3/2438                     | <i>Free-1</i> | 5.75/27.25       | 5.72/10.94       | 5.83/11.01  | - / -            | 6.32/66.53       | <b>3.37/9.55</b>  |
| <i>City-night-free</i>                                                                          | 9.8/581                       | <i>HQ-1</i>   | 5.82/8.87        | 3.25/6.33        | 2.98/9.15   | <b>2.17/7.23</b> | 5.26/73.98       | 2.47/5.68         |
| <i>Highway1</i>                                                                                 | 33.5/1113                     | <i>Free-1</i> | 3.55/16.73       | 4.06/24.78       | 3.82/17.98  | - / -            | 4.18/33.75       | <b>3.52/14.27</b> |
| <i>Highway2</i>                                                                                 | 40.6/1314                     | <i>HQ-2</i>   | 2.96/14.72       | 2.65/12.15       | 2.83/11.79  | 2.27/6.19        | 4.37/25.80       | <b>2.13/6.72</b>  |
| <i>Tunnel1</i>                                                                                  | 0.8/51                        | <i>HQ-1</i>   | 1.97/3.89        | 2.98/9.92        | 3.28/13.77  | - / -            | - / -            | <b>1.75/4.34</b>  |
| <i>Tunnel2</i>                                                                                  | 1.7/99                        | <i>Free-2</i> | <b>2.52/4.70</b> | 9.63/33.68       | 10.34/38.77 | - / -            | - / -            | 2.79/5.3          |

TABLE IV  
 PICKING TIMES OF DIFFERENT ODOMETRY FOR CITY-DAY-BUSY1

| Method     | LIO    | VIO   | VDM  | VKM | GNSS | MM |
|------------|--------|-------|------|-----|------|----|
| Pick times | 180938 | 58627 | 1570 | 18  | 148  | 54 |

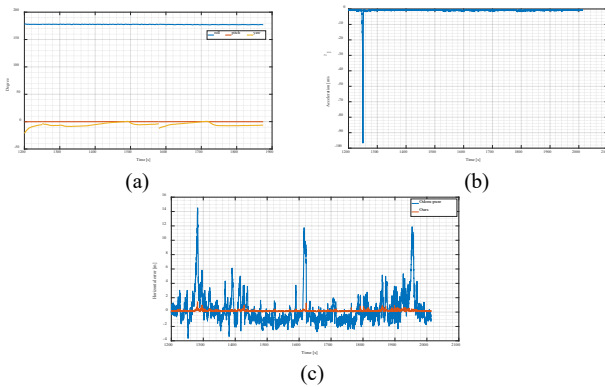


Fig. 10. The robustness evaluation against inertial sensor outliers. (a) is the attitude outputs from IMU, where the yaw direction is not consistent. (b) presents the linear acceleration output, where an outlier exists. (c) compares the horizontal error of Odom-pure and our system w.r.t. ground truth.

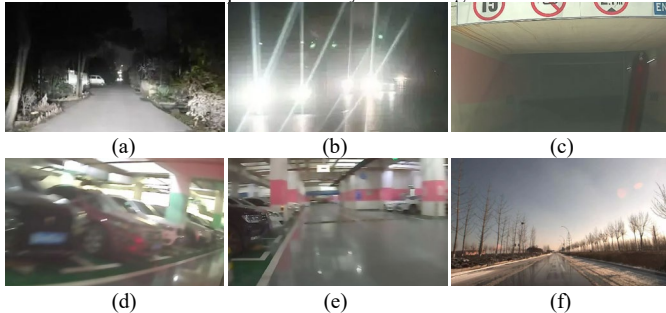


Fig. 11. Six typical visual camera failure examples. (a) is the low lighting in *City-night-free*. (b) shows the gleaming head lights in *City-night-busy*. (c) presents the fast-changing lighting conditions when entering the underground parking lot of *U-parking1*. (d) presents the grievous blurred image due to low lighting conditions of *U-parking1*. (e) shows the reflections on the ground of *U-parking2*. (f) demonstrates the snowy weather of *Highway2*, which will cast many reflections on the ground, or even blind the sensor.

We denote LIO, VIO, VDM, VKM and MM as the LiDAR-inertial, Visual-inertial, vehicle dynamics and kinematics, as well as map matching odometry. The output frequency of LIO,

VIO, VDM, and VKM is at 100 Hz, whereas the MM and in-vehicle GNSS is at 50 Hz. We list the picking times of each odometry in TABLE IV, where LIO is the dominant selection. As shown in Fig. 8, the LIO has three long-during failures along the path. In contrast with Fast-lio2 which has a total paralysis towards the first failure, our LIO can reinitialize and recover from the failure.

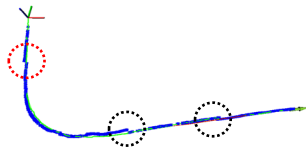
### 5.3 Robustness

This module presents our system's robustness and supports the claim that the proposed system is redundant and robust to different sensor failure cases.

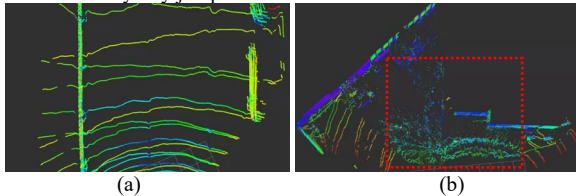
The first failure cases are about loss of data streams. Since a large volume of data is transmitted in the vehicle, some sensor information is inevitable to suffer from transmission loss. As a filter-based system without failure detection, Fast-lio2 has a high dependence of sensor data quality. The IMU has a five-second data lost in *Tunnel1*, and the real-time odometry of Fast-lio2 has a large vertical divergence within this period (more than 100 m). Besides, this error is non-reversible when no global correction is available. Since our modified version of ORB-SLAM2 and Vins-mono still relies on the preintegration result of vehicle dynamics, they provide no results when no IMU data is sent. In comparison, our system switch to vehicle dynamics or vehicle kinematics for pose estimation when both LiDAR-inertial and Visual-inertial system is not working. The maximum error happens at the switching point, where our system is verifying and deciding which system to change to. The left camera of the stereo camera has no output for one-minute while traversing inside the campus for *Campus-to-city1*. As a visual-centric solution, both the ORB-SLAM2 and Vins-mono generate no odometry output within this period. On the contrary, our system is robust to this failure with redundant odometry and has the best performance among the others.

The second focuses on sensor outliers. This situation mainly happens for the inertial sensors and GNSS. For the sequence *Pure-campus1*, the well-grown trees fully cover the lanes, leading to many GNSS outliers as shown in Fig. 9. We can infer that the switch manner of our system can effectually choose the

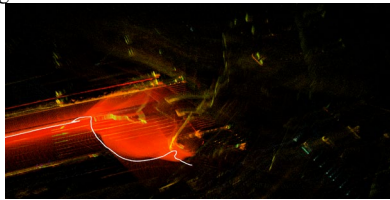
best odometry candidate at the right time, where our trajectory is smooth and follows the straight driving line. Since the inertial measurements from CAN bus and IMU all suffer from outliers,



**Fig. 12.** Visual illustration of the pose estimation failures caused by changing lighting conditions (red dashed circle) and blurred image (black dashed circle). The blue trajectory is from the ORB-SLAM2, whereas the green trajectory is ours. It is seen many tiny jumps exist on the curve of ORB-SLAM2.



**Fig. 13.** Visual illustration of the Leishen CH32 LiDAR failures in the rainy days. (a) shows the irregular scan lines against heavy rains, where the circular scan rings are not smooth even no obstacles are detected. (b) presents the large scan noise against wet vegetations. It is seen no circular scan lines can be found in the red rectangle.



**Fig. 14.** Visualization of Fast-lio2 failure due to blockage of nearby vehicles. The white line is the real-time trajectory of Fast-lio2, it has a large curve at this degenerated district (drive straight forward).

such as large accelerations and inconsistent gyro outputs, the Odom-pure will generate many outliers even with GNSS assistance. Take a near-straight driving segment of *Highway1* for example, the raw input has some clearly wrong data as plotted in Fig. 10(a) and Fig. 10(b), leading to some outliers for Odom-pure. In comparison, our system can choose the proper candidate odometry, and provide a smooth error curve.

The third is about environment dependent sensor failures. As shown in Fig. 11, visual cameras are prone to fail at bad illumination scenarios. Some of these failures is detectable, such as the low lighting scenario presented in Fig. 11(a), where insufficient amount of feature points is extracted. However, the sudden change of illumination condition, Fig. 11(b) and Fig. 11(c), is usually hard to detect, and may cause pose estimation errors. Besides, the heavily blurred image in Fig. 11(d), reflections on ground in Fig. 11(e) and Fig. 11(f) will also lead to inconsistent odometry results as visualized in Fig. 12. In contrary, our system can detect such errors and generate a smooth trajectory. We also find the LiDAR generates inaccurate range measurement in bad weather. As visualized in Fig. 13, many noises exist on the scan lines in rainy days, leading to less-reliable correspondence tracking. Listed in TABLE III, Fast-lio2 has a large maximum error for *City-night-free* due to the LiDAR sensor noise. In comparison, our system can detect and discard this outlier with redundant odometry.

The fourth is about individual pose estimation failures. Since the vehicles and pedestrians in visual image are all detected and

excluded for pose estimation, our system is robust to highly dynamic scenes. However, our system may fail at the feature-poor districts without degeneracy analysis. Since the sensor mounting height is limited, the four LiDARs can be easily disturbed by nearby vehicles. As shown in Fig. 14, Fast-lio2 fails at a crossing of *City-day-busy1*, where many vehicles are stopping around. Our LiDAR-inertial system also fails at this scene, but the global state estimation is not influenced with redundant odometry. Besides, our system is also self-recovery, where the LiDAR-inertial system is reinitialized when long-time large errors are detected. On the contrary, this temporary error is non-reversible for Fast-lio2, and the whole system died.

#### 5.4 Time Complexity

The efficiency of our system is evaluated by collecting the average processing time of each frame. Unlike many other works which compute the time consumption for each step, we merely care about the final odometry output. Fig. 15 presents the average time cost of generating each odometry result for all the sequences.

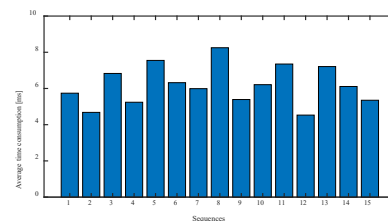
It is seen our system does not become slower with increased travel distance. For the redundant odometry, we can infer that LIO highly increases the system burden. For similar scenes (e.g. *Pure-campus1* and *Pure-campus2*) the average time usage has a 20% growth when LIO is added to the system. The dominant computation part of LIO is the residual computation and nearest neighbor searching.

#### 5.5 Application in Motion Controlling

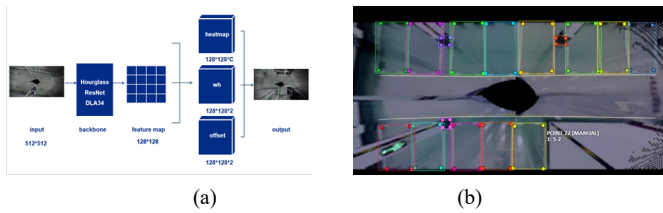
This module is designed to show that our system can be used with perception results, allowing a better motion controlling. We use *Free-2* to develop an auto parking assist system (APA). APA utilizes the parking line detection results from 4 surround view cameras, the occupancy avoidance from 5 radars and 12 ultrasonic sensors, and chassis information from CAN bus to realize an automated parking system. We first get the BEV image from the IPM results. Then we utilize CenterNet [48] to detect the four key points of a parking area, the network architecture and detection results are visualized in Fig. 16.

TABLE V  
THE PARKING BOUNDARY DISTANCE COMPARISON

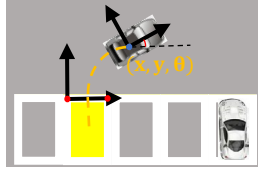
|          | Left boundary |           | Right boundary |           |
|----------|---------------|-----------|----------------|-----------|
|          | Max [cm]      | Mean [cm] | Max [cm]       | Mean [cm] |
| Original | 9.78          | 3.47      | 6.26           | 3.35      |
| Ours     | 4.63          | 1.94      | 4.37           | 1.98      |



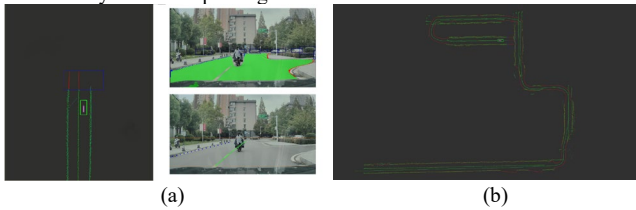
**Fig. 15.** The average computing time of our system for each sequence, the number on x-axis indicates the related sequence in TABLE II (top to bottom).



**Fig. 16.** The applied network structure in (a) and the detected key points from the BEV image in (b).



**Fig. 17.** The workflow of APA. When the user selects a parking area on the human interface (the yellow one), the APA system will setup a local coordinate based on the detected key points (two red points). Then the IV pose  $(x, y, \theta)$  w.r.t the given parking area will be continuously sent to the control module until the IV is fully inside the parking area.



**Fig. 18.** The visual recognition and semantic mapping process. (a) shows the real-time recognition, segmentation, and local mapping process. (b) presents a segment of generated visual semantic map (1 km long).



**Fig. 19.** The real-time constructed semantic high precision point cloud map of an underground parking lot. The color green indicates the lanes, parking lines, or road markings detected by LiDAR and surround view cameras. The red line is the real-time vehicle trajectory generated by our system.

The detection only approach is not robust for this task [49] due to potential false positive or false negative detections. We hereby leverage the Kalman filter to solve this problem, where the detected key points are treated as noisy observations and the pose estimation results from above are viewed as states. Since the key points cannot be detected all the time, the observations are only updated intermittently. The workflow is visualized in Fig. 17. We care about the vehicle’s center to the parking spot boundary in the APA process. Therefore, we select the same parking spot on the user interface, and let the APA system automatically parking into the given spot 10 times.

We follow the metric evaluation method in [18] to quantitatively assess the system performance, where the difference between the vehicle’s center to the parking spot boundary in the real world and on the human interface are measured. Listed in TABLE V, we compare the difference of original APA in *Free-2* and our version. It is seen that our method has a 40% improvement over the original APA.

## 5.6 Application in Map Construction

In this module, we seek to present that our system can be integrated into reconstructing both visual semantic maps and high precision point cloud maps.

We use the forward camera of *HQ-2* to perform segmentation of lanes, road markings, and drivable areas as visualized in Fig. 18(a). The 2D lanes and markings are lifted into 3D space in the body frame using IPM. Based on the pose estimation results from our system, these features can be directly transferred into global coordinates as shown in Fig. 18(b).

Similarly, we can leverage the cooperative detection results from LiDAR and cameras of *HQ-1* to generate high precision semantic point cloud maps as shown in Fig. 19. In addition, we employ the detected loop closure to correct the accumulated drift. We maintain local maps for every 20 meters along the path. For each latest local map, we compare it with maintained local maps. Two local maps are registered by the ICP method. They are treated as a match if the transformation fails into a threshold.

Then we get the relative pose between these two local maps, which will be used to correct the accumulated drift.

## 6. CONCLUSION

In this paper, we presented a robust and resilient odometry framework for IVs. Our framework first tests the confidence and monitors the health of each input data stream. The temporary sensor outliers or data loss will be discarded for pose estimation, in case of long during failures, our system will warn the drivers in the user interface. Then the multiple streams are sent to parallel running odometry including vehicle kinematics and dynamics preintegration, LiDAR-inertial, Visual-inertial, GNSS, as well as map matching. All the algorithms share the general sensor setup as well as the lightweight design. The individual results then undergo a quality verification, with the clearly wrong odometry removed. Besides, the redundant pose estimation can detect the temporary errors of certain estimation engines, and triggers the re-initialization process.

We evaluated our system on both research purpose and mass-produced IVs and supported all claims made in this paper. The experimental results suggest that the proposed system is resilient and robust to individual sensor and odometry failures, e.g., loss of data streams, measurement noises, illumination variations, LiDAR degeneracy, and highly dynamic scenes. Besides, our approach achieves better performance than all the baselines in most scenarios, especially for the large suppression of the maximum errors.

The current candidate selection assumes the vehicle dynamic can provide reliable reference for short period, which may be inaccurate for some applications. Therefore, we plan to use reinforcement learning to design better odometry proposal selection criteria in the future work.

## REFERENCES

- [1] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [2] J. Zhang and S. Singh, “LOAM: Lidar Odometry and Mapping in Real-time,” in *Robotics: Science and Systems*, 2014, vol. 2, no. 9.
- [3] X. Ban, H. Wang, T. Chen, Y. Wang, and Y. Xiao, “Monocular visual odometry based on depth and optical flow using deep learning,” *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–19, 2020.
- [4] L. Pei *et al.*, “IVPR: An instant visual place recognition approach based on structural lines in Manhattan world,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4173–4187, 2019.
- [5] C. Zhang, L. Chen, and S. Yuan, “ST-VIO: Visual-inertial odometry combined with image segmentation and tracking,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8562–8570, 2020.

- [6] Y. Wang, Y. Lou, W. Song, and Z. Tu, "A Tightly-Coupled Framework for Large-Scale Map Construction with Multiple Non-Repetitive Scanning LiDARs," *IEEE Sens. J.*, 2022.
- [7] J. Zhang and S. Singh, "Laser-visual-inertial odometry and mapping with high robustness and low drift," *J. Field Robot.*, vol. 35, no. 8, pp. 1242–1264, 2018.
- [8] W. Wang, J. Liu, C. Wang, B. Luo, and C. Zhang, "Dv-loam: Direct visual lidar odometry and mapping," *Remote Sens.*, vol. 13, no. 16, p. 3340, 2021.
- [9] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R2LIVE: A Robust, Real-time, LiDAR-Inertial-Visual tightly-coupled state Estimator and mapping," *ArXiv Prepr. ArXiv210212400*, 2021.
- [10] T. Shan, B. Englot, C. Ratti, and D. Rus, "LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping," *ArXiv Prepr. ArXiv210410831*, 2021.
- [11] R. C. Luo, C.-C. Yih, and K. L. Su, "Multisensor fusion and integration: approaches, applications, and future research directions," *IEEE Sens. J.*, vol. 2, no. 2, pp. 107–119, 2002.
- [12] A. Santamaria-Navarro, R. Thakker, D. D. Fan, B. Morrell, and A. Aghamohammadi, "Towards resilient autonomous navigation of drones," in *The International Symposium of Robotics Research*, 2019, pp. 922–937.
- [13] A. Reinke, X. Chen, and C. Stachniss, "Simple But Effective Redundant Odometry for Autonomous Vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9631–9637.
- [14] K. J. Wu, C. X. Guo, G. Georgiou, and S. I. Roumeliotis, "Vins on wheels," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5155–5162.
- [15] Z. Hong, Y. Petillot, and S. Wang, "Radarslam: Radar based large-scale slam in all weathers," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5164–5170.
- [16] M. Maaref, J. Khalife, and Z. M. Kassas, "Lane-level localization and mapping in GNSS-challenged environments by fusing lidar data and cellular pseudoranges," *IEEE Trans. Intell. Veh.*, vol. 4, no. 1, pp. 73–89, 2018.
- [17] S. Jung, J. Kim, and S. Kim, "Simultaneous localization and mapping of a wheel-based autonomous vehicle with ultrasonic sensors," *Artif. Life Robot.*, vol. 14, no. 2, pp. 186–190, 2009.
- [18] T. Qin, T. Chen, Y. Chen, and Q. Su, "Avp-slam: Semantic visual mapping and localization for autonomous vehicles in the parking lot," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5939–5945.
- [19] 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)*, 2018, pp. 4758–4765.
- [20] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1885–1892.
- [21] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lid2: Fast direct lidar-inertial odometry," *IEEE Trans. Robot.*, 2022.
- [22] W. Xu and F. Zhang, "Fast-lid: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, 2021.
- [23] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "Lic-fusion: Lidar-inertial-camera odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5848–5854.
- [24] X. Zuo *et al.*, "Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5112–5119.
- [25] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.
- [26] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016.
- [27] Z. Yan, W. Guan, S. Wen, L. Huang, and H. Song, "Multirobot cooperative localization based on visible light positioning and odometer," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021.
- [28] Y. Huang, C. Xue, F. Zhu, W. Wang, Y. Zhang, and J. A. Chambers, "Adaptive recursive decentralized cooperative localization for multirobot systems with time-varying measurement accuracy," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–25, 2021.
- [29] B. Xu, S. Li, A. A. Razzaqi, Y. Guo, and L. Wang, "A novel measurement information anomaly detection method for cooperative localization," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–18, 2021.
- [30] B. Xu, Y. Guo, L. Wang, and J. Zhang, "A novel robust Gaussian approximate smoother based on EM for cooperative localization with sensor fault and outliers," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2020.
- [31] M. Maimone, Y. Cheng, and L. Matthies, "Two years of visual odometry on the mars exploration rovers," *J. Field Robot.*, vol. 24, no. 3, pp. 169–186, 2007.
- [32] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, 2012.
- [33] D. E. Smith and J. M. Starkey, "Effects of model complexity on the performance of automated vehicle steering controllers: Model development, validation and comparison," *Veh. Syst. Dyn.*, vol. 24, no. 2, pp. 163–181, 1995.
- [34] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.
- [35] Y. Wang, Y. Lou, Y. Zhang, W. Song, F. Huang, and Z. Tu, "A Robust Framework for Simultaneous Localization and Mapping with Multiple Non-Repetitive Scanning Lidars," *Remote Sens.*, vol. 13, no. 10, p. 2015, 2021.
- [36] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Inf. Fusion*, vol. 14, no. 1, pp. 57–77, 2013.
- [37] D. He, W. Xu, and F. Zhang, "Embedding manifold structures into kalman filters," *ArXiv E-Prints*, p. arXiv-2102, 2021.
- [38] Y. Cai, W. Xu, and F. Zhang, "ikd-Tree: An Incremental KD Tree for Robotic Applications," *ArXiv Prepr. ArXiv210210808*, 2021.
- [39] B. Nagy, P. Foehn, and D. Scaramuzza, "Faster than FAST: GPU-accelerated frontend for high-speed VIO," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 4361–4368.
- [40] W. Wen and L.-T. Hsu, "Towards Robust GNSS Positioning and Real-time Kinematic Using Factor Graph Optimization," *ArXiv Prepr. ArXiv210601594*, 2021.
- [41] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, p. 1729881419841532, 2019.
- [42] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *European Conference on Computer Vision*, 2020, pp. 194–210.
- [43] A. Segal, D. Haehnel, and S. Thrun, "Generalized-icp," in *Robotics: science and systems*, 2009, vol. 2, no. 4, p. 435.
- [44] Wang Y, Lou Y, Song W, et al. A LiDAR-inertial SLAM tightly coupled with dropout-tolerant GNSS fusion for autonomous mine service vehicles[J]. *IEEE Transactions on Instrumentation and Measurement*, 2023, 72: 1-15.
- [45] Wang Y, Song W, Zhang Y, et al. Four years of multimodal odometry and mapping on the rail vehicles[J]. *Journal of Field Robotics*, 2024, 41(2): 227-257.
- [46] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [47] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.
- [48] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6569–6578.
- [49] W. Wang, Y. Song, J. Zhang, and H. Deng, "Automatic parking of vehicles: A review of literatures," *Int. J. Automot. Technol.*, vol. 15, no. 6, pp. 967–978, 2014.