PAPER • OPEN ACCESS

Adaptive Adjustment of Factor's Weight for a Multi-Sensor SLAM

To cite this article: Zihan Zhu *et al* 2023 *J. Phys.: Conf. Ser.* **2451** 012004

Adaptive Adjustment of Factor's Weight for a Multi-Sensor SLAM

Zihan Zhu[1,2], Yi Zhang[1,2], Weijun Wang[1,2, 3, 4], Wei Feng[1,2,3, 4, 5,*], Haowen Luo[1,2], Yaojie Zhang[1,2]

[1]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences; 518055, China

[2]University of Chinese Academy of Sciences, Beijing, 100049, China.

[3]Guangdong Provincial Key Lab of Construction Robotics and Intelligent Manufacturing, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China.

[4]Guangdong Provincial International Joint Research Center for Robotics and Intelligent Construction, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China.

[5]Guangdong Provincial Key Lab of Robotics and Intelligent System, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, 518055, China.

*E-mail: wei.feng@siat.ac.cn

Abstract. A multi-sensor fusion simultaneous localization and mapping(SLAM) method based on factor graph optimization that can adaptively modify the weight of the graph factor is proposed in this study, to enhance the localization and mapping capability of autonomous robots in tough situations. Firstly, the algorithm fuses multi-lines lidar, monocular camera, and inertial measurement unit(IMU) in the odometry. Second, the factor graph is constructed using lidar and visual odometry as the unary edge and binary edge constraints, respectively, with the motion determined by IMU odometry serving as the primary odometry in the system. Finally, different increments of IMU odometry, lidar odometry and visual odometry are computed as favor factors to realize the adaptive adjustment of the factor's weight. The suggested method has greater location accuracy and a better mapping effect in complex situations when compared to previous algorithms.

1. Introduction

Because of the progress of robotics, mobile robots are commonly utilized in construction and logistics. A crucial technology enabling mobile robots to locate and map in uncharted regions is simultaneous localization and mapping (SLAM). With the increasing variety of sensors available for mobile robots, how to fuse different types of sensors effectively to improve the SLAM performance in challenging scenarios is an important problem [1].

Lidar-based SLAM algorithms are widely used because of their high accuracy and high reliability. However, owing to degeneracies and outliers, such as working in long and straight corridors, they struggle to give accurate motion estimates [2]. Monocular camera-based SLAM algorithms are inexpensive, but in scenes with strong light variations or few texture features like working indoors and outdoors, it is difficult to extract and match features or do optical flow [3]. An inertial measurement unit (IMU) can estimate accurate pose derivation in a short time, but it is prone to cumulating drift errors. Single-sensor SLAM algorithms may suffer from odometry degradation in challenging scenarios, affecting localization and mapping accuracy. Therefore, multi-sensor SLAM is an important trend.

According to the optimization, the multi-sensor fusion SLAM algorithms can split into two kind of groups, tightly coupled methods and loosely coupled methods [4]. Because of their simplicity and extensibility, loosely linked optimization techniques are favoured because they estimate parameters individually before fusing them at the data level, feature level, or decision level [5]. While tightly coupled optimization methods estimate the observed landmarks and the robot states at the same time, which is proved to be advantageous for more accuracy but requires more computing resources.

Motivated by the discussion above, a factor graph optimization-based multi-sensor SLAM approach that can adjust the graph factor's weight adaptively is proposed in this paper. The method uses the motion calculated by IMU odometry as the main odometry, the other two odometries as the constraints of the graph, and the constraints are weight adaptively adjustable as the favourite factor. The method is experimentally verified to have good localization accuracy compared with other algorithms.

2. Overview of the system

A multi-sensor SLAM system is proposed in this article, which fuses multi-lines lidar, monocular camera, and IMU as odometry, and estimates robot state by factor graph optimization. The tightly coupled optimization method can adaptively adjust the factors' weight by favor factor to estimate the robot's pose. According to the type of sensor, the system can be divided into three modules: IMU odometry (IO) is the main odometry, lidar-inertial odometry (LIO) still with visual-inertial odometry (VIO) are the adding odometry, as shown in Figure 1.

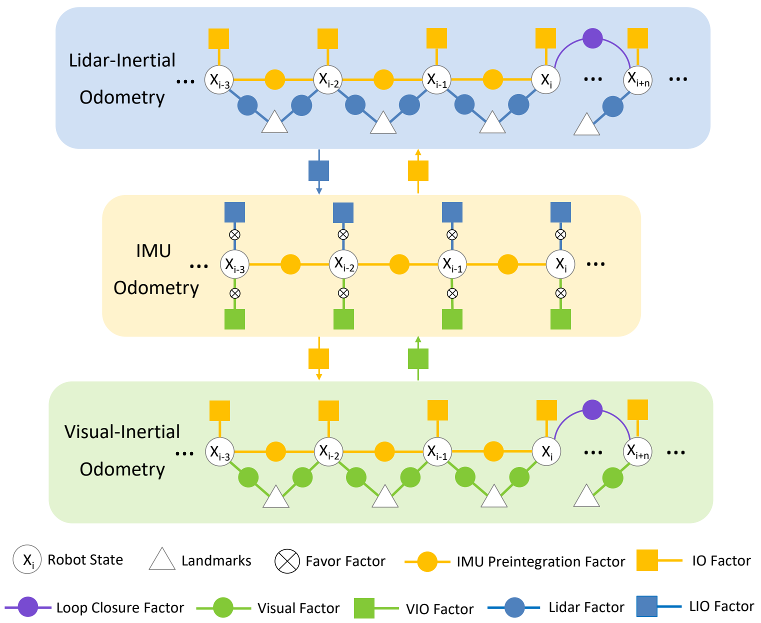

Figure 1. Factor graph of the system.

If the bias drift can be well limited by other sensors, the IMU delivers smooth measurements like position, speed, and posture with noise, which can make its estimate highly accurate. IO is therefore intended to be the main module. When subjected to other constraints, IO can estimate accurate robot pose, while attending pre-integration constraints during the next two keyframes. The tightly coupled optimization is based on GTSAM, using factor graphs and Bayes networks. The robot's movement state is set as vertex and estimates obtained from VIO and LIO are set as respectively. When LIO and VIO factors are included into the graph optimise, the favor factors are calculated to adjust the factors' weight.

Firstly, we define notation throughout the article. World frame is denoted as W, IMU frame is denoted as I, lidar frame is denoted as L , and camera frame is denoted as C. The IMU frame is recorded as the robot based frame. Therefore, the objective is to calculate the IMU frame's position in relation to the world frame. The robot state $\boldsymbol{x}_k$ can be expressed as:

$$\boldsymbol{x}_k = [\boldsymbol{p}_k^W, \boldsymbol{v}_k^W, \boldsymbol{q}_k^W, \boldsymbol{b}_a, \boldsymbol{b}_g], k \in [0, n] \tag{1}$$

where the k means robot state estimation at the keyframe k. $\boldsymbol{p}_k^W, \boldsymbol{v}_k^W, \boldsymbol{q}_k^W$ are respectively the position, velocity, and quaternion of the robot. $\boldsymbol{b}_a$, $\boldsymbol{b}_g$ are respectively the IMU accelerometer and gyroscope biases.

3. Methodology

A factor graph is used in tightly coupled optimization. We primarily introduce four categories of factors here: (a) IO factor; (b) LIO factor; (c) VIO factor; (d) Favor factor

3.1. IO factor

The structure of the IO factor is shown in Figure 1. The projected state is shown in the factor graph of IO contains $\mathbf{p}_{b_k}^w$, $\mathbf{v}_{b_k}^w$, $\mathbf{q}_{b_k}^w$, $\mathbf{b}_a$, $\mathbf{b}_g$, which meaning is described in Section 2. The two adjacent states are connected by IMU preintegration factor, while the LIO factor and VIO factor add by the connection of the favor factor as a constraint. The keyframes decide how densely the nodes are arranged. IMU's kinematic model can be described as follows:

$$\hat{\boldsymbol{\omega}}_t = \boldsymbol{\omega}_t + \boldsymbol{b}_g + \boldsymbol{\eta}_g \tag{2}$$

$$\hat{\boldsymbol{a}}_t = \boldsymbol{R}_t^{IW}(\boldsymbol{a}_t - \boldsymbol{g}) + \boldsymbol{b}_a + \boldsymbol{\eta}_a \tag{3}$$

where subscript t means measurement at time t, $\boldsymbol{\omega}_t$, $\boldsymbol{a}_t$ are respectively the angular velocity and acceleration, $\boldsymbol{\eta}_g$, $\boldsymbol{\eta}_a$ are respectively the noise of accelerometer and gyroscope, $\boldsymbol{R}^{IW}$ represents rotation from world frame to the IMU frame, $\boldsymbol{g}$ means gravity.

According to the pose at last frame i, the state of the frame j can be integrated by measurement during this period, the relative motion error in the IMU frame is as follows:

$$\begin{aligned}\boldsymbol{e}_{ij}^I &= \boldsymbol{z}_{ij}^I - h_{ij}^I(\boldsymbol{x}_i, \boldsymbol{x}_j)\\ &= \begin{bmatrix}\hat{\boldsymbol{p}}_j\\ \hat{\boldsymbol{v}}_j^i\\ \hat{\boldsymbol{q}}_j^i\\ \boldsymbol{b}_{aj}\\ \boldsymbol{b}_{gj}\end{bmatrix} \ominus \begin{bmatrix}\boldsymbol{R}_w^i\left(\boldsymbol{p}_j^W - \boldsymbol{p}_i^W + \dfrac{1}{2}\boldsymbol{g}\Delta t^2 - \boldsymbol{v}_i^W \Delta t\right)\\ \boldsymbol{R}_W^i(\boldsymbol{v}_j^W + \boldsymbol{g}\Delta t - \boldsymbol{v}_i^W)\\ \boldsymbol{q}_i^{W^{-1}} \otimes \boldsymbol{q}_j^W\\ \boldsymbol{b}_{ai}\\ \boldsymbol{b}_{gi}\end{bmatrix}\end{aligned} \tag{4}$$

where rotation is represented by $\boldsymbol{R}$ and quaternions $\boldsymbol{q}$. $\ominus$ is the IMU residuals' minus operation. Take note of the graph's joint optimization of the bias errors for the accelerometer and gyroscope.

3.2. LIO factor

The structure of the LIO factor is shown in Figure 1. There are four types of constraints in LIO: IO constraint, IMU pre-integration constraint, lidar odometry constraint, and closed-loop constraint. Four kinds of constraints are jointly optimized in factor graph optimization [6]. Lidar odometry is produced by scanning and matching point clouds, or by comparing the most recent keyframe point cloud data with the neighbourhood map following feature extraction. To lessen the quantity of data handled, keyframes are chosen. A new state node is added to the factor graph and keyframe information is created when the posture change rises over a certain threshold.

3.3. VIO factor

The structure of the VIO factor is shown in Figure 1. There are four types of constraints in VIO: IO constraint, IMU pre-integration constraint, visual odometry constraint, and closed-loop constraint. Four kinds of constraints are jointly optimized in factor graph optimization. Before optimization, the system needs to judge if visual degradation occurs.

3.4. Favor factor

In the process of factor graph optimization, the weight reflects the confidence of nodes and the strength of constraints. The degenerated data affect the state estimation of the whole factor graph. Therefore, adding favor factors to adjust the factor's weight is necessary.

Due to a short period, the pose state variation between the adjacent keyframes is not large, and the IMU odometry is robust to changes in the external environment and only focuses on its pose changes. It can be assumed that the robot state estimate obtained by the IMU odometry at this time is a high-precision value, which is used as a reference value to calculate the weighting factors of other sensors. When the lidar odometry or visual odometry is degraded, the degraded factor weight is set to 0. In this case, the factor will not participate in the graph optimization. When odometries are not degraded, there is still a difference in the pose estimation. The favorite factor $\boldsymbol{f}_k^L$, $\boldsymbol{f}_k^V$ are designed to adjust the factor weight of the LIO and VIO at the keyframe k adaptively. The favor factor algorithm is designed as follows:

Algorithm 1 Calculation of Favor Factor

Input: $\boldsymbol{p}_{k-1}^L, \boldsymbol{p}_k^L, \boldsymbol{p}_{k-1}^V, \boldsymbol{p}_k^V, \boldsymbol{p}_{k-1}^I, \boldsymbol{p}_k^I$.
Output: $\boldsymbol{f}_{k-1,k}^L, \boldsymbol{f}_{k-1,k}^V$.

```
1: initialize: Set f^L_{k-1,k} = 1, f^V_{k-1,k} = 1;
2: for k = 1, 2, ..., n do
3:     Compute p^L_{k-1,k}, p^V_{k-1,k}, p^I_{k-1,k} from key frame k-1 to k
4:     Compute f^L_{k-1,k} = ||p^L_{k-1,k}|| / ||p^I_{k-1,k}||, f^V_{k-1,k} = ||p^V_{k-1,k}|| / ||p^I_{k-1,k}||;
5:     if f^L_{k-1,k} > 1 then
6:         Compute f^L_{k-1,k} = 1/f^L_{k-1,k};
7:     end if
8:     if f^V_{k-1,k} > 1 then
9:         Compute f^V_{k-1,k} = 1/f^V_{k-1,k};
10:    end if
11: end for
```

4. Experiment

In this section, an experiment is designed to verify the absolute pose error (APE/meter) of our algorithm compared with A-LOAM [7], VINS-MONO [8], and LIO-SAM [9] which are famous lidar-based, visual-based, and lidar-initial based SLAM algorithms [10]. To quantify the location and mapping performance, the root means square error (RMSE/meter) and mean absolute error (MAE/meter) are determined. The simulation is implemented on Ubuntu 18.04. The hardware device is a computer equipped with Intel (R) Core (TM) i7-10700 CPU @ 2.90 GHz, RAM 16 G.

4.1. Dataset

Based on the M2DGR [11] dataset, door_01, and door_02 scenarios are selected to test the performance. The scenario is collected by a ground robot with a Velodyne VLP-32C lidar, a Realsense D435i visual-initial sensor, and other sensors. At door_01, the robot moves from a park out of the building into the hall. At door_01, the robot moves from the park out of the building into the hall and returns to the park.

4.2. Experiment on door_01 and door_02

The Red line represents the trajectories computed by our algorithms and other colors represent the result of other algorithms. The left figure displays the entire trajectory, while the right figure displays the specifics of the corner in the red dotted line circle. The APE of different SLAM methods is evaluated in Table 1.

Table 1. Absolute pose error of SLAM algorithms on door_01, door_02.

SLAM Algorithms	Door_01		Door_02	
	RMSE(m)	MAE(m)	RMSE(m)	MAE(m)
A-LOAM	0.2496	0.2296	0.3081	0.2801
VINS-MONO	1.2519	1.0973	0.6764	0.5759
LIO-SAM	0.2486	0.3780	1.0169	0.8038
Ours	**0.2437**	**0.2225**	**0.2296**	**0.2178**

On the door_01 scenario, the trajectory is shown in Figure 2. The RMSE and MAE of our algorithm are 0.2437 meters and 0.2225 meters, which are both the least APE compared with other algorithms. As shown in the right figure of Figure 2, our method has the nearest trajectory with door_01_groundtruth at the corner.

On the door_02 scenario, the trajectory is shown in Figure 3. The RMSE and MAE of our algorithm are 0.2296 meters and 0.2178 meters, which are both the least APE compared with other algorithms. As shown in the right figure of Figure 3, our method has the nearest trajectory with door_01_groundtruth at the corner.

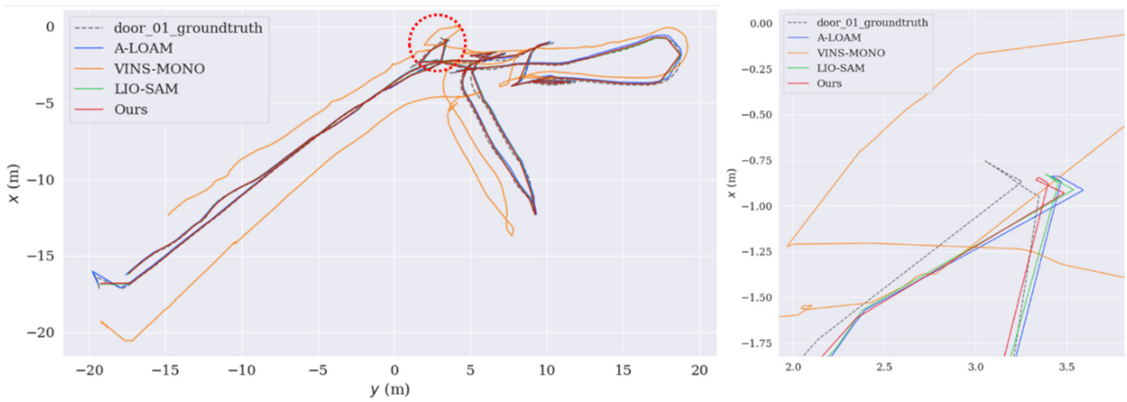

Figure 2. The trajectories of different algorithms in door_01. The left figure displays the entire trajectory, while the right figure displays the specifics of the trajectories at the corner in the red dotted line circle.

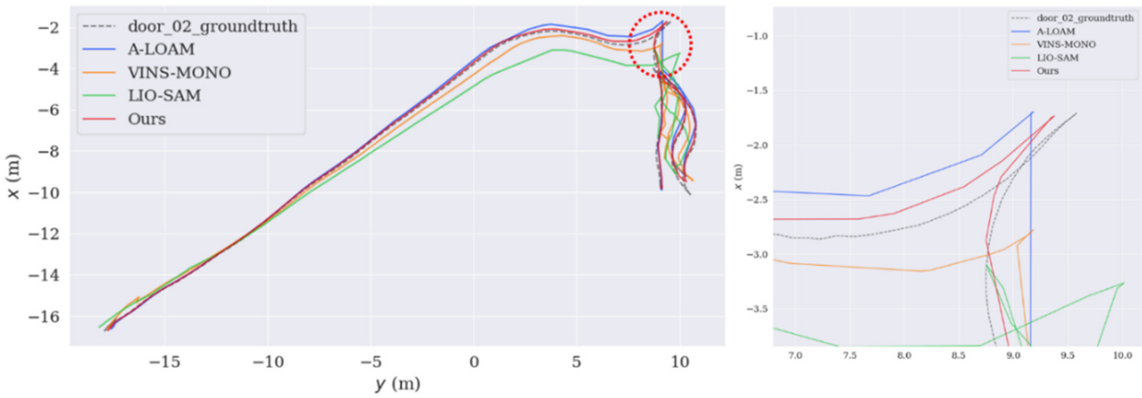

Figure 3. The trajectories of different algorithms in door_02. The left figure displays the entire trajectory, while the right figure displays the specifics of the trajectories at the corner in the red dotted line circle.

5. Conclusion

A factor graph optimization-based multi-sensor SLAM approach that can adjust the graph factor's weight adaptively is proposed to figure out the challenging environments. The algorithm fuses fuses multi-lines lidar, monocular camera, and IMU in the system and makes lidar odometry and visual odometry the constraints to create the factor graph. The increments of lidar odometry, visual odometry, and IMU odometry are used as the favor factor, the constraint weight adjustment coefficients, to adjust the factor's weight adaptively. Our method is tested in the M2DGR dataset's door_01 and door_02 scenarios and has the least APE compared with other famous proposed algorithms.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (U1813222), National Natural Science Foundation of China (U20A20283), Shenzhen International Cooperation Research Program (GJHZ20200731095009029), Department of Science and Technology of Guangdong Province (Grant No.2019TQ05Z654), CAS Science and Technology Service Network Plan (STS) - Dongguan Special Project (Grant No. 20211600200062).

References

[1] Shan, T. Englot, B. Ratti, C. Rus, D. 2021. LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Xian, China. 5692-5698.

[2] Xu, X. Zhang, L. Yang, J. Cao, C. Wang, W. Ran, Y. Tan, Z. Luo, M. (2022) A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR. Remote Sensing, Vol. 14 Issue 12: 27.

[3] Lai, T. (2022) A Review on Visual-SLAM: Advancements from Geometric Modelling to Learning-Based Semantic Scene Understanding Using Multi-Modal Sensor Fusion. Sensors, Vol. 22 Issue 19: 7265.

[4] Chiang, K. Chang, H. Li, Y. Tsai, G. Tseng, C. Tien, Y. Hsu, P. (2019) Assessment for INS/GNSS/odometry/barometer integration in a loosely-coupled and tightly-coupled scheme in a GNSS-degraded environment. IEEE Sensors Journal, Vol. 20 Issue 6: 3057-3069.

[5] Zhou, X. Zhou, H. She, D. Li, S. He, F. Ling, H. 2021. Dynamic Indoor Localization Algorithm Based on Compressed Sensing and Inertial Navigation. In: 2021 International Conference on Computer, Internet of Things and Control Engineering (CITCE). Guangzhou, China.49-59.

[6] Yu, J. Yu, C. Lin, C. Wei, F. 2021. Improved Iterative Closest Point (ICP) Point Cloud Registration Algorithm based on Matching Point Pair Quadratic Filtering. In: 2021

International Conference on Computer, Internet of Things and Control Engineering (CITCE). Guangzhou, China. 1-5.
[7] Zhang, J. Singh, S. 2014. LOAM: Lidar Odometry and Mapping in Real-time. In: Robotics: Science and Systems Conference. Berkeley, CA, US.
[8] Qin, T. Li, P. Shen, S. (2018) VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. IEEE Transactions on Robotics, Vol. 34 Issue 4: 1004-1020.
[9] Shan, T.Englot, B. Meyers, D. Wang, W. Ratti, C. Rus, D. 2020. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In: 2020IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS). Las Vegas, US. 5135-5142.
[10] Zhao, S. Zhang, H. Wang, L. Nogueira. Scherer, S. 2021. Super odometry: IMU-centric LiDAR-visual-inertial estimator for challenging environments. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Prague, Czech Republic. 8729-8736.
[11] Yin, J. Li, A. Li, T. Yu, W. Zou, D. (2022) M2DGR: A Multi-Sensor and Multi-Scenario SLAM Dataset for Ground Robots. IEEE Robotics and Automation Letters, Vol. 7 Issue 2: 2266-2273.