2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Tightly Coupled Vector Field Inertial Localization

Naoki Akai[1,2]

***Abstract*— This paper presents a tightly coupled Vector Field Inertial Localization (VFIL) method that integrates the capabilities of Vector Field Sensors (VFSs) and an Inertial Measurement Unit (IMU). While vector fields, such as magnetic or Wi-Fi signal intensities, provide vector data at each point, leveraging VFSs for accurate localization is challenging due to their limited measurement range. Within the VFIL framework, the IMU predicts the pose, and the VFS compensates any prediction errors. To enhance constraints with the VFS, it is essential to store vector data based on the predicted trajectory. While this trajectory can be deduced from accumulated IMU readings, significant error is often included in these accumulations. Our solution is to register the vector data to a vector field map and simultaneously correct the accumulation error. We achieve this by adopting a factor-graph-based optimization method that concurrently estimates pose, velocity, and biases in both IMU and VFS measurements. To demonstrate effectiveness of VFIL, we conduct simulations and real-world experiments, comparing it against particle-filtering and pose-graph-based optimization methods. Results reveal that VFIL consistently offers superior pose estimation accuracy compared to the compared methods owing to the tightly coupled estimation.**

I. Introduction

In this paper, we present a localization and/or trajectory estimation method using an Inertial Measurement Unit (IMU) and a Vector Field Sensor (VFS), where a VFS measures vector data of a point where it locates, e.g., magnetic intensity and Wi-Fi signal strength sensors. Note that the naming of VFS was inspired by these previous works [1], [2]. Performing accurate localization with a VFS is not easy owing to its limited measurement range. To make better constraints with a VFS, vector data should be stored according to sensor trajectory. The sensor trajectory can be obtained by accumulating IMU measurements; however, large error is often included in the accumulation. The stored vector data should be registered to a vector field map while compensating the accumulation error. To this end, we present a factor-graph-based optimization method [3] to perform tightly coupled Vector Field Inertial Localization (VFIL). In VFIL, sensor pose (or trajectory) and velocity and biases in both IMU and VFS measurements are simultaneously estimated.

The factor graph is applied to many robotic applications in recent and its superior performance has been shown. For example, the factor graph is applied to SLAM [4], [5], legged robot state estimation [6], and GNSS-based trajectory estimation [7]. In recent works, the factor graph is also applied for localization with VFSs. For example, Yang *et al*. presented a localization method with Wi-Fi and an ultra wide band sensor [8]. Li *et al*. presented a factor-graph-based localization method with the local magnetic field factor [9]. The local magnetic field factor is very similar to the factor used in our proposal; however, we extend it to more general factor that can treat vector data and its measurement biases.

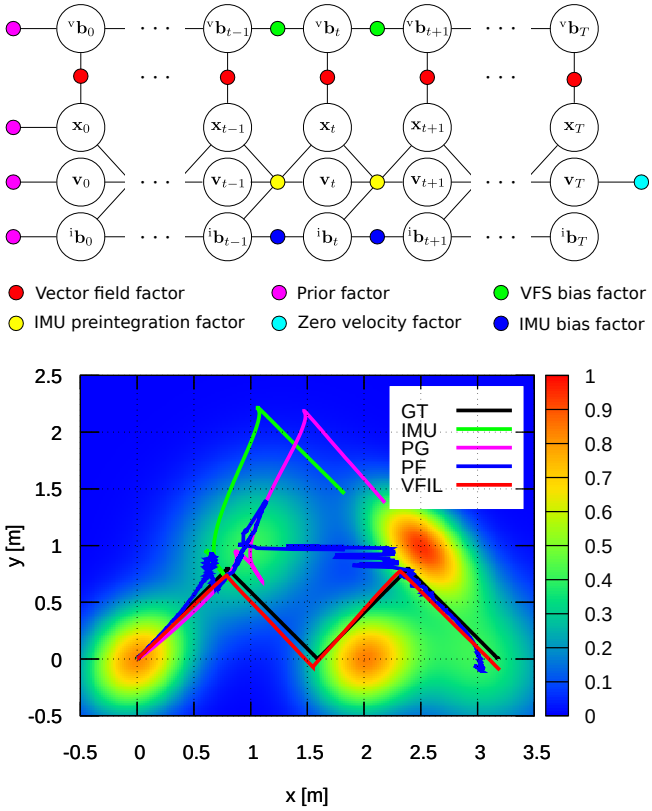

Fig. 1. Top: The factor graph of the presented method. Pose $\mathbf{x}$, velocity $\mathbf{v}$, and IMU and VFS measurement biases, ${}^{\mathrm{i}}\mathbf{b}$ and ${}^{\mathrm{v}}\mathbf{b}$, are simultaneously estimated. Bottom: The localization results by the pose graph optimization (PO), particle filtering (PF), and presented method (VFIL) in the simulation. The green line represents an accumulation result of IMU measurements. VFIL outperforms the comparison methods in terms of the estimation accuracy in all the experiments.

The top of Fig. 1 illustrates a factor graph of the presented method. In the factor graph, pose and velocity of a VFS, $\mathbf{x}$ and $\mathbf{v}$, and biases in IMU and VFS measurements, ${}^{\mathrm{i}}\mathbf{b}$ and ${}^{\mathrm{v}}\mathbf{b}$, are introduced as hidden variables. The pose, velocity, and IMU measurement biases are connected via an IMU preintegration factor [10]. A vector field factor that provides a constraint between the sensor pose and the vector field map while considering VFS measurement biases. We also assume that both biases in IMU and VFS measurements do not drastically change and its assumption is implemented using the bias factors assigned between the continuous biases. In addition, a zero velocity factor made from zero velocity

[1]Naoki Akai is with the Institute of Innovation for Future Society, Nagoya University, Nagoya 464-8603, Japan akai@nagoya-u.jp
[2]Naoki Akai is with the LOCT Co., Ltd., Nagoya 464-0805

detection methods such as presented [11] is optionally used. The zero velocity factor is effective to compensate the IMU accumulation error, and the easy introduce of the factor is an advantage of the use of the factor graph.

The bottom of Fig. 1 shows an example of trajectory estimation results by three methods including the presented method. In this simulation, scalar value measurements like magnetic intensity norm were imitated. The black and green lines indicate the ground truth and the accumulated trajectory of the IMU measurements. In the IMU measurements, time variant biases and white noises were added, resulting in the IMU trajectory was largely drifted. Note that similar biases and noises were added to the VFS measurements. We applied two methods based on the pose graph optimization (PG) and particle filtering (PF) as comparison methods; however, these methods could not estimate the trajectory well. PG can handle the continuous connectivity between nodes, i.e., poses, and the smooth trajectory could be estimated. However, it could not compensate the accumulation error correctly because the VFS measurements including the biases did not contain enough information to make better constraints. The estimate by PF was quite inaccurate because large noises were added in the particle update process to cope with the inaccurate IMU measurements. The presented method depicted with the red line overcame these issues and achieved accurate and smooth trajectory estimation. The contribution of this paper is presenting the first concept of the VFIL framework.

The rest of this paper is organized as follow. Section II summarizes related works. Section III describes overview and implementation details of the presented method. Section IV shows experimental results using simulation and actual data. Section V concludes this work.

II. Related work

To perform localization accurately, multiple constrains should be obtained at the same time. For example, at least three constrains are necessary for 2D localization. LiDAR and/or camera is able to obtain multiple constrains; however, VFSs cannot obtain multiple constrains in one measure. Hence, ICP [12] that is a popular localization approach cannot be simply applied to VFS-based localization.

Some authors proposed magnetic-sensor-based localization methods. Haverinen and Kemppainen proposed a line-magnetic-map-based localization method [13]. In the method, magnetic sensor measurements are stored according to the travel distance and sensor position is estimated on the line by MCL [14]. It should be noted that MCL, or other Bayesian filtering methods such as Kalman filter, can be simply applied to VFS-based localization because previous possibility regarding the existence can be retained. Rahok *et al*. also proposed a similar localization method and applied it to mobile robot navigation [15]. In the method, multiple line magnetic maps are built and longitudinal and lateral errors are corrected based on magnetic fluctuation patterns. We extended this method to incorporate with the LiDAR-based localization and achieved autonomous navigation in real world [16]. However, building multiple line maps is significant time consuming and the line-map-based methods have limitation in terms of localization accuracy since it cannot provide enough constraints for 2D localization.

Frassl *et al*. presented a 2D-magnetic-map-based localization method based on MCL [17]. To this end, a dense magnetic map is required and they built it using the motion capture system that accurately tracks the robot having a magnetic sensor. In our previous work, we also presented the similar localization method based on MCL and built the dense magnetic map using Gaussian processes regression [18]. The use of Gaussian processes was inspired by the work of Ferris *et al*. [19]. A similar approach to our previous proposal with Wi-Fi signal strength is presented in [20]. These works indeed showed that MCL can achieve localization; however, MCL does not work in some cases, for example, INS accuracy is low. In particular, MCL cannot consider entire time sequence since it assumes the Markov property.

The factor graph that is a probabilistic modeling method is widely used in the recent robotics field [3]. Modeling with the factor graph is similar to Bayesian-network-based modeling, but it can cope with bidirectional relationships. For example, factor-graph-based SLAM is able to estimate IMU measurement biases while performing localization and mapping [4], [5]. In addition, a factor-graph-based trajectory estimation method with GNSS that is able to treat several biases simultaneously is also presented [7]. Modeling with the factor graph enables us to handle multiple constraints against different states concurrently and it is shown that factor-graph-based perception increases accuracy and robustness. It should be noted that the pose graph optimization [21] can be formulated using the factor graph, but the factor graph is able to handle more complex relationships.

The factor graph is also used to incorporate magnetic, Wi-Fi, and other types of VFS sensor measurements. Sun *et al*. presented an attitude estimation method using a magnetometer [22]. Guo *et al*. presented a pedestrian dead reckoning method with Wi-Fi round-trip time and received signal strength [23]. Yang *et al*. presented a localization method combining Wi-Fi and an ultra wide band sensor [8]. These works utilized the factor graph for tight coupling of the sensors and demonstrated that coupling based on the factor graph can enhance estimation accuracy. Li *et al*. presented a similar method to our proposal [9] and the local magnetic factor is introduced to compensate the pedestrian dead reckoning error. However, they do not consider bias of the VFS measurement. In addition, our proposal is more general extension of the method because we treat general vector data.

III. Proposed method

A. Problem setting

The factor graph used in our work is shown in the top of Fig. 1. In the method, we try to simultaneously estimate a VFS pose $\mathbf{x}_t$ and velocity $\mathbf{v}_t$ and biases in both IMU and VFS measurements, where acceleration, angular velocity, and

VFS sensor biases at time t are denoted as ${}^{\mathrm{a}}\mathbf{b}_t$, ${}^{\mathrm{g}}\mathbf{b}_t$, and ${}^{\mathrm{v}}\mathbf{b}_t$, and the IMU biases are aggregated into ${}^{\mathrm{i}}\mathbf{b}_t^\top = ({}^{\mathrm{a}}\mathbf{b}_t^\top, {}^{\mathrm{g}}\mathbf{b}_t^\top)$. $\mathbf{x}_t$ is composed of a rotation matrix R_t and a translation vector $\mathbf{t}_t$ and $\mathbf{v}_t$ is a translation velocities according to a map frame. A VFS measures a vector data corresponding to each pose and vector data corresponding to $\mathbf{x}_t$ is denoted as $\mathbf{z}_t$. IMU measures linear accelerations and angular velocities and these measurements at time t are denoted $\mathbf{a}_t$ and $\boldsymbol{\omega}_t$.

To perform estimate by the presented method, a vector field map that stores vector data of a target environment is needed to be built in advance. In this work, we will conduct simulation and real experiments. The vector map can be easily prepared in the simulation cases; however, building a dense vector filed map is time consuming in the real environment case. In the real environment case, we adopt the mapping method presented in [18] and use the result as the map.

In this work, we implemented the presented method using GTSAM[1]. In below, we describe how to design the factors assigned to the graph.

B. Vector filed factor

A unary factor corresponding to a VFS measurement likelihood with a Gaussian noise model is denoted as

$$f_t(\mathbf{x}_t, {}^{\mathrm{v}}\mathbf{b}_t; \mathbf{z}_t) = \exp\left(-\frac{1}{2}\|h(\mathbf{x}_t) - {}^{\mathrm{v}}\mathbf{b}_t - \mathbf{z}_t\|^2_\Sigma\right), \quad (1)$$

where $h(\mathbf{x}_t)$ returns a vector data of the vector field map corresponding to $\mathbf{x}_t$ and Σ is a covariance matrix. To create the unary factor in GTSAM, the Jacobian of the measurement model should be determined. The Jacobian can be easily determined from the vector field map if the dense map is available since it can be determined by only considering differences between adjacent cell values. However, the Jacobian may not be static during the optimization because the state will be moved. Hence, the Jacobian should be modified according to the optimization result. We interpolate values of the vector field map using the cubic polynomial based on surrounding measured values and determine the Jacobian based on the interpolation result. The measurement model $h(\cdot)$ is also implemented using the cubic polynomial.

C. IMU preintegration factor

To make the IMU factor, we use the IMU preintegration technique [10] that efficiently incorporates IMU constraints into the factor graph. The update process of the IMU sensor pose can be denoted as:

$$R_{t+\Delta t} = R_t \exp\left((\boldsymbol{\omega}_t - {}^{\mathrm{g}}\mathbf{b}_t - {}^{\mathrm{g}}\boldsymbol{\eta}_t)\Delta t\right) \quad (2)$$

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \mathbf{g}\Delta t + R_t(\mathbf{a}_t - {}^{\mathrm{a}}\mathbf{b}_t - {}^{\mathrm{a}}\boldsymbol{\eta}_t)\Delta t \quad (3)$$

$$\mathbf{t}_{t+\Delta t} = \mathbf{t}_t + \mathbf{v}_t\Delta t + \frac{1}{2}\left(\mathbf{g} + R_t(\mathbf{a}_t - {}^{\mathrm{a}}\mathbf{b}_t - {}^{\mathrm{a}}\boldsymbol{\eta}_t)\right)\Delta t^2 \quad (4)$$

where $\exp(\cdot)$ is the exponential map, $\mathbf{g}$ is the gravity vector, and ${}^{\mathrm{a}}\boldsymbol{\eta}_t$ and ${}^{\mathrm{g}}\boldsymbol{\eta}_t$ are white noises in the IMU measurements.

[1] https://github.com/borglab/gtsam

From the IMU preintegration results, the following costs are determined between continuous poses:

$$\Delta R_{ij} = R_i^\top R_j \exp(\delta\boldsymbol{\phi}_{ij}), \quad (5)$$

$$\Delta\mathbf{v}_{ij} = R_i^\top(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) + \delta\mathbf{v}_{ij}, \quad (6)$$

$$\Delta\mathbf{t}_{ij} = R_i^\top(\mathbf{t}_j - \mathbf{t}_i - \mathbf{v}_i\Delta t_{ij} - \frac{1}{2}\mathbf{g}\Delta t_{ij}^2) + \delta\mathbf{t}_{ij}, \quad (7)$$

where $\delta\boldsymbol{\phi}_{ij}$, $\delta\mathbf{v}_{ij}$, and $\delta\mathbf{t}_{ij}$ are white noise in the preintegration process.

D. Prior factor

If we assume that the initial state regarding the pose and velocity is available, the prior factors can be assigned to them. The prior of the biases can also be assigned to the bias variables. In implementation, we assign the zero bias factors to the bias variables because we cannot estimate the initial bias; however, their variances were set to large more than other priors.

E. Zero velocity factor

We apply a zero velocity detection method such as presented in [11] to compensate the trajectory obtained by the IMU measurement accumulation. If a zero velocity moment is detected, a zero velocity factor is assigned to a corresponding velocity variable as a prior factor. It should be noted that the zero velocity factor is optional but is effective for the IMU error compensation.

F. Bias factor

We assume that bias values are not changed rapidly. Hence, we assign constant bias factors to the time continuous bias variables.

IV. Experiments

We conducted simulated- and actual-sensor-based experiments. It should be noted that we focused on the 2D localization and/or trajectory estimation problem in this work. This section details the experimental setups, comparison methods, and results.

A. Comparison methods

In the experiments, we compare the presented method with the particle-filtering- and pose-graph-optimization-based methods. This subsection describes the comparison methods briefly.

1) Particle filter: In the Particle Filter (PF), a state variable is denoted as $\mathbf{x} = (R, \mathbf{t}, \mathbf{v})$ and the IMU measurements are denoted as a_x, a_y, and ω_z, where a_x and a_y are linear accelerations and ω_z is yaw rate. The particle state is updated with noise added IMU measurements

$$a'_x \sim \mathcal{N}(a_x, \alpha_1 a_x^2 + \alpha_2 a_y^2 + \alpha_3\omega_z^2), \quad (8)$$

$$a'_y \sim \mathcal{N}(a_y, \alpha_4 a_x^2 + \alpha_5 a_y^2 + \alpha_6\omega_z^2), \quad (9)$$

$$\omega'_z \sim \mathcal{N}(\omega_z, \alpha_7 a_x^2 + \alpha_8 a_y^2 + \alpha_9\omega_z^2), \quad (10)$$

where $\alpha_1 - \alpha_9$ are arbitrary positive constants and are determined experimentally.

After updating the states, likelihood is calculated as

$$\zeta \frac{1}{(\sqrt{2\pi})^m \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(h(\mathbf{x})-\mathbf{z})^\top {}^{\mathrm{v}}\Sigma^{-1}(h(\mathbf{x})-\mathbf{z})\right) + (1-\zeta)\mathrm{unif}(\mathbf{z}), \tag{11}$$

where ζ is an arbitrary positive constant including from 0 to 1, m is size of vector data, ${}^{\mathrm{v}}\Sigma$ is a covariance of a measurement, and $\mathrm{unif}(\mathbf{z})$ is a uniform distribution defined on a measurement space.

The bias variables cannot be treated in PF because its implementation is based on the Bayesian network. In addition, whole time sequence data cannot be treated simultaneously because PF assumes the Markov property.

2) Pose graph optimization: In the Pose Graph (PG) optimization, the following cost function is minimized.

$$\sum_{ij\in\mathcal{E}} \left(\mathbf{x}_j - f(\mathbf{x}_i, U_{ij})\right)^\top {}^{\mathrm{i}}\Sigma^{-1} \left(\mathbf{x}_j - f(\mathbf{x}_i, U_{ij})\right) + \sum_{i\in\mathcal{V}} \left(h(\mathbf{x}_i)-\mathbf{z}_i\right)^\top {}^{\mathrm{v}}\Sigma^{-1} \left(h(\mathbf{x}_i)-\mathbf{z}_i\right), \tag{12}$$

where U_{ij} is a set of IMU measurements between i and j time steps, f is a motion model regarding IMU, i.e., IMU preintegration process, ${}^{\mathrm{i}}\Sigma$ is a covariance of IMU motion, and $\mathcal{E}$ and $\mathcal{V}$ are sets of edge and node. It should be noted that 2D poses are only used in PG, i.e., $\mathbf{x}$ is denoted as (x, y, θ), where x and y are 2D position and θ is a heading angle. The PG optimization is performed at every time when a new node is added to the graph.

B. Simulation experiments

1) IMU simulation: We first determined the time variant linear and angular velocities of IMU in the map coordinates. Then, we rotate the linear velocities to the IMU body coordinates and simulate its acceleration. White noise and time variant biases are added to the generated accelerations and angular velocities to simulate the IMU measurements. The IMU preintegration was performed using the noise added measurements.

2) VFS simulation: In this work, we conducted two types of simulation experiments that assume the use of magnetic and Wi-Fi intensity sensors as a VFS. In the magnetic-sensor-type experiment, we imitate its measurements with the Gaussian mixture model

$$z = \frac{1}{N}\sum_{i=1}^{N} \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i) + \epsilon, \tag{13}$$

where $\boldsymbol{\mu}_i \in \mathbb{R}^2$ and $\Sigma_i \in \mathbb{R}^{2\times 2}$ are mean and covariance of ith normal distribution and ϵ is a white noise with variance σ^2. It should be noted that magnetic sensors can measure three axes magnetic field intensities, but we use their norm, $z \in \mathbb{R}$, since measurement results of three decomposed intensities depend on sensor attitude. To compute the norm, the magnetic sensor calibration such as presented in [24] should be used.

TABLE I
POSE ESTIMATION ERRORS IN METERS AND DEGREES.

		PF	PG	VFIL
Mag. sim.	Pos.	0.516 (0.477)	0.905 (0.681)	0.064 (0.025)
	Ang.	4.043 (2.268)	7.076 (1.928)	0.838 (0.432)
Wi-Fi sim.	Pos.	0.058 (0.033)	0.078 (0.036)	0.044 (0.021)
	Ang.	2.992 (1.793)	14.614 (1.798)	3.321 (1.729)

In the Wi-Fi-intensity-sensor-type experiment, we imitate its measurement with the free-space path loss model

$$z_i = -20\left(\log_{10}(d_i) + \log_{10}(f) + \log_{10}\left(\frac{4\pi}{c}\right)\right) + \epsilon, \tag{14}$$

where d_i is a distance from the sensor to ith Access Point (AP), f is the frequency, and c is the luminous flux. We assume that m access points are available and a measurement by the simulated Wi-Fi intensity sensor is denoted as $\mathbf{z} \in \mathbb{R}^m$.

3) Simulation results: The bottom of Fig. 1 shows trajectory estimation results by PG (purple), PF (blue), and VFIL (red), respectively. The black and green trajectories are the ground truth and the raw IMU accumulation result. Note that we did not give a specific unit such as G to the simulated magnetic sensor measurements.

In the PF processing, the IMU measurements were used to update the particle state, and the noise values were set large to cope with inaccuracy of the IMU measurements. Consequently, the particles were distributed largely and its estimate sometimes jumped. In particular, PF could not estimate smooth trajectory since PF cannot consider entire relation of the estimated trajectory.

PG could estimate smooth trajectory; however, its estimation accuracy was quite low. PG indeed modified the accumulation error of the IMU trajectory little bit, but its modification was not enough. This is because the VFS could not provide much information to perform accurate modification without the bias estimation.

However, VFIL could achieve accurate and smooth trajectory estimation. Figure 2 shows bias estimation results in the IMU (top) and VFS (bottom) measurements. The solid and dashed lines depict the estimated and simulated biases. The bias estimation performance could not be said as better; however, the bias estimation indeed contributed to the performance improvement compared to PG. Note that the estimate bias shown in the bottom of Fig. 2 was calculated by $h(\mathbf{x}_t) - {}^{\mathrm{v}}b_t$.

Table I summarizes the positional and angular estimated errors in meters and degrees by each method. The values inside the brackets are the standard deviations. As can be seen the table, VFIL outperformed the other methods in terms of the estimation accuracy.

Figure 3 shows the estimated trajectories in the case where the Wi-Fi sensor measurements were simulated, and Table I also summarizes their estimation errors. Note that we also did not assign a specific unit to the simulated measurements. In the simulation, 5 APs were simulated and the map shows

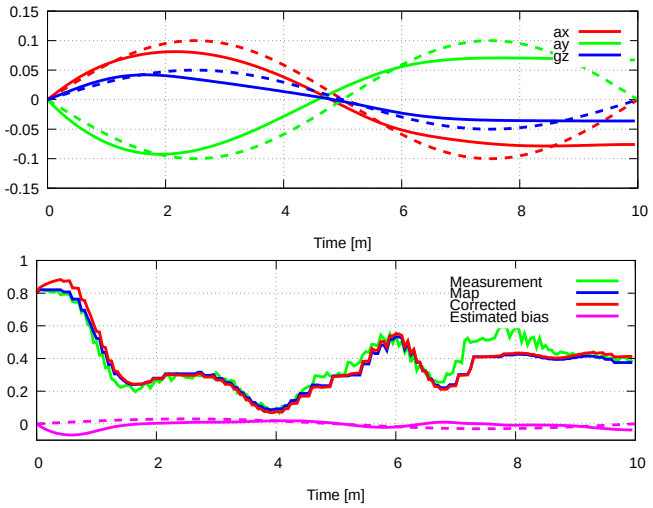

Fig. 2. The bias estimation results in the IMU (top) and VFS (bottom) sensor measurements. The solid and dashed line depict the estimated and simulated biases.

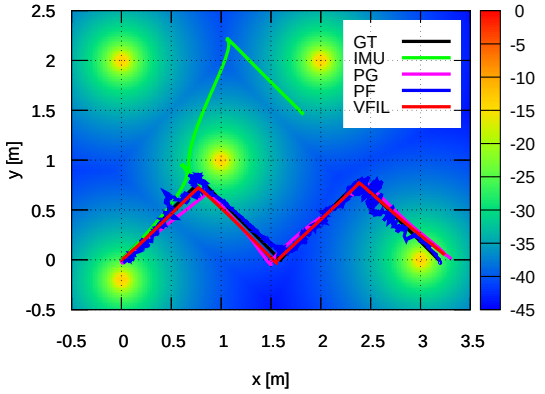

Fig. 3. The simulation results under the case where 5 APs' Wi-Fi signal intensities were simulated.

the highest Wi-Fi signal strength value at each position. All the methods achieved estimation more accurately than the results shown in Fig. 1 because the VFS provided better constrains according to increase of the measured values. The result showed that VFIL outperforms the other methods even when the performance of other methods were improved.

C. Experiment in real environment

We also conducted an experiment in an real environment. The top left of Fig. 4 shows the equipment used. We used MTi-100 as a VFS sensor and it can measure accelerations, angular velocities, and magnetic intensities. VLP16 that is a 3D LiDAR was used to build an environment map and measure ground truth trajectory of the VFS. We put the equipment on a trolley and moved it manually to collect data. gmapping[2] and als_ros[3] were used for the map building and ground truth generation.

[2]http://wiki.ros.org/gmapping
[3]https://github.com/NaokiAkai/als_ros

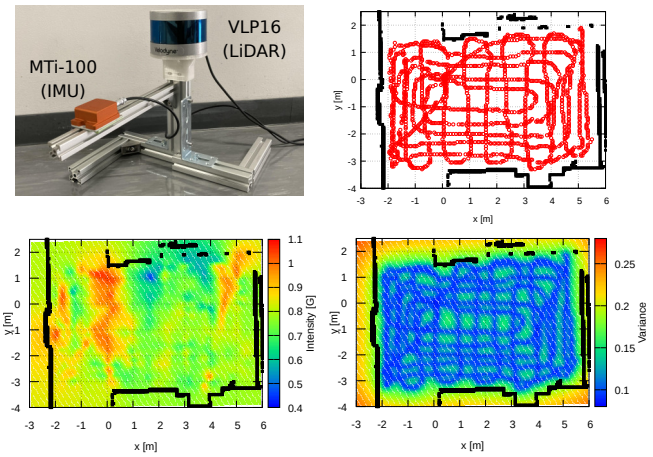

Fig. 4. The experimental setup. The top left shows the equipment. The top right shows measured points of the magnetic field with the equipment. The bottom figures indicate the magnetic intensity and its variance map regressed using Gaussian processes [18].

To build the magnetic map, we used the method presented in [18]. In short, the method first measures the magnetic field while respecting the 2D-LiDAR-based localization result and regresses the dense magnetic map using Gaussian processes. The top right of Fig. 4 shows measured points of the magnetic field. The red points are the measured points and black points are occupied points on the 2D map built using gmapping, i.e., walls. We selected an indoor environment because it typically contains much magnetic fluctuations. The bottom figures are the regressed intensity and variance maps. The variance map is used to defined the vector field factor shown in Equation (1).

Figure 5 shows the trajectory estimation results (the bottom figure is an enlarge figure of the top figure to show the results by IMU and PG). The estimate by PF was also unstable and PG cannot modify the accumulation errors as shown in the previous subsection. VFIL cannot exactly track the ground truth; however, its estimation accuracy was quite higher than other methods. We confirmed that VFIL also outperforms the other methods in the real environment experiment.

D. Limitation

The tight coupling is indeed effective to improve the estimation accuracy. However, it requires accurate estimate of the error variances of the used variables. For example, we confirmed that results by VFIL are unstable when inaccurate error variances are assigned. In particular, VFIL performance decreased even when variance estimation accuracy against only one variable is low. Hence, we need to carefully analyze the variances to ensure stability of VFIL.

In addition, sometimes bias estimation yielded inaccurate estimation results. For example, miss bias estimation caused large trajectory jump and quite bad localization estimate was obtained. This reason is also the same to inaccurate variance estimate. To achieve better estimate by VFIL, accurate and precise parameter tuning is necessary and it is sometimes

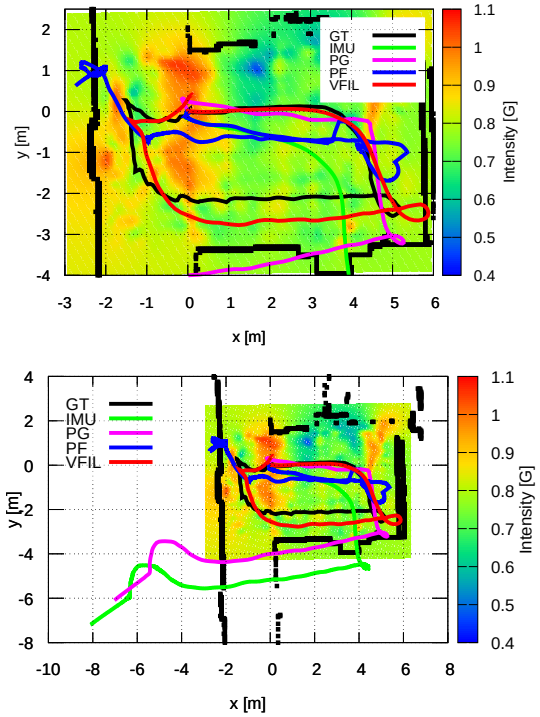

Fig. 5. The trajectory estimation results in the real environment experiment.

time consuming.

V. Conclusion

This paper has presented a tightly coupled Vector Field Inertial Localization (VFIL) method, where a vector field means a field that provides a vector data at each point, e.g., magnetic and/or Wi-Fi intensity field, and we refer sensors for measuring them to Vector Field Sensors (VFSs). VFIL is formulated as a factor-graph-based optimization framework and simultaneously estimates pose (or trajectory), velocity, and biases in IMU and VFS measurements. In addition, VFIL can consider entire time sequence owing to the effect of the factor-graph-based formulation while the Bayesian-network-based modeling cannot do. We conducted simulated- and actual-sensor-based experiments and compared VFIL with particle-filtering- and pose-graph-optimization-based methods. The experimental results showed that the simultaneous estimation and entire consideration contribute to improve the estimation accuracy and VFIL outperforms both the methods. Our future work is investigating the VFIL performance in diverse environments with different type sensors.

Acknowledgment

This work was supported by KAKENHI under Grant 23K03773.

References

[1] J.-S. Gutmann, G. Brisson, E. Eade, P. Fong, and M. Munich. Vector field SLAM. In *2010 IEEE International Conference on Robotics and Automation*, pages 236–242, 2010.

[2] J-S. Gutmann, E. Eade, P. Fong, and M.E. Munich. Vector field SLAM-localization by learning the spatial variation of continuous signals. *IEEE Transactions on Robotics*, 28(3):650–667, 2012.

[3] Frank Dellaert and Michael Kaess. *Factor Graphs for Robot Perception*. Now Publishers Inc., August 2017.

[4] H. Ye, Y. Chen, and M. Liu. Tightly coupled 3D lidar inertial odometry and mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3144–3150, 2019.

[5] K. Koide, M. Yokozuka, S. Oishi, and A. Banno. Globally consistent and tightly coupled 3d lidar inertial mapping. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5622–5628, 2022.

[6] D. Wisth, M. Camurri, and M. Fallon. Robust legged robot state estimation using factor graph optimization. *IEEE Robotics and Automation Letters*, 4(4):4507–4514, 2019.

[7] T. Suzuki. GNSS Odometry: Precise trajectory estimation based on carrier phase cycle slip estimation. *IEEE Robotics and Automation Letters*, 7(3):7319–7326, 2022.

[8] L. Yang, N. Wu, B. Li, W. Yuan, and L. Hanzo. Indoor localization based on factor graphs: A unified framework. *IEEE Internet of Things Journal*, 10(5):4353–4366, 2023.

[9] Z. Li, J. Shang, and H. Shi. Factor graph with local constraints: A magnetic field/pedestrian dead reckoning integrated navigation method based on a constrained factor graph. *Electronics*, 12(18):3832, 2023.

[10] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 33(1):1–21, 2017.

[11] I. Skog, P. Handel, J.-O. Nilsson, and J. Rantakokko. Zero-velocity detection—an algorithm evaluation. *IEEE Transactions on Biomedical Engineering*, 57(11):2657–2666, 2010.

[12] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[13] J. Haverinen and A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10):1028–1035, 2009.

[14] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1322–1328, 1999.

[15] S.A. Rahok, Y. Shikanai, , and K. Ozaki. Navigation using an environmental magnetic field for outdoor autonomous mobile robots. *Advanced Robotics*, 25(13-14):1751–1771, 2011.

[16] N. Akai, K. Inoue, and K. Ozaki. Autonomous navigation based on magnetic and geometric landmarks on environmental structure in real world. *Journal of Robotics and Mechatronics*, 26(2):158–165, 2014.

[17] M. Frassl, M. Angermann, M. Lichtenstern, P Robertson, B.J. Julian, and M. Doniec. Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 913–920, 2013.

[18] N. Akai and K. Ozaki. Gaussian processes for magnetic map-based localization in large-scale indoor environments. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4459–4464, 2015.

[19] B. Ferris, D. Hähnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In *Proceeding of robotics: science and systems*, 2006.

[20] R. Miyagusuku, A. Yamashita, and H. Asama. Improving gaussian processes based mapping of wireless signals using path loss models. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4610–4615, 2016.

[21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. pages 3607–3613, 2011.

[22] K. Sun, Q. Zeng, J. Liu, W. Qiu, and J. Shi. Modified attitude factor graph fusion method for unmanned helicopter under atmospheric disturbance. *Chinese Journal of Aeronautics*, 35(6):285–297, 2022.

[23] G. Guo, R. Chen, X. Niu, K. Yan, S. Xu, and L. Chen. Factor graph framework for smartphone indoor localization: Integrating data-driven PDR and Wi-Fi RTT/RSS ranging. *IEEE Sensors Journal*, 23(11):12346–12354, 2023.

[24] M. Angermann, M. Frassl, M. Doniec, B.J. Julian, and P. Robertson. Characterization of the indoor magnetic field for applications in localization and mapping. In *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9, 2012.