

Article

Continuous High-Precision Positioning in Smartphones by FGO-Based Fusion of GNSS–PPK and PDR

Amjad Hussain Magsi ^{1,*}, Luis Enrique Díez ¹ and Stefan Knauth ²

¹ Faculty of Engineering, University of Deusto, Avda. Universidades 24, 48007 Bilbao, Spain; luis.enrique.diez@deusto.es

² Hochschule für Technik Stuttgart, Faculty of Computer Science, Geomatics and Mathematics, Schellingstraße 24, 70174 Stuttgart, Germany

* Correspondence: amjad.hussain@deusto.es

Abstract: The availability of raw Global Navigation Satellites System (GNSS) measurements in Android smartphones fosters advancements in high-precision positioning for mass-market devices. However, challenges like inconsistent pseudo-range and carrier phase observations, limited dual-frequency data integrity, and unidentified hardware biases on the receiver side prevent the ambiguity resolution of smartphone GNSS. Consequently, relying solely on GNSS for high-precision positioning may result in frequent cycle slips in complex conditions such as deep urban canyons, underpasses, forests, and indoor areas due to non-line-of-sight (NLOS) and multipath conditions. Inertial/GNSS fusion is the traditional common solution to tackle these challenges because of their complementary capabilities. For pedestrians and smartphones with low-cost inertial sensors, the usual architecture is Pedestrian Dead Reckoning (PDR)+ GNSS. In addition to this, different GNSS processing techniques like Precise Point Positioning (PPP) and Real-Time Kinematic (RTK) have also been integrated with INS. However, integration with PDR has been limited and only with Kalman Filter (KF) and its variants being the main fusion techniques. Recently, Factor Graph Optimization (FGO) has started to be used as a fusion technique due to its superior accuracy. To the best of our knowledge, on the one hand, no work has tested the fusion of GNSS Post-Processed Kinematics (PPK) and PDR on smartphones. And, on the other hand, the works that have evaluated the fusion of GNSS and PDR employing FGO have always performed it using the GNSS Single-Point Positioning (SPP) technique. Therefore, this work aims to combine the use of the GNSS PPK technique and the FGO fusion technique to evaluate the improvement in accuracy that can be obtained on a smartphone compared with the usual GNSS SPP and KF fusion strategies. We improved the Google Pixel 4 smartphone GNSS using Post-Processed Kinematics (PPK) with the open-source RTKLIB 2.4.3 software, then fused it with PDR via KF and FGO for comparison in offline mode. Our findings indicate that FGO-based PDR+GNSS–PPK improves accuracy by 22.5% compared with FGO-based PDR+GNSS–SPP, which shows smartphones obtain high-precision positioning with the implementation of GNSS–PPK via FGO.

Keywords: factor graph optimization; fusion; GNSS; Kalman Filter; PDR; post-processing kinematics; smartphone



Citation: Magsi, A.H.; Díez, L.E.; Knauth, S. Continuous High-Precision Positioning in Smartphones by FGO-Based Fusion of GNSS–PPK and PDR. *Micromachines* **2024**, *15*, 1141. <https://doi.org/10.3390/mi15091141>

Academic Editors: Changhui Jiang, Shuai Chen, Yuwei Chen and Jianxin Jia

Received: 10 August 2024

Revised: 2 September 2024

Accepted: 9 September 2024

Published: 11 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Global Navigation Satellite System (GNSS) provides Positioning, Navigation, and Timing (PNT) services across diverse fields such as transportation, autonomous vehicles, disaster forecasting, aerospace, and defense. Since these services have traditionally relied on specialized expertise, costly and high-precision professional GNSS receivers. Therefore, their widespread use has been limited. However, nowadays, chip-size GNSS receivers have empowered devices like smartphones, tablets, smartwatches, and other portable gadgets to provide positioning services broadly. The use of smartphones for

pedestrian positioning and navigation has become more widespread due to their mass availability and ease of access for individuals [1,2].

Before 2016, Android smartphones only provided position and velocity data, limiting research on smartphone-based positioning [3]. In May 2016, Google enabled access to raw GNSS data via the Android API, allowing for advancements in high-precision positioning. However, access to raw GNSS observations from smartphones does not solve the problem of high-accuracy localization systems via smartphones, as they are typically equipped with low-cost, poorly performed polarized GNSS antennas and unknown hardware biases on the receiver, resulting in noisy data and frequent cycle slips, especially in GNSS-challenging areas such as urban canyons, complex environments, tunnels, underpasses, etc., where these issues typically arise due to multipath interference and non-line-of-sight (NLOS) conditions [4,5]. Additionally, PNT signals from satellites are considerably impacted by various errors stemming from atmospheric conditions, including ionospheric delays, dry and wet tropospheric delays, as well as satellite clock offsets and ephemeris inaccuracies [6]. Therefore, the All Source Positioning and Navigation (APNS) program under the USA-based Defense Advanced Research Projects Agency (DARPA) has noted that single-source navigation is no longer reliable, prompting researchers to explore alternative navigation solutions [7,8]. This led researchers to fuse alternative navigation solutions to mitigate these problems. Different sensors are already available in modern smartphones, but among these, the Microelectrical Mechanical Systems (MEMS) Inertial Measurement Unit (IMU) holds significant importance due to its complementary behavior with GNSS, offering low errors in the short term and a high update rate without requiring additional infrastructure in the environment [9]. MEMS IMUs face limitations due to their high noise output, preventing the development of Strapdown Inertial Navigation Systems (SINS) and their integration with GNSS. This accumulates errors that are increased with the elapsed time, while the PDR system accumulates errors based on the detected steps, as they rely on pedestrian walking patterns [10]. Therefore, for the pedestrian navigation context with smartphones, PDR+GNSS fusion is preferable.

The fusion of PDR and GNSS in smartphones is still an open research topic. These fusion architectures mainly involve several steps, such as integration levels, fusion algorithms and frequencies, constellation, and GNSS processing techniques, as shown in Figure 1. Most of the researchers used KF and its variant models as the main fusion technique for these fusion architectures, such as Jiang et al. [11], who used the Krill Herd Algorithm (KHA) to optimize particle filters for GNSS/PDR integration, enhancing positioning accuracy. Zhu et al. [12] improved GNSS/PDR fusion positioning based on smartphone IMU error analysis. Zhang et al. [13] assessed GNSS/PDR fusion on smartphones in complex settings. Yan et al. [14] modified the KF using varied sampling frequencies from smartphone GNSS and IMU data for fusion. However, achieving better accuracy with a KF is challenging due to its first-order Markov assumption, Gaussian noise assumption, and reliance on the most recent observation.

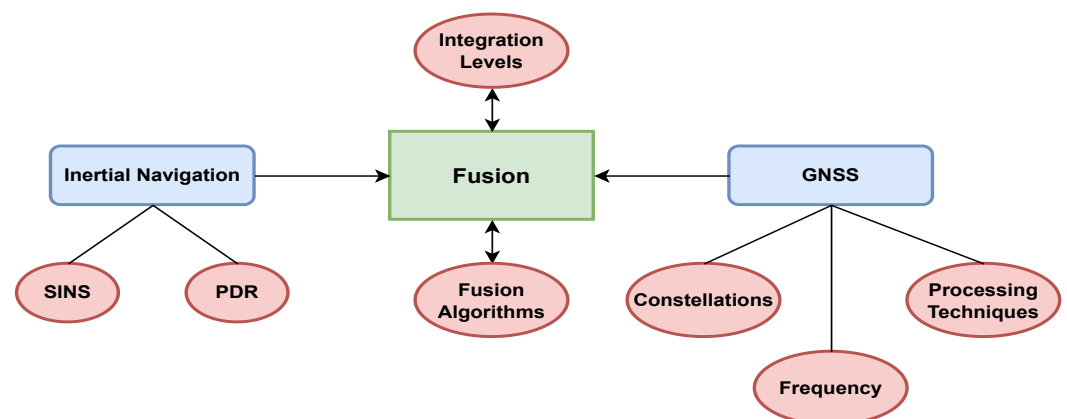


Figure 1. Inertial+GNSS fusion architectures.

Recently, FGO-based multi-sensor fusion has gained attraction among researchers. Initially, FGO demonstrated superior accuracy in SLAM-based complex problems in robotics and has been applied to INS/GNSS fusion architectures as well [15–22]. However, in the pedestrian navigation domain, significant analysis is still needed, and the application of FGO is in its early stages, presenting open research opportunities. Recently, Jiang et al. [23] implemented FGO for PDR+GNSS for a single walking dataset in open-sky conditions and compared its performance against KF. Later, ref. [24] introduced a PDR/GNSS integration method using FGO to boost accuracy in challenging GNSS conditions. Later, ref. [25] tested FGO for PDR/GNSS on smartphones by placing the smartphone in different positions, such as box, handheld, and pocket, and tested the fusion performance with open sky and urban canyon conditions. In our previous paper [26], we compared the performance of FGO and KF under two different motion scenarios (walking and running) in partially open sky conditions. We found that FGO performance deteriorated with varying motions, with running significantly affecting FGO performance more than walking. This degradation was attributed to differences in motion parameters such as step length, stride length, and speed. Therefore, in another paper [27], we conducted a more detailed analysis of FGO by examining a range of pedestrian motions from different individuals, including variations in gender, gait patterns, and heights. Additionally, we investigated how transient variations in how the smartphone is held affect errors in the PDR block. We then evaluated FGO's performance against these errors and compared it with KF and Smooth KF under these sources of PDR errors.

To the best of our knowledge, on the one hand, no work has tested the fusion of GNSS PPK and PDR on smartphones. And, on the other hand, the works that have evaluated the fusion of GNSS and PDR by means of FGO have always performed it using the GNSS SPP technique. Therefore, the aim of this work is to combine the use of the GNSS PPK technique and the FGO fusion technique to evaluate the improvement in accuracy that can be obtained on a smartphone compared with the usual GNSS SPP and KF fusion strategies.

The remaining sections detail the methodology, covering the PDR development and PDR+GNSS (SPP/PPK) fusion architectures via KF and FGO. The experimental evaluation section outlines the experiment's description, setup, and results, followed by the conclusion and future work section.

2. Methodology

The PDR development and its fusion with GNSS–SPP and GNSS–PPK by KF and FGO are presented in this section.

2.1. PDR

By using the number of detected steps, step length, and heading estimation calculation, the PDR can be developed, and its mechanism shown in Figure 2 can be mathematically written as follows:

$$\mathbf{Pos}_{k+1} = \mathbf{Pos}_k + \Delta\mathbf{Pos}_{k|k+1} \quad (1)$$

In this equation, \mathbf{Pos}_{k+1} and \mathbf{Pos}_k show position at steps $k + 1$ and k , respectively, and $\Delta\mathbf{Pos}_{k|k+1}$ is a positional increment from k to $k + 1$, which can be formulated as follows in east and north directions.

$$\Delta\mathbf{Pos}_{k|k+1} = \begin{bmatrix} \Delta\mathbf{Pos}_{k|k+1}^E \\ \Delta\mathbf{Pos}_{k|k+1}^N \end{bmatrix} = \begin{bmatrix} \mathbf{SL}_{k|k+1} \cdot \cos(\phi) \\ \mathbf{SL}_{k|k+1} \cdot \sin(\phi) \end{bmatrix} \quad (2)$$

ϕ is the heading angle, and $\mathbf{SL}_{k|k+1}$ is the step length between two steps.

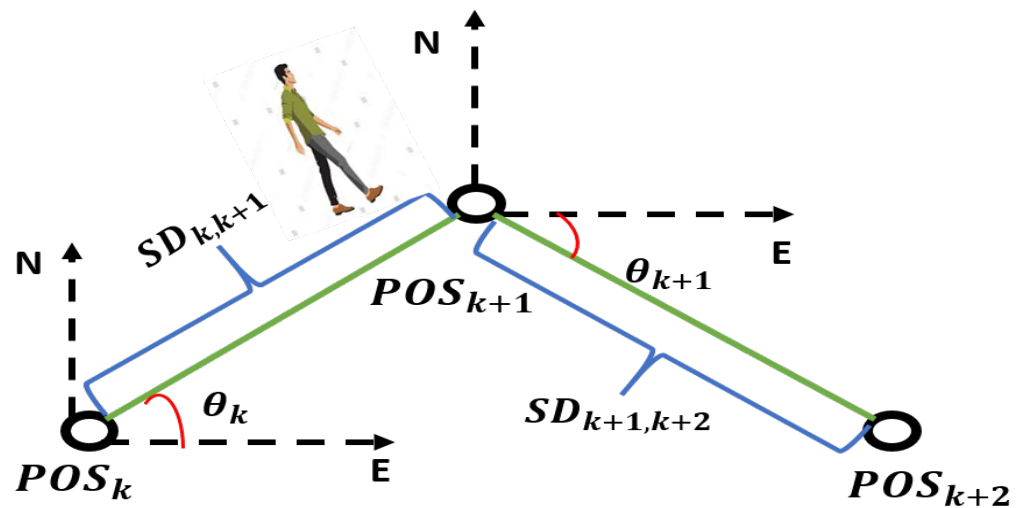


Figure 2. PDR mechanism.

The steps taken were detected by measuring the peak of acceleration magnitude, which can be written as

$$Acc_{mag_k} = \sqrt{Acc_{x_k}^2 + Acc_{y_k}^2 + Acc_{z_k}^2} \tag{3}$$

where Acc_{mag_k} denotes the magnitude of the triaxial accelerometer, and Acc_{x_k} , Acc_{y_k} , and Acc_{z_k} denote the accelerometer measurements in x , y , and z directions at time step k .

The step length was estimated using Weinberg’s algorithm [28], which can be mathematically formulated as follows:

$$SL_{k|k+1} = C \cdot [\max(Acc_{mag_k}) - \min(Acc_{mag_k})]^{1/4} \tag{4}$$

$SL_{k|k+1}$ denotes the estimated step length between two successive acceleration peaks, while C is a constant factor to adjust the estimated step length, and the step heading was estimated via the Attitude and Heading Reference System (AHRS) from the smartphone. The factor value of C was determined through empirical testing. We conducted experiments by adjusting C under various conditions to see how it influenced the accuracy and consistency of the results. Through this process, we identified a value for C that yielded the best performance for our specific application.

2.2. GNSS

Modern Android smartphones can now provide raw GNSS observations, including pseudorange measurements, carrier phase data, and navigation messages. This capability allows for high-accuracy positioning directly on smartphones by processing these raw data to minimize associated errors [29–31]. In this study, we have utilized Post-Processing Kinematic (PPK) and compared it with Single-Point Positioning (SPP). A brief description of these techniques is provided below.

Single-Point Positioning (SPP): Also known as standalone positioning, it is a straightforward method that relies solely on the code–phase pseudorange measurements to determine a receiver’s position. This method does not incorporate the more precise carrier phase measurements and typically does not apply error corrections [32,33]. The basic pseudorange equation used in SPP is as follows:

$$\rho_i = c(t_r - t_s) + d_{orb} + d_{iono} + d_{trop} + \epsilon \tag{5}$$

where

- ρ_i is the pseudorange from satellite i ;

- c is the speed of light;
- t_r and t_s are the receiver and satellite clock biases, respectively;
- d_{orb} represents satellite orbit errors;
- d_{iono} and d_{trop} are the ionospheric and tropospheric delays;
- ϵ encompasses various noise sources and multipath errors.

In SPP, the receiver calculates its position by solving for the location using pseudorange data from multiple satellites. However, due to the nature of code–phase measurements, this method is subject to several sources of error:

- **Clock offsets:** the receiver’s clock is less accurate compared with the atomic clocks on GNSS satellites, leading to timing errors.
- **Ephemeris errors:** inaccuracies in the satellite’s transmitted position.
- **Ionospheric and tropospheric delays:** signal propagation is affected by the Earth’s atmosphere, causing delays.
- **Multipath and NLOS (non-line-of-sight):** signals reflecting off surfaces before reaching the receiver, leading to errors in the measured pseudorange.

Because SPP does not correct for these errors and relies only on code data, the resulting position estimates are typically less accurate, often with errors in the range of several meters.

Post-Processed Kinematics (PPK): In contrast, PPK positioning is a more sophisticated method that uses carrier phase measurements, which are significantly more accurate than code–phase measurements. PPK applies a series of corrections to the raw GNSS data to achieve better accuracy, which is basically a post-processed RTK GNSS processing technique [34,35]. We opted for this approach because the SAPOS base station services in Germany did not provide real-time correction data. Instead, we had to download the RINEX observation files from the base station and process them afterward using RTKLIB Explorer. Additionally, we applied several other corrections, including ionospheric and tropospheric delay corrections, ambiguity fixing ratios, and adjustments to elevation angles. The basic equation for carrier phase measurement is as follows:

$$\Phi_i = \rho_i + \lambda N_i + c(t_r - t_s) + d_{orb} + d_{iono} + d_{trop} + \epsilon_\Phi \quad (6)$$

where

- Φ_i is the carrier phase measurement;
- λ is the carrier wavelength;
- N_i is the integer ambiguity (number of whole carrier wavelengths between the satellite and the receiver);
- Other terms are similar to those in the pseudorange equation, with ϵ_Φ representing the measurement noise specific to carrier phase.

Key aspects of PPK include the following:

- **Ambiguity resolution:** PPK involves resolving the integer ambiguity N_i , which is crucial for achieving high precision. This often requires a reference station (a nearby base station) with known coordinates to help resolve these ambiguities.
- **Error corrections:** PPK applies corrections for ionospheric and tropospheric delays, satellite clock errors, and ephemeris errors. These corrections are typically derived from precise models or additional reference stations.

By correcting for these errors and resolving ambiguities, PPK can achieve positioning accuracy down much better than SPP.

2.3. PDR Fusion with GNSS–SPP and PPK with KF in Smartphones

The fusion of PDR and GNSS (both SPP and PPK) using KF has been performed following an error state approach and a loosely coupled architecture. \mathbf{X} can be the state vector for transitioning the state addressing position errors in both the x and y coordinates of PDR and can be written as follows:

$$\mathbf{X} = \begin{pmatrix} \Delta\text{PDR}_x \\ \Delta\text{PDR}_y \end{pmatrix}$$

The formulation of this transformation further can be written as below:

$$\mathbf{X}_{k+1} = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k + \mathbf{W}_{k+1} \quad (7)$$

In this equation, \mathbf{X}_{k+1} and \mathbf{X}_k are the state vectors. \mathbf{W}_{k+1} is a process noise vector with the assumption of normal Gaussian distribution as $\mathbf{W}_{k+1} \sim \mathcal{N}(0, Q_k)$, where Q_k is the covariance matrix of the process noise and $\mathbf{F}_{k|k+1}$ is the transformation matrix at epochs $k + 1$ and k , which can be expressed mathematically as

$$\mathbf{F}_{k|k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (8)$$

The measurement model is

$$\mathbf{Z}_{k+1} = \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1} + \mathbf{V}_{k+1} \quad (9)$$

where \mathbf{Z}_{k+1} represents the measurement vector. \mathbf{V}_{k+1} is the measurement noise vector with Gaussian distribution assumption expressed as $\mathbf{V}_{k+1} \sim \mathcal{N}(0, R)$, where R represents the covariance matrix of the measurement noise and \mathbf{H}_{k+1} represents the observation matrix.

$$\mathbf{H}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (10)$$

The real observations represent the difference between PDR and GNSS–SPP for PDR+GNSS–SPP and PDR and GNSS–PPK for PDR+GNSS–PPK. Therefore, it can be rewritten and described as

$$\mathbf{Z}_{k+1} = \mathbf{Pos}_{k+1}^{\text{GNSS}} - \mathbf{Pos}_{k+1}^{\text{PDR}} \quad (11)$$

Prediction from KF will be as follows:

$$\mathbf{X}_{k+1}^- = \mathbf{F}_{k|k+1} \cdot \mathbf{X}_k \quad (12)$$

$$\mathbf{P}_{k+1}^- = \mathbf{F}_{k|k+1} \cdot \mathbf{P}_k \cdot (\mathbf{F}_{k|k+1})^T + \mathbf{Q}_k \quad (13)$$

The prediction state \mathbf{X}_{k+1}^- is derived via the transformation model. The matrix \mathbf{P}_k denotes the predicted state covariance matrix at epoch k , while \mathbf{P}_{k+1}^- represents the prediction at epoch $k + 1$. Subsequently, the correction step of the Kalman filter updates the state estimation as follows:

$$\hat{\mathbf{X}}_{k+1} = \mathbf{X}_{k+1}^- + \mathbf{K}_{k+1} \cdot (\mathbf{Z}_{k+1} - \mathbf{H}_{k+1} \cdot \mathbf{X}_{k+1}^-) \quad (14)$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T \cdot [\mathbf{H}_{k+1} \cdot \mathbf{P}_{k+1}^- \cdot (\mathbf{H}_{k+1})^T + \mathbf{R}_{k+1}]^{-1} \quad (15)$$

$$\mathbf{P}_{k+1} = (\mathbf{I}_{2 \times 2} - \mathbf{K}_{k+1} \cdot \mathbf{H}_{k+1}) \cdot \mathbf{P}_{k+1}^- \quad (16)$$

Here, $\hat{\mathbf{X}}_{k+1}$ denotes the revised state estimation at time step $k + 1$, \mathbf{K}_{k+1} symbolizes the Kalman gain matrix at time step $k + 1$, and $\mathbf{I}_{2 \times 2}$ denotes the 2×2 identity matrix.

2.4. PDR Fusion with GNSS–SPP and GNSS–PPK with FGO in Smartphones

A factor graph serves as a probabilistic model that employs a graph structure to illustrate the probabilistic relationships among various variables. In essence, it consists of two distinct types of nodes: the unknown state variables and the factor nodes, which encode the conditional probabilities associated with certain sets of state variables. These nodes are connected by undirected edges, linking each state variable node to its corresponding

factor node. An edge exists between these nodes only if there's a functional dependence between the variable and factor. By employing a factor graph, one can break down a global probability function into the product of several local probability functions, thereby simplifying the resolution of the overall function.

In sensor integration, our focus typically lies on estimating a collection of state variables X based on a given set of measurements Z . Under the assumption of a first-order Markov model, we can express the conditional probability density function in the following manner:

$$P(X|Z) = \prod_{i=1}^k \frac{P(z_i|x_i)P(x_i|x_{i-1}, u_i)}{P(z_i)} P(x_0) \tag{17}$$

where z_i represents the measurements observed at epoch i (e.g., GNSS measurements), x_i represents the system state at i , and u_i denotes the control input (e.g., PDR measurements).

The most often used estimator for these unknown state variables X is the maximum a posteriori, or MAP estimate:

$$\hat{X} = \arg \max P(X|Z) = \arg \max \prod_{i=1}^k P(z_i|x_i)P(x_i|x_{i-1}, u_i) \tag{18}$$

In FGO-based integration, all these likelihood and transition probabilities are treated as the factorization of the global probability:

$$\hat{X} = \arg \max \prod_{j=1}^n f(x_j) \tag{19}$$

Each factor models a constraint and must include a measure of uncertainty. The most common model is a Gaussian noise:

$$P(z_i|x_i) \propto \exp\left(-\frac{1}{2}\|h_i(x_i - z_i)\|^2 \nabla_i\right) \tag{20}$$

$$P(x_i|x_{i-1}) \propto \exp\left(-\frac{1}{2}\|\Phi_i(x_{i-1} - x_i)\|^2 \Omega_i\right) \tag{21}$$

where the function $\Phi_i(\cdot)$ describes the relationship between the preceding states x_{i-1} and x_i , the function $h_i(\cdot)$ represents the relationship between the state x_i and the measurement z_i , and the covariance matrices are denoted by ∇_i and Ω_i .

Taking the negative log and dropping the factor $1/2$ transforms the problem into the minimization of an error function, which is a nonlinear least-squares problem.

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \|\Phi_i(x_{i-1}) - x_i\|_{\Omega_i}^2 + \sum_{i=1}^k \|h_i(x_i) - z_i\|_{\nabla_i}^2 \right) \tag{22}$$

In our case, there are two factors: PDR measurements and the GNSS measurements (GNSS-SPP in case of PDR+GNSS-SPP) and (GNSS-PPK in case of PDR+GNSS-PPK) fusion architectures as shown in Figure 3.

Factor by PDR position: this factor is obtained from the positions derived by the PDR block.

$$\begin{aligned} \left\| e_{k+1}^{PDR} \right\|_{\Sigma_{k+1}^{PDR}}^2 &= \|X_{k+1} - f(X_k)\|_{\Sigma_{k+1}^{PDR}}^2 \\ &= \left\| x_{k+1} - \left(x_k + \begin{bmatrix} \mathbf{SL}_{t,t+1} \cdot \cos(\phi) \\ \mathbf{SL}_{t,t+1} \cdot \sin(\phi) \end{bmatrix} \right) \right\|_{\Sigma_{k+1}^{PDR}}^2 \end{aligned} \tag{23}$$

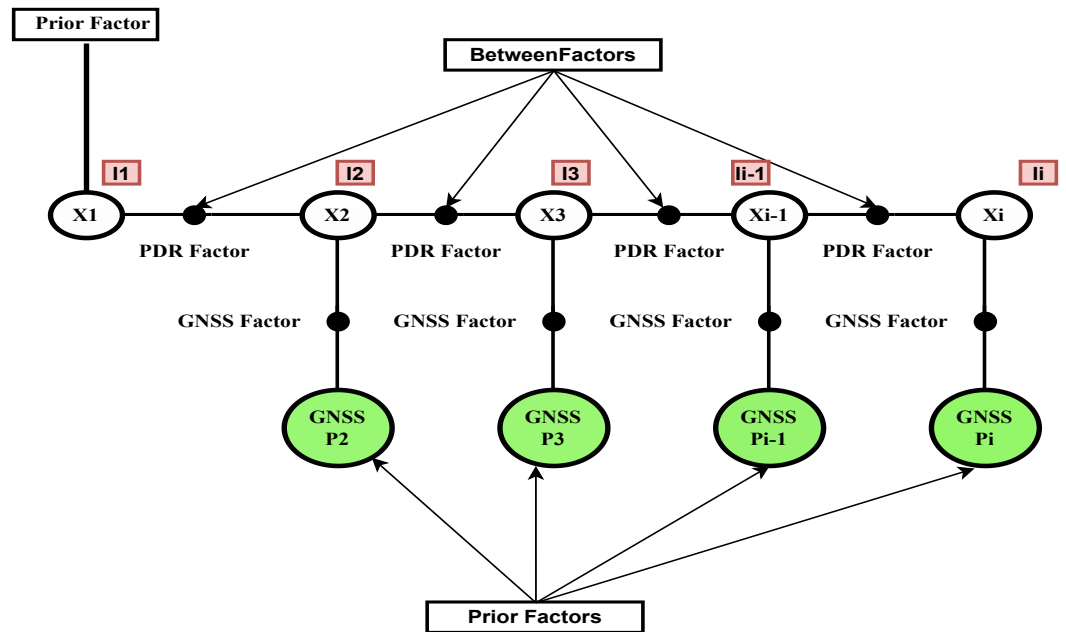


Figure 3. Factor graph for PDR+GNSS-SPP fusion architecture.

GNSS-SPP and PPK factor: GNSS-SPP factor is obtained by the smartphone raw GNSS observation with SPP solution.

$$\|e_{k+1}^{GNSS}\|_{\Sigma_{k+1}^{GNSS}}^2 = \|X_{k+1} - GNSS_{k+1}\|_{\Sigma_{k+1}^{GNSS}}^2 \tag{24}$$

Therefore,

$$\hat{X} = \arg \min \left(\sum_{i=1}^k \left[\|e_{i+1}^{PDR}\|_{\Sigma_{i+1}^{PDR}}^2 + \|e_{i+1}^{GNSS}\|_{\Sigma_{i+1}^{GNSS}}^2 \right] \right) \tag{25}$$

The GNSS-PPK factor is created by replacing the GNSS with GNSS-PPK data.

We used the Georgia Tech Smoothing and Mapping (GTSAM) library [36] with the Levenberg-Marquardt (LM) optimizer [37].

3. Experimental Evaluation

3.1. Experimental Description

The experiment aimed to evaluate the reliability of the Google Pixel 4 smartphone’s GNSS for high-precision localization by comparing its accuracy with that of a high-accuracy Ublox ZED F9P RTK GNSS receiver (Thalwil, Switzerland), which served as the ground truth. The Ublox antenna was placed on top of the pedestrian’s head with a rod on the backpack, and the smartphone was held in a handheld position by the pedestrian, as shown in Figure 4. The experiment was conducted in Korntal Munchingen, Stuttgart, Germany. A closed path of around 920 m, characterized by a partially open sky with trees and houses, was chosen for data collection, which can be seen in Figure 5. The data were collected at a normal walking speed by completing two laps (1.8 km) on the pre-defined path in approximately 27 min. A Google Pixel 4 smartphone was used to collect the smartphone’s built-in GNSS and inertial sensor measurements. For ground truth, a Ublox ZED F9P RTK GNSS receiver was used, which is capable of providing centimeter-level accuracy. The pedestrian carried the smartphone in front of his chest, while the Ublox’s ANN-MB-00 GNSS antenna was mounted on a backpack, as can be seen in Figure 4.



Figure 4. Data collection setup.



Figure 5. Difference between the ground truth, SPP-GNSS, and PPK-GNSS data.

The raw GNSS observations from the Android smartphone were logged using the GNSSLogger app [38], which conveniently allows data to be recorded directly in the RINEX format. For processing these observations with the PPK GNSS technique, we utilized the open-source RTKLIB Explorer2.4.3 software [39]. The RINEX observation file obtained from the smartphone contained numerous errors inherent to raw data, including satellite clock errors, inaccuracies in the satellite orbit and position, ionospheric delays, wet and dry tropospheric delays, receiver clock offsets, as well as multipath effects and non-line-of-sight (NLOS) errors caused by obstructions such as trees and nearby buildings.

To mitigate these errors, we employed the PPK technique, which applies corrections to the noisy observation file using data from a nearby base station. The RTKLIB software also offers built-in corrections for ionospheric and tropospheric delays, satellite clock errors,

and other necessary adjustments. In this study, we corrected the RINEX observation file by selecting signals from multiple GNSS constellations, including GPS, GLONASS, Galileo, and BeiDou. However, we disabled the SBAS correction system because our focus was on evaluating the fusion of PPK-based GNSS data with Pedestrian Dead Reckoning (PDR) rather than on SBAS corrections.

For satellite clocks and orbits, we relied on broadcast ephemerides rather than precise clock products, as our study did not focus on Precise Point Positioning (PPP). Ionospheric errors were corrected using the broadcast ionospheric model, with the impact of these errors further reduced by the use of dual-frequency observations, which are less susceptible to ionospheric disturbances compared with single-frequency data. For tropospheric corrections, we used an estimated Zenith Tropospheric Delay (ZTD) model, which is given by

$$\text{ZTD} = 0.0022768 \cdot \frac{P}{f(\phi, H)} + \text{ZWD}$$

where P is the atmospheric pressure at the receiver's location (in hPa). $f(\phi, H)$ is a function of the latitude ϕ and height H above sea level, accounting for the elevation and atmospheric conditions.

Carrier phase ambiguity resolution was implemented only for the GPS constellation, with ambiguity resolution disabled for other constellations. This selective ambiguity resolution allowed us to achieve integer-cycle resolution for GPS signals. These parameters, which are preconfigured in the RTKLIB software, were used to correct the observations. Additionally, we applied both forward and backward filtering, resulting in a combined filter approach that enhanced the accuracy and reliability of the corrections applied to the GNSS data. Moreover, smartphone low-cost chip size GNSS chip receivers and antennas are particularly susceptible to user-side errors such as clock biases, hardware inaccuracies, and antenna offsets [5,35]. Although we are aware of these issues, they were not addressed in this paper.

For processing the data, we utilized the L1, L2, and E5b frequencies. A 15-degree elevation mask was applied, with receiver dynamics enabled and tidal corrections disabled. For integer ambiguity resolution, we employed specific strategies for different satellite systems: continuous-resolution was used for GPS, while auto-calculation was selected for GLONASS (GLO) and BeiDou (BDS) was kept ON, as can be seen in Figure 6.

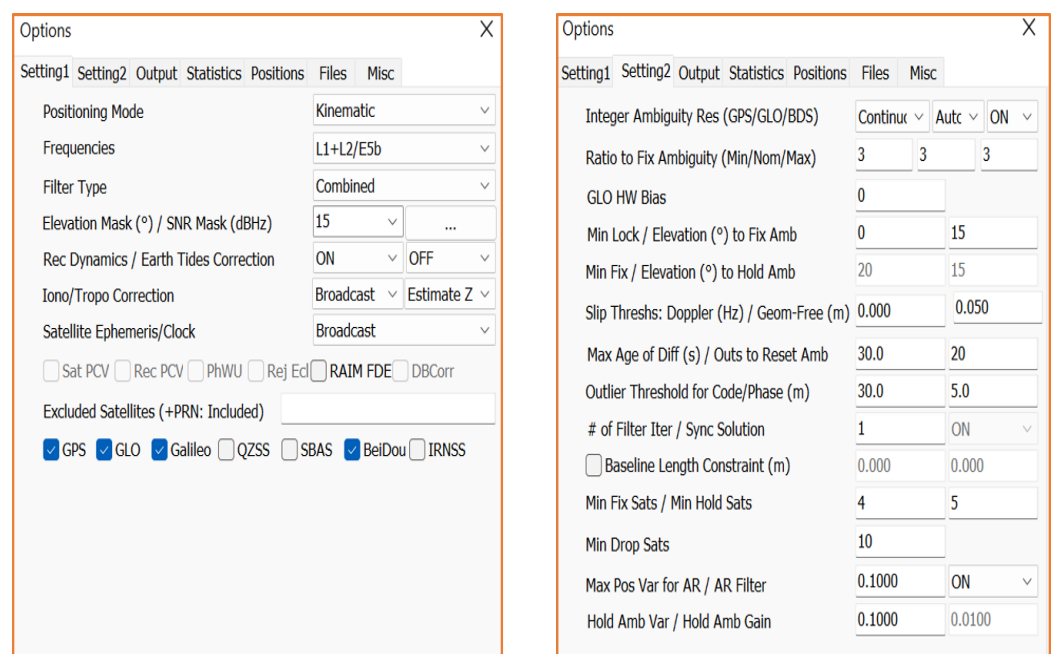


Figure 6. RTKLIB settings.

These settings were chosen after testing various configurations, as they provided a more reliable fixed solution. Specifically, we set the ratio for fixing ambiguity at 3. The elevation mask for fixing ambiguities was also set at 15 degrees, with the minimum fix elevation and minimum elevation to hold ambiguity configured at 20 degrees and 15 degrees, respectively. Additionally, an outlier threshold of 5 m was established for carrier phase measurements.

The base station, located approximately 17 km from the experimental site, provided Radio Technical Commission for Maritime Services (RTCM) correction data through the Satellite Positioning Service (SAPOS) in Baden-Württemberg, Germany. The RTKLIB parameters that we used in the software settings described above for post-processing the smartphone raw GNSS observations are listed in Table 1. With these parameters, we achieved almost 11% fixed and 89% float solutions for smartphone GNSS. In contrast, through Ublox ZED F9P, the raw GNSS data were logged in its .ubx (Ublox) format. Therefore, we first converted that into RINEX format using the RTKConv tool from the RTKLIB. By processing this RINEX data with RTCM correction data of the same base station and with the same RTKLIB parameters as listed in Table 1, we yielded a fixed solution rate of 96.9%. The difference between the implementation of PPK on the datasets from Ublox and smartphone GNSS can be further seen in Figure 5. This high fixed solution rate supports the use of the Ublox receiver as ground truth, demonstrating its reliability and efficiency compared with the smartphone GNSS.

Table 1. RTKLIB parameters.

Positioning Mode	Kinematic
Filter Type	Combined
GNSS Constellations	Disable QZSS, IRNSS, and SBAS
Ionospheric Correction	Broadcast
Tropospheric Correction	Estimate ZTD
Satellite Clock	Broadcast
Integer AR	Continuous for GPS

Accurate synchronization of ground truth and smartphone data: The data from both devices were timely synchronized by maintaining the sampling frequency of both devices. Moreover, while the Ublox receiver and the smartphone were kept approximately one meter apart during data collection, the resulting positional offset was considered but not specifically corrected in this study. This choice was made to maintain a straightforward setup, with our primary focus on the fusion strategy between PDR and GNSS using the PPK technique.

KF and FGO parameters: The efficacy of the FGO and KF fusion algorithms hinges on their adept utilization of parameter covariance settings. Throughout this study, we preserved uniform parameter settings for both fusion techniques. Regarding the KF, we set the KF parameters as follows when we parametrize it in our case:

$$P = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}, \quad Q_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

where P is the initial state covariance matrix, Q_k is the process noise covariance matrix, and R is the measurement noise covariance matrix.

In the domain of FGO, our approach involves two factors: PDR and GNSS. It is important to recognize that both of these factors come with uncertainties and noise. We have used GTSAM's odometry and prior noise model classes to handle these uncertainties. We set the same values for both KF and FGO.

```
odometryNoise = noiseModel.Diagonal.Sigmas([2; 2; 0.1])
```

```
priorNoise = noiseModel.Diagonal.Sigmas([1.4; 1.4; 0.1])
```

3.2. Results

3.2.1. Position Error

In the semi-open sky condition, the SPP solution of smartphone GNSS was affected by the influence of trees and houses. This effect can be seen in Figure 7 in the form of horizontal position error peaks shown in red. Therefore, it can also be seen that the fusion of this GNSS–SPP with PDR from both KF and FGO is producing better results as compared with only GNSS–SPP by mitigating the peak errors of GNSS–SPP. However, we achieved lower positioning errors with FGO compared with KF. This is because of FGO’s iterative optimization of historical measurements, which results in improved accuracy, while KF relies on current and last measurements only. We computed the Cumulative Distribution Function (CDF) of these positioning errors obtained by FGO, KF, and GNSS–SPP, as shown in Figure 8, and from this, it is clear that FGO-based fusion provides overall better accuracy and precision than KF. It can also be observed from the fused trajectories presented in Figure 9 that FGO is more robust than KF.

For PPK, we enhanced smartphone raw GNSS measurements using RTCM messages from a base station, processed with RTKLIB software using various parameters shown in Table 1. Consequentially, GNSS error peaks were lower than SPP, as shown in Figure 10 by the red color. Fusing these GNSS–PPK with PDR produced better, smoother, and more precise output than PDR+GNSS–SPP by both fusion algorithms, KF and FGO. However, again, FGO produced better output than KF and only GNSS–PPK, which can be clearly seen from the CDF of horizontal position errors in Figure 8. The trajectories from FGO, KF, and GNSS–PPK can be seen in Figure 11.

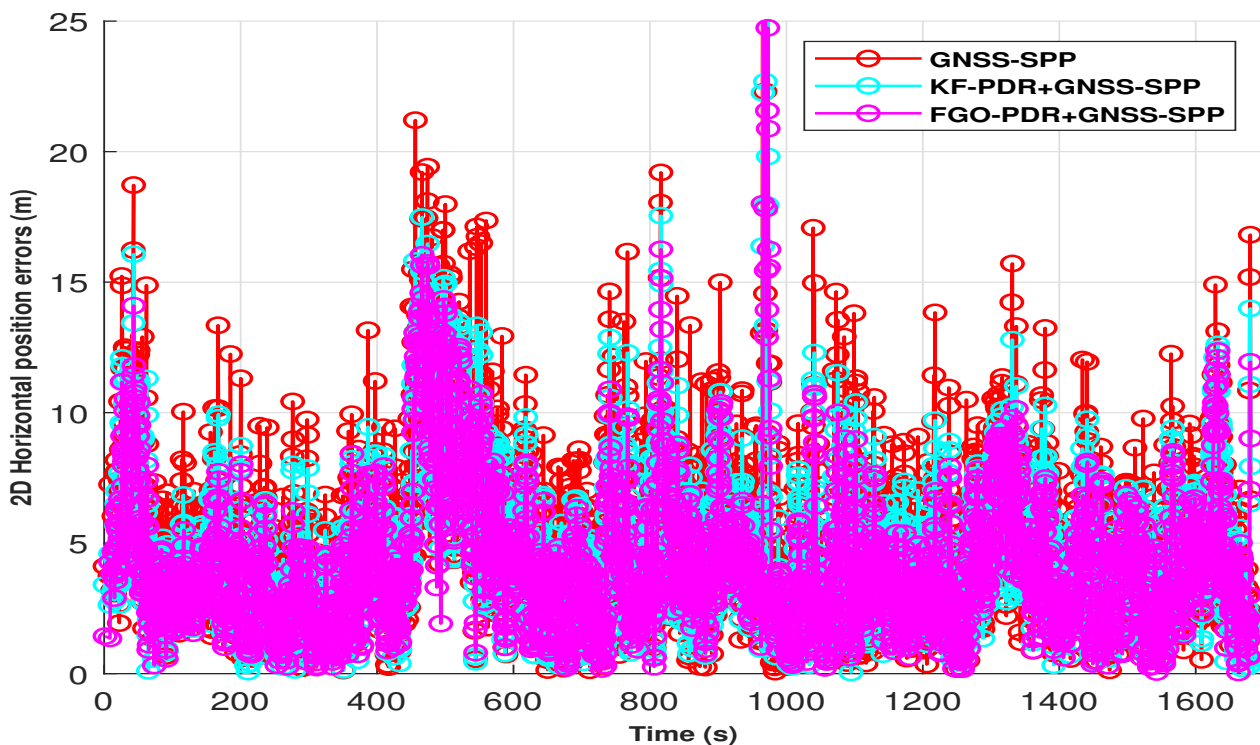


Figure 7. Two-dimensional horizontal positioning errors in PDR+GNSS–SPP fusion architecture.

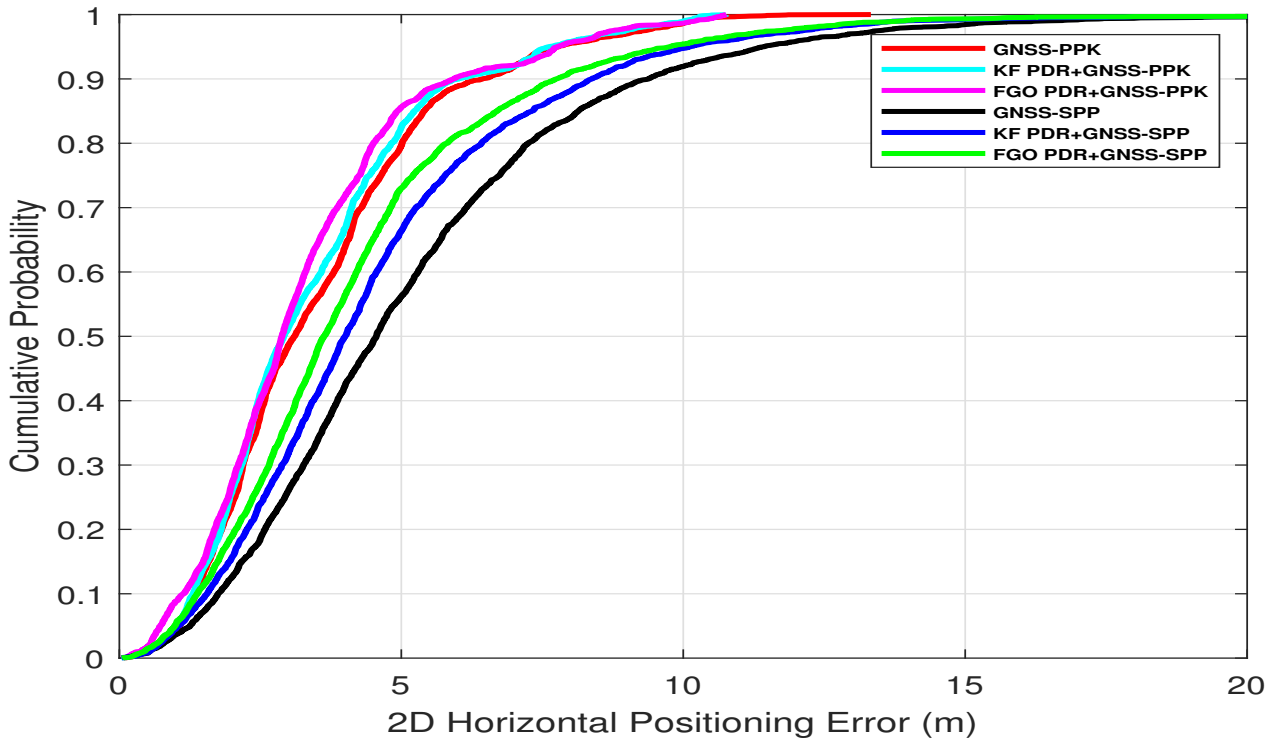


Figure 8. Combined overall CDF from both fusion architectures using FGO and KF.

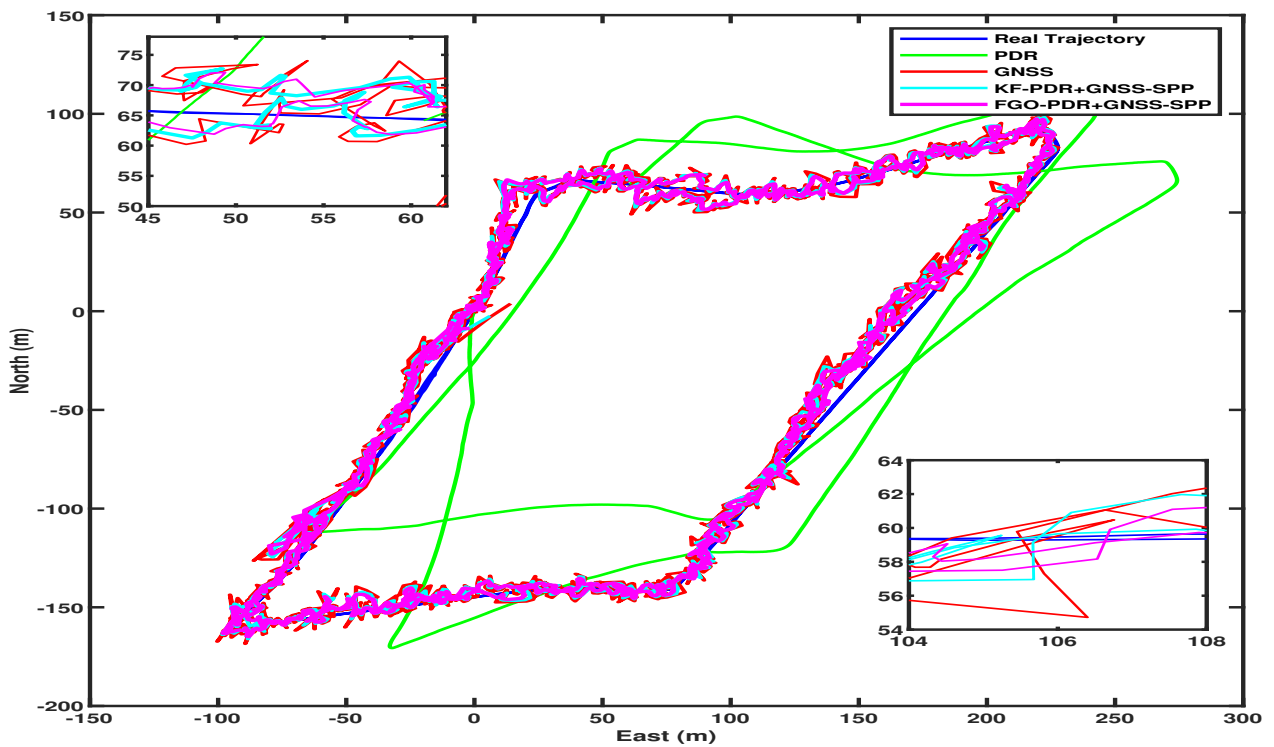


Figure 9. Comparison of trajectories in PDR+GNSS-SPP fusion architectures.

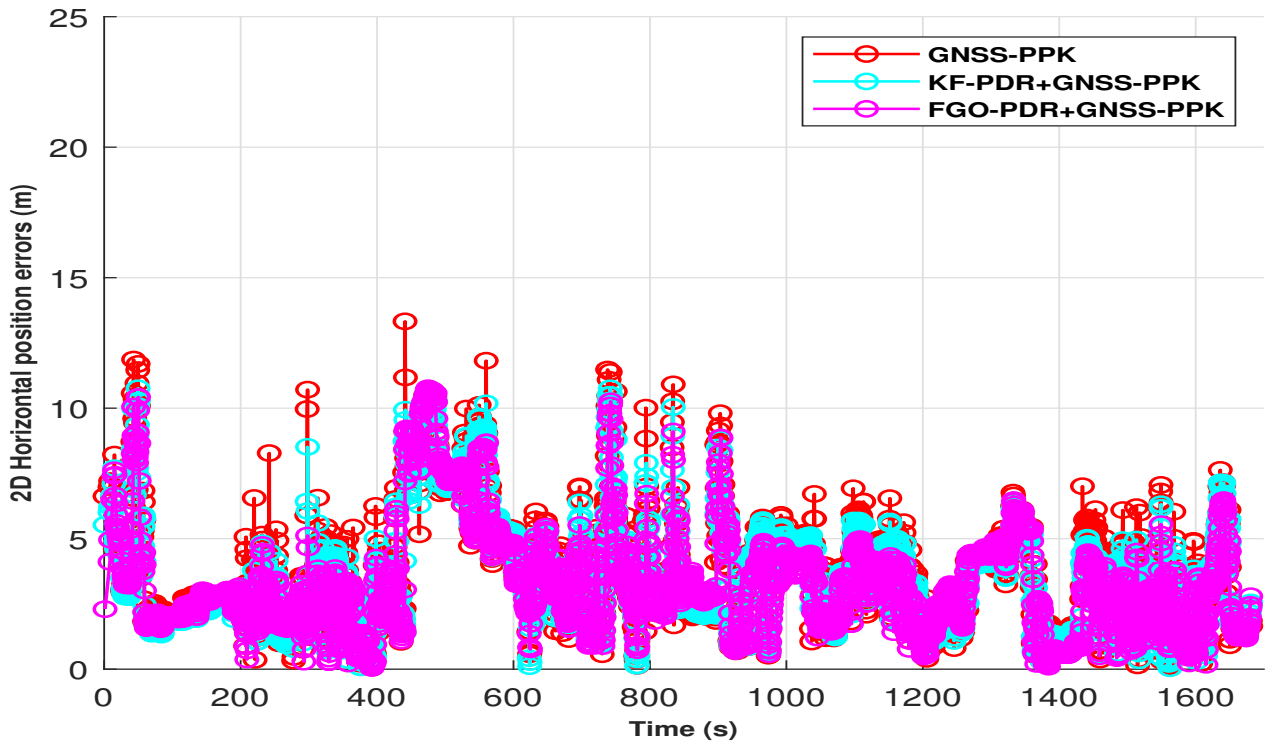


Figure 10. Two-dimensional horizontal positioning errors in PDR+GNSS–PPK fusion architecture.

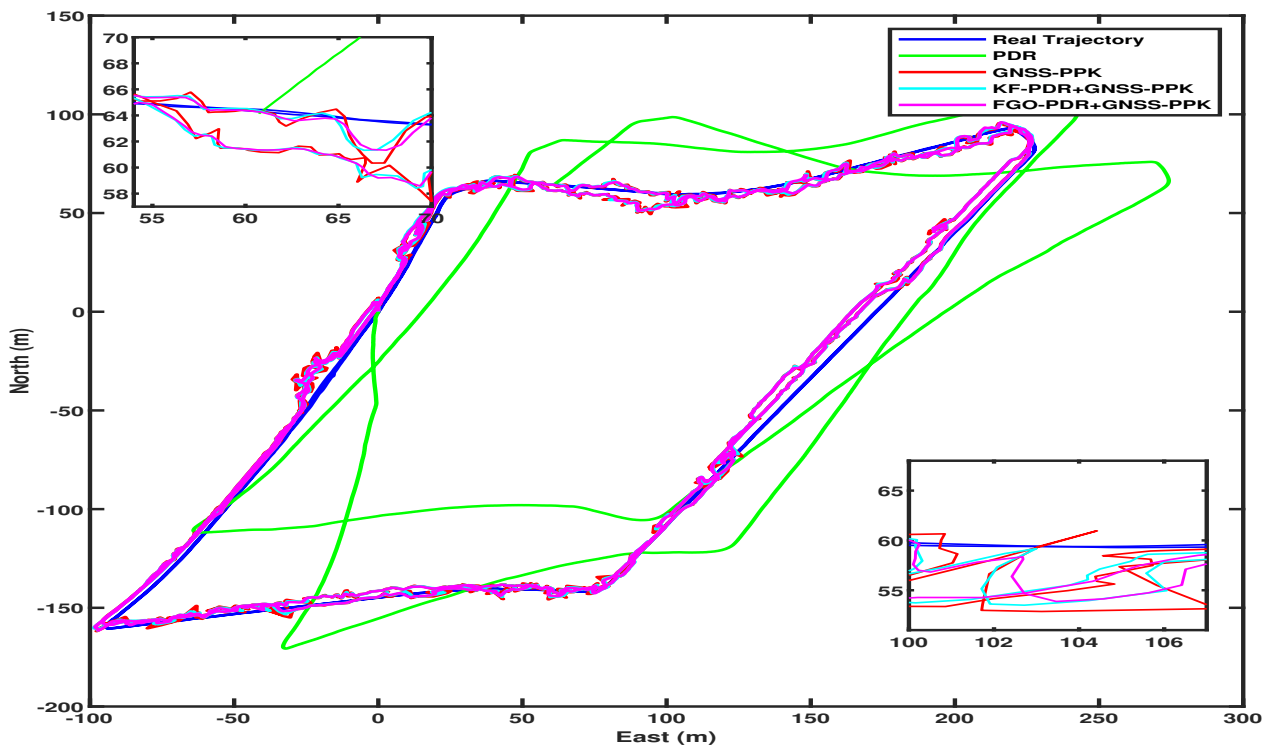


Figure 11. Comparison of trajectories in PDR+GNSS–PPK fusion architecture.

3.2.2. Computational Load

FGO-based PDR+GNSS–PPK has demonstrated superior positional accuracy compared with other methods discussed in this study. However, it is crucial to assess the computational demands of the FGO approach due to its iterative optimization process. FGO performs optimization at each epoch, incorporating the entire dataset, which can significantly impact computational efficiency. To evaluate the computational load, we

analyzed the performance of FGO at each epoch, measuring the amount of computational power utilized. This analysis includes both the per-epoch computational load and the cumulative computational load, which accounts for the computational resources used across all previous epochs. The results of this evaluation are illustrated in Figures 12 and 13. It can be noted that Figure 12 has some peaks at several epochs. This is because of unexpected variations in the data, such as noise or outliers, which require additional computations. The LM optimizer may also encounter conditions at specific epochs that demand more iterations or complex calculations to converge. Additionally, system-related overheads, like memory management or concurrent processes, could contribute to these temporary spikes. The slight increase in computational load per epoch might result from subtle changes in system state, solver efficiency, or the gradual accumulation of numerical errors over time.

Figure 13 shows that the computational load of FGO increases with each new epoch. This happens because FGO needs to optimize all factors from scratch at each epoch, which requires more computing power. By the end, the computational load has grown significantly, reaching around 5320 s. This increase is due to FGO's iterative process, which re-optimizes the entire dataset at every epoch. This repeated optimization and consideration of all factors makes FGO quite resource-intensive, which makes it less practical for real-time applications where efficiency is important.

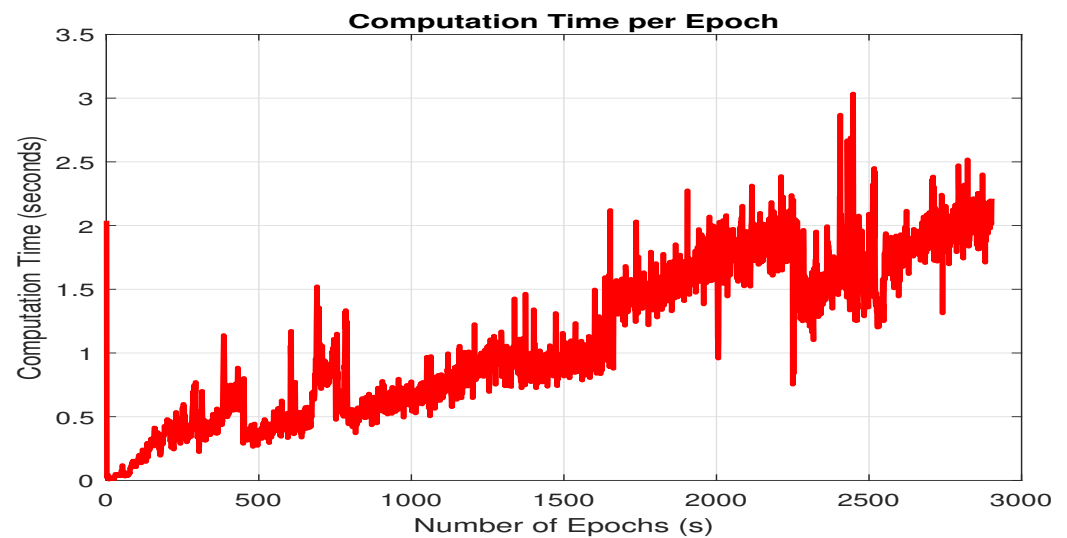


Figure 12. Computational time consumed by FGO-based PDR+GNSS-PPK fusion architecture.

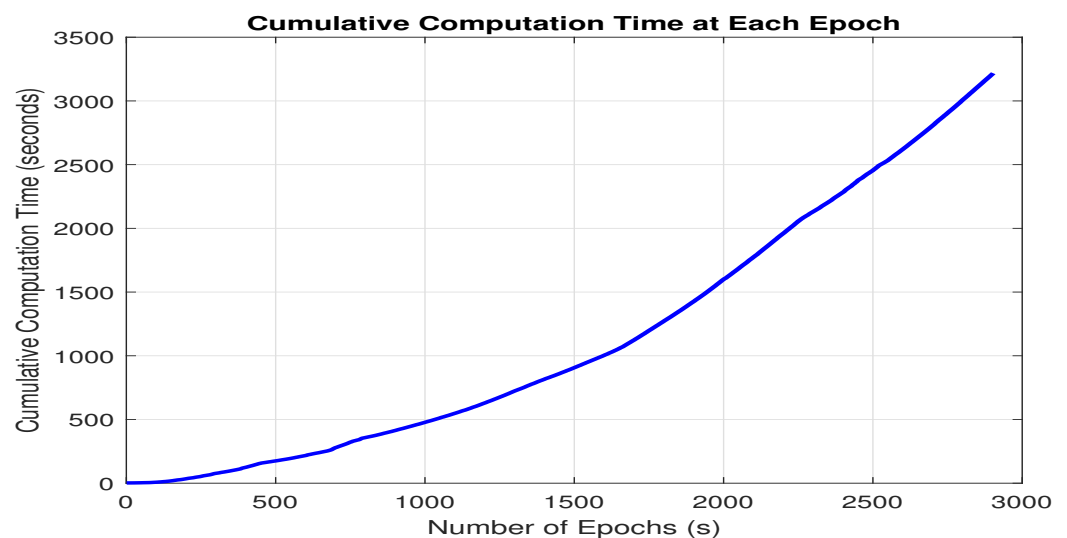


Figure 13. Cumulative computational time consumed by FGO-based PDR+GNSS-PPK fusion.

3.3. Discussion

To characterize the positional performance of PDR+GNSS-PPK with FGO, we computed the mean, median, and standard deviation (STD). Subsequently, we compared its mean % improvement with PDR+GNSS-SPP, GNSS-SPP, GNSS-PPK, KF-PDR+GNSS-SPP, and KF-PDR+GNSS-PPK, as detailed in Table 2. FGO with PDR+GNSS-PPK has achieved 22.5% better 2D mean error compared with FGO fusion with PDR+GNSS-SPP, which shows the higher capability of smartphone positioning in partial-open sky conditions, but still, for PPK, we have received positional errors in the range of 2–3 m. This is due to the fact that the fixed ambiguity solution from smartphone raw GNSS observation was very low, which is 11%. This is mainly due to the limitations of the smartphone’s low-quality chip size GNSS receiver, which resulted in continuous cycle slips across all satellites for significant portions of the data and also resulted in hardware clock discontinuities at every epoch, likely caused by the phone’s duty cycling of GNSS tracking during data collection, which can be seen in Figure 14, which shows simultaneous observations from a Google Pixel 4, where the red ticks indicate the flagged cycle slips. With few to no valid carrier phase observations in the degraded files, these solutions often exhibited very large position errors. Similarly, it was highlighted by other studies as well that a smartphone’s GNSS performance is prone to several limitations [5,34,40–43].

Table 2. Two-dimensional error analysis.

Approach	GNSS Processing Techniques						% FGO Mean Improvement over GNSS and KF
	PPK			SPP			
	Mean Error	Median Error	STD	Mean Error	Median Error	STD	
GNSS (only)	3.56 m	3.11 m	2.10 m	5.23 m	4.55 m	3.56 m	7.28% (PPK) and 18.5% (SPP)
KF-PDR+GNSS	3.44 m	2.92 m	2.04 m	4.60 m	4.00 m	3.16 m	4.36% (PPK) and 7.60% (SPP)
FGO-PDR+GNSS	3.29 m	2.88 m	2.03 m	4.25 m	3.62 m	3.02 m	as compared with SPP 22.5%

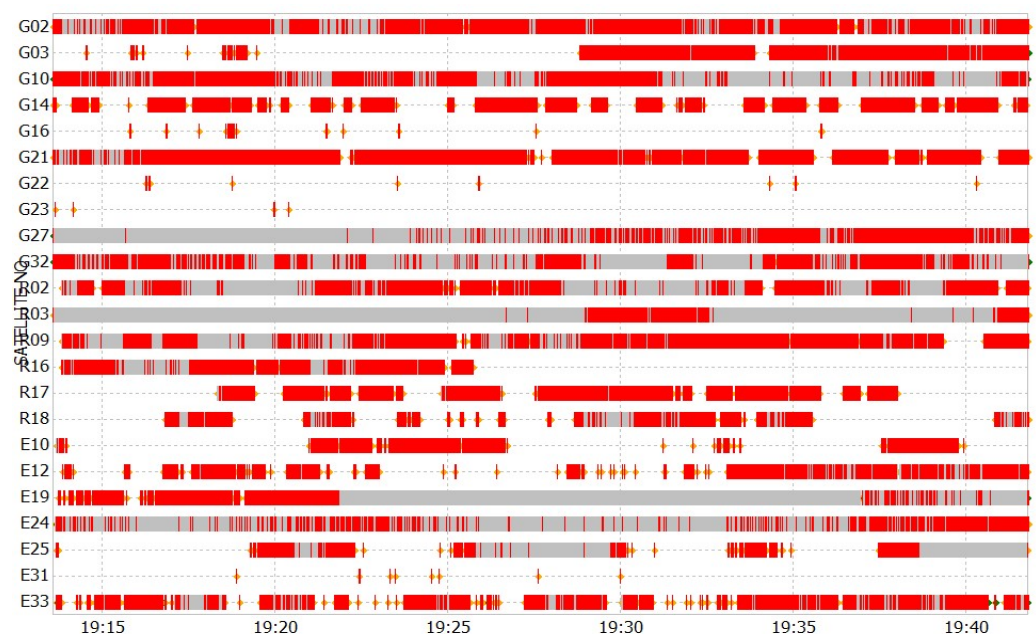


Figure 14. Google Pixel 4 raw GNSS observations.

Despite the limited corrections applied to the raw GNSS observations, we observed improved positioning accuracy compared with KF-based fusion. This enhancement is ad-

vantageous for various applications. However, FGO's computational performance is quite high, making it impractical for real-time applications. This limitation arises from FGO's batch processing mode, which optimizes the entire dataset in each iteration. A potential solution to manage this issue could be the use of a sliding window-based optimization approach. While our offline implementation showcases promising results, that can be useful for some applications. However, further examination is needed to assess its feasibility and practicality in diverse operational scenarios.

4. Conclusions and Future Work

Our study implemented GNSS–PPK and its fusion with PDR using FGO under partially open sky conditions, surrounded by homes and trees. We compared its performance with GNSS–SPP fusion with PDR via both FGO and KF, PDR+GNSS–PPK via KF, as well as only GNSS–SPP and GNSS–PPK. From the results obtained, we observe that for enhanced accuracy, the FGO-based PDR+GNSS–PPK offers a superior option for seamless localization. However, the computational load for FGO-based PDR+GNSS–PPK is much higher to be implemented in real-time in smartphones; the computational load at each epoch can be seen in Figure 12, and the overall accumulative computational time consumed can be seen in Figure 13. Moreover, it can be noted that PPK is a post-processed GNSS technique, which makes it unsuitable for real-time applications. Therefore, in the future, we plan to test our system in real-time by setting up our own base station. We believe that establishing a base station for real-time testing is a crucial next step. Consequently, we believe that the following works are essential for future research:

- (1) It is essential to explore the balance between computational load and the performance of FGO–PDR/GNSS systems. Implementing a sliding window for optimization could be a promising approach to reduce computational demands while preserving the system's performance.
- (2) We want to assess our fusion architecture's performance using more datasets in a variety of complex environments, including real-time smartphone locations and deep urban canyons, using sliding window-based FGO to maintain a healthy balance between computational complexity and accuracy.

Author Contributions: Conceptualization, A.H.M.; methodology, A.H.M. and L.E.D.; software RTKLIB Explorer 2.4.3, A.H.M.; validation, L.E.D.; writing—original draft preparation, A.H.M.; writing—review and editing, A.H.M., L.E.D., and S.K.; supervision, L.E.D. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported in part by the Research Training Grants Programme of the University of Deusto and in part by REPIN++ under Grant RED2022-134355-T.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

FGO	Factor Graph Optimization
GNSS	Global Navigation Satellite Systems
IMU	Inertial Measurement Unit
KF	Kalman Filter
NLOS	Non-Line-of-Sight
PNT	Positioning Navigation and Timing
PDR	Pedestrian Dead Reckoning
SPP	Single-Point Positioning
PPK	Post-Processing Kinematics

References

1. Paziewski, J. Recent advances and perspectives for positioning and applications with smartphone GNSS observations. *Meas. Sci. Technol.* **2020**, *31*, 091001. [\[CrossRef\]](#)
2. Weng, D.; Chen, W.; Ji, S.; Wang, J. Intelligent Urban Positioning Using Smartphone-Based GNSS and Pedestrian Network. *IEEE Internet Things J.* **2024**, *11*, 22537–22549. [\[CrossRef\]](#)
3. Banville, S.; Van Diggelen, F. Precise positioning using raw GPS measurements from Android smartphones. *GPS World* **2016**, *27*, 43–48.
4. Specht, C.; Szot, T.; Dąbrowski, P.; Specht, M. Testing GNSS receiver accuracy in Samsung Galaxy series mobile phones at a sports stadium. *Meas. Sci. Technol.* **2020**, *31*, 064006. [\[CrossRef\]](#)
5. Yun, J.; Lim, C.; Park, B. Inherent limitations of Smartphone GNSS Positioning and effective methods to increase the accuracy utilizing dual-frequency measurements. *Sensors* **2022**, *22*, 9879. [\[CrossRef\]](#)
6. Karaim, M.; Elsheikh, M.; Noureldin, A.; Rustamov, R. GNSS error sources. *Multifunct. Oper. Appl. GPS* **2018**, *32*, 137–144.
7. Burke, J. DARPA positioning, navigation, and timing (PNT) technology and their impacts on GPS users. In Proceedings of the 23rd Meeting Space-Based Positioning, Navigation, and Timing (PNT) Advisory Board, Alexandria, VA, USA, 6–7 June 2019.
8. Zhuang, Y.; Sun, X.; Li, Y.; Huai, J.; Hua, L.; Yang, X.; Cao, X.; Zhang, P.; Cao, Y.; Qi, L.; et al. Multi-sensor integrated navigation/positioning systems using data fusion: From analytics-based to learning-based approaches. *Inf. Fusion* **2023**, *95*, 62–90. [\[CrossRef\]](#)
9. Elmezayen, A.; Elsheikh, M.; Iqbal, U.; Noureldin, A. Examining the Capabilities of Smartphone-Based High Precision Positioning in Challenging Suburban Environments. In Proceedings of the 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, CO, USA, 19–23 September 2022; pp. 2329–2338.
10. Kuang, J.; Niu, X.; Chen, X. Robust pedestrian dead reckoning based on MEMS-IMU for smartphones. *Sensors* **2018**, *18*, 1391. [\[CrossRef\]](#)
11. Jiang, C.; Chen, Y.; Liu, Z.; Xia, Q.; Chen, C.; Hyyppä, J. A Probabilistic Method-Based Smartphone GNSS Fault Detection and Exclusion System Utilizing PDR Step Length. *Remote Sens.* **2023**, *15*, 4993. [\[CrossRef\]](#)
12. Zhu, F.; Tao, X.; Liu, W.; Shi, X.; Wang, F.; Zhang, X. Walker: Continuous and precise navigation by fusing GNSS and MEMS in smartphone chipsets for pedestrians. *Remote Sens.* **2019**, *11*, 139. [\[CrossRef\]](#)
13. Zhang, R.; Mi, J.; Li, J.; Wang, Q. A Continuous PDR and GNSS Fusing Algorithm for Smartphone Positioning. *Remote Sens.* **2022**, *14*, 5171. [\[CrossRef\]](#)
14. Yan, W.; Zhang, Q.; Wang, L.; Mao, Y.; Wang, A.; Zhao, C. A modified kalman filter for integrating the different rate data of gyros and accelerometers retrieved from android smartphones in the GNSS/IMU coupled navigation. *Sensors* **2020**, *20*, 5208. [\[CrossRef\]](#)
15. Li, W.; Cui, X.; Lu, M. A robust graph optimization realization of tightly coupled GNSS/INS integrated navigation system for urban vehicles. *Tsinghua Sci. Technol.* **2018**, *23*, 724–732. [\[CrossRef\]](#)
16. Wen, W.; Bai, X.; Kan, Y.C.; Hsu, L.T. Tightly coupled GNSS/INS integration via factor graph and aided by fish-eye camera. *IEEE Trans. Veh. Technol.* **2019**, *68*, 10651–10662. [\[CrossRef\]](#)
17. Wen, W.; Pfeifer, T.; Bai, X.; Hsu, L.T. Comparison of extended Kalman filter and factor graph optimization for GNSS/INS integrated navigation system. *arXiv* **2020**, arXiv:2004.10572.
18. Wen, W.; Pfeifer, T.; Bai, X.; Hsu, L.T. Factor graph optimization for GNSS/INS integration: A comparison with the extended Kalman filter. *NAVIGATION J. Inst. Navig.* **2021**, *68*, 315–331. [\[CrossRef\]](#)
19. Wen, W.; Kan, Y.C.; Hsu, L.T. Performance comparison of GNSS/INS integrations based on EKF and factor graph optimization. In Proceedings of the 32nd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2019), Miami, FL, USA, 16–20 September 2019; pp. 3019–3032.
20. Li, Q.; Zhang, L.; Wang, X. Loosely coupled GNSS/INS integration based on factor graph and aided by ARIMA model. *IEEE Sens. J.* **2021**, *21*, 24379–24387. [\[CrossRef\]](#)
21. Xin, S.; Wang, X.; Zhang, J.; Zhou, K.; Chen, Y. A Comparative Study of Factor Graph Optimization-Based and Extended Kalman Filter-Based PPP-B2b/INS Integrated Navigation. *Remote Sens.* **2023**, *15*, 5144. [\[CrossRef\]](#)
22. Boguspayev, N.; Akhmedov, D.; Raskaliyev, A.; Kim, A.; Sukhenko, A. A Comprehensive Review of GNSS/INS Integration Techniques for Land and Air Vehicle Applications. *Appl. Sci.* **2023**, *13*, 4819. [\[CrossRef\]](#)
23. Jiang, C.; Chen, Y.; Chen, C.; Jia, J.; Sun, H.; Wang, T.; Hyyppä, J. Implementation and performance analysis of the PDR/GNSS integration on a smartphone. *GPS Solut.* **2022**, *26*, 81. [\[CrossRef\]](#)
24. Jiang, C.; Chen, Y.; Chen, C.; Chen, S.; Meng, Q.; Bo, Y.; Hyyppä, J. Cooperative Smartphone GNSS/PDR for Pedestrian Navigation. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *70*, 2301–2305. [\[CrossRef\]](#)
25. Jiang, C.; Chen, Y.; Chen, C.; Jia, J.; Sun, H.; Wang, T.; Hyyppä, J. Smartphone PDR/GNSS Integration via Factor Graph Optimization for Pedestrian Navigation. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 8504112. [\[CrossRef\]](#)
26. Hussain, A.; Díez, L.E. Comparison of FGO and KF for PDR-GNSS Fusion on Smartphone for Diverse Pedestrian Movements. In Proceedings of the 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Nuremberg, Germany, 25–28 September 2023; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6.

27. Magsi, A.H.; Díez, L.E. Comparison of FGO and KF for PDR-GNSS Fusion under Different PDR Errors. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 8506312. [[CrossRef](#)]
28. Weinberg, H. Using the ADXL202 in pedometer and personal navigation applications. *Analog Devices AN-602 Appl. Note* **2002**, *2*, 1–6.
29. Li, G.; Geng, J. Characteristics of raw multi-GNSS measurement error from Google Android smart devices. *GPS Solut.* **2019**, *23*, 90. [[CrossRef](#)]
30. Fu, G.M.; Khider, M.; Van Diggelen, F. Android raw GNSS measurement datasets for precise positioning. In Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020), Online, 22–25 September 2020; pp. 1925–1937.
31. Lachapelle, G.; Gratton, P.; Horreft, J.; Lemieux, E.; Broumandan, A. Evaluation of a low cost hand held unit with GNSS raw data capability and comparison with an android smartphone. *Sensors* **2018**, *18*, 4185. [[CrossRef](#)]
32. Jiménez-Martínez, M.J.; Farjas-Abadia, M.; Quesada-Olmo, N. An approach to improving GNSS positioning accuracy using several GNSS devices. *Remote Sens.* **2021**, *13*, 1149. [[CrossRef](#)]
33. Paziewski, J.; Pugliano, G.; Robustelli, U. Performance assessment of GNSS single point positioning with recent smartphones. In Proceedings of the IMEKO TC-19 International Workshop on Metrology for the Sea, Naples, Italy, 5–7 October 2020; pp. 5–7.
34. Everett, T.; Taylor, T.; Lee, D.K.; Akos, D.M. Optimizing the use of RTKLIB for smartphone-based GNSS measurements. *Sensors* **2022**, *22*, 3825. [[CrossRef](#)]
35. Darugna, F.; Wübbena, J.; Ito, A.; Wübbena, T.; Wübbena, G.; Schmitz, M. RTK and PPP-RTK using smartphones: From short-baseline to long-baseline applications. In Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019), Miami, FL, USA, 16–20 September 2019; pp. 3932–3945.
36. Dellaert, F. *Factor Graphs and GTSAM: A Hands-On Introduction*; Technical Report; Georgia Institute of Technology: Atlanta, GA, USA, 2012.
37. Gavin, H.P. *The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems*; Department of Civil and Environmental Engineering Duke University: Durham, NC, USA, 2019; Volume 19.
38. GNSSLogger. 2016. Available online: <https://github.com/google/gps-measurement-tools> (accessed on 15 May 2024).
39. Takasu, T. RTKLIB: An Open Source Program Package for GNSS Positioning. Version 2.4.3. 2023. Available online: <https://www.rtklib.com> (accessed on 15 May 2024).
40. Zangenehjad, F.; Gao, Y. GNSS smartphones positioning: Advances, challenges, opportunities, and future perspectives. *Satell. Navig.* **2021**, *2*, 1–23. [[CrossRef](#)]
41. Wu, Q.; Sun, M.; Zhou, C.; Zhang, P. Precise point positioning using dual-frequency GNSS observations on smartphone. *Sensors* **2019**, *19*, 2189. [[CrossRef](#)]
42. Glaner, M.F.; Weber, R. Breaking the one-meter accuracy level with smartphone GNSS data. *Eng. Proc.* **2023**, *54*, 16. [[CrossRef](#)]
43. Takasu, T.; Yasuda, A. Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB. In Proceedings of the International Symposium on GPS/GNSS, Seogwipo-si, Republic of Korea, 4–6 November 2009; Volume 1, pp. 1–6.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.