

LVIO-SAM: A Multi-sensor Fusion Odometry via Smoothing and Mapping

Xinliang Zhong, Yuehua Li*, Shiqiang Zhu, Wexuan Chen, Xiaoqian Li, and Jason Gu

Abstract— State estimation with sensors is critical for the mapping and navigation of mobile robots. Since different sensors have different performances in the environment, how to fuse different sensors together will be a problem. In this paper, we propose a multi-sensor fusion odometry, LVIO-SAM, which fuses LiDAR, stereo camera and inertial measurement unit (IMU) via smoothing and mapping. The pre-integration motion estimation from the IMU eliminates the skew of the point cloud and produces initial guesses for the optimization of lidar odometry. The obtained lidar odometry is used to estimate the bias of the IMU and as the initial value of the triangulation of the visual odometry. The visual odometry is used as a between factor for the motion estimation of the entire system. To ensure the real-time performance, we separately marginalize the old lidar scans and visual 3D points for pose optimization. The method has been widely evaluated on datasets gathered from simulation environment and public datasets. To benefit the community, we open source our simulation environment and codes on our Github¹.

Index Terms— multi-sensor fusion, state estimation, visual-inertial odometry, lidar-inertial odometry

I. INTRODUCTION

Enabling mobile robots to perform tasks and interact with people in the real world has been a long-term goal that has driven scientists and engineers forward [1]. One of the fundamental tasks to be solved is to give a robot the sense of position and environment. The Simultaneous localization and mapping (SLAM) algorithm is designed to solve this task. State estimation is the core of SLAM algorithm. With the popularity of cameras and the development of visual SLAM algorithms, the community has also developed a deep connection with the field of computer vision and methods such as Bundle Adjustment (BA) [2]. The vision-based state estimator is not robust enough in practice due to the influence of illumination, small field of view (FoV), motion blur and so on. Recent advances in lidar hardware have promoted the research on lidar based state estimation [3], [4], [5], [6]. The wide FoV, density, range, and accuracy of lidar sensors make them suitable for navigation, localization, and mapping tasks. In most cases, SLAM systems based on LiDAR have proven their advantages in accuracy and robustness. However, the pure LiDAR approach will fail in some degraded situations, such as through featureless tunnels or straight corridors.

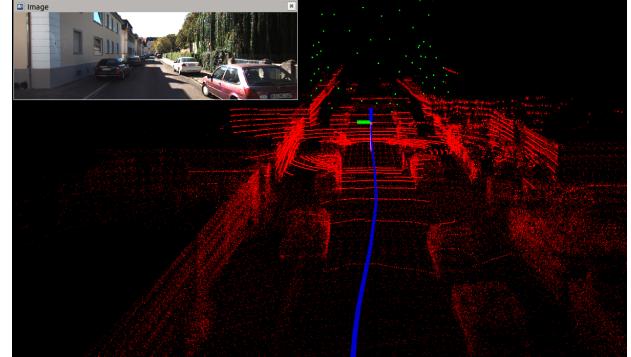


Figure 2: Output of the proposed method in KITTI dataset [28]. The blue path is computed trajectory and the LiDAR maps are colored red, and the 3D points of visual is colored green.

The visual-based system is efficient and lightweight. However, the range of observations is limited and they depend on good data associations to perform well. As a result, they fail in environments without many visual features. The IMU produces high-frequency measurements, which are reasonable for short intervals to correct deviations, but drift quickly.

In this paper, we propose a multi-sensor fusion odometry via smoothing and mapping, LVIO-SAM, to address the aforementioned problems. We use raw IMU measurements to de-skew the point cloud and estimate sensor motion during lidar scanning. In addition to de-skewing the point cloud, the pre-integration motion can also be used as an initial guess for lidar odometry optimization. The obtained lidar odometry is used to estimate the bias of the IMU in the factor graph, and at the same time as the initial value of the 3D points triangulation in the visual odometry. To ensure the real-time performance, we maintain a sliding window of key frame poses and marginalize the old lidar scans and visual 3D points.

The main contributions of our work are:

- We take advantage of the measurement results of LiDAR, stereo camera and inertial sensors to fuse them in a tightly coupled manner.
- We maintain a sliding window which is updated using lidar and visual odometry rather than observations like edge/plane feature of lidar scans

All authors are with the Intelligent Robot Research Center of Zhejiang Lab, Hang Zhou, China. {xinliang,zxl, liyh, zhushq, wxchen, lixiaoqian, jgu}@zhejianglab.com

*Corresponding author: Yuehua Li.

¹ <https://github.com/TurtleZhong/LVIO-SAM>.

This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0102700, Stable support project of State Administration of Science, Technology and Industry for National Defence Grant, PRC under grant No.HTKJ2019KL502005.

and the tracked visual feature points to streamline the fusion system.

Our system is carefully engineered and open sourced to benefit the whole robotics community.

II. RELATED WORK

we review existing works closely related to our work, including visual odometry (VO), lidar-inertial odometry (LIO), and lidar-visual-inertial odometry (LVIO) methods.

Visual odometry (VO) can be divided into two categories: feature-based methods and direct methods. feature-based methods typically extract sparse points feature and minimize the reprojection error with feature points while direct methods minimize the photometric errors, with no need for feature points. In our work, we limit our consideration to stereo cameras. Among the most popular publicly available VO pipelines, S-PTAM [7], ORB-SLAM2 [8], and Vins-Fusion [9], [10] are feature-based, while S-DSO [11] is direct methods. VINS-Fusion is an optimization-based multi-sensor state estimator, which achieves accurate self-localization for autonomous applications (drones, cars, and AR/VR).

The existing work of lidar-inertial odometry (LIO) can be divided into two categories: loosely-coupled and tightly-coupled. The loosely-coupled method processes the two sensors separately to infer their motion constraints, while the tightly-coupled method directly fuses lidar and inertial measurement through joint optimization. Lidar based odometry are becoming increasingly popular thanks to the preliminary work of Zhang et al. [3], who proposed the LOAM algorithm. LOAM performs point feature to edge/plane scan-matching to find correspondences between lidar scans. IMU measurements are used to de-skew lidar scans and give a motion prior of scan matching. LeGO-LOAM [4] and A-LOAM [12] are the LOAM algorithm family, One is optimized for ground vehicle mapping tasks and the other is an implementation of LOAM which uses Eigen and Ceres Solver [13] to simplify code structure. Since IMU measurements are not used in the optimization step, those algorithm can be classified as a loosely-coupled method. Compared with loosely coupled methods, tightly coupled methods show higher robustness and accuracy, so they have attracted increasing research interest in recent years. In LIO-Mapping [6] the author proposed a graph-based optimization framework which jointly optimizes the pre-integrated IMU measurements and the 3D plane factor derived from the closest point-to-plane representation. LINS [14] is the first tightly-coupled LIO that solves the six degrees of freedom (6-DOF) ego-motion via iterated error-state Kalman filter (ESKF) [15]. Compared with the former algorithm, LIO-SAM [16] optimizes a sliding window of keyframe poses in a factor graph to achieve higher accuracy and robustness. Similarly, LiLi-OM [17] is designed for both conventional and solid-state LiDARs based on sliding window optimization.

To avoid degradation and make the system more robust, LVIO methods was explored in some more recent works [18], [19], [20], [21], [22]. In LIMO [19], the author proposed a visual odometry system based on beam

adjustment. They corrected the depth to ensure the correct scale by re-projecting lidar measurements into the image space and associating them with visual features. VIL-SLAM [18] combines the VIO with a lidar based odometry as a separate subsystem for tightly combining different sensor modes rather than joint optimization. Similarly, LVI-SLAM [22] is composed of two subsystems, a visual-inertial system (VIS) and a lidar-inertial system (LIS). The two sub-systems can function independently when failure is detected in one of them, or jointly when enough features are detected. Multi-State Constraint Kalman Filter (MSCKF) framework [23] is adopted by many approach. LIC-fusion [20] framework combining sparse visual features, IMU measurements and LiDAR plane and edge features with online spatial and temporal calibration based on the MSCKF framework. Whereas, in a recent follow-up work, LIC-Fusion 2.0 [21], the authors introduce a sliding window based plane feature tracking method for efficient processing of 3D LiDAR point clouds. R2LIVE [24] is a robust, real-time tightly coupled multi-sensor fusion framework. It estimates the state within the framework of error-state iterated Kalman-filter, and further improve the overall precision with factor graph optimization. LVI-SAM[22] is composed of two sub system: a visual-inertial system (VIS) and a lidar-inertial system (LIS). Our method is closest to it, but we centralize the data of three sensors in one system and only build a factor graph using all three factors. Our system uses the IMU data once compared to its twice.

The proposed LVIO-SAM method fuses LiDAR, stereo camera and IMU via smoothing and mapping. visual odometry is used as between factor constrain of the whole factor graph. It maintains a sliding window of key frame poses like LIO-SAM. By fusing different types of sensor measurements, we achieve state estimation of higher accuracy and robustness.

III. METHOD

A. System Overview

We aim to estimate orientation, position, linear velocity of a robot equipped with lidar, stereo cameras, and IMU in real-time with high accuracy and robustness. We denote the world frame as \mathbf{W} and the body frame of robot as \mathbf{B} . For convenience, we also assume that the IMU frame coincides with the robot body frame. The robot state can be written as:

$$\mathbf{x}_i \triangleq [\mathbf{R}_i, \mathbf{p}_i, \mathbf{v}_i, \mathbf{b}_i^g, \mathbf{b}_i^a] \quad (1)$$

where $\mathbf{R}_i \in SO(3)$ is the orientation, $\mathbf{p}_i \in \mathbb{R}^3$ is the position vector. $\mathbf{v}_i \in \mathbb{R}^3$ represent the linear velocity. and the last two elements \mathbf{b}_i^g and \mathbf{b}_i^a are the usual IMU gyro and accelerometer biases. The transform from \mathbf{B} to \mathbf{W} is represented as $\mathbf{T} = [\mathbf{R}|\mathbf{P}]$, $\mathbf{T} \in SE(3)$.

The structure of the whole system is shown in Fig.2. The system receives sensor data from a 3D lidar \mathbf{F} , an IMU \mathbf{I} and stereo cameras \mathbf{C} at different times and frequencies. The full set measurements at time t_k in the sliding window can be written as:

$$\mathbf{Z}_k = [\{\mathbf{F}_i\}, \{\mathbf{I}_j\}, \{\mathbf{C}_l\}] \quad (2)$$

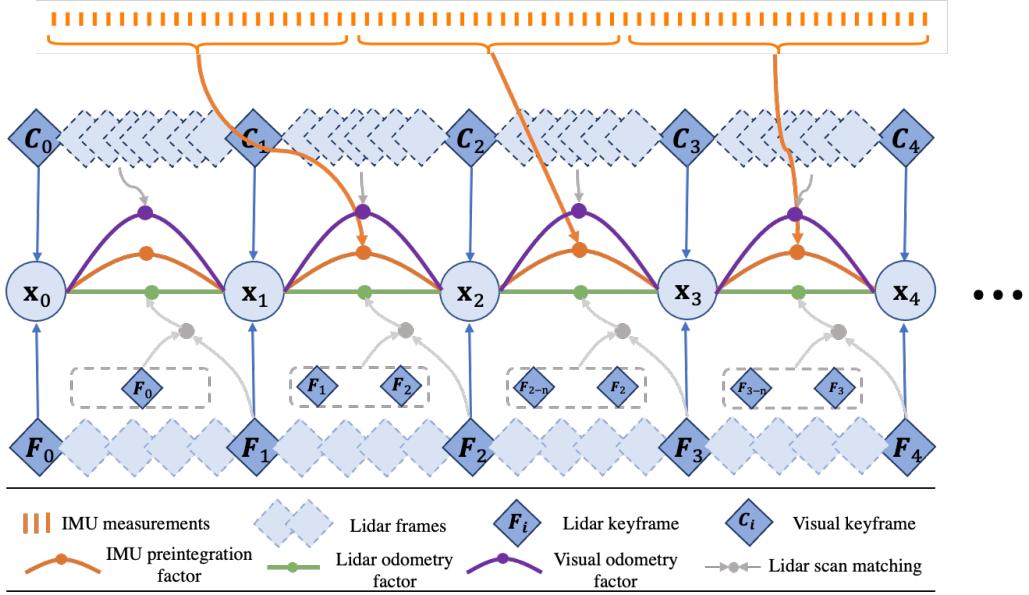


Figure. 2: The system structure of LVIO. The system receives input from a 3D lidar, a stereo camera and an IMU. Three types of factors are introduced to construct the factor graph.: (a) IMU preintegration factor, (b) lidar odometry factor, and (c) visual odometry factor. The generation of these factors is discussed in Section III.

Our goal is to estimate the robot's state X_k and its trajectory using the measurements of these sensors. This can be considered as maximum a posterior (MAP) problem:

$$X_k = \arg \max p(X_k | Z_k) \propto p(X_0)p(Z_k | X_k) \quad (3)$$

The Eq. (3) is equivalent to solving a nonlinear least-squares problem with the assumption of a Gaussian noise model [25]. We build a factor graph which contains four types of factors. Specifically, the four types of factors are: IMU pre-integration factors, lidar odometry factors, visual odometry factors, and a state prior factor. The factor graph is optimized using incremental smoothing and mapping with GTSAM [26]. The following sections describe the process of generating these factors.

B. IMU Preintegration Factor

The true values of acceleration and angular velocity from the IMU raw measurements are defined using Eqs. (4) and (5).

$$\hat{\mathbf{a}}_t = \mathbf{R}_t^{BW}(\mathbf{a}_t - \mathbf{b}_t^a - \mathbf{n}_t^a) + \mathbf{g} \quad (4)$$

$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t - \mathbf{b}_t^\omega - \mathbf{n}_t^\omega \quad (5)$$

where, \mathbf{a} and $\boldsymbol{\omega}$ is the true acceleration and angular velocity from the bias \mathbf{b}_t and white noise \mathbf{n}_t sensor readings \mathbf{a}_t and $\boldsymbol{\omega}_t$ in \mathbf{B} with the rotation from \mathbf{W} to \mathbf{B} , \mathbf{R}_t^{BW} at time t . \mathbf{g} is the gravity vector in \mathbf{B} .

The rotation, velocity and position of the robot at time $t + \Delta t$ can be formulated as follows:

$$\mathbf{R}_{t+\Delta t} = \mathbf{R}_t \exp((\boldsymbol{\omega}_t - \mathbf{b}_t^\omega - \mathbf{n}_t^\omega)\Delta t) \quad (6)$$

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \mathbf{g}\Delta t + \mathbf{R}_t(\mathbf{a}_t - \mathbf{b}_t^a - \mathbf{n}_t^a)\Delta t \quad (7)$$

$$\begin{aligned} \mathbf{p}_{t+\Delta t} &= \mathbf{p}_t + \mathbf{v}_t\Delta t + \frac{1}{2}\mathbf{g}\Delta t^2 \\ &\quad + \frac{1}{2}\mathbf{R}_t(\mathbf{a}_t - \mathbf{b}_t^a - \mathbf{n}_t^a)\Delta t^2 \end{aligned} \quad (8)$$

where, the bias of acceleration and angular velocity remain constant between t and $t + \Delta t$.

We now can follow [27] to preintegrate the high frequency IMU data between two consecutive lidar frames as shown in Fig. 3. The preintegrate measurements between time i and j can be computed using:

$$\Delta \mathbf{v}_{ij} = \mathbf{R}_i^\top(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) \quad (9)$$

$$\Delta \mathbf{p}_{ij} = \mathbf{R}_i^\top(\mathbf{p}_j - \mathbf{p}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2) \quad (10)$$

$$\Delta \mathbf{R}_{ij} = \mathbf{R}_i^\top \mathbf{R}_j \quad (11)$$

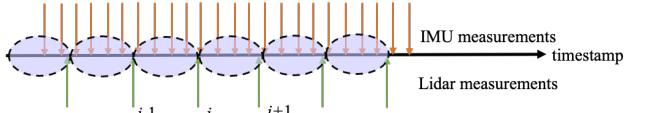


Figure. 3: preinteration of high frequency IMU measurements.

The IMU bias is optimized jointly with the lidar odometry factors and the visual odometry factors in the factor graph.

C. Lidar Odometry Factor

If a lidar scan is received, we extract planar and edge feature by evaluating the roughness of points over a local region. Points with small roughness value are classified as planar features. While edge features are categorized by large roughness value. We denote \mathcal{F}_i^p and \mathcal{F}_i^e the planar and edge features that extract at time i respectively. A lidar frame \mathcal{F}_i , where $\mathcal{F}_i = \{\mathcal{F}_i^p, \mathcal{F}_i^e\}$ is consist of all the features extracted time i . Instead of add every lidar odometry factor to the factor graph, we adopt a key frame strategy. A lidar frame \mathcal{F}_{i+1} is selected as key frame when the pose difference robot between pose \mathbf{x}_{i+1} and previous state \mathbf{x}_i . Following the most keyframe-based method in visual odometry, we extract the n most recent keyframes for motion estimation. We denote \mathcal{M}_i as the maps in \mathbf{W} with the set of recent keyframes $\{\mathcal{F}_{i-n}, \dots, \mathcal{F}_i\}$ associated with the transformations $\{\mathbf{T}_{i-n}, \dots, \mathbf{T}_i\}$. \mathcal{M}_i is composed of planar

maps \mathcal{M}_i^p and \mathcal{M}_i^e edge maps. To compute the transformation of lidar frame \mathcal{F}_{i+1} , we first transform $\{\mathcal{F}_{i+1}^p, \mathcal{F}_{i+1}^e\}$ to global frame \mathbf{W} and obtain $\{\hat{\mathcal{F}}_{i+1}^p, \hat{\mathcal{F}}_{i+1}^e\}$ using the IMU prediction transform. We then follow [3] to find its planar or edge correspondence in \mathcal{M}_i . The point-to-line and point-to-plane distance can be written as:

$$\mathbf{d}_{p_k} = \frac{|(\mathbf{p}_{i+1,k}^p - \mathbf{p}_{i,u}^p) \times (\mathbf{p}_{i,u}^p - \mathbf{p}_{i,w}^p)|}{|(\mathbf{p}_{i,u}^p - \mathbf{p}_{i,v}^p) \times (\mathbf{p}_{i,u}^p - \mathbf{p}_{i,w}^p)|} \quad (12)$$

$$\mathbf{d}_{e_k} = \frac{|(\mathbf{p}_{i+1,k}^e - \mathbf{p}_{i,u}^e) \times (\mathbf{p}_{i+1,k}^e - \mathbf{p}_{i,v}^e)|}{|\mathbf{p}_{i,u}^e - \mathbf{p}_{i,v}^e|} \quad (13)$$

where k, u, v and w are the feature indices to the correspondence set. For a planner feature $\mathbf{p}_{i+1,k}^p$ in $\hat{\mathcal{F}}_{i+1}^p$, $\mathbf{p}_{i,u}^p$, $\mathbf{p}_{i,v}^p$ and $\mathbf{p}_{i,w}^p$ are the corresponding planar patch in \mathcal{M}_i^p . Similarly, $\mathbf{p}_{i,u}^e$ and $\mathbf{p}_{i,v}^e$ are the corresponding edge patch in \mathcal{M}_i^e to edge feature $\mathbf{p}_{i+1,k}^e$ in $\hat{\mathcal{F}}_{i+1}^e$. The lidar pose can be solved using the GaussNewton method by minimizing the above distance. At last, \mathbf{T}_{i+1}^W , the lidar pose of frame $i + 1$ is inserted to the factor graph as lidar odometry constrain.

D. Visual Odometry Factor

When a pair of stereo images arrives, we first perform feature extraction. Corner like features are extracted in the left image C_k^l and tracked using the optical flow method on the right image C_k^r . Similarly, we track the features using the same method between two consecutive image frames. In order to balance computing resources and accuracy, we follow [10] using sliding window to solve the 3D points and the camera poses of the robot. Fig. 4 shows the feature tracking process and the constrain between temporal and spatial image frames.

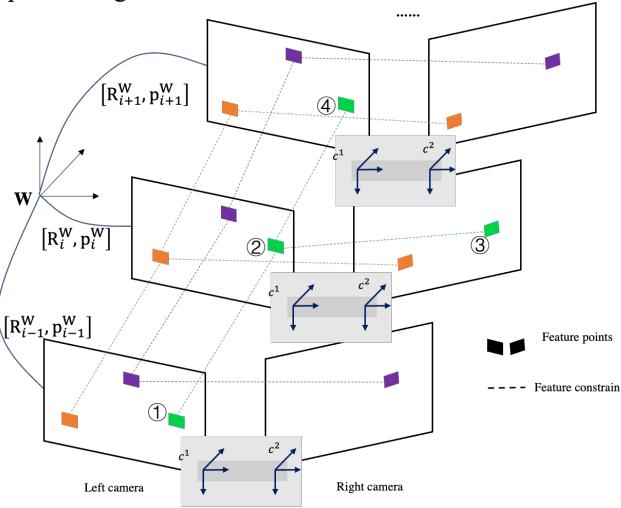


Figure 4 feature tracking process and the constrain between temporal and spatial image frames.

A landmark P , whose position in \mathbf{W} is $x_p \in \mathbb{R}^3$, is observed by multiple states, $\{S\}_p$. It maps to the image coordinates $\mathbf{z} = [u, v]$ through the camera projection model $\pi(x_p): \mathbb{R}^3 \rightarrow \mathbb{R}^2$. Suppose the feature (colored green in Fig. 4) is first observed in frame i , the feature was tracked in several frames. Consequently, a constrain function can be written as:

$$e_{ij} = \mathbf{z}_j - \pi(\mathbf{R}_i^j \pi^{-1}(\mathbf{z}_i) + \mathbf{p}_i^j) \quad (12)$$

where e_{ij} is the reprojection error of feature between frame i and j . $\pi^{-1}(*): \mathbb{R}^2 \rightarrow \mathbb{R}^3$ is the inverse of $\pi(*)$. \mathbf{R}_i^j and \mathbf{p}_i^j is the rotation and translation from frame i to j respectively.

Due to space limitations, we refer the reader to the description from [10], [12] for the detailed derivation of Eq. 12.

In order to ensure the sparsity of the whole factor graph, we only use the pose of keyframe rather than the 3D landmarks as the visual factor. Due to the three sensors have different output frequency, and they are not time synchronized. To integrate the visual odometry into the system, we interpolate a virtual visual pose based on the timestamp of the lidar. Two virtual factors form a between factor for the entire optimization process as shown in Fig. 5.

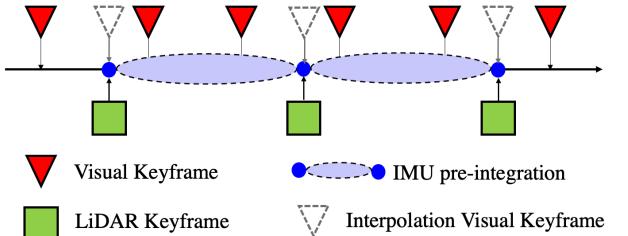


Figure 5. Illustration of visual odometry constraints integrated into the entire system. To ensure the sparsity, only the pose of the keyframe is used for graph optimization.

E. Data Flow Throughout the System

To further illustrate the data interaction between the three sensors, let's make a brief summary. The high-frequency IMU is used to de-skew lidar scans on the one hand, and on the other hand is used for pre-integration as a pre-integration factor to be added to the factor graph. Lidar odometry is computed using the point-to-line and point-to-plane constrains. In the visual odometry process, we extract sparse feature points and tracked them in spatial and temporal image frames. The 3D feature points and visual keyframe pose are jointly optimized in the sliding window with the reprojection error constrain. The lidar pose is used to as the initial value in the triangulation process to improve the visual odometry optimization process. We then construct a factor graph with IMU pre-integration, lidar odometry and visual odometry. The IMU bias is optimized jointly with the lidar odometry factor and visual odometry factors in the factor graph.

IV. EXPERIMENT

We evaluated our algorithm on simulation environment datasets and KITTI datasets [28]. Since very few datasets meet the input requirements of our algorithm. We modify the Gazebo world proposed in [29] and adding our own sensors to test our proposed method. All methods are implemented in C++ and executed using the robot operating system (ROS) in Ubuntu Linux on a laptop equipped with an Intel i7-8700 CPU. We note that only the CPU is used for computation, without parallel computing enabled.

A. CMU Campus Datasets

In this experiment, we test the mapping quality of our proposed method [30]. Since high precision large-scale map of environments relies on accurate odometry. We collect data of CMU Campus in the simulated environment as shown in Fig. 6. We compare LVIO-SAM with existing lidar and lidar-inertial odometry methods, including A-LOAM, LIO-Mapping.

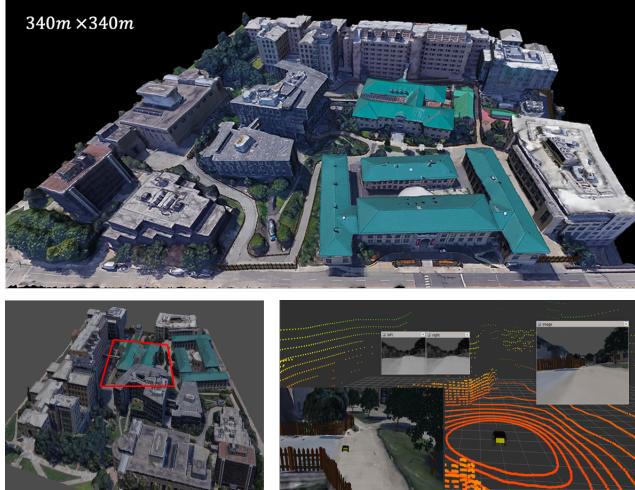


Figure 6. Simulation environment for multi-sensor fusion localization and mapping. The top of figure shows the CMU campus model and the bottom-left corner shows the robot’s range of motion. The data of all sensors is displayed in the bottom-right corner.

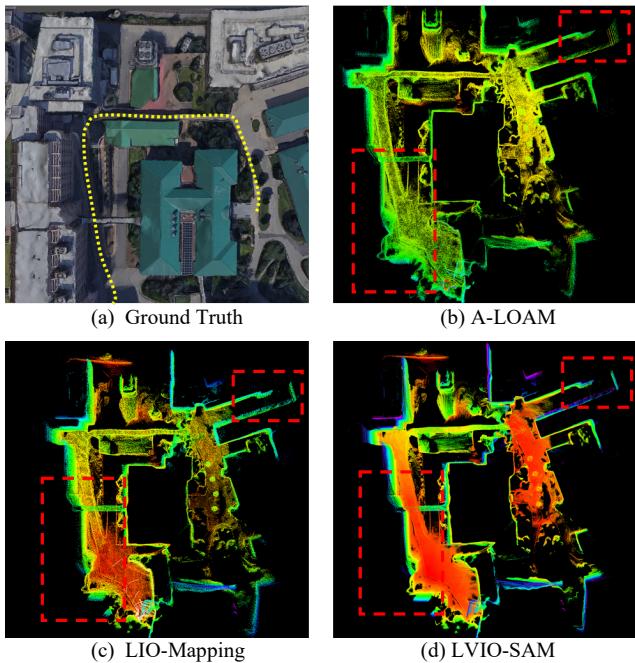


Figure 7. Mapping results of A-LOAM, LIO-Mapping and LVIO-SAM using CMU dataset. The map of A-LOAM and LIO-Mapping in (b) and (c) show numerous blurry structures due to inaccurate pose estimation.

In this test, the map of A-LOAM, shown in Fig. 7(b), diverges at multiple locations when the robot encounters

ups and downs. LIO-Mapping outperforms A-LOAM. However, its map, shown in Fig. 7(c) show blurry structures due to inaccurate pose estimation. Our proposed method, LVIO-SAM outperforms both methods and generated a more consistent map compared with the ground truth, shown in Fig. 7(a).

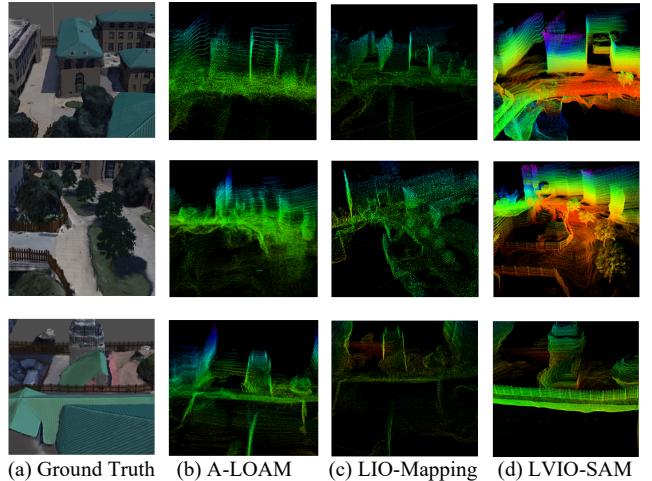


Figure 8. The detail of mapping results. The three rows from top to bottom represent the results of the building plan, trees, and fences.

The consistency of a plane and the integrity of an object are good evaluation criteria for the quality of localization and mapping. Fig. 8 shows the detail mapping results of building plan, trees, and fences. We can clearly distinguish the outline of the building, the structure of the trees. Particularly, we can identify every piece of fence without blur. The experimental results show that LVIO-SAM is accurate enough to reconstruct a dense 3D, high precision map of outdoor environments.

B. KITTI Dataset

We evaluate our proposed system using KITTI [28] Datasets. The datasets contain stereo images (1382x512, 10FPS), 3D Lidar (Velodyne HDL-64, 10Hz), IMU (50Hz) and GPS. All the sensors are synchronized. The ground truth states are provided by Inertial Navigation System (OXTS RT 3003). We add a loop-closure module in our proposed method, referred to as LVIO-SAM-Loop. The complete visualizations of some sequences can be found at the link below².

TABLE I. RMSE IN KITTI DATASETS IN METERS

Sequences [09_30]	Length [km]	RMSE[m]			
		A- LOAM	LIO- Mapping	LVIO- SAM	LVIO- SAM- Loop
drive_0018	2.21	3.49	2.63	1.55	0.97
drive_0027	0.69	0.92	1.38	0.55	0.72
drive_0028	4.13	16.57	20.46	12.46	5.30
drive_0033	1.71	10.94	64.31	12.64	12.44
drive_0034	0.92	2.32	4.38	2.71	2.13

² <https://youtu.be/ci5QTtYvJAY>

We evaluate APE (absolute pose error) between several algorithm by *evo* tool [31]. The RMSE (Root Mean Square Errors) of absolute pose error for KITTI datasets is show in Table. I. For all sequences, the proposed method shows very competitive results, which demonstrates that LVIO-SAM is accuracy in motion estimation. Also, a loop-closure module can decrease the absolute pose error which means improving the global consistency of trajectory estimation. In sequence *drive_0033* and *drive_0034*, we found that the RMSE of A-LOAM and LVIO-SAM are very close. The possible reason of this problem is the IMU intrinsic parameters, which is very important in pre-integration part while the KITTI dataset do not provide. This experiment demonstrates that proposed method is very competitive in trajectory estimation.

V. CONCLUSION

In this paper, we propose a multi-sensor fusion odometry which fuses 3D lidar, stereo cameras and IMU. In order to ensure the real-time performance of the system, we maintain a sliding window which is updated using lidar and visual odometry rather than 3D points (lidar scans and visual points). Experiment results show that our system is competitive.

In the future, we will focus on building a data set containing stereo images, 3D lidar scans and IMU data on real sensors with hardware synchronization, opening to the community and elevating our proposed method on them. We will further simplify the input measurements of the system and replace the stereo images with monocular images. To improve the robustness of the system, we will add loop closure methods based on deep learning to our system. such a system is significant for accuracy and robust robot localization task.

REFERENCES

- [1] Gu, Jason, et al. "Sensor fusion in mobile robot: some perspectives." Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No. 02EX527). Vol. 2. IEEE, 2002.
- [2] Taketomi T, Uchiyama H, Ikeda S. Visual SLAM algorithms: a survey from 2010 to 2016[J]. IPSJ Transactions on Computer Vision and Applications, 2017, 9(1): 1-11.
- [3] Zhang J, Singh S. LOAM: Lidar Odometry and Mapping in Real-time[C]/Robotics: Science and Systems. 2014, 2(9).
- [4] Shan T, Englot B. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 4758-4765.
- [5] Geneva P, Eckenhoff K, Yang Y, et al. LIPS: Lidar-inertial 3d plane slam[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 123-130.
- [6] Ye H, Chen Y, Liu M. Tightly coupled 3d lidar inertial odometry and mapping[C]//2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019: 3144-3150.
- [7] Pire T, Fischer T, Castro G, et al. S-ptam: Stereo parallel tracking and mapping[J]. Robotics and Autonomous Systems, 2017, 93: 27-42.
- [8] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgbd cameras[J]. IEEE transactions on robotics, 2017, 33(5): 1255-1262.
- [9] Qin T, Li P, Shen S. Vins-mono: A robust and versatile monocular visual-inertial state estimator[J]. IEEE Transactions on Robotics, 2018, 34(4): 1004-1020.
- [10] Qin T, Pan J, Cao S, et al. A general optimization-based framework for local odometry estimation with multiple sensors[J]. arXiv preprint arXiv:1901.03638, 2019.
- [11] Wang R, Schworer M, Cremers D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 3903-3911.
- [12] Qin T, <https://github.com/HKUST-Aerial-Robotics/A-LOAM>.
- [13] Agarwal, S., Mierle, K.: Others: Ceres solver, <https://code.google.com/p/ceres-solver/>.
- [14] Qin C, Ye H, Pranata C E, et al. Lins: A lidar-inertial state estimator for robust and efficient navigation[C]//2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020: 8899-8906.
- [15] Sola J. Quaternion kinematics for the error-state Kalman filter[J]. arXiv preprint arXiv:1711.02508, 2017.
- [16] Shan T, Englot B, Meyers D, et al. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020: 5135-5142.
- [17] Li K, Li M, Hanebeck U D. Towards high-performance solid-state-lidar-inertial odometry and mapping[J]. IEEE Robotics and Automation Letters, 2021, 6(3): 5167-5174.
- [18] Shao W, Vijayarangan S, Li C, et al. Stereo visual inertial lidar simultaneous localization and mapping[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 370-377.
- [19] Graeter J, Wilczynski A, Lauer M. Limo: Lidar-monocular visual odometry[C]//2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2018: 7872-7879.
- [20] Zuo X, Geneva P, Lee W, et al. Lic-fusion: Lidar-inertial-camera odometry[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 5848-5854.
- [21] Zuo X, Yang Y, Geneva P, et al. Lic-fusion 2.0: Lidar-inertial-camera odometry with sliding-window plane-feature tracking[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020: 5112-5119.
- [22] Shan T, Englot B, Ratti C, et al. LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping[J]. arXiv preprint arXiv:2104.10831, 2021.
- [23] Mourikis A I, Roumeliotis S I. A multi-state constraint Kalman filter for vision-aided inertial navigation[C]//Proceedings 2007 IEEE International Conference on Robotics and Automation. IEEE, 2007: 3565-3572.
- [24] Lin J, Zheng C, Xu W, et al. R2LIVE: A Robust, Real-time, LiDAR-Inertial-Visual tightly-coupled state Estimator and mapping[J]. arXiv preprint arXiv:2102.12400, 2021.
- [25] Dellaert F, Kaess M. Factor graphs for robot perception[J]. 2017.
- [26] Dellaert F. Factor graphs and GTSAM: A hands-on introduction[R]. Georgia Institute of Technology, 2012.
- [27] Forster C, Carlone L, Dellaert F, et al. On-manifold preintegration for real-time visual-inertial odometry[J]. IEEE Transactions on Robotics, 2016, 33(1): 1-21.
- [28] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite[C]//2012 IEEE conference on computer vision and pattern recognition. IEEE, 2012: 3354-3361.
- [29] Zhang J, Hu C, Chadha R G, et al. Falco: Fast likelihood-based collision avoidance with extension to human-guided navigation[J]. Journal of Field Robotics, 2020, 37(8): 1300-1313.
- [30] Zhang, Weidong, et al. "Learning to predict high-quality edge maps for room layout estimation." IEEE Transactions on Multimedia 19.5 (2016): 935-943.
- [31] Grupp M. *evo*: Python package for the evaluation of odometry and slam[J]. Note: <https://github.com/MichaelGrupp/evo> Cited by: Table, 2017, 7.