

Eagleeye: A Lane-Level Localization Using Low-Cost GNSS/IMU

Aoki Takanose¹, Yuki Kitsukawa^{1,2}, Junichi Meguro³, Eijiro Takeuchi^{4,5}, Alexander Carballo^{4,5}, Kazuya Takeda^{1,4,5}

Abstract— In this paper, we propose Eagleeye, an open-source software, that performs lane level localization in an urban environment. A low-cost GNSS receiver, IMU, and velocity sensor are used for position estimation. The feature of this method is that it is optimized to take full advantage of the averaging effect using time series data longer than a few tens of seconds. This optimization improves the estimation performance by reducing the GNSS multipath in urban areas. In order to verify the effectiveness of the system, we conducted accuracy evaluation of the proposed method and performance comparison tests with expensive position estimation systems. As a result of the test, we confirmed that the proposed method can estimate the relative position results with an accuracy of 0.5 m per 100m and the absolute position performance with an accuracy of 1.5 m. In addition, it was confirmed that the performance of the proposed method was equivalent to that of an expensive system. Therefore, it is considered that the proposed method can effectively estimate the location even in an urban environment.

I. INTRODUCTION

Autonomous vehicles and advanced driver assistance applications require vehicle position information. Accurate and robust position estimation techniques are required for vehicle navigation and control in complex road environments involving vehicles and pedestrians. These position estimation technologies will be used not only for automated vehicles [1-3], but also for transportation robots such as AGVs (Automated guided vehicle) [4,5] mapping systems such as SLAM (Simultaneous Localization and Mapping) technology [6,7], and driver assistance systems [8]. In particular, for automatic driving, we believe that a position estimation performance with an error of at least 0.3m or less is required, referring to the references [9].

Among the efforts on robot and vehicle position estimation, the mainstream methods are related to SLAM using 2D and 3D LiDAR [10]. As an example, the 3D Normal Distributions Transform [11,12] converts the input map into 3D normal distributions and the input scan is matched against the normal distributions, achieving low positioning errors and cost to performance trade-off, and is currently in use in self-driving vehicles as the main localization algorithm in Autoware open source self-driving software [13,14]. However, there are some situations where LiDAR localization is not good and may fail (e.g. low feature environments). In addition, high precision localization of scan matching based solutions comes at a cost: 3D LiDARs are expensive and high precision 3D pointcloud maps may be even more expensive to obtain.

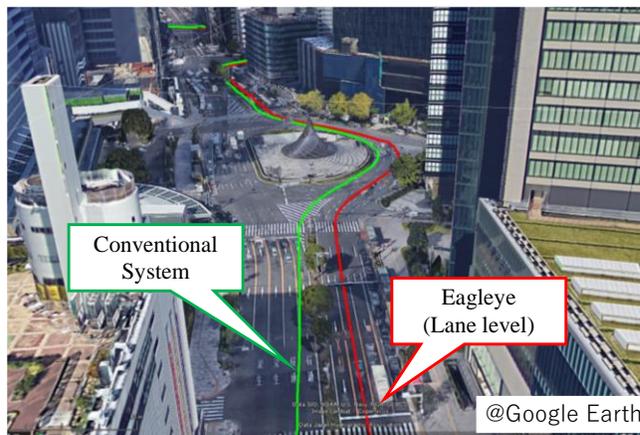


Figure 1 The Eagleeye algorithm is a lane-level position estimation system. It can estimate the position within a lane as shown in the figure.

These problems can be solved by GNSS (Global Navigation Satellite System) and IMU (Inertial Measurement Unit). By utilizing the position results estimated by GNSS/IMU, the failure of LiDAR localization can be interpolated. In order to do so, the GNSS/IMU system needs to have at least lane level accuracy. Conventionally, a high accuracy GNSS/IMU system would cost tens of thousands of dollars or more. Considering the widespread use of this system, we expect that lowering the cost of this system will enable its application to general vehicles.

Therefore, in this paper, we propose a low-cost, accurate, and robust location estimation algorithm. We call this algorithm Eagleeye, and have made it available as open source [15]. Eagleeye is implemented in ROS, and it is an algorithm based on [16,17]. The Eagleeye algorithm uses a low-cost GNSS receiver, a three-axis MEMS-IMU, and a vehicle speed sensor connected via a CAN (Controller Area Network) bus, thus reducing the total cost to a few hundred dollars. The Eagleeye algorithm has the following features:

- Sensor error estimation based on GNSS doppler.
- Vehicle motion estimation based on GNSS doppler.
- Rejection of Multipath Positioning Results Considering Vehicle Motion.

The algorithm achieves a relative position error of 0.5 m per 100 m and an absolute position error of 1.5 m, which is lane level accuracy (Figure1).

¹Graduate School of Informatics, Nagoya University, Aichi, Japan.

²MapIV Inc. Nagoya University Open Innovation Center, 1-3,Mei-eki 1-chome, Nakamura-Ward, Nagoya, 450-6610, Japan.

³Graduate School of Mechatronics Engineering, Meijo University, Aichi, Japan.

⁴Institute of Innovation for Future Society, Nagoya University, Furocho,

Chikusa-ku, Nagoya 464-8601, Japan.

⁵TierIV Inc., Nagoya University Open Innovation Center, 1-3,Mei-eki 1-chome, Nakamura-Ward, Nagoya, 450-6610, Japan.

Email: takanose.aoki@g.sp.m.is.nagoya-u.ac.jp, yuki.kitsukawa@tier4.jp, meguro@meijo-u.ac.jp, takeuchi@g.sp.m.is.nagoya-u.ac.jp, alexander@g.sp.m.is.nagoya-u.ac.jp, kazuya.takeda@nagoya-u.jp

In this paper, we evaluate the performance of the Eagleye algorithm to verify its performance and also compare its performance with that of POSLV[18], an expensive position estimation system.

II. RELATED WORKS

Applications using GNSS are becoming more and more diverse. In recent years, multi-GNSS has become available due to the increase of satellite systems [19-21]. The main types of multi-GNSS are Global Positioning System (GPS) of the United States, Galileo of the European Union, Global Navigation Satellite System (GLONASS) of Russia, and BeiDou Navigation Satellite System (BDS) of China. Multi-GNSS solves the problem of decreasing the number of observation satellites and improves the utilization rate. As a result, the RTK-GNSS method can provide cm-class position estimation in a favorable environment, but it requires continuous network communication with a reference station. Therefore, there is a method called PPP (Precise Point Positioning) that can provide high-precision position estimation without requiring communication with a reference station [22]. PPP improves performance by using highly accurate orbital information broadcast from satellites, but it requires meticulous initialization to ensure accurate positioning. In addition, the biggest challenge for RTK-GNSS and PPP is multipath in urban environments. Multipath occurs when satellite signals are reflected or diffracted by obstructions such as elevated tracks and tall buildings in urban areas. The multipath of satellite signals causes a significant degradation in the accuracy of GNSS position estimation.

In order to improve the robustness of GNSS, a method of integrating GNSS with various sensors has been proposed. GNSS is not capable of positioning in tunnels or under elevated structures where satellite signals cannot be received. By integrating GNSS and IMU, it is possible to estimate the position continuously even when satellite signals cannot be received. There are two main types of integration between GNSS and IMU: tight coupling [23,24], which integrates the raw values of the sensors, and loose coupling [25,26], which integrates the results of each sensor. For the integration of GNSS and IMU, Kalman filter is commonly used because it assumes that the error noise is normally distributed. However, it is known that the error noise of GNSS positioning results and satellite signal information is non-normally distributed when multipath occurs. In addition, when the satellite signal is blocked for a long time, the IMU needs to estimate the position by long-term integration. However, low-cost IMUs, such as MEMS IMUs, are prone to accumulation errors, which reduces the accuracy of position estimation. Therefore, the accuracy of position estimation in such a system is highly dependent on the accuracy of each individual sensor. As a countermeasure, there are methods to increase the performance of individual sensors by using expensive sensors. For this reason, most autonomous vehicles and mapping systems are equipped with expensive sensors.

III. SYSTEM OVERVIEW

A. Summary of the Eagleye Algorithm

The Eagleye algorithm aims to achieve lane level position estimation in an urban environment. Figure 2 shows an overview of the Eagleye algorithm. The Eagleye algorithm is one of the methods to integrate GNSS and IMU as described in Section II. Eagleye has two major features that are different from conventional methods. The first is that Eagleye does not estimate the parameters sequentially like the Kalman filter, but by accumulating time series data of tens of seconds or more. This is because the Kalman filter-based method is likely to fail in urban environments where multipath is more frequent, as described in Section II. The second is that each parameter is estimated individually while removing multipath using time series data of tens of seconds or more. Instead of simultaneously estimating the position, attitude, and velocity of the vehicle, we carefully integrate them one by one in a compatible combination. By adopting these methods, we have improved the performance of Eagleye. This paper describes in detail the wheel speedometer error estimation, heading angle estimation, sideslip angle estimation, and position estimation employed in the Eagleye algorithm.

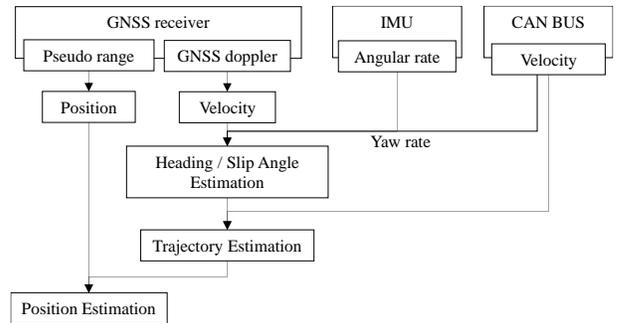


Figure 2 Block diagram of the Eagleye algorithm

B. Wheel speedometer error estimation

Generally, wheel speedometers attached to vehicles calculate wheel speed from the number of revolutions of the drive shaft and the diameter of the tire. The wheel speed is calculated using the vehicle specific reference value for tire diameter. However, due to changes in tire pressure and other factors, the actual tire diameter may differ from the reference value (Figure 3). The Eagleye algorithm assumes that there will be an error due to the ratio of the actual tire diameter to the reference value. This ratio is corrected as the scale factor SF of the measured wheel speed. The actual speed \bar{V} to the measured wheel speed V_{wheel} is related by equation (1).

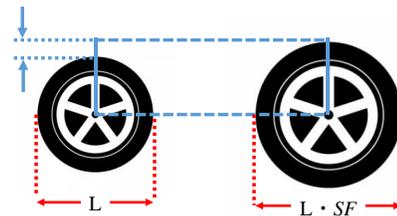


Figure 3 Diagram of how tire width changes

$$\bar{V} = V_{\text{wheel}} \cdot SF \quad (1)$$

The Eagleye algorithm uses GNSS Doppler to estimate the SF . The GNSS Doppler can estimate the velocity very accurately in a good environment. Therefore, the velocity from GNSS Doppler V_{gnss} can be considered as the actual velocity in a good environment. With this assumption, the scale factor SF can be estimated by equation (2).

$$SF = \frac{V_{\text{gnss}}}{V_{\text{wheel}}} \quad (2)$$

However, GNSS Doppler velocity V_{gnss} is not necessarily accurate. Since the actual tire diameter rarely fluctuates dynamically, it is often safe to continue using the estimated scale factor SF . Therefore, the Eagleye algorithm accumulated only data under favorable conditions, and uses the average obtained value as the scale factor SF .

C. Heading angle and Yaw rate error estimation

The heading angle is estimated by integrating the GNSS Doppler and the IMU yaw rate. The GNSS Doppler can calculate the velocity vector with respect to north as shown in Figure 4. From this velocity vector, the heading angle relative to north can be determined. In the Eagleye algorithm, the heading angle is estimated as follows:

$$\Psi_{\text{gyro}} = \Psi_{\text{init}} - \int_{t-k}^t \dot{\Psi} dt \quad (3)$$

$$\Psi_{\text{init}} = \operatorname{argmin} \sum (\Psi_{\text{gnss}} - \Psi_{\text{gyro}})^2 \quad (4)$$

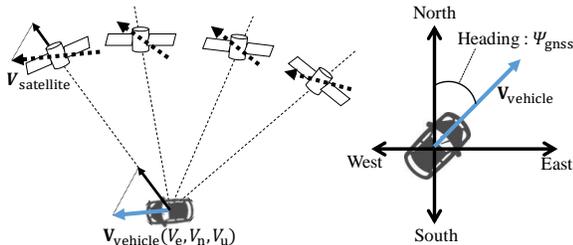


Figure 4 Relationship between velocity vector and heading angle by GNSS Doppler

where Ψ_{gyro} is the relative heading angle accumulated from the yaw rate of the gyro, $\dot{\Psi}$ is the yaw rate and Ψ_{gnss} is the heading angle obtained from GNSS Doppler. In the integration, the last 30 seconds of GNSS Doppler and yaw rate data are used to estimate the heading angle. The 30 seconds of accumulated data is minimized according to equation (4), and the resulting heading angle, Ψ_{init} , is the estimated result. In the absence of GNSS reception, the yaw rate is estimated by integrating it from the heading angle Ψ_{init} . The heading angle Ψ_{gnss} from the GNSS can be affected by multipath and cause errors in urban environments [27]. As a result, the estimated heading angle Ψ_{init} is also subject to error. The Eagleye algorithm improves the estimation performance by removing this multipath affected result. If the residual difference between the stored heading Ψ_{gyro} and the GNSS heading Ψ_{gnss} is large, the GNSS is considered to have suffered from

multipath and the data is eliminated (Figure 5). In this way, the optimal heading angle Ψ_{init} can be estimated by removing the bad data and minimizing it again according to equation (4).

Instantaneous determination of multipath in GNSS is considered to be difficult. The Eagleye algorithm uses long time series data, a feature that allows it to reject multipath results. This feature allows us to improve the accuracy of the heading angle compared to conventional methods.

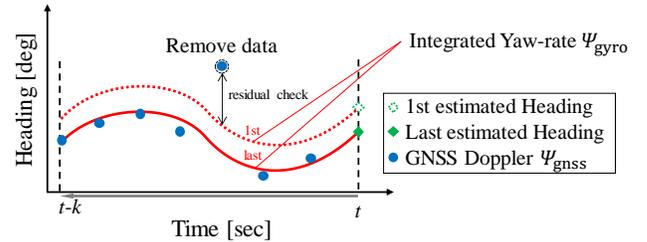


Figure 5 Heading angle estimation with outliers removed

On the other hand, the yaw rate measured by the IMU contains an error. If the raw yaw rate values are accumulated over a long period of time, there may be a cumulative error. Therefore, it is necessary to accurately estimate and compensate for the amount of yaw rate error. The yaw rate error is produced by the bias offset of the IMU. This amount of offset is estimated by approximating the differences between the long time (several minutes) yaw rate integration value and the estimated heading angle Ψ_{init} (Figure 6). Estimating the yaw rate error allows for long time integration. Therefore, even when GNSS cannot be received for a long period of time, it is expected to have the effect of keeping the accuracy of the heading angle interpolated by the yaw rate at a high level of accuracy.

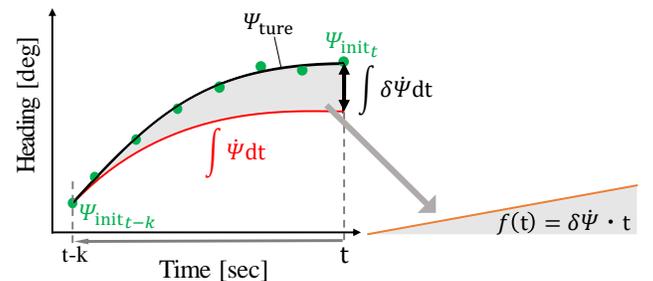


Figure 6 Yaw rate error estimation by temporal approximation

D. Side slip angle estimation

When a vehicle is turning, there is a gap between the direction of the vehicle and the angle of the tires, called the sideslip angle. Due to this sideslip angle, the moving direction of the vehicle, which is important for position estimation, cannot be estimated correctly. The IMU can measure the direction in which the vehicle is facing. Therefore, it is necessary to estimate the sideslip angle to improve the accuracy of position estimation. Conventional methods [28, 29] set up an observer to measure the parameters needed to estimate the sideslip angle. However, measuring all of them accurately is difficult.

The Eagleye algorithm for estimating the sideslip angle uses a two-wheel vehicle model to simplify the

problem. According to the two-wheel vehicle model, the sideslip angle β has the relationship shown in equation (5).

$$\beta = -\frac{mL_f}{2LK_r} \dot{\Psi}V \quad (5)$$

where m is the mass of the vehicle, L_f is the distance between the front wheel axle and the vehicle's center of gravity, L is the distance between the front wheel axle and the rear wheel axle, K_r is the cornering power of the rear wheels and V is the speed of the center of gravity. Even with the two-wheel vehicle model, a large number of parameters are required to obtain the sideslip angle, as shown in Equation (5).

Notice here that the GNSS Doppler is the relative velocity between the satellite and the receiver. In other words, the velocity vector from GNSS Doppler indicates the direction of travel of the vehicle. Therefore, when the vehicle turns, the difference between the heading angle Ψ_{gnss} from the GNSS Doppler and the heading angle Ψ_{gyro} from the yaw rate integration becomes the sideslip angle. (see Figure 7 and Equation 6).

$$\beta = \Psi_{gyro} - \Psi_{gnss} \quad (6)$$

Some parameters in equation (5) are vehicle-specific parameters and can be considered to change little dynamically. Therefore, we define the vehicle-specific parameters as coefficients K as follows:

$$K = -\frac{mL_f}{2LK_r} \quad (7)$$

From equation (7), if K can be obtained, the sideslip angle can be estimated. Using equation (5) to equation (7), it can be obtained as follows:

$$\begin{aligned} \beta &= \Psi_{gyro} - \Psi_{gnss} = K\dot{\Psi}V \\ K &= \frac{\Psi_{gyro} - \Psi_{gnss}}{\dot{\Psi}V} \end{aligned} \quad (8)$$

However, each parameter in the right-hand side of equation (8) may contain errors. Therefore, in order to reduce these errors, the least-squares method is used to estimate the coefficient K . If the coefficient K can be estimated accurately, it is possible to estimate the sideslip angle when there is no GNSS reception.

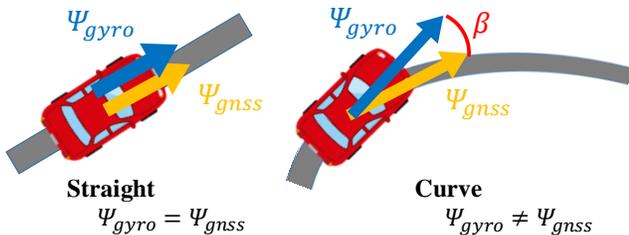


Figure 7 Appearance and relationship of sideslip angle generation

E. Relative and absolute position estimation

Using the parameters estimated in sections B to D, it is possible to calculate the velocity vector of the vehicle. By accumulating these velocity vectors, it is possible to estimate the relative position the vehicle has traveled. In Eagleye, this velocity vector is called the vehicle motion vector, and the relative position is called the vehicle trajectory. The vehicle trajectory can be expressed as follows, decomposed into east and north directions.

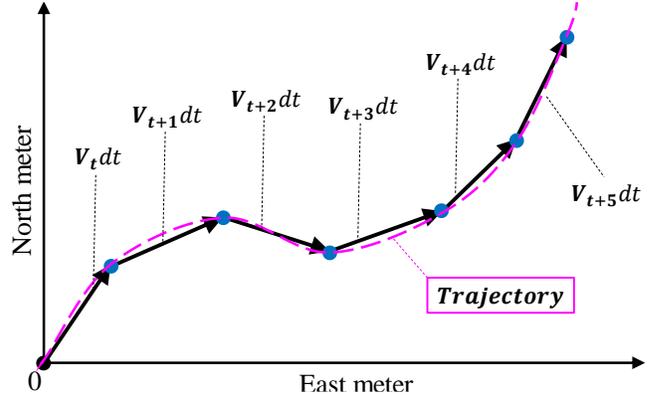


Figure 8 Overview of calculate trajectory

$$T_e = T_{e_{init}} + \sum \{SF \cdot V_{wheel} \cdot \cos(\Psi + \beta) \cdot dt\} \quad (9)$$

$$T_n = T_{n_{init}} + \sum \{SF \cdot V_{wheel} \cdot \sin(\Psi + \beta) \cdot dt\} \quad (10)$$

Where T_e is the vehicle trajectory in the east direction, T_n is the vehicle trajectory in the north direction, SF is the scale factor of wheel speed, V_{wheel} is the wheel speed, Ψ is the estimated heading angle, and β is the estimated sideslip angle. The vehicle trajectory estimated by equations (9) and (10) can be estimated with a performance of 0.5 m per 100 m.

The next section describes absolute position estimation by integrating GNSS positioning results and vehicle trajectory. The Eagleye algorithm uses the exact shape of the vehicle trajectory to estimate the current position using that shape as a constraint. The algorithm estimates the position in a similar way to the heading angle. The integrated position of the GNSS position result and the vehicle trajectory is estimated by the following equation.

$$B_e = \operatorname{argmin} \sum (P_e - T_e)^2 \quad (11)$$

$$B_n = \operatorname{argmin} \sum (P_n - T_n)^2 \quad (12)$$

where P_e, P_n is the GNSS position, T_e, T_n is the trajectory positions and B_e, B_n is the estimated position. The vehicle trajectory is integrated with the GNSS position using the least-squares method. The vehicle trajectory and GNSS position are compared, and the results with large residuals are removed as multipath. Again, the trajectory and GNSS positioning results are integrated and the residuals are compared. This process is repeated until the maximum value of the residuals is less than the threshold value, which indicates that the outlier removal

from the GNSS positioning results has been completed, and the result is the final position estimation result.

Figure 9 shows the location estimation using real data. Figure 9 shows the result of applying the proposed method to an environment with many high-rise buildings around Nagoya Station. In such an environment, the GNSS positioning results in Figure 9a) deviate greatly from the actual route and integrating these positioning results and trajectories results in a position estimation that deviates from the actual route. In Figure 9b), however, it can be confirmed that the position estimation is close to the actual driving route by using the shape of the trajectory as a constraint and selecting the GNSS positioning results.

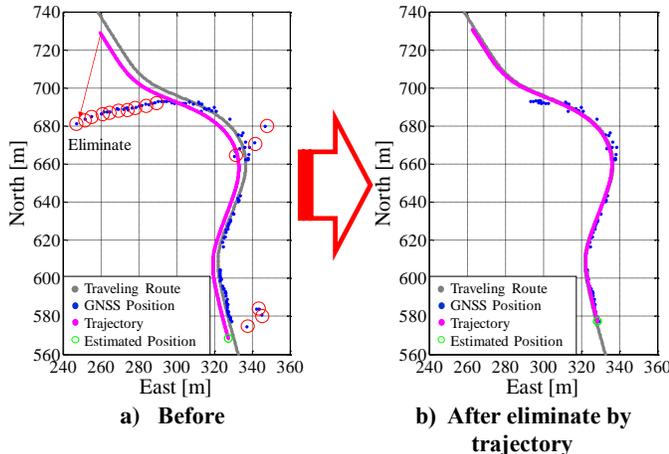


Figure 9 Integration of vehicle trajectory and GNSS for position estimation

IV. EVALUATION TESTS

A. Outline of evaluation testing

In order to confirm the effectiveness of the proposed method, we conducted evaluation tests in various environments. For the evaluation test, we will use the open dataset made available by Meijo University [30]. According to Reference [30], Table 1 shows the list of sensors used in the evaluation tests, and Figure 10 shows the vehicle used in the tests. The sensors used in the evaluation tests were low-cost ones, and the GNSS receiver was a U-blox F9P with a reception cycle of 5 [Hz]. For the MEMS IMU, Tamagawa Seiki's MEMS IMU AU7684 was used with an acquisition period of 50[Hz]. As a comparison, the POSLV220 was used as an expensive GNSS/IMU system equipped with a 3-axis fiber optic gyro and a GNSS receiver for surveying. The post-processing results of the POSLV are used as the true value in the evaluation. The real-time results of the POSLV under the same conditions as those of the proposed method are used for comparison in the evaluation.

Table 1 Equipment used for evaluation [30]

Equipment	Manufacturer	Model(cycle)
GNSS Antennas	Aero	AT1645-540T
GNSS receiver	U-blox	F9P
IMU	Tamagawa	AU7684
Speed	Toyota Sienta	CAN
Reference	Applanix	POSLV220



Figure 10 Experimental car exterior [30]

The evaluation test consists of relative position evaluation and absolute position evaluation items. First, the relative position evaluation verifies the performance of the estimated vehicle trajectory. The evaluation method is to calculate the error in 100m dead reckoning. Figure 11 shows an overview of the evaluation test. First, the initial position is adjusted to the position result of the POSLV post-processing. Then, a dead reckoning of 100 m is performed, and the difference between the final position and the POSLV post-processing position is defined as the error. This is done every 10 meters during one lap of the course, and the statistic of the error is evaluated as the achievement rate of the cumulative frequency distribution. Dead reckoning is performed using the real-time results of the vehicle trajectory and POSLV in the proposed method, and the raw values of the IMU. In order to confirm the effectiveness of the proposed method in estimating the lateral slip angle, we evaluate the method with and without the lateral slip angle.

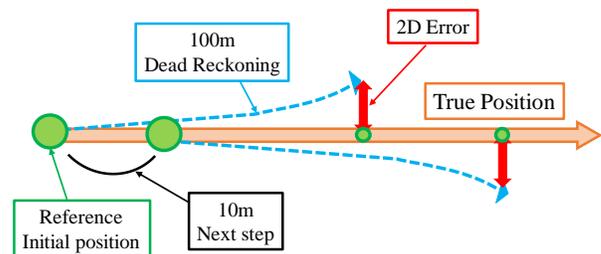


Figure 11 Overview of relative position evaluation

Next, in the absolute position evaluation, the difference between the estimated position result and the POSLV post-processing result is defined as the error. In the same way as in the relative position evaluation, the absolute position evaluation is also performed by evaluating the statistics of the error as the achievement rate of the cumulative frequency distribution. The evaluation items are the proposed method and the real-time position estimation results of POSLV. Since the POSLV, which is a comparison target, uses a single positioning method, the proposed method also uses a single positioning method to match the conditions. In addition, the proposed method uses DGNSS positioning with correction information for position estimation and evaluation. Figure 12 shows the locations where the evaluation tests were conducted. Two test courses (14.0 km and 12.5 km) were conducted around Odaiba, Tokyo.

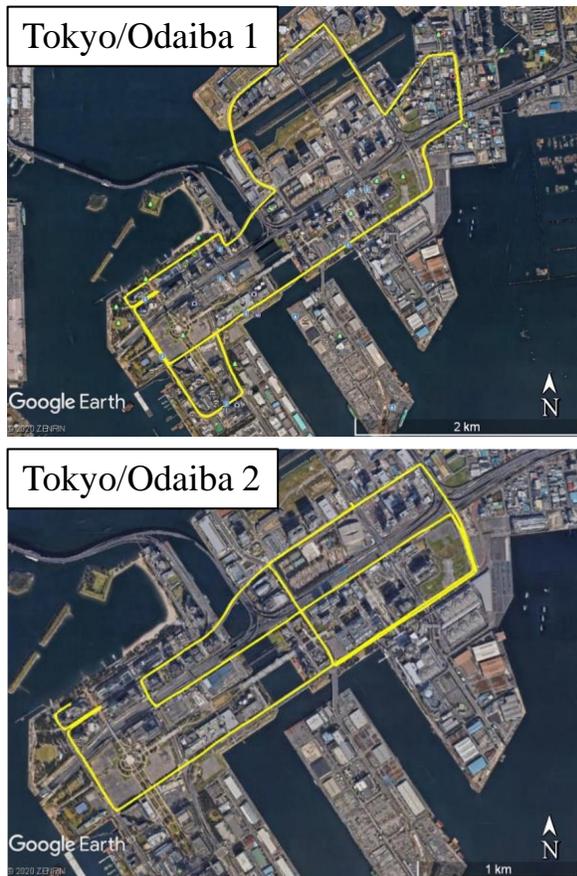


Figure 12 List of courses used in the evaluation test

B. Relative position evaluation test

Figure 13 shows the results of relative position evaluation. In Figure 13, comparing the proposed method (Propositional) with the one using the raw IMU values (IMU Raw), the accuracy is significantly improved in both courses. In addition, when comparing the proposed method with the real-time results of POSLV (POSLV RT), it can be confirmed that the same level of accuracy is obtained. From these two comparisons, it can be confirmed that the proposed method provides the same performance as the expensive IMU used in the POSLV with the correction effect of the low-cost IMU.

Next, we discuss the trajectory performance of the proposed method. First, it is confirmed that the estimation accuracy is improved by 5% to 10% by estimating the lateral slip angle in the proposed method. The trajectory of the proposed method can be estimated with an accuracy of 0.5 m per 100 m in 80% of the cases, and with an accuracy of 1.0 m in 95% of the cases. Therefore, the evaluation tests show that the proposed method can estimate the trajectory with high accuracy even in an urban multipath environment.

C. Absolute position evaluation test

Figure 14 shows the results of the absolute position evaluation. Figure 14 shows the comparison between the proposed method (Proposal Single) and the real-time results of POSLV (POSRV RT Single). In both results, the

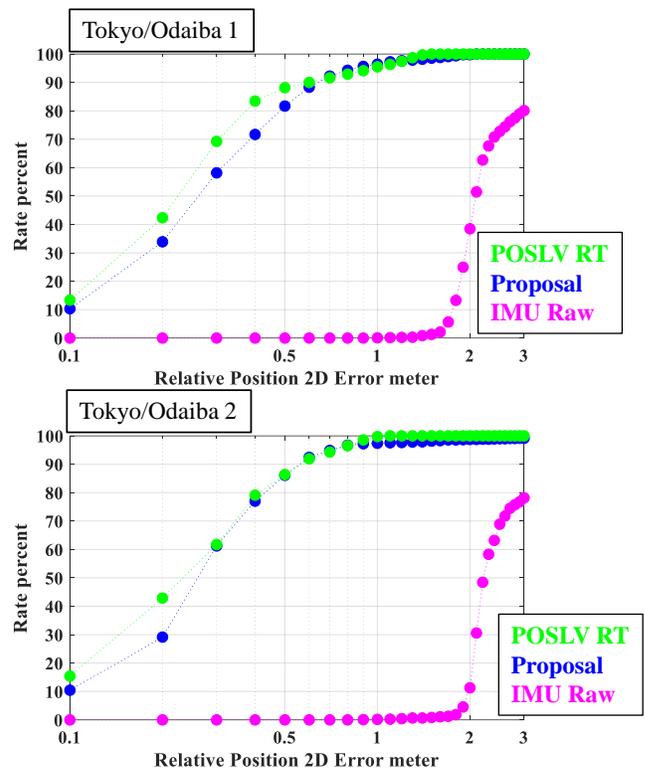


Figure 13 List of results of relative position evaluation

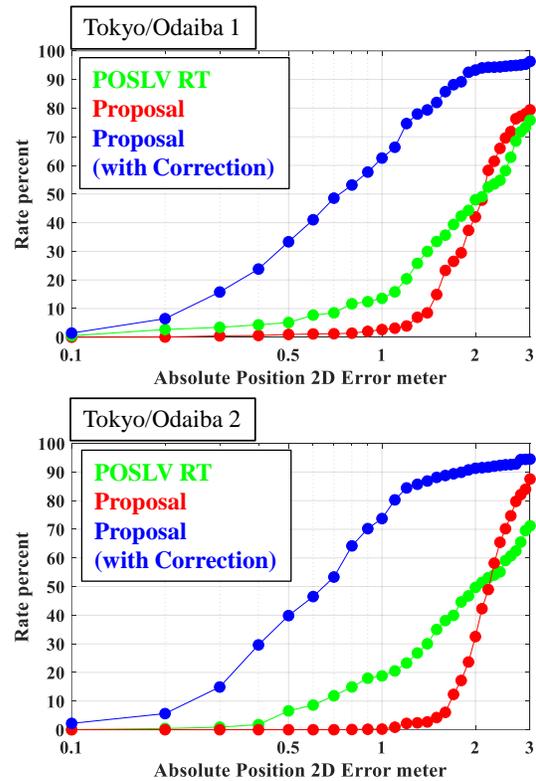


Figure 14 List of results of absolute position evaluation

achievement rate of 1.5 m accuracy is low, indicating that lane level positioning accuracy has not been achieved. Figure 15 shows a part of the estimation results around Odaiba, Tokyo.

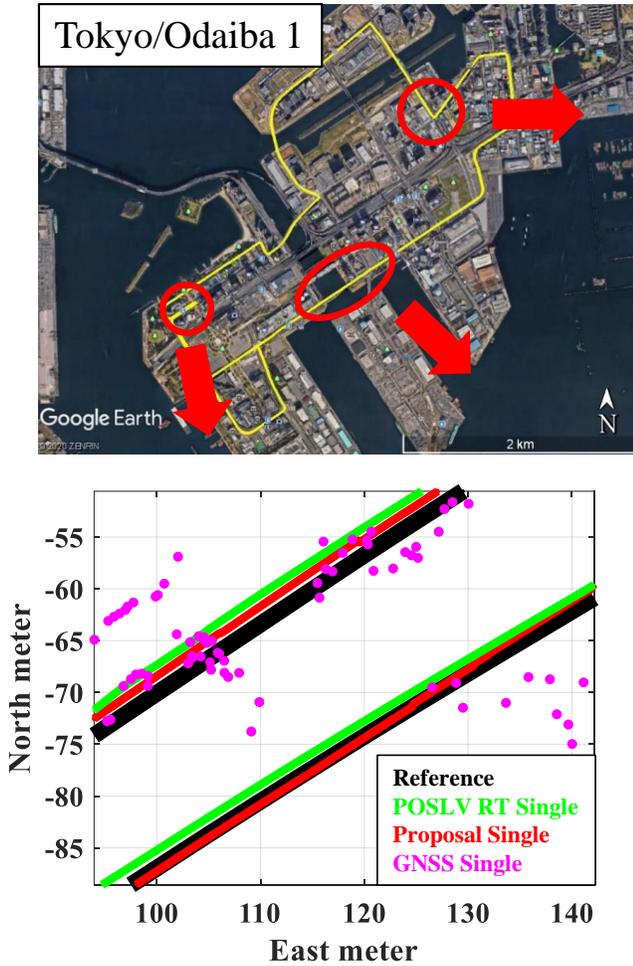


Figure 15 Some of the results of position estimation in the Odaiba area

Figure 15 shows that the proposed method does not have any outlier errors while the single positioning method has errors due to multipath. However, it can be seen that the proposed method and POSLV have an offset error from the true value. This may be because the single positioning method used for position estimation has an error of several meters due to the ionosphere and troposphere. The accuracy of the proposed method and POSLV was not improved due to the influence of the error. In the proposed method, the ionospheric and tropospheric errors are removed by using the correction information and the DGNS positioning results are re-estimated (Figure 13: Proposed with Correction). By using the correction information, we can confirm that the accuracy has been improved in all courses. In this test conducted in the Odaiba area of Tokyo, the ionosphere and troposphere had a large impact on the stand-alone positioning results, resulting in large errors. In fact, the GNSS positioning results were found to be offset by about 1.5 m to the north-northwest on average. Therefore, if the POSLV is also measured using the correction information, the same effect as that obtained by the proposed method can be obtained and the accuracy can be improved. In addition, it can be confirmed that 85% to 90% of the POSLV positions are estimated with 1.5 m accuracy, which is the lane level, in all courses using the proposed method. The performance of the proposed method is similar

in suburban and urban areas, indicating that it reduces the effect of multipath as well as relative position evaluation.

V. CONCLUSION

In this paper, we propose a method for accurate and robust position and attitude angle estimation in an urban environment. This method is characterized by the fact that it does not perform a global optimization to estimate the position and attitude angle simultaneously but integrates them in a compatible combination. In addition, by using tens to hundreds of seconds of data for these estimations, it is possible to remove multipaths received by the satellite. As a result, the proposed method can achieve accurate and robust position and attitude estimation even in urban areas using low-cost sensors.

In the evaluation test, we verified the effectiveness of the proposed method. In addition, we compared the real-time performance of the proposed method with that of a POSLV equipped with expensive sensors. In the relative position evaluation, it was confirmed that the proposed method could estimate the position with the same performance as the real-time performance of the POSLV with the correction of the low-cost IMU. In absolute position estimation, the performance of the proposed method is comparable to the real-time performance of the POSLV. In absolute position estimation, we confirmed that the proposed method can

estimate the position at the lane level with an accuracy of 1.5 m (85% to 90%) by using correction information in GNSS positioning. The results of the evaluation tests show that the system is capable of low-cost and high-accuracy position estimation even in an urban environment. We believe that this location estimation system, Eagleye, will be more widely applied to vehicles such as self-driving cars and mapping systems.

In the future, we are considering introducing this Eagleye algorithm into Autoware. We believe that the introduction of the Eagleye algorithm will help to increase the accuracy and reliability of the location estimation system. For example, by using Eagleye's relative trajectory, which is more accurate than the vehicle speed and attitude angle used as input values in Autoware's Extended kalman filter, we can improve the robustness and accuracy of the system. Also, in recent years, RTK-GNSS has become readily available. We would like to further improve the estimation performance of the Eagleye algorithm by introducing RTK-GNSS. In addition, not only real-time estimation, but also post-processing can be introduced to improve the accuracy, which is expected to be utilized for map generation using SLAM.

ACKNOWLEDGEMENT

The test was conducted using an open dataset available at Meijo University [30].

REFERENCES

- [1] Waymo, Self-Driving Car, [online] Available: <https://waymo.com/>
- [2] G., Ros, et al., "Vision-based offline-online perception paradigm for autonomous driving", Proc. IEEE WACV, pp. 231-238, 2015.
- [3] J., Levinson, et al., "Towards fully autonomous driving: Systems and algorithms", Proc. IEEE IV, pp. 163-168, 2011
- [4] Benjamin, C., H, and Dennis, F. and Wuensche, H., J., "Precise Object-Relative Positioning for Car-Like Robots", IEEE ITSC 2016
- [5] Liu, J., Cai, B., Wang, J., "Track-constrained GNSS/Odometer-based Train Localization using a Particle Filter", IEEE IV 2016
- [6] Jiali, B., Yanlei, G, and Shunsuke, K., " Vehicle positioning with the integration of scene understanding and 3D map in urban environment", IEEE IV 2017
- [7] Farouk G., Ghayath EL-HAJ-SHHADE, Marie-Anne MITTET, Fawzi Nashashibi, "LIDAR-Based road signs detection For Vehicle Localization in an HD Map", IEEE IV 2019
- [8] Ahmed, H., Pablo, M., P., Fernando, G. and José, M. A., Autonomous Cooperative Driving Using V2X Communications in Off-Road Environment, Proc. of IEEE ITSC 2017.
- [9] J. Ziegler, et al, "Making Bertha Drive – An Autonomous Journey on a Historic Route", IEEE Intelligent Transportation Systems Magazine, Vol. 6, no. 2, pp. 8-20, 2014.
- [10] S. Thrun, W. Burgard, D. Fox (2005). Probabilistic robotics. Cambridge, Mass.: MIT Press. ISBN: 0262201623 9780262201629.
- [11] E. Takeuchi and T. Tsubouchi. "A 3-D scan matching using improved 3-D normal distributions transform for mobile robotic mapping." IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3068-3073, 2006.
- [12] M. Magnusson, A. Lilienthal, and T. Duckett. "Scan registration for autonomous mining vehicles using 3D-NDT." Journal of Field Robotics 24.10 (2007): 803-827
- [13] Sinpei, K., Shota, T., Yuta, M., Seiya, M., Manato, H., Yuki, K., Abraham, M.m Tomohito, A., Yusuke, F. and Takuya, A., "Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems", in Proc. of ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS), 2018.
- [14] S. Kato, E. Takeuchi, Y. Ishiguro, Y. Ninomiya, K. Takeda, and T. Hamada. "An open approach to autonomous vehicles." IEEE Micro, 35(6), 60-68, 2015.
- [15] Eagleye,[Online] Available:<https://github.com/MapIV/eagleye>
- [16] Junichi, M., Takuya, A., Syunsuke, M. and Aoki, T., "Low-cost Lane-level positioning in Urban Area Using Optimized Long Time Series GNSS and IMU Data", IEEE ITSC,2018
- [17] Aoki, T., Kanamu, T., Takuya, A. and Junichi, M., "Improvement of RTK-GNSS with Low-Cost Sensors Based on Accurate Vehicle Motion Estimation Using GNSS Doppler", Proc. IEEE IV, pp. 163-168, 2020
- [18] POSLV, High-precision location estimation system, [online] Available: <https://www.applanix.com/products/poslv.htm>
- [19] Odolinski, R.; Teunissen, P.J.G. Low-cost, high-precision, single-frequency GPS–BDS RTK positioning. GPS Solut. 2017, 21, 1315–1330.
- [20] Teunissen, P.J.G.; Odolinski, R.; Odijk, D. Instantaneous BeiDou+GPS RTK positioning with high cut-off elevation angles. J. Geod. 2013, 88, 335–350.
- [21] Odolinski, R.; Teunissen, P.J.G.; Odijk, D. Combined BDS, Galileo, QZSS and GPS single-frequency RTK. GPS Solut. 2014, 19, 151–163.
- [22] H., Eric, Tseng, et al., Estimation of Land Vehicle Roll and Pitch Angles, International Journal of Vehicle Mechanics and Mobility, vol. 45, 2007.
- [23] Song, J.W.; Park, C.G. Enhanced pedestrian navigation based on course angle error estimation using cascaded Kalman filters. Sensors 2018, 18, 1281, doi:10.3390/s18041281.
- [24] Falco, G.; Einicke, G.A.; Malos, J.T.; Dovis, F. Performance analysis of constrained loosely coupled GPS/INS integration solutions. Sensors 2012, 12, 15983–16007.
- [25] Qifen, L.; Lun, A.; Junpeng, A.; Hsu, L.T.; Kamijo, S.; Gu, Y. Tightly coupled RTK/MIMU using single frequency BDS/GPS/QZSS receiver for automatic driving vehicle. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS), Monterey, CA, USA, 23–26 April 2018.
- [26] Rabbou, M.A.; El-Rabbany, A. Tightly coupled integration of GPS precise point positioning and MEMS-based inertial systems. GPS Solut. 2015, 19, 601–609, doi:10.1007/s10291-014-0415-3.
- [27] Kojiro, T., Yoshiko, K. and Eiji, T., "Trajectory Estimation Improvement Based on Time-series Constraint of GPS Doppler and INS in Urban Areas" ,Proc. of IEEE/ION PLANS 2012, 700/705,(2012).
- [28] Kanwar, Bharat, Singh, "Vehicle Sideslip Angle Estimation Based on Tire Model Adaptation",electronics 2019,8,199
- [29] Selmanj, D., Matteo, C., Giulio, P., Sergio M.Savaresi,"Vehicle sideslip estimation A kinematic based approach" Control Eng. Pract. 2017, 67, 1–12.
- [30] Open Data Set: [online] Available: https://github.com/MeijoMeguroLab/open_data