1st Place Winner of the Smartphone Decimeter Challenge: Two-Step Optimization of Velocity and Position using Smartphone's Carrier Phase Observations

Taro Suzuki, *Chiba Institute of Technology, Japan*

BIOGRAPHY

Taro Suzuki is a chief researcher at Chiba Institute of Technology, Japan. He received his B.S., M.S., and Ph.D. in Engineering from Waseda University in 2007, 2009, and 2012 respectively. From 2012 to 2014, he worked as a postdoctoral researcher at Tokyo University of Marine Science and Technology. From 2015 to 2019, he worked as an assistant professor at Waseda University. His current research interests include GNSS precise positioning in urban environments.

ABSTRACT

This paper describes a method that was declared as the winner of the Google smartphone decimeter challenge 2022 (GSDC2022), conducted on the competition platform Kaggle from May 3 to July 30, 2022. GSDC2022 was conducted to estimate the driving trajectory of a vehicle based on the raw global navigation satellite system (GNSS) data from smartphones. The GNSS data from smartphones have lower signal levels and higher noise compared to commercial GNSS receivers. Consequently, direct application of existing high-precision positioning methods, such as precise point positioning and real-time kinematic GNSS, is challenging. This study devised a two-step optimization method to optimize the velocity and position of a smartphone. In the first step, the states to be optimized were 3D velocity and receiver clock drift, which were estimated through the framework of factor graph optimization (FGO) using GNSS Doppler observations. The outliers of the optimized velocity were excluded. Further, the sections where the velocity were not obtained were interpolated to estimate the velocity of the entire driving trajectory. In the next step, the states to be optimized were 3D position and receiver clock bias, and the velocity and clock drift obtained in the previous step were used as loose constraints between the states. In addition, the time-differenced carrier phase (TDCP) was used as a high-precision relative position constraint, and an error-corrected pseudorange using a GNSS base station was added as an absolute position constraint. Following the implementation of the proposed method and its evaluation performed at the competition, the final score, mean(50%, 95%), was 1.382 and 1.229 m for public and private, respectively, which won the first place at GSDC2022.

I. INTRODUCTION

Smartphones have become indispensable for pedestrian and vehicle navigation applications. They are equipped with a global navigation satellite system (GNSS) and an inertial measurement unit (IMU), which are combined to estimate location. With continuous development, GNSS chipsets used in smartphones can now support multiple frequencies and multiple GNSS, and the availability and accuracy of GNSS positioning has been improving. Although smartphones have long been equipped with GNSS, users have only been able to access the location information output by GNSS receivers. In 2016, the Android operating system released an application program interface to access the raw GNSS measurement data from GNSS chipsets installed in smartphones. Consequently, raw GNSS data (pseudorange, pseudorange rate (Doppler), and accumulated delta range (carrier phase)), could be acquired. This facilitated the development of position estimation algorithms for smartphones using raw GNSS measurements.

However, there exist many challenges associated with smartphone positioning compared with the positioning using commercial GNSS receivers. The GNSS antennas of smartphones exhibit lower performance than those of GNSS receivers used for surveying, and the noise of GNSS observations is very large. Consequently, the application of existing high-precision positioning methods to smartphones, such as precise point positioning and real-time kinematic GNSS, is challenging. The usual positioning accuracy of the smartphone is approximately 3 to 10 m, which results in significant challenges for advanced navigation and other applications such as a sidewalk and lane-level navigation for pedestrians and vehicles.

To overcome this situation, in 2021, the Google smartphone decimeter challenge (GSDC) was held on the international competition platform Kaggle. In 2022, the Google smartphone decimeter challenge 2022 (GSDC2022) was held again from

May 3 to July 30, 2022, using a new dataset based on the smartphone's raw GNSS measurements. This paper describes the winning solution at GSDC2022.

II. GSDC2022 OVERVIEW

At GSDC, each dataset included the raw GNSS measurements collected by the several Android smartphone devices, together with the ground truth trajectories collected by the high-grade GNSS and inertial navigation unit (IMU) integration system for reference. Figure 1 shows a picture of GNSS data collection at GSDC [1]. The GNSS logs of multiple smartphones mounted on the dashboard of a vehicle were provided in two datasets: training data with reference position and test data for evaluation in the competition. The smartphone logs included observations such as GNSS pseudorange, pseudorange rate, and accumulated delta range, etc. Further, the smartphone's IMU data, such as acceleration and angular velocity, were also provided.

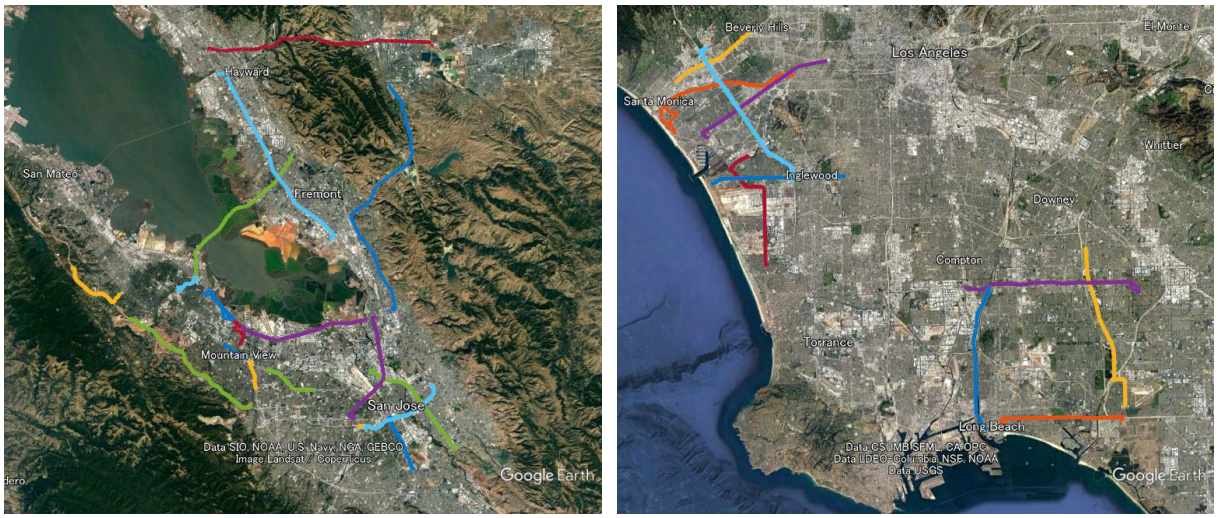
Figure 1: GNSS data collection in Google smartphone decimeter challenge. Multiple smartphones are placed on the dashboard of the vehicle to acquire raw GNSS data [1].

Driving data for five types of smartphones (Google Pixel4, Google Pixel5, Google Pixel6, Samsung Galaxy S20 Ultra, and Xiaomi Mi8) were provided at the GSDC2022. While at GSDC2021, the driving data for the San Francisco area was provided, at GSDC2022, that for the Los Angeles area was added. Figure 2 shows the GSDC2022 data from the 36 runs included in the test dataset. At GSDC2021, each run included data acquired simultaneously by multiple models of smartphones; however, at GSDC2022, only one type of smartphone was used for each run. This implies that the technique of improving accuracy by assembling the positions estimated independently for each smartphone was no longer available.

The driving data provided can be roughly divided into two categories: highway driving in an open-sky environment and street driving in an area lined with trees and buildings. The Los Angeles driving data included several long tunnels and elevated sections, including several 10-s GNSS signal blockages. In approximately half of the runs included in the test data, the data were run on different courses that were not included in the training data.

III. STRATEGY

At GSDC2021, the author proposed a location estimation method based on smartphone observations using global optimization with factor graph optimization (FGO) [2, 3]. In this study, the method developed in 2021 was modified to improve the location estimation accuracy, particularly in environments where the accuracy of pseudorange deteriorates under trees and elevated structures. The pseudorange of smartphones is noisy, and achieving highly accurate position estimation using the pseudorange alone is challenging. However, if the carrier phase can be tracked continuously, the relative position change (velocity) can be estimated with high accuracy from the time-differenced carrier phases (TDCP). Because of the limited availability of valid TDCPs, velocity can be estimated from more robust Doppler observations, although they are less accurate than TDCP. However, Doppler observations remain unavailable when satellites are shielded for long periods of time, such as in elevated structures or tunnels.



(a) San Francisco area (b) Los Angeles area

Figure 2: Driving trajectories included in test data provided by GSDC 2022. A total of 36 runs were provided, divided into two parts: one run in the San Francisco area and another in the Los Angeles area.

(a) Test/2022-04-01-US-LAX-3/]XiaomiMi8 (b) Test/2022-02-23-US-LAX-3/XiaomiMi8

Figure 3: Los Angeles vehicle driving trajectory included in Google smartphone decimeter challenge 2022. (a) stops under elevated tracks and (b) runs through long tunnels where GNSS is completely blocked.

Figure 3 shows a section of the Los Angeles travel path in the test data that stops under an elevated railway and travels in a tunnel. As evident, there are certain sections where the use of Doppler is difficult, resulting in large outliers and missing data in the velocity estimates.

Therefore, in this study, a two-step optimization was used to estimate the position of a smartphone. Figure 4 shows the flow of the proposed method. The first step was to estimate the velocity, where the 3D velocity of the smartphone and receiver clock drift were estimated employing FGO using Doppler observations, which although less accurate are readily available. From the 3D velocity estimated by the optimization, outliers in velocity were detected, excluded, and interpolated to obtain an estimate of the 3D velocity at all times. Thereafter, the states to be optimized were 3D position and the receiver clock bias of each satellite system. The 3D velocity and clock drift obtained in the previous step were used as loose constraints between the states in this

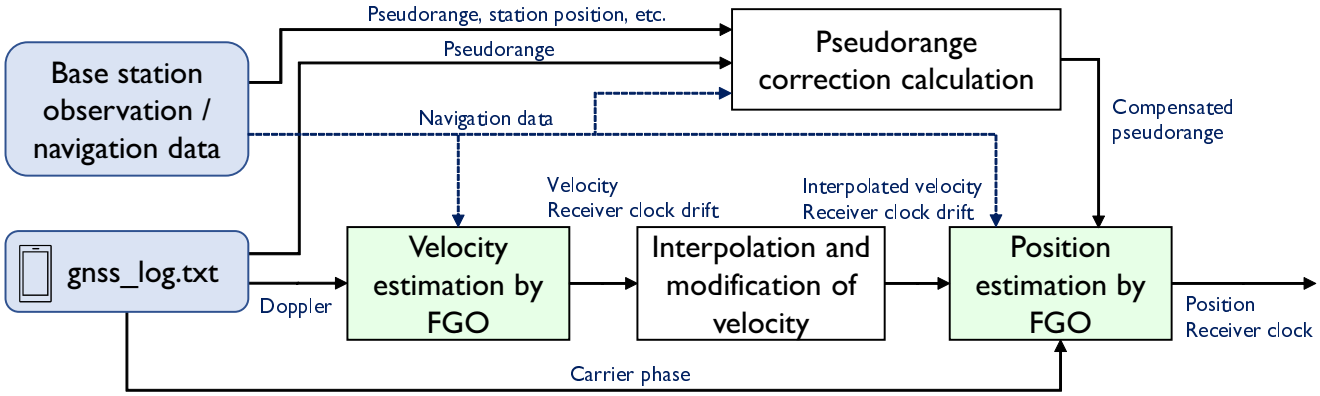


Figure 4: Flow of the proposed method. It consists of two optimization steps: velocity estimation and position estimation by factor graph optimization (FGO).

step. In addition, if valid TDCPs were obtained, TDCP-based constraints between the states were added, and as an absolute position constraint, the pseudorange of the smartphone, whose error was corrected using the GNSS base station, was used. Even when no TDCP is available, this method added a state-to-state constraint based on the velocity estimated using Doppler in the first step, and the position could be obtained even in the tunnel.

IV. RELATED RESEARCHES

The core of the proposed approach was to use a global optimization method based on FGO [4]. FGO can apply a variety of complex nonlinear constraints and simultaneously optimize all state variables (the entire driving trajectory). A factor graph is a graphical representation wherein there is an unknown state variable (variable nodes) and a factor (factor nodes) that is a function of the state variable; the edges connecting the factor and variable nodes can be considered as constraints on the state variable by the factor. Thus, the state estimation problem is reduced to an optimization problem using the objective function constructed by the factor graph representation.

Graph-based optimization has been extensively studied in the field of robotics; however, it has recently been actively studied in the GNSS field as well. The factor graph was proposed to model factorizations [4]. One of the earliest studies proposed a robust optimization method using only GPS pseudorange observations based on factor graph optimization [5], which improved in positioning accuracy compared to least-squares-based positioning. Subsequently, extensions to real-time position estimation [6] and combinations with IMU and other sensors have been studied [7–9]. In [10], a comparative evaluation of GNSS positioning based on graph-based optimization with that based on least-squares-based positioning and extended Kalman filter was discussed. The study by [11] implemented precise point positioning using carrier phase measurements with graph optimization and compared it with Kalman filter based implementation. The results indicated that factor graph-based optimization exhibited better performance compared to the commonly used filtering methods such as an extended Kalman filter. This is owing to the global optimization using all the observations from past to present in the graph structure.

V. VELOCITY ESTIMATION STEP

In the velocity estimation step, the state $\mathbf{V}$ estimated in epoch i was the 3D velocity and receiver clock drift. The state $\mathbf{V}$ can be expressed as:

$$\mathbf{V}_i = \begin{bmatrix} \mathbf{v}_i \\ \dot{\mathbf{t}}_i \end{bmatrix} \tag{1}$$

$$\mathbf{v}_i = \begin{bmatrix} v_{x,i} & v_{y,i} & v_{z,i} \end{bmatrix}^T \quad \dot{\mathbf{t}}_i = \dot{t}_{\mathrm{gps},i} \tag{2}$$

where $\dot{t}_{\mathrm{gps},i}$ is the receiver clock drift with respect to the GPS time in m/s. The 3D velocity $\mathbf{v}_i$ is represented in the earth-centered earth-fixed (ECEF) coordinate system. Here, only the GPS receiver clock drift was included in the state because the inter-system bias, including the receiver clock bias, between GPS and other satellite systems can be considered constant.

The graph structure of the proposed FGO is shown in Figure 5. The circled markers in the figure represent the factor nodes,

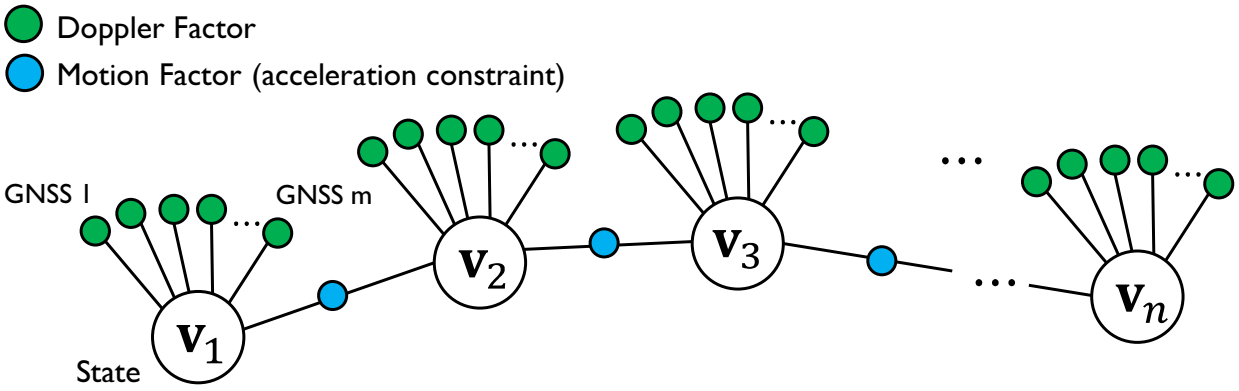


Figure 5: Graph structure of proposed method in velocity estimation step. We use the Doppler factor and motion factor.

where the variable nodes are connected to the Doppler factors from each satellite, and the motion factor is connected between variable nodes as an acceleration constraint.

1. Doppler Factor

The observed Doppler frequency can be converted to a pseudorange rate. GNSS pseudorange rate from Doppler of satellite k in i-th epoch ρ_i^k can be modeled as follows.

$$\dot{\rho}_i^k = \left(\mathbf{v}_{\mathrm{s},i}^k - \mathbf{v}_i\right)\mathbf{u}_i^k + \dot{t}_i \tag{3}$$

where $\mathbf{v}_{\mathrm{s},i}^k$ is the satellite velocity in ECEF coordinate system. $\mathbf{u}_i^k = \left[u_{x,i}^k, u_{y,i}^k, u_{z,i}^k\right]$ is the unit line-of-sight vector from the receiver to the satellite k in ECEF.

The error function of Doppler factor is represented as follows.

$$e_{\mathrm{d},i}^k = \mathbf{H}_{\mathrm{v},i}^k \mathbf{V}_i - \left(\dot{\rho}_i^k - \mathbf{v}_{\mathrm{s},i}^k \mathbf{u}_i^k\right) \tag{4}$$

Here, the measurement matrix $\mathbf{H}_{\mathrm{v},i}^k$ can be formulated as:

$$\mathbf{H}_{\mathrm{v},i}^k = \begin{bmatrix} \mathbf{u}_i^k & 1 \end{bmatrix} \tag{5}$$

Doppler observations are less affected by multipath than pseudorange observations. Compared to carrier phase observations, Doppler observations are more robust and can be used even when carrier phase has cycle slips.

2. Motion Factor

Here the nodes are the velocity and clock bias, and the constraints between the nodes acted as acceleration constraints. Although acceleration observations can be obtained from the smartphone's IMU, they were not used in this study because they require a coordinate transformation based on the smartphone's orientation, as well as synchronization with GPS time. The error function was simply defined as a constraint on the maximum value of acceleration such that the velocity changed smoothly, as follows

$$\mathbf{e}_{\mathrm{m},i} = \mathbf{V}_{i+1} - \mathbf{V}_i \tag{6}$$

Here, by appropriately providing $\mathbf{\Omega}_{\mathrm{m},i}$, the information matrix, during optimization based on the maximum value of the actual acceleration of the vehicle, large jumps in the velocity estimated by optimization can be suppressed and a continuous velocity can be estimated.

3. Optimization

The objective function to be optimized is as follows.

$$\widehat{\mathbf{V}} = \underset{\mathbf{V}}{\operatorname{argmin}} \sum_i \|\mathbf{e}_{\mathrm{m},i}\|^2_{\mathbf{\Omega}_{\mathrm{m},i}} + \sum_i \sum_k \left\|e^k_{\mathrm{d},i}\right\|^2_{\Omega^k_{\mathrm{d},i}} \tag{7}$$

The M-estimator was used to reject the multipath error of Doppler measurements [12]. Further, the GTSAM [13] was used for graph optimization backend, and RTKLIB [14] were used for GNSS general computation.

4. Outlier Removal and Interpolation

Interpolation was performed to eliminate outliers in the estimated velocity by the FGO and to obtain velocity estimates at all epochs. The estimated velocities in the ECEF coordinate system were transformed to the east-north-up coordinate system. Further, the threshold for the estimated velocity estimates in the up direction excluded outliers as the vehicle was bound on the ground.

Furthermore, to estimate the velocity at epochs where Doppler cannot be obtained, such as in tunnels, the velocity at all epochs was interpolated from the previous and next velocity information. Subsequently, this estimated velocity was used in the next position optimization step.

VI. POSITION ESTIMATION STEP

In the position estimation step, the state $\mathbf{X}$ estimated in epoch i was the 3D position relative to the initial position in ECEF coordinates and multi-GNSS clock biases. The state $\mathbf{X}$ can be expressed as:

$$\mathbf{X}_i = \begin{bmatrix} \mathbf{r}_i \\ \mathbf{t}_i \end{bmatrix} \tag{8}$$

$$\mathbf{r}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \quad \mathbf{t}_i = \begin{bmatrix} t_{\mathrm{gps},i} & t_{\mathrm{glo},i} & t_{\mathrm{gal},i} & t_{\mathrm{bds},i} \end{bmatrix}^T \tag{9}$$

where $t_{\mathrm{gps},i}$ is the receiver clock bias of the GPS satellites, and $t_{\mathrm{glo},i}$, $t_{\mathrm{gal},i}$, and $t_{\mathrm{bds},i}$ are the system biases including the time bias of the GLONASS, Galileo, and BeiDou relative to the GPS time, respectively. The state was defined by the difference from the initial values of the 3D position and clock bias for linearization. In addition, the initial values of the 3D position and clock bias were the results of ordinary least-squares-based positioning using pseudorange observations.

Figure 6 illustrates the proposed graph structure. Three types of factors: pseudorange, TDCP, and velocity/clock drift factors were used.

1. Velocity/Clock Drift Factor

The velocity/clock drift factor is the relative constraint between time-series nodes. In the previous step, the 3D velocity and clock drift at all epochs were estimated based on Doppler observations; thus, the estimated velocity and clock drift were used as loose constraints between states in the position estimation step. If TDCP observations are not available, this velocity/clock drift factor will be the only constraint between states. The error function of the velocity/clock drift factor is denoted as:

$$\mathbf{e}_{\mathrm{v},i} = (\mathbf{X}_{i+1} - \mathbf{X}_i) - \mathbf{V}_i \Delta t \tag{10}$$

where Δt is the time step. The velocity/clock drift factor was added between all nodes, such that even in epochs where the pseudorange or TDCP was not observed at all, such as in a tunnel, the position was estimated based on the velocity estimated in the previous step.

2. Pseudorange Factor

In the pseudorange factor, the pseudorange compensated using the pseudorange error computed at a base station was used to estimate the absolute position. GNSS pseudorange of satellite k in i-th epoch ρ^k_i can be modeled as follows.

$$\rho^k_i = r^k_i + t_i - \delta T^k_i + I^k_i + T^k_i \tag{11}$$

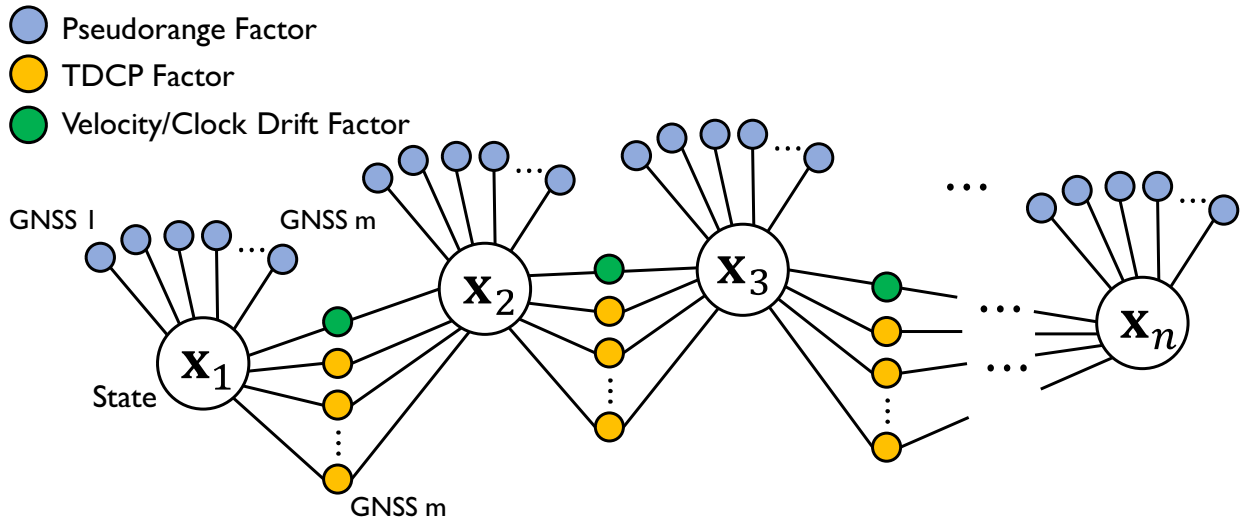


Figure 6: Graph structure of proposed method in position estimation step. We use the three types of factors, pseudorange factor, TDCP factor, and velocity/clock drift factor.

where r_i^k is the geometric satellite-to-receiver distance, which is calculated using initial node. Further, δT_i^k as the clock bias of the satellite, I_i^k is the ionospheric delay, and T_i^k is the tropospheric delay.

To completely eliminate satellite orbit, clock, tropospheric, and ionospheric delays in the smartphone pseudorange observations, the GNSS pseudorange error at the GNSS base station was calculated to correct the pseudorange. The error function of the pseudorange factor is represented as follows.

$$e_{\mathrm{pr},i}^k = \mathbf{H}_{\mathrm{p},i}^k \mathbf{X}_i - \left(\rho_i^k - r_i^k - \varepsilon_i^k\right) \tag{12}$$

where ε_i^k is the pseudorange correction value, including the tropospheric and ionospheric delay, clock error of the satellite, and base station receiver, calculated at a base station whose position is known. Here, the measurement matrix $\mathbf{H}_{\mathrm{p},i}^k$ can be formulated as:

$$\mathbf{H}_{\mathrm{p},i}^k = \begin{bmatrix} \mathbf{u}_i^k & 1 & \delta_{\mathrm{glo},i}^k & \delta_{\mathrm{gal},i}^k & \delta_{\mathrm{bds},i}^k \end{bmatrix} \tag{13}$$

where $\delta_{\mathrm{glo},i}^k$, $\delta_{\mathrm{gal},i}^k$, and $\delta_{\mathrm{bds},i}^k$ are equal to 1 when the k-th GNSS measurement is GLONASS, Galileo, or BeiDou, respectively.

3. TDCP Factor

Both Doppler and TDCP can be used to estimate the velocity. However, Doppler observations from smartphones were noisier than those from commercial GNSS receivers, and velocity computed from pseudorange rate based on TDCP measurement was much more accurate than those calculated from Doppler.

The TDCP measurement between epochs i and $i+1$ $\Delta\Phi_i^k$ is expressed as follows:

$$\lambda\Delta\Phi_i^k = \lambda\left[\Phi_{i+1}^k - \Phi_i^k\right] \simeq \Delta r_i^k + \Delta t_i \tag{14}$$

where λ is the signal wavelength, Φ_i^k is the measured carrier phase in cycles, and r_i^k is the satellite-receiver geometric distance. If the time difference is short, the ionospheric and tropospheric delays and satellite orbital clock errors in the carrier phase can be canceled. Here, $\lambda\Delta\Phi_i^k$ represents the exact receiver-satellite distance change between epochs i and $i+1$.

The error function of TDCP factor is as follows.

$$e_{\mathrm{td},i}^k = \mathbf{H}_{\mathrm{p},i}^k \left(\mathbf{X}_{i+1} - \mathbf{X}_i\right) - \left(\lambda\Delta\Phi_i^k - \Delta L_i^k\right) \tag{15}$$

Here, ΔL_i is the change in distance owing to satellite motion from the antenna position.

However, although TDCP is accurate, its availability is lower than the Doppler owing to cycle slip and half-cycle ambiguity problems. In this study, based on the carrier phase tracking status output by the smartphone, the TDCP factor was added to the graph only when the carrier phase was continuously tracked.

4. Optimization

The final objective function to be optimized, using all the factors, is as follows.

$$\widehat{\mathbf{X}} = \underset{\mathbf{X}}{\operatorname{argmin}} \sum_i \|\mathbf{e}_{\mathrm{v},i}\|^2_{\mathbf{\Omega}_{\mathrm{v},i}} + \sum_i \sum_k \left\|e^k_{\mathrm{pr},i}\right\|^2_{\Omega^k_{\mathrm{pr},i}} + \sum_i \sum_k \left\|e^k_{\mathrm{td},i}\right\|^2_{\Omega^k_{\mathrm{td},i}} \tag{16}$$

Similar to that in the velocity estimation step, robust estimation by M-estimator with Huber function was applied to the pseudorange and TDCP factors. As the pseudorange observations contain multipath errors and the TDCP observations contain many cycle slips, the M-estimator was used to exclude outliers. The hyperparameters of the Huber function were determined by trial and error. Furthermore, the smartphone position optimized in the position estimation step was directly used as the final estimate.

VII. EVALUATION USING TRAINING DATA

The training data, for which reference locations were provided, was used to evaluate the positioning accuracy of the proposed method. Three runs were selected from all training data to compute the positioning error. Figure 7 shows the driving trajectories and the positioning results, referred to as "baseline," provided by the competition organizer. This baseline was the result of least-squares-based processing of the pseudorange observed by the smartphone to estimate the position.

Figure 7(a) represents an open-sky environment with elevated tracks in places. Figure 7(b) represents driving in the city, and the GNSS signal is blocked by trees. Figure 7(c) is a run in Los Angeles, where the GNSS signal is blocked for a long period by an elevated structure. The positioning error of the proposed method was compared with that of the baseline position provided in these runs.

(a) Train/2021-03-16-US-MTV-3/ GooglePixel5

(b) Train/ 2021-03-16-US-MTV-1/ XiaomiMi8

(c) Train/ 2021-12-07-US-LAX-1/ SamsungGalaxyS20Ultra

Figure 7: Driving trajectories in each driving area for the performance evaluation of the proposed method.

Figure 8 shows the time-series position estimation error. The blue line indicates the baseline position estimation error, and the red line indicates the proposed method. In highway driving (Figure 8(a)), the TDCP factor estimated the relative position with high accuracy, and the pseudorange factor corrected the absolute position. Further, the baseline position sometimes exhibited a sudden error of approximately 10 m, which occurred when the vehicle went under an elevated road while driving on the highway. However, the proposed method suppressed the increase in error by interpolating the velocity estimated from Doppler. The final position score, mean(50%, 95%), was 0.396 m in case of highway driving, thus achieving decimeter accuracy.

Figure 8(b) shows the positioning error of street driving. The pseudorange and carrier phase noise were large, and the position estimation accuracy was worse than that of highway driving. Further, the baseline exhibited frequent position errors exceeding 10 m; however, the proposed method could estimate the position accurately. The final position score, mean(50%, 95%), was

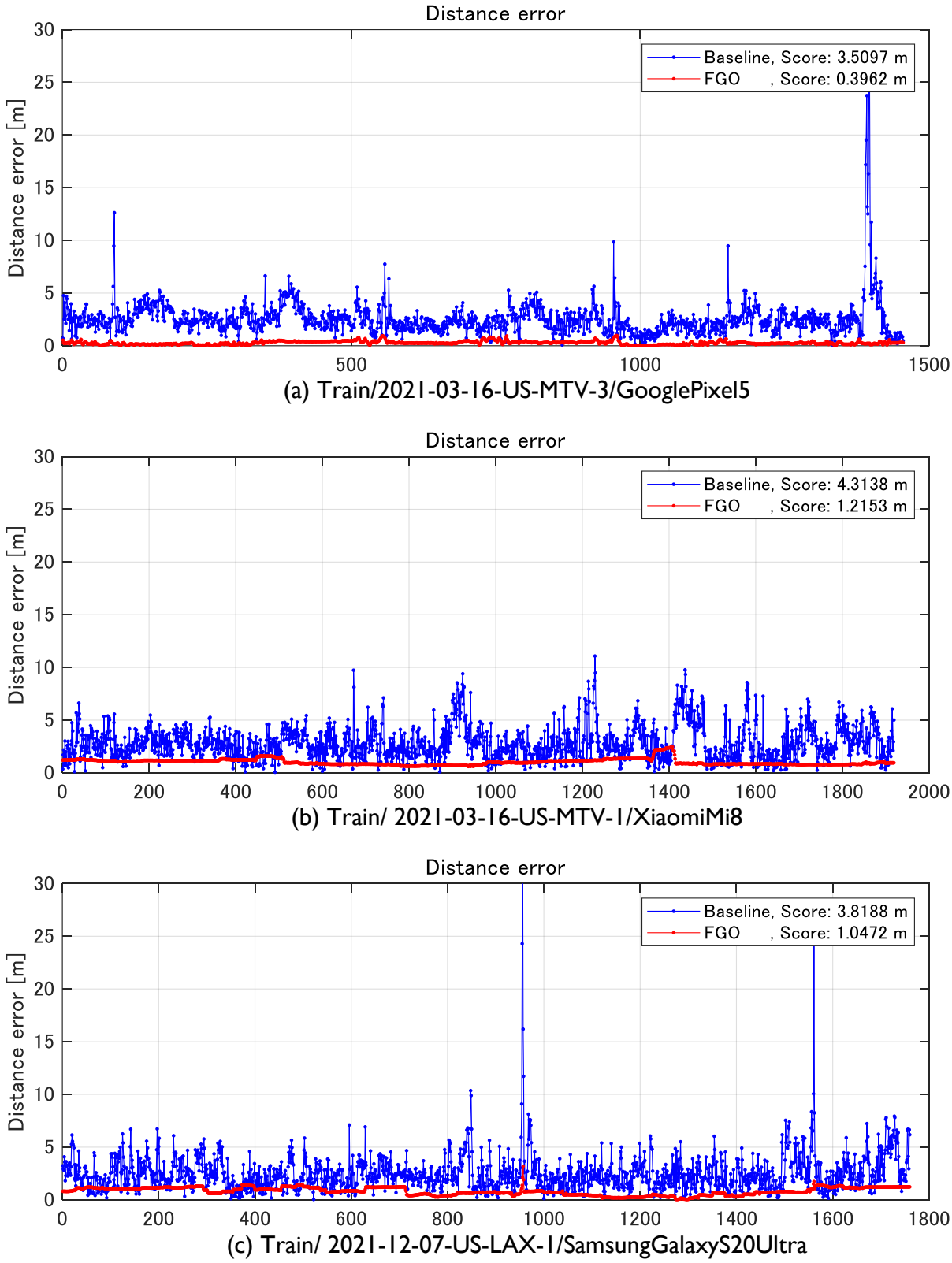


Figure 8: Comparison of positioning error between baseline and proposed method.

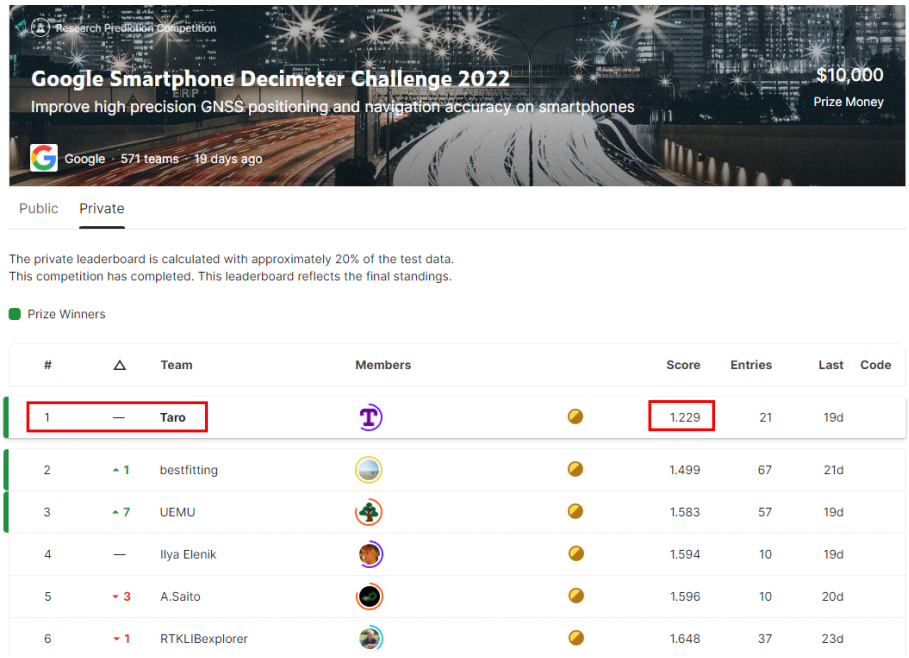


Figure 9: Final Google smartphone decimeter challenge 2022 leaderboard.

1.215 m.

Figure 8(c) shows the positioning error of driving in the Los Angeles area. Although the error increased at the point where the vehicle went under the overpass, the proposed method significantly suppressed the increase in error when compared to the baseline. The final position score, mean(50%, 95%), was 1.047 m.

These results show that the proposed method significantly improves positioning accuracy with smartphones.

VIII. EVALUATION USING TEST DATA

Using the above method, the location of the smartphone was estimated and the competition was tackled. Figure 9 shows the final leaderboard of the GSDC2022. The public score for the proposed method was 1.382 m, ranking first. The final private score was 1.229 m, which also was in the first place. Thus, the proposed method can be used to estimate the location of smartphones with high accuracy.

IX. CONCLUSION

This study described a method for estimating the position of a smartphone used in the Google smartphone decimeter challenge held in 2022. The proposed method used factor graph optimization to estimate the entire trajectory of a smartphone by creating various factors from the smartphone's GNSS observations. A two-step optimization method that estimates velocity and position in separate steps was proposed. As a result of the Google smartphone decimeter challenge with the proposed method, the final private score was 1.229 m, which won the first place.

The training data with position references for machine learning was provided; however, the proposed method did not use machine learning. Thus, the incorporation of machine learning into the graph optimization framework is a future challenge.

REFERENCES

[1] G. M. Fu, M. Khider, and F. van Diggelen. (2021) Google smartphone decimeter challenge. [Online]. Available: https://www.kaggle.com/c/google-smartphone-decimeter-challenge/

[2] T. Suzuki, "First place award winner of the smartphone decimeter challenge: global optimization of position and velocity by factor graph optimization," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 2974–2985.

[3] T. Suzuki, "Robust vehicle positioning in multipath environments based on graph optimization," in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 4223–4233.

[4] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[5] N. Sunderhauf and P. Protzel, "Switchable constraints for robust pose graph SLAM," in *IEEE International Conference on Intelligent Robots and Systems*, 2012, pp. 1879–1884.

[6] N. Sünderhauf, M. Obst, G. Wanielik, and P. Protzel, "Multipath mitigation in GNSS-based localization using robust optimization," in *IEEE Intelligent Vehicles Symposium*, 2012, pp. 784–789.

[7] D. Chen and G. X. Gao, "Probabilistic graphical fusion of LiDAR, GPS, and 3D building maps for urban UAV navigation," *Navigation, Journal of the Institute of Navigation*, vol. 66, no. 1, pp. 151–168, jan 2019.

[8] W. Li, X. Cui, and M. Lu, "A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles," *Tsinghua Science and Technology*, vol. 23, no. 6, pp. 724–732, dec 2018.

[9] R. M. Watson and J. N. Gross, "Robust navigation in gnss degraded environment using graph optimization," in *30th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2017*, vol. 5. Institute of Navigation, 2017, pp. 2906–2918.

[10] W. Wen, T. Pfeifer, X. Bai, and L.-T. Hsu, "Factor graph optimization for gnss/ins integration: A comparison with the extended kalman filter," *NAVIGATION*, vol. 68, no. 2, pp. 315–331, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/navi.421

[11] R. M. Watson and J. N. Gross, "Evaluation of kinematic precise point positioning convergence with an incremental graph optimizer," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2018, pp. 589–596.

[12] T. Suzuki, "Gnss odometry: Precise trajectory estimation based on carrier phase cycle slip estimation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7319–7326, 2022.

[13] F. Dellaert, "Factor Graphs and GTSAM: A Hands-on Introduction," Tech. Rep., 2012.

[14] T. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB," in *Proc. of The International Symposium on GPS/GNSS*, Jeju, Korea, 2009.