

Real-Time Graph-Based Optimization for GNSS-Doppler Integrated RTK-GNSS/IMU/DR Positioning System in Urban Area

Aoki Takanose

*Graduate School of Informatics
Nagoya University*

Furo-cho, Chikusa-ku,
Nagoya, Aichi, Japan

takanose.aoki@g.sp.m.is.nagoya-u.ac.jp

Eijiro Takeuchi

TierIV Inc.

Mei-eki 1-chome, Nakamura-ku,
Nagoya, Aichi, Japan

takeuchi@g.sp.m.is.nagoya-u.ac.jp

Alexander Carballo

*Department of Electrical,
Electronic and Computer Engineering
Gifu University*

Yanagido 1-1, Gifu, Gifu, Japan
alex@gifu-u.ac.jp

Junichi Meguro

*Department of Mechatronics Engineering
Meijo University*

Shiokamaguchi 1-501, Tenpaku-ku,
Nagoya, Aichi, Japan

meguro@meijo-u.ac.jp

Kazuya Takeda

*Graduate School of Informatics
Nagoya University*

Furo-cho, Chikusa-ku,
Nagoya, Aichi, Japan

kazuya.takeda@nagoya-u.jp

Abstract—Autonomous driving of vehicles and robots requires highly accurate position information, and RTK-GNSS is expected to be utilized for this purpose. In this paper, we propose a robust and real-time operation method by introducing graph optimization into the integrated RTK-GNSS/IMU method. The proposed method is an extension of a method using vehicle trajectories that can estimate positions with lane-level accuracy even in urban areas. The position is estimated by removing GNSS multipaths from the shape of a vehicle trajectory of several hundred meters and averaging the remaining GNSS results. This method does not take into account the errors in the vehicle trajectory and cannot fully benefit from the high accuracy positioning solution of RTK-GNSS. To solve this problem, we introduce graph optimization to the base method, which treats the error state as a probabilistic model. However, general graph optimization methods have problems with processing time and outlier elimination. The proposed method solves these problems by restricting the time series data to be optimized and using a two-step optimization structure. Evaluations show that the proposed method is effective because it satisfies the requirements for real-time operation and improves accuracy compared to conventional methods.

Index Terms—RTK-GNSS/IMU, Graph Optimization, multipath, Localization, outlier removal

I. INTRODUCTION

Research and development of automatic driving for mobility, including vehicles and mobile robots, has been active. One of the necessary elements for automatic driving is the recognition of the ego-vehicle localization [1]–[3]. High positioning accuracy is required for automatic driving [4]–[6]. In particular, taking vehicles as an example, the required positioning accuracy is about 0.3 m [6]. Various position estimation methods have been proposed to achieve the required

accuracy [7]–[10]. Some of them are camera-based methods [7], [8] and LiDAR-based methods [9], [10]. RTK-GNSS is a high-precision positioning method that outputs centimeter-accurate positioning solutions. We think that RTK-GNSS has the potential to be useful for automatic driving [11], [12]. In this study, we aim to improve the accuracy by combining RTK-GNSS and IMU.

Various methods for combining GNSS and IMU have been proposed [13]–[15]. The traditional method is to use Kalman filter fusion. The Kalman filter performs best when the noise can be modeled as white Gaussian noise. However, low-cost IMUs have large bias error variations that contribute to the cumulative error of the IMU. Accurate estimation is difficult if the bias error variability cannot be modeled. Furthermore, in urban areas, GNSS suffers from multipath, which causes large position errors. GNSS error due to multipath is known to be non-normally distributed [16]. For these reasons, a method to reduce the effects of IMU bias and GNSS multipath is needed in urban areas.

We focus on a method [17] that achieves lane-level accurate position estimation even in urban areas. The method [17] is characterized by statistical processing of time-series data ranging from a few seconds to several hundred seconds. The method [17] estimates vehicle motion based on GNSS Doppler. The estimation of vehicle motion is useful for IMU bias estimation and vehicle trajectory generation. The method [17] can use vehicle trajectories to remove multipath-induced outliers. This is because the shape of the vehicle trajectory generated over several hundred meters, and the shape of the trajectory, allows GNSS solutions that do not match previous paths to be determined as being multipath. Finally, the re-

maining GNSS is integrated with the vehicle trajectory by the least-squares method to obtain the position. It is shown that the method [17] improves robustness to multipath.

On the other hand, the method [17] assumes that there is no error in the vehicle trajectory. The accuracy of the vehicle trajectory is much higher than that of the GNSS positioning solution. Therefore, the error of the vehicle trajectory is assumed to be negligibly small. However, when centimeter-accurate positioning solutions such as RTK-GNSS are available, the error in the vehicle trajectory becomes non-negligible. As a result, the benefits of RTK-GNSS are not sufficient to improve performance.

In this study, we focus on extensibility to graph optimization by treating errors in vehicle trajectories with a probabilistic model. Graph optimization has been proposed as an approach to solve the SLAM problem as a typical example [18]–[20]. An example of graph-based SLAM is the pose graph optimization method, which uses a probabilistic model to optimize the robot’s position and pose. The pose graph optimization method improves accuracy by modifying the shape of the entire map to minimize the accumulated errors in map construction. We expect that the same effect can be obtained by probabilistically treating vehicle trajectories in the same way.

However, the general usage of graph optimization is mainly for batch processing, and there is little research on real-time operation. Complex graphical models with outlier removal also require more processing time. The proposed method enables real-time operation by simplifying the graph structure and limiting the time-series data to be optimized. In this study, we attempt to develop an integrated method that takes into account errors in vehicle trajectories and fully exploits the performance of RTK-GNSS by applying the proposed method.

II. RELATED WORK

Various methods of position estimation using graph optimization have been proposed in the past [21]–[23]. This chapter describes in detail a method that has an outlier removal function in graph optimization [24]–[27].

The combined GNSS/IMU method can be roughly divided into loose coupling and tight coupling [28], [29]. Loose coupling is a method that integrates the GNSS positioning calculation results with each sensor. In contrast, the tight coupling method uses pseudorange, Doppler shift, and other observables from GNSS receivers for sensor integration. In [25], both methods are implemented and compared with a traditional method, the extended Kalman filter. The evaluation shows that the graph optimization method performs better than the EKF method, and the tight coupling method performs better. The [25] tight coupling method also has a multipath decision for each satellite. It is reported that the longer the time-series data used for graph optimization, the smaller the effect of multipath GNSS. However, it has been confirmed that the more time-series data to be optimized, the more processing time required for optimization. Therefore, the method [25] is not suitable for real-time applications in automated driving.

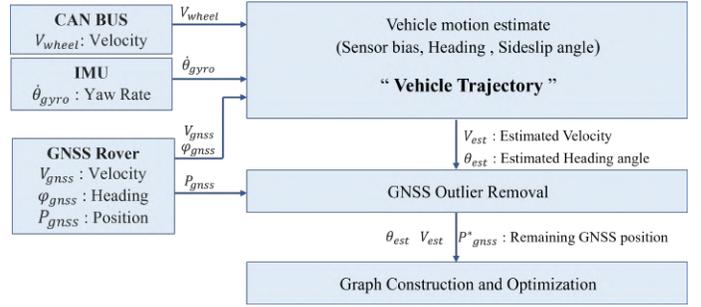


Fig. 1. Overview of the proposed method

Method [26], [27], which applies graph optimization to GNSS positioning algorithms, introduces a switch variable that represents the reliability of the GNSS observation. The switch variable is used for the pseudo-distance of each satellite in Method [26] and for detecting the cycle-slip of the carrier wave in Method [27]. The switch variable is designed to be less reliable in the case of multipath or loss of GNSS signal. Low-reliability GNSS observations are given less weight in positioning calculations or are not used at all. This method suppresses the degradation of position estimation performance. However, the addition of switch variables to the state variables tends to increase the complexity of the graphical model. Similar to method [25], method [26], [27] is expected to require more processing time due to the large number of optimization variables. Therefore, these methods are assumed to be batch processing.

III. PROPOSED METHOD

A. Overview of the proposed method

By introducing graph optimization, we aim to improve the accuracy of position estimation by considering errors in the vehicle trajectory of the method [17]. Fig. 1 shows an overview of the proposed method. The proposed method has three main structures. First, the vehicle motion is estimated by the method [17]. At the same time, the errors of the IMU and wheel speed sensors are estimated. Next, a graph is constructed using each sensor data and the estimated sensor errors. Finally, the graph is optimized to obtain position estimation results.

The challenges of the proposed method are to operate in real-time and to improve its robustness against GNSS multipath. As described in section II, increasing the complexity of the graph structure, such as tight coupling and the introduction of switch variables, increases the processing time. The proposed method uses a loose coupling method, which is a light processing method. Outlier removal is not performed during graph optimization, which is a time-consuming process, but is performed in the previous step. Furthermore, the processing time is maximally reduced by limiting the range of graph optimization.

This paper focuses on the similarities between the methods [17] and general graph optimization. Therefore, the optimization models of each method are described in sections III-B–III-C. Next, based on the similarities shown in sections III-D,

we describe extensions of Method [17] to graph optimization. Next, we explain the key points of the proposed method: outlier removal and limitation of the optimization range.

B. Methodology of method [17] optimization

The method [17] does not optimize the position and orientation simultaneously, but rather separately. This is to ensure highly accurate estimation of sensor errors in the IMU and wheel speedometer. The method [17] uses only compatible sensor data for the composite. The vehicle trajectory estimated by this method can be generated with an accuracy of about 0.5m error per 100m.

This section describes in detail the optimization model for position and heading angle estimation in the method [17]. First, heading angle estimation is explained. For heading angle estimation, the GNSS Doppler and IMU yaw rate are estimated using the least-squares method. The optimization model used for estimation is shown in (1) and (2).

$$\theta_{imu_t} = \theta_{imu_0} + \int_0^t \dot{\theta} dt \quad (1)$$

$$\theta_{est_t} = \underset{\theta_{imu_0}}{\operatorname{argmin}} \sum_{i=0}^t \{\theta_{imu_i} - \theta_{gnss_i}\}^2 \quad (2)$$

θ_{imu_t} is the heading angle based on yaw-rate integration, θ_{imu_0} is the initial value of yaw-rate integration, θ is the yaw-rate, and (1) represents the yaw-rate integration model. θ_{gnss_t} is the heading angle calculated from GNSS Doppler. By estimating the initial value θ_{imu_0} that minimizes the sum of squares of the residuals from θ_{imu_t} and θ_{gnss_t} , the final heading angle estimate E can be obtained. In other words, formula (2) represents an optimization model using the least-squares method to find a plausible heading angle.

As with the heading angle, the position is estimated by the least-squares method. The optimization model used for estimation is shown in (3)~(5).

$$T_{x_t} = T_{x_0} + \int_0^t V_t \cdot \cos\theta_t dt \quad (3)$$

$$T_{y_t} = T_{y_0} + \int_0^t V_t \cdot \sin\theta_t dt \quad (4)$$

$$\mathbf{T}_t = \underset{\mathbf{T}_0}{\operatorname{argmin}} \sum_{i=0}^t \{\mathbf{T}_i - \mathbf{P}_{gnss_i}\}^2 \quad (5)$$

Where $\mathbf{T}_t(T_{x_t}, T_{y_t})$ is the vehicle trajectory, \mathbf{T}_0 is the initial value of the vehicle trajectory, V is the vehicle speed, and $\mathbf{P}_{gnss_t}(P_{x_t}, P_{y_t})$ is the GNSS positioning solution. As with the heading angle, formula (3) and (4) are the models for generating the vehicle trajectory, and formula (5) is the optimization model for finding a plausible position. In this way, the method [17] estimates position and heading angle separately. Another feature of the method is that the optimization handles time series data for tens of seconds for heading estimation and hundreds of meters for position estimation.

If the GNSS contains multipath, the estimation performance will deteriorate. The method [17] detects multipath when

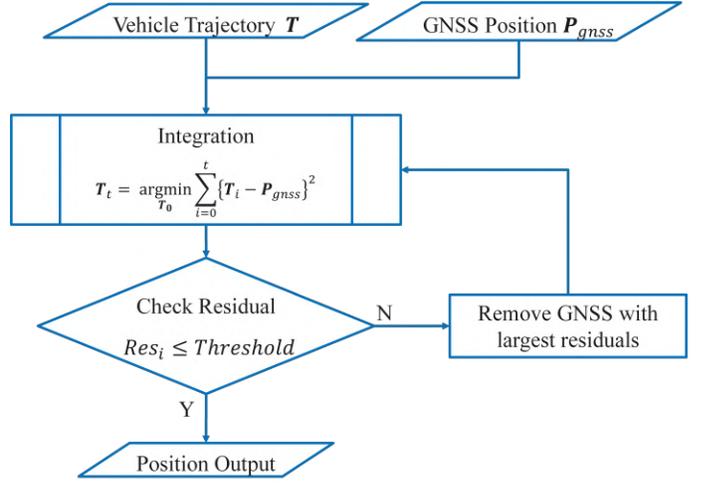


Fig. 2. GNSS Outlier Removal Chart

calculating the residuals in (2) and (5). Fig. 2 shows a flowchart of multipath detection, and Fig. 3 shows outlier removal, using a vehicle trajectory as an example. As shown in (6), if the residual is greater than a threshold value, the detection is considered to be an outlier of the GNSS positioning solution subject to multipath. The GNSS positioning solution determined to be an outlier is removed from the least-squares calculation to improve robustness. As shown in Fig. 2, outliers are eliminated in the order of the residuals in (6). If all remaining GNSS positioning solutions are within the threshold, multipath removal is considered complete and the final position and heading angle are output. Lane-level position estimation is achieved using this method.

$$Res_i = \begin{cases} \{\theta_{imu_i} - \theta_{gnss_i}\}^2 \\ or \\ \{\mathbf{T}_i - \mathbf{P}_{gnss_i}\}^2 \end{cases} = \begin{cases} 0 \text{ for } Res_i > Th \\ Res_i \text{ for } Res_i < Th \end{cases} \quad (6)$$

However, this method relies heavily on the accuracy of the yaw-rate integration and the vehicle trajectory. The optimization in (2) and (5) estimates only plausible initial values, respectively. Errors that occur in the process of integration from the initial values are not taken into account. In other words, the yaw-rate integration value and the vehicle trajectory are treated as if they were "rigid bodies that are not allowed to deform". As shown in Fig. 3, the initial values of the vehicle trajectory were shifted in parallel so that the evaluation function in (5) becomes smaller. At the time of integration, if a highly accurate positioning solution such as RTK-GNSS is left, the error in the vehicle trajectory directly becomes a position estimation error. Therefore, an integration method that takes vehicle trajectory errors into account is necessary to take advantage of the performance of RTK-GNSS.

C. Methodology of general graph optimization

We discuss SLAM as an example of general graph optimization. One of the problems with SLAM is that the cumulative map error becomes large due to errors included when updating

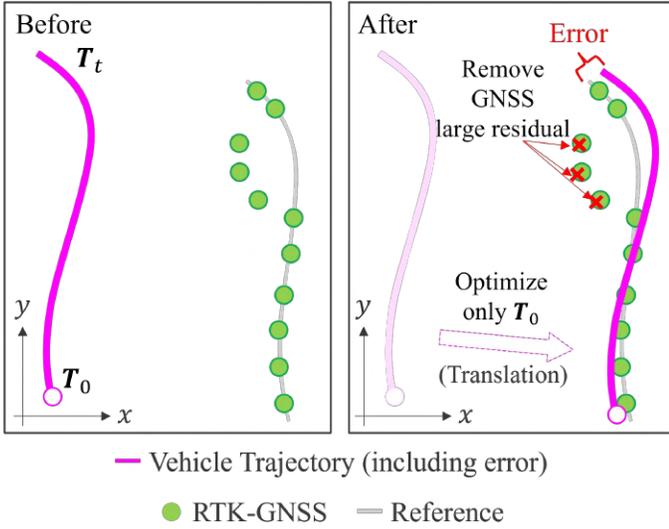


Fig. 3. Overview and issues of optimization in method [17]

the map. One solution is to introduce graph optimization. Graph optimization improves map construction accuracy by deforming the map to minimize the cumulative error. This paper describes graph optimization in detail.

Graph optimization is a method for finding the maximum posterior probability of a moving robot in state $\mathbf{x}_{0:t}$ and map m , given a time series of observations $\mathbf{z}_{1:t}$. In particular, the method that optimizes the state $\mathbf{x}_{0:t}$ and the map m separately is called pose graph optimization. First, the state $\mathbf{x}_{0:t}$ of the robot is optimized. Then, a map is constructed based on the optimized states.

According to [19], the model of pose graph optimization is in (7).

$$\mathbf{x}_{0:t} = \underset{\mathbf{x}_{0:t}}{\operatorname{argmax}} \prod_{i=0}^t p(\mathbf{x}_i | \mathbf{x}_{i-1}) \prod_{j=1}^t p(\mathbf{z}_j | \mathbf{x}_j) \quad (7)$$

Where $p(\mathbf{x}_t | \mathbf{x}_{t-1})$ is the state transition probability and $p(\mathbf{z}_t | \mathbf{x}_t)$ is the observation probability. If these probabilities are assumed to be normally distributed, they can be expressed as Equation (8), (9).

$$\mathbf{x}_{0:t} = \underset{\mathbf{x}_{0:t}}{\operatorname{argmin}} F(\mathbf{x}_{0:t}) \quad (8)$$

$$F(\mathbf{x}_{0:t}) = \sum_{i=0}^t \|\mathbf{x}_i - g_i(\mathbf{x}_{i-1})\|_{\Sigma_{g_i}}^2 + \sum_{j=1}^t \|\mathbf{z}_j - h_j(\mathbf{x}_j)\|_{\Sigma_{h_j}}^2 \quad (9)$$

In (8) and (9), $g_t(\cdot)$ and $h_t(\cdot)$ are the state transition function and the observation function, respectively, Σ_{g_t} and Σ_{h_t} are the covariance, and $\|\cdot\|_{\Sigma}^2$ is the Mahalanobis distance. F is called the error function. In other words, the problem is to solve a nonlinear least-squares method to minimize the error

function. Solving this problem yields an optimal solution and thus a plausible robot state.

D. Graph optimization of the proposed method

The similarities between Method [17] and graph optimization in the proposed method are explained. In Method [17], several relationships become apparent when the states \mathbf{x} and observables \mathbf{y} are defined as follows.

$$\mathbf{x} = [x, y, \theta]^T \quad (10)$$

$$\mathbf{y} = [x_{gnss}, y_{gnss}, \theta_{gnss}]^T \quad (11)$$

In the state, \mathbf{x} , \mathbf{y} is the position and θ the heading angle. Focusing on the error function $F(\mathbf{x})$ in (9), the second term calculates the residual between the observed quantity and the state. This part corresponds to the calculation of the residuals in (2) and (5) in the method [17]. Comparing the states to be optimized, the method [17] uses only the initial values, while the graph optimization uses from the initial values to the current time. In other words, the method [17] was an optimization model under the constraint that the first term of the error function is not considered in the graph optimization and that only the initial values are optimized.

The proposed method extends the optimization model of Method [17] by adding residuals in state transitions and by setting the optimization variable from the initial value to the current time. The residuals of the state transitions indicate the relationship between the previous and next states. This corresponds to (1), (3), and (4) in the method [17]. In other words, by adding the residuals of the state transitions to the error function, errors in the vehicle trajectory can be taken into account. Finally, the vehicle trajectory is integrated with the shape of the trajectory deformed, which is expected to improve position estimation performance.

For graph optimization of the proposed method, the state transition function and the observation function are shown in (12) and (13).

$$g_t(\mathbf{x}_{t-1}) = \begin{bmatrix} x_{t-1} + \int_{t-1}^t V_t \cdot \cos\theta_t dt \\ y_{t-1} + \int_{t-1}^t V_t \cdot \sin\theta_t dt \\ \theta_{t-1} + \int_{t-1}^t \dot{\theta}_t dt \end{bmatrix} \quad (12)$$

$$h_t(\mathbf{x}_t) = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} \quad (13)$$

The state-space model defined by (10)~(13) is optimized according to (9). The optimization is performed using g^2o [30], a library for solving nonlinear least-squares methods.

In general graph optimization, the variables to be optimized increase as time passes. The more variables to optimize, the longer the processing time increases. To avoid this problem, the proposed method places a limit on the variables to be optimized. Specifically, the proposed method optimizes only

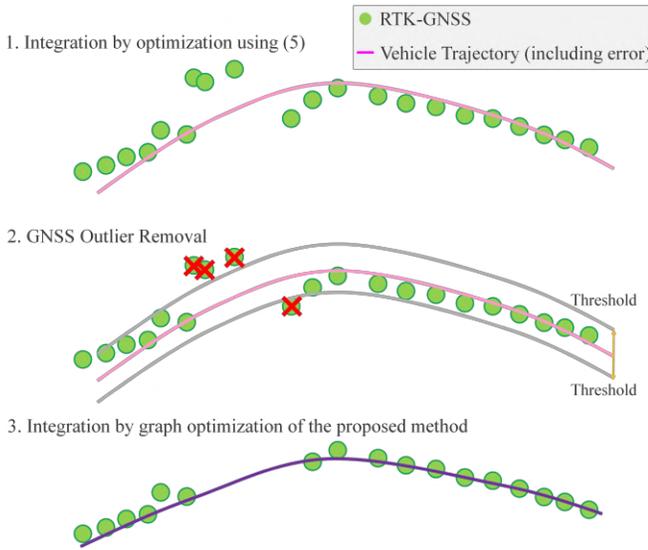


Fig. 4. Optimization algorithm of the proposed method

the data for several hundred meters from the current time. This method solves the problem of processing time increase as time passes.

In addition, to ensure real-time operation, the optimization timing is at GNSS reception. In graph optimization, the state is adjusted more significantly during observation updates than during state transitions. In other words, the observation update by GNSS causes a large change in the state. In addition, GNSS has a longer sensor frequency than IMUs, etc., which allows a margin of time until the time deadline for the next step. These efforts enable the proposed method to operate in real-time.

However, the current graph optimization cannot handle GNSS outliers due to multipath. Graph optimization using (8) assumes that all probabilities are normally distributed. GNSS multipath errors that occur in urban areas are non-normally distributed. Therefore, direct application of general graph optimization may cause performance degradation. In particular, unlike the method [17], the optimization solution may be attracted to GNSS outliers because they deform the vehicle trajectory. Also, increasing the number of variables to be optimized for outlier removal, as in conventional methods, increases processing time. The proposed method aims to achieve both outlier removal and real-time operation. Therefore, the outlier removal method of [17] is employed before graph optimization, as shown in Fig. 4. This method allows outlier removal without increasing the processing time of graph optimization. It has been confirmed that the outlier removal method in [17] works in real-time. Therefore, real-time operation is possible as long as the graph optimization process is completed by the scheduled time. Whether real-time operation is possible or not will be verified in the evaluation.



Fig. 5. Evaluation course

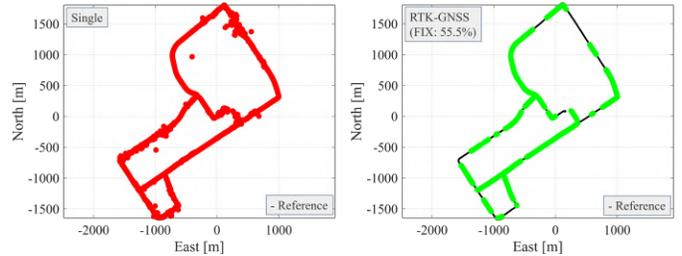


Fig. 6. GNSS Positioning Distribution

IV. EVALUATE EXPERIMENT

A. Experiment Overview

The proposed method is evaluated using real data. The evaluation uses the open-dataset [31] published by Meijo University. Fig. 5 shows the evaluation course. The test environment is Odaiba, Tokyo, where multipaths occur frequently. The proposed method is validated with a low-cost sensor. The GNSS receiver is the Ublox F9P and the IMU is the Analog Devices ADIS16475-2. Wheel speed is obtained from the vehicle's CAN-BUS. POSLV220 post-processing results are used for the reference.

The evaluation compares the position estimation performance of the proposed method with that of general graph optimization and the method [17]. Two types of GNSS positioning methods are used. One is single point positioning, which is susceptible to multipath. The other is a receiver-based FIX solution of RTK-GNSS. For comparative evaluation, the cumulative distribution of errors is used. The horizontal axis shows the 2D-error, and the vertical axis shows the cumulative frequencies converted into percentages.

B. GNSS positioning evaluation

Before evaluating the proposed method, we evaluate the GNSS positioning results. Fig. 5 shows the GNSS positioning distribution and Table I shows the GNSS positioning error. The results of the single point positioning show that the positioning solution deviates from the course significantly. The maximum error of the solution was 917 meters. This result shows that the single point positioning is strongly affected by multipath.

TABLE I
EVALUATION RESULTS OF EACH GNSS POSITIONING METHOD

	Error Mean [m]	Error Standard Deviation [m]
Single Point Positioning	4.3	16.4
RTK-GNSS FIX	0.07	0.29

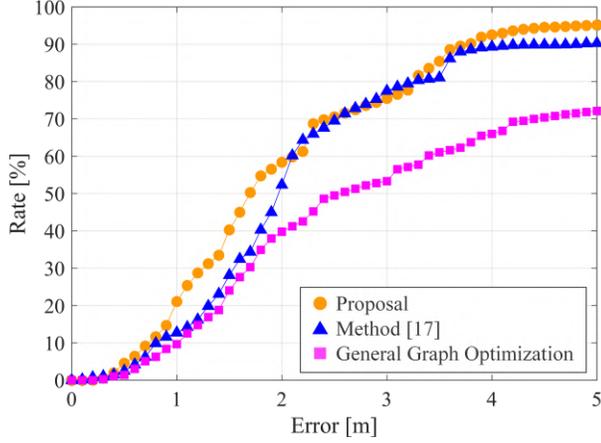


Fig. 7. Position evaluation results using single point positioning

Even in locations where the multipath effect is small, offset errors due to the ionosphere, troposphere, and other factors are included. The accuracy of the single point positioning is low because it includes these errors.

The RTK-GNSS results are very accurate compared to the single point positioning. The error average is 0.07 m, and this result shows that RTK-GNSS is a centimeter-accurate positioning method. On the other hand, unlike the single point positioning, there are some areas where there is no positioning solution. The RTK-GNSS FIX solution was obtained for 55.5% of the GNSS reception epochs. This is because the FIX solution cannot be output until the carrier ambiguity is determined. In an environment with large multipath, the accuracy of ambiguity determination is reduced, and thus the positioning solution cannot be obtained.

C. Position estimation evaluation

Fig. 7 shows the cumulative distribution using the single point positioning solution as the GNSS positioning method. And Fig. 8 9 shows the position estimation results for each method. Since the single positioning method is less susceptible to multipath, GNSS outliers are noticeable. According to Fig. 7, the results for general graph optimization are inaccurate. This is because general graph optimization is not equipped with outlier removal functions. The inability to reduce the effect of GNSS multipath degrades the accuracy of position estimation. Fig. 8 also shows that general graph optimization results in position estimation that is affected by GNSS outliers.

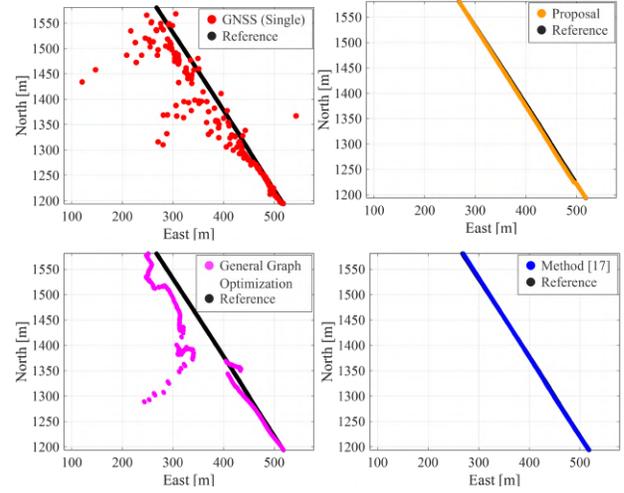


Fig. 8. Position estimation results for each method (many multipaths)

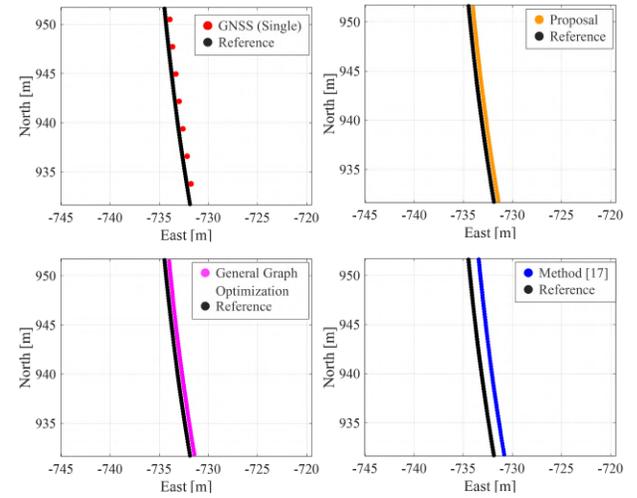


Fig. 9. Position estimation results for each method (less multipaths)

On the other hand, the proposed method is more accurate than general graph optimization. The proposed method with the outlier removal function can suppress GNSS multipath errors. This result indicates that the proposed method is effective in removing outliers. The performance of the proposed method is comparable to that of the method [17]. As shown in Fig. 9, the proposed method tends to perform better in environments with fewer multipaths in single point positioning. When the accuracy of the single point positioning solution is higher than that of the vehicle trajectory, the method [17] degrades the accuracy of position estimation due to errors in the trajectory shape. In contrast, the proposed method and general graph optimization produce position results closer to the reference. Therefore, it can be seen that under high GNSS accuracy, graph optimization can provide position estimation with reduced vehicle trajectory errors.

Fig. 10 also shows the cumulative distribution of FIX

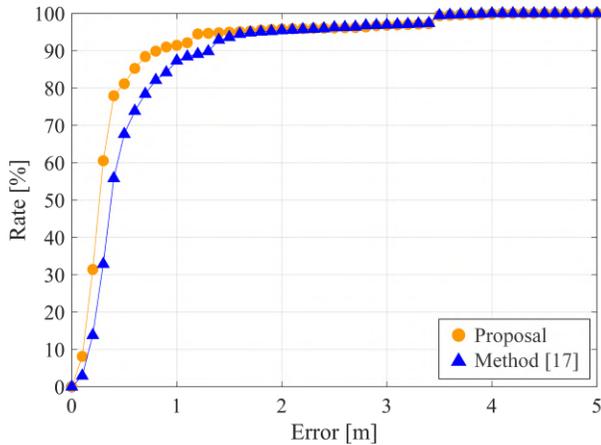


Fig. 10. Position evaluation results using FIX of RTK-GNSS

solutions for RTK-GNSS as input. In the RTK-GNSS evaluation, because there are no significant outliers in the FIX solution, the results of the general graph optimization and the proposed method are exactly the same. Therefore, the results of the general graph optimization are not shown in Fig. 10. Fig. 10 shows that the proposed method is more accurate than the method [17]. Focusing on the accumulation of position estimates within an error of 0.3 m, the performance improved from 32.8% to 60.5%. This result indicates that the benefits of RTK-GNSS are utilized enough by considering vehicle trajectory errors and integrating them, which is the objective of this study. As with single point positioning, the proposed method is expected to be able to remove outliers even if they are included in the RTK-GNSS.

These results indicate that the proposed method can contribute to improving the accuracy of position estimation by extending the method [17]. Moreover, we confirmed that the proposed method is effective in suppressing the degradation of position estimation performance in urban environments even when GNSS outliers due to multipath are included in the estimation. Therefore, the proposed method is effective in improving the robustness against multipath and the position estimation performance.

D. Processing time evaluation

Verify whether the proposed method is capable of real-time operation. Since the proposed method is intended to be applied to automatic driving, real-time operation is required. The proposed method is implemented in ROS and parallel processing is assumed. The real-time operation of the proposed method has been verified for the parts that are common to the methods [17]. Therefore, it is necessary to investigate the processing time required for the outlier removal function and optimization computation in the proposed method.

Table II shows the measured processing time of the proposed method for outlier removal and graph optimization. The optimization of the proposed method is executed at the time of GNSS reception. In this evaluation, the GNSS runs at 5

Hz, so the processing should be completed within 200 msec. Table II shows that each function and the total time required for each function is within the required time. The time margin is more than 10 times. Therefore, the proposed method can operate in real-time.

V. CONCLUSION

Accurate self-positioning is important for automatic driving of vehicles and mobile robots. In this study, we aimed to propose a highly accurate position estimation method by combining RTK-GNSS and IMU. The use of RTK-GNSS is essential to improve the accuracy of position estimation. The method [17] can remove outliers of multipath GNSS positioning solutions using vehicle trajectories. However, the method [17] did not improve the accuracy when combined with RTK-GNSS because it did not consider the error of the vehicle trajectory.

In this study, we focused on the possibility of extending the method [17] to graph optimization by redefining the vehicle trajectory error in a probabilistic model. Graph optimization enables integration that takes into account errors in vehicle trajectories. However, graph optimization in general is based on batch processing, and real-time operation is difficult. Methods with outlier elimination also increase the processing time due to the complexity of the graph structure. The proposed method simplifies the graph structure and divides the optimization into two stages, enabling outlier removal without increasing processing time. In addition, optimization range is limited and optimization calculations are executed at the time of GNSS reception, which enables real-time operation.

To validate the proposed method, it is evaluated in an urban area where multipath occurs frequently. The validation of the proposed method is compared using single point positioning (which is vulnerable to multipath) and RTK-GNSS as input to the GNSS. The results of the single point positioning show that the proposed method, which can remove GNSS outliers, performs well compared to general graph optimization. In the evaluation using RTK-GNSS, it is confirmed that the proposed method achieves more accurate position estimation than the method [17] as originally intended. The evaluation results show the effectiveness of the proposed method as a graph optimization method that can operate in real-time and has the ability to remove outliers in GNSS positioning solutions due to multipath.

Currently, as with Pose Graph optimization, the state is composed of position and pose. Graph optimization is not introduced in the vehicle trajectories generated by the method [17]. The accuracy of the vehicle trajectory is improved by estimating the error of each sensor. Further performance improvement is expected by introducing graph optimization to the estimation of the error amount for each sensor. In the future, we will consider a method that can operate in real-time by extending the sensor error to the state of graph optimization.

TABLE II
PROCESSING TIME EVALUATION FOR OPTIMIZATION

	Outlier removal		Optimization		Total		Real Time Require [msec]
	Max [msec]	Mean [msec]	Max [msec]	Mean [msec]	Max [msec]	Mean [msec]	
Single Point Positioning	4.74	1.01	14.42	8.35	18.28	9.35	<200
RTK-GNSS FIX	1.21	0.46	15.36	7.64	16.26	8.11	<200

ACKNOWLEDGMENT

The paper is evaluated using the open data set published by Meijo University [31].

REFERENCES

- [1] Jessica Van Brummelen, Marie O'Brien, Dominique Gruyer, and Homayoun Najjaran. Autonomous vehicle perception: The technology of today and tomorrow. *Transportation Research Part C: Emerging Technologies*, 89:384–406, 2018.
- [2] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE Access*, 8:58443–58469, 2020.
- [3] Sampo Kuutti, Saber Fallah, Konstantinos Katsaros, Mehrdad Dianati, Francis Mccullough, and Alexandros Mouzakitis. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. *IEEE Internet of Things Journal*, 5(2):829–846, 2018.
- [4] Karl Rehrl and Simon Gröchenig. Evaluating localization accuracy of automated driving systems. *Sensors*, 21(17), 2021.
- [5] Kelvin Wong, Ehsan Javanmardi, Mahdi Javanmardi, and Shunsuke Kamijo. Estimating autonomous vehicle localization error using 2d geographic information. *ISPRS International Journal of Geo-Information*, 8(6), 2019.
- [6] Tyler GR Reid, Sarah E Houts, Robert Cammarata, Graham Mills, Siddharth Agarwal, Ankit Vora, and Gaurav Pandey. Localization requirements for autonomous vehicles. *arXiv preprint arXiv:1906.01061*, 2019.
- [7] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6684–6692, 2017.
- [8] Henning Lategahn, Markus Schreiber, Julius Ziegler, and Christoph Stiller. Urban localization with camera and inertial measurement unit. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 719–724. IEEE, 2013.
- [9] Keisuke Yoneda, Hossein Tehrani, Takashi Ogawa, Naohisa Hukuyama, and Seiichi Mita. Lidar scan feature for localization with highly precise 3-d map. In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 1345–1350. IEEE, 2014.
- [10] Yuan Xu, Yuriy S Shmaliy, Yueyang Li, Xiyuan Chen, and Hang Guo. Indoor ins/lidar-based robot localization with improved robustness using cascaded fir filter. *IEEE Access*, 7:34189–34197, 2019.
- [11] Markus Dorn, Julian O Filwary, and Manfred Wieser. Inertially-aided rtk based on tightly-coupled integration using low-cost gnss receivers. In *2017 European Navigation Conference (ENC)*, pages 186–197. IEEE, 2017.
- [12] Joong-hee Han, Chi-ho Park, Young-Jin Park, and Jay Hyoun Kwon. Preliminary results of the development of a single-frequency gnss rtk-based autonomous driving system for a speed sprayer. *Journal of Sensors*, 2019:1–9, 2019.
- [13] Jinsil Lee, Minchan Kim, Jiyun Lee, and Sam Pullen. Integrity assurance of kalman-filter based gnss/imu integrated systems against imu faults for uav applications. In *Proceedings of the 31st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2018)*, pages 2484–2500, 2018.
- [14] Siavash Hosseinyalamdary. Deep kalman filter: Simultaneous multi-sensor integration and modelling; a gnss/imu case study. *Sensors*, 18(5):1316, 2018.
- [15] Peihui Yan, Jinguang Jiang, Fangning Zhang, Dongpeng Xie, Jiaji Wu, Chao Zhang, Yanan Tang, and Jingnan Liu. An improved adaptive kalman filter for a single frequency gnss/mems-imu/odometer integrated navigation module. *Remote Sensing*, 13(21):4317, 2021.
- [16] Vincent Havyarimana, Zhu Xiao, Pierre Claver Bizimana, Damien Hanyurwimfura, and Hongbo Jiang. Toward accurate intervehicle positioning based on gnss pseudorange measurements under non-gaussian generalized errors. *IEEE Transactions on Instrumentation and Measurement*, 70:1–12, 2020.
- [17] Aoki Takanose, Yuki Kitsukawa, Junichi Meguro, Eijiro Takeuchi, Alexander Carballo, and Kazuya Takeda. Eagleeye: A lane-level localization using low-cost gnss/imu. In *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pages 319–326, 2021.
- [18] Niko Sünderhauf and Peter Protzel. Towards a robust back-end for pose graph slam. In *2012 IEEE International Conference on Robotics and Automation*, pages 1254–1261, 2012.
- [19] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.
- [20] Frank Dellaert. Factor graphs: Exploiting structure in robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:141–166, 2021.
- [21] Zheng Gong, Peilin Liu, Fei Wen, Rendong Ying, Xingwu Ji, Ruihang Miao, and Wuyang Xue. Graph-based adaptive fusion of gnss and vio under intermittent gnss-degraded environment. *IEEE Transactions on Instrumentation and Measurement*, 70:1–16, 2021.
- [22] Vadim Indelman, Stephen Williams, Michael Kaess, and Frank Dellaert. Factor graph based incremental smoothing in inertial navigation systems. In *2012 15th International Conference on Information Fusion*, pages 2154–2161, 2012.
- [23] Sina Taghizadeh, Mohsen Nezhadshahbodaghi, Reza Safabakhsh, and Mohammad Reza Mosavi. A low-cost integrated navigation system based on factor graph nonlinear optimization for autonomous flight. *GPS Solutions*, 26(3):78, 2022.
- [24] Wei Li, Xiaowei Cui, and Mingquan Lu. A robust graph optimization realization of tightly coupled gnss/ins integrated navigation system for urban vehicles. *Tsinghua Science and Technology*, 23(6):724–732, 2018.
- [25] Weisong Wen, Tim Pfeifer, Xiwei Bai, and Li-Ta Hsu. Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter. *NAVIGATION: Journal of the Institute of Navigation*, 68(2):315–331, 2021.
- [26] Niko Sünderhauf, Marcus Obst, Gerd Wanielik, and Peter Protzel. Multipath mitigation in gnss-based localization using robust optimization. In *2012 IEEE Intelligent Vehicles Symposium*, pages 784–789, 2012.
- [27] Taro Suzuki. Gnss odometry: Precise trajectory estimation based on carrier phase cycle slip estimation. *IEEE Robotics and Automation Letters*, 7(3):7319–7326, 2022.
- [28] Gianluca Falco, Marco Pini, and Gianluca Marucco. Loose and tight gnss/ins integrations: Comparison of performance assessed in real urban scenarios. *Sensors*, 17(2):255, 2017.
- [29] Omar Garcia Crespillo, Daniel Medina, Jan Skaloud, and Michael Meurer. Tightly coupled gnss/ins integration based on robust m-estimators. In *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 1554–1561. IEEE, 2018.
- [30] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. In *2011 IEEE International Conference on Robotics and Automation*, pages 3607–3613, 2011.
- [31] Junichi Meguro, Taro Suzuki, Nobuaki Kubo, and Takahiro Kashimoto. GNSS/IMU Open Data Set for Autonomous Driving, 2022.