remote sensing

Article

An Improved Velocity-Aided Method for Smartphone Single-Frequency Code Positioning in Real-World Driving Scenarios

Zhaowei Han [1,2], Xiaoming Wang [1,3,*], Jinglei Zhang [1], Shiji Xin [1,2], Qiuying Huang [1,2] and Sizhe Shen [1,2]

[1] Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; hanzhaowei22@mails.ucas.ac.cn (Z.H.); zhangjinglei@aircas.ac.cn (J.Z.); xinshiji21@mails.ucas.ac.cn (S.X.); huangqiuying22@mails.ucas.ac.cn (Q.H.); shensizhe22@mails.ucas.ac.cn (S.S.)

[2] School of Electronic, Electrical and Communicating Engineering, University of Chinese Academy of Sciences, Beijing 101408, China

[3] College of Resources and Environment, University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: wxm@aoe.ac.cn

Abstract: The availability of Global Navigation Satellite System (GNSS) raw observations in smartphones has driven research into low-cost GNSS solutions, especially in challenging urban environments, which have garnered significant attention from scholars in recent years. This study proposes an improved smartphone-based velocity-aided positioning method and conducts vehicle-mounted experiments in urban roads representing typical scenarios. The results show that when transitioning from low- to high-multipath environments, the number of visible satellites and carrier phase observations are highly sensitive to environmental factors, with frequent multipath effects. The introduction of robust pre-fit and post-fit residual algorithms has proven to be an effective quality control method. Additionally, using more refined observation models and appropriate parameter estimation algorithms led to a slight 6% improvement in velocity performance. The improved Kalman filter position estimation model (KFSPP-P) strategy, by incorporating velocity uncertainty into the state estimation process, overcomes the limitations of conventional velocity-aided smartphone positioning methods (KFSPP-V) in complex urban environments. In low-multipath environments, the accuracy of the KFSPP-P strategy is comparable to that of KFSPP-V, with an approximate 8% improvement in horizontal accuracy. However, in more challenging environments, such as tree-lined roads and urban environments, the KFSPP-P strategy shows significant improvements, particularly enhancing horizontal positioning accuracy by approximately 50%. These advancements demonstrate the potential of using smartphones to provide reliable positioning services in complex urban environments.

Keywords: smartphone; Doppler; time-differenced carrier phase (TDCP); velocity; Kalman filter; complex and open scene; positioning

Citation: Han, Z.; Wang, X.; Zhang, J.; Xin, S.; Huang, Q.; Shen, S. An Improved Velocity-Aided Method for Smartphone Single-Frequency Code Positioning in Real-World Driving Scenarios. *Remote Sens.* **2024**, *16*, 3988. https://doi.org/10.3390/rs16213988

Academic Editors: Li-Ta Hsu, Juliette Marais and Omar García Crespillo

Received: 23 September 2024
Revised: 23 October 2024
Accepted: 24 October 2024
Published: 27 October 2024

1. Introduction

High-precision positioning with smartphones is becoming increasingly important for advanced applications such as lane-level navigation, autonomous vehicles, and precise mapping, significantly enhancing location-based services [1]. The growing demand for accurate positioning, coupled with the release of Android 7.0 in 2016, which provided access to Global Navigation Satellite System (GNSS) raw measurements, has driven significant research into developing robust positioning algorithms for smartphones. However, achieving high positioning accuracy with smartphones is challenging due to noisy GNSS observations and susceptibility to strong multipath effects, primarily caused by low-cost chips and inferior linearly polarized (LP) antennas [2]. Numerous studies have explored the error characteristics of smartphone GNSS observations compared to those recorded by geodetic-grade receivers. It has been reported that GNSS observations from smartphones typically have a carrier-to-noise density (C/N0) that is 10 dB-Hz lower than that of survey-grade receivers [3] and exhibit rapid variations due to poor multipath suppression and non-uniform gain patterns [4]. Additionally, code noise in smartphones is about ten times larger than in geodetic receivers and contains more gross errors. Håkansson [5] also observed significant drift between code and carrier phase observations collected from a Nexus Android tablet under various multipath conditions, a phenomenon similarly noted in Xiaomi 8 and Huawei P10 devices [6].

In addition to analyzing observation data quality, numerous previous studies have demonstrated that using Real-Time Kinematic (RTK) and Precise Point Positioning (PPP) techniques can enhance smartphone positioning performance in open environments. Realini et al. [7] demonstrated that decimeter-level accuracy can be achieved with a Google/HTC Nexus 9 tablet using static short-baseline RTK under open-sky conditions. Similarly, Dabove and Di Pietra et al. [8] showed that applying the RTK technique to single-frequency GNSS observations from a smartphone could yield 0.6 m horizontal accuracy and 3 m vertical accuracy in static experiments when the smartphone was placed on an outdoor rooftop. In addition to RTK, the positioning performance of smartphones and tablets using PPP has been evaluated in several studies. Laurichesse et al. [9] achieved sub-meter-level positioning with a single-frequency PPP model using real-time orbit, clock, and ionosphere corrections, with a 1 min convergence time in a static open-sky environment, and meter-level accuracy in dynamic mode using a Samsung S8 smartphone. Wu et al. [10] reported similar performance using an ionosphere-free PPP model in a static open-sky condition after convergence. Li et al. [11] developed an uncombined PPP algorithm to address the code–carrier inconsistency observed in the P40 smartphone, testing it under open-sky conditions. Their results showed a positioning accuracy of 0.2 m in static conditions and 1 m in dynamic conditions. However, in real-world environments, smartphones are often used in kinematic modes in suburban and urban environments, where signal blockages from buildings, vehicles, and pedestrians are common. Zhang et al. [12] demonstrated that the Smart-RTK technique can achieve decimeter-level horizontal positioning accuracy under static open conditions and in relatively open walking environments. However, in vehicular urban environments, the presence of elevated highways and obstructions leads to significant fluctuations in Smart-RTK positioning results. Shinghal et al. [13] demonstrated that while the PPP technique can achieve 0.3 m 3D accuracy in static open-sky conditions, the position error can increase to approximately 7 m in kinematic experiments conducted in sub-urban environments with high multipath effects, such as parking lots, a phenomenon that has also been frequently observed with the Mi8 [14,15]. These studies demonstrate that the environment is a key factor affecting smartphone positioning. In addition, the application of RTK on smartphones is constrained by its short coverage range due to the degradation of distance-dependent biases [12]. While PPP does not require a reference station, its long convergence times are not suitable for mass market users [13,16,17].

Therefore, it is imperative to develop universally applicable methods to enhance smartphone positioning accuracy, particularly given that a substantial proportion of smartphones still rely on single-frequency code observations [18,19]. Numerous studies have shown that Doppler and Time-Differenced Carrier Phase (TDCP) measurements have been widely applied in code-based positioning algorithms [20–22]. Geng et al. [23] showed that, under open static and open dynamic conditions, Single Point Positioning (SPP) positioning accuracy improved by 2–3 m using Hatch-filtered optimization compared to traditional SPP. However, Hatch filtering relies on high-precision Doppler or carrier phase measurements to smooth code and is more suited to static or low-speed environments. This is particularly challenging for smartphones with low-cost chips, where noise and cycle slips frequently occur, impacting the final positioning accuracy [3]. Mazher et al. [24] demonstrated that compared to Hatch filtering, Kalman filtering provides better noise control and greater robustness against cycle slips in kinematic modes. Bai et al. [25] demonstrated that utilizing Doppler frequency shift to detect cycle slips and selecting corresponding epochs of carrier phase to construct TDCP loop closure can effectively improve positioning performance. Liu et al. [26] proposed an improved pedestrian navigation algorithm with velocity assistance, showing that horizontal positioning accuracy is around 2 m in open static conditions and 3–5 m in open dynamic conditions, delivering relatively continuous positioning results. However, ideal open static conditions do not reflect the real-world usage scenarios of all smartphones. Zhang et al. [3] further noted that while the TD filter can achieve continuous and smooth positioning results in open dynamic pedestrian scenarios, it encounters significant deviations in environments with tree obstructions, making it difficult to obtain continuous positioning results.

In summary, the main challenges faced by smartphones today include the limitations imposed by complex urban environments on the application of most algorithms, the disparity in frequency support that limits the use of dual-frequency PPP/RTK in smartphone positioning [27,28], and the current research on velocity-aided algorithms for smartphones, which mainly focuses on directly using velocity for state estimation, with little consideration of incorporating velocity uncertainty into the Kalman filter. To address this, this paper proposes an improved velocity-aided positioning strategy, aimed at enhancing the positioning performance of smartphones in complex urban environments. The structure of this paper is as follows: Section 2 describes the methods used in this study, including the algorithms for detecting outliers in GNSS observations, the velocity model, and the positioning model. Section 3 provides details on the experimental setup and route. Section 4 presents the analysis of the data quality, velocity performance, and positioning performance of smartphones in different scenarios. Finally, the discussion and conclusions are provided in the last section.

2. Materials and Methods

As shown in Figure 1, the Materials and Methods Section consist of three main parts: algorithms for detecting outliers in GNSS observations (data preprocessing), the Improved Doppler and TDCP Kalman Filter-Based Velocity Estimation Model (KF-DT2), and the Improved Kalman Filter Position Estimation Model (KFSPP-P). Firstly, raw GNSS observations from smartphones are collected and converted into Receiver INdependent Exchange (RINEX) format, with robust outlier detection algorithms applied to preprocess the observations and obtain robust observations. Next, the KF-DT2 strategy is used to perform Kalman filter velocity estimation by combining Doppler and TDCP observations. Finally, the constant acceleration KFSPP-P strategy, which employs a stochastic model weighted by C/N0, is used for Kalman filter-based position estimation.

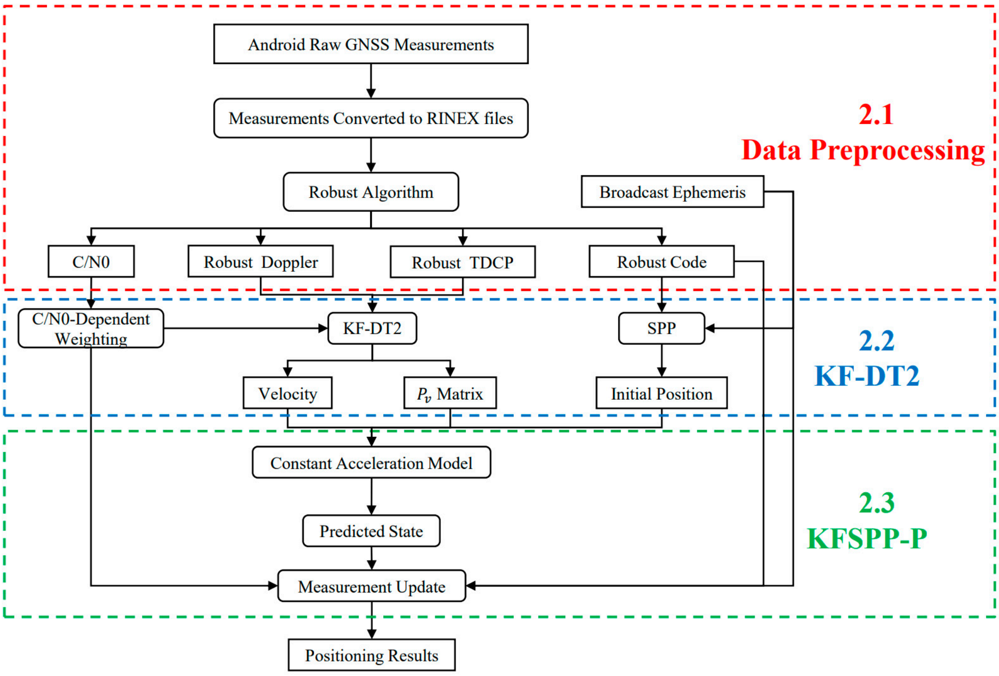

Figure 1. Flowchart of the KFSPP-P processing procedure.

2.1. Algorithms for Detecting Outliers in GNSS Observations (Data Preprocessing)

Smartphones equipped with small linearly polarized antennas and low-power GNSS chips are highly susceptible to interference from multipath effects, signal obstructions, and reflections in complex urban environments. This affects the quality of both Doppler and TDCP observations, significantly limiting their effectiveness in urban settings. Therefore, to achieve more accurate velocity estimation in such environments, it is essential to incorporate robust algorithms during velocity calculations. In response to this, we have developed a robust outlier detection algorithm based on the IQR criterion, designed to perform both pre- and post-fit residual checks. The feasibility of this algorithm is demonstrated in the experimental section of this study.

When using Doppler and TDCP observations for velocity estimation, clock drift is a crucial parameter that must be estimated. However, the frequent jumps in carrier phase observations in smartphones can lead to inconsistencies in receiver clock drift. By adjusting the FullBiasNanos parameter, the generation of carrier phase observations has been improved, effectively mitigating clock drift inconsistencies [29,30]. As a result, in our generation of carrier phase observations, this inconsistency has also been taken into account.

Inspired by [26], the receiver clock biases calculated from different satellites in the same epoch absorb measurement noise, multipath effects, and other errors. Therefore, these clock biases should cluster around the true receiver clock bias, and identifying and removing the outliers from this cluster effectively enables pre-fit quality control. Clock drift, as the rate of change in clock bias, follows the same characteristics as clock bias within small sampling intervals ($\leq$1 Hz). Thus, the receiver clock drifts computed from the same epoch should similarly cluster around the true clock drift value. In smartphones, the noise level is generally higher compared to geodetic-grade receivers, regardless of whether code, Doppler, or TDCP observations are used. TDCP observations, in particular, are affected by cycle slips, which can significantly disrupt parameter estimation despite the elimination of integer ambiguity. Therefore, pre-fit clock drift consistency checks are crucial for quality control in smartphone velocity estimation.

Firstly, the receiver's predicted velocity for the current epoch is calculated based on the approximate position coordinates derived from the current epoch and the filtered position coordinates from the previous epoch:

$$v_R = (P_t - P_{t-1})/\Delta t \tag{1}$$

where v_R represents the predicted velocity value for the current epoch; P_t represents the approximate coordinates for the current epoch; P_{t-1} represents the filtered position coordinates from the previous epoch; and Δt is the time interval. The predicted velocity value v_R is then substituted into the Doppler observation Equation (4) and the TDCP observation Equation (5) to estimate the receiver clock drift:

$$\begin{cases} clk_{Doppler} = \lambda * D - (e * v_R - e * v_S - c * \delta t_s) \\ clk_{TDCP} = \lambda * \Delta\widetilde{\Phi} - \Delta D + \Delta g + [e * (v_R * \Delta t)] \end{cases} \tag{2}$$

where $clk_{Doppler}$ and clk_{TDCP} are the clock drift values obtained from Doppler observations and carrier phase observations, respectively; D represents the Doppler observation for the current epoch; λ represents the wavelength; e represents the unit vector in the line-of-sight direction; v_R is the predicted receiver velocity for the current epoch; v_S is the satellite velocity for the current epoch; δt_s is the satellite clock drift; c represents the speed of light; Δ represents the difference operator; $\widetilde{\Phi}$ represents the carrier phase observation; ΔD represents the range variation, which is proportional to the average Doppler shift caused by the relative motion between the satellite and receiver along the line of sight; Δg accounts for the changes in relative satellite–receiver geometry due to alterations in the line-of-sight vector direction; and Δt is the time interval.

Due to the susceptibility of low-cost GNSS chips in smartphones to noise in complex environments, significant outliers can occur. If detection methods based on variance or mean (such as Pauta or 3-sigma) are used, the excessive noise or cycle slips may affect the calculation of the mean and variance, leading to incorrect outlier identification [31]. Therefore, after obtaining the receiver clock drift cluster for the current epoch, a more robust IQR outlier detection method is selected to identify the discrete values in the cluster.

$$\begin{cases} clk \in (Q_1 - C_1 * IQR,\ Q_3 + C_1 * IQR) & normal \\ clk \notin \ (Q_1 - C_1 * IQR,\ Q_3 + C_1 * IQR) & outlier \end{cases} \tag{3}$$

where clk represents the estimated receiver clock drift; Q_1 and Q_3 represent the 25th and 75th percentiles of the sorted cluster data, respectively; IQR represents the interquartile range, defined as $Q_3 - Q_1$; $(Q_1 - C_1 * IQR)$ represents the lower bound for outlier detection; and $(Q_3 + C_1 * IQR)$ represents the upper bound for outlier detection, where C_1 is typically set to 1.5 [32].

In the process described above, only the pre-fit outlier estimation is completed, which is effective for removing obvious outliers. However, some unmodeled residuals may still exist in certain observations, requiring post-fit data quality control after parameter estimation. We apply an iterative cycle algorithm based on absolute value IQR testing to the post-fit residuals. When a satellite is flagged as invalid during post-fit testing, it is removed, and parameter estimation is repeated. This process continues until no more satellites are flagged, at which point the final results are produced. Additionally, a similar outlier detection method was applied to the code observations to ensure that only clean data were utilized for positioning.

2.2. Improved Doppler and TDCP Kalman Filter-Based Velocity Estimation Model (KF-DT2)

Doppler and TDCP observations are crucial for smartphones in calculating accurate velocity information using raw GNSS data [29,33]. The phenomenon in which the frequency of the received signal changes due to the relative motion between the signal transmitter and receiver is called the Doppler effect. The relative velocity between the signal source and receiver can be derived from the Doppler frequency shift, and the instantaneous velocity can be calculated using Doppler observations. The observation equation is as follows:

$$\lambda * D = -e * (v_S - v_R) + c * \delta t_u - c * \delta t_s + \delta d_{iono} + \delta d_{trop} + \eta \tag{4}$$

where δt_u represents the receiver clock drift; η includes multipath error and noise; and δd_{iono} and δd_{trop} represent the rate of change in ionospheric and tropospheric delays over time. Since the actual sampling interval is short ($\leq$1 Hz), the changes in ionospheric and tropospheric delays are very slow and, thus, the rate of change for these delays can be neglected. The unknown parameters in Equation (4) include the receiver's three-dimensional velocity and clock drift, totaling four unknowns.

The TDCP observations are derived from the carrier phase differences between epochs at the same frequency for the same satellite within a small sampling interval. This process eliminates the impact of integer ambiguities. Similar to Doppler observations, under a small sampling interval ($\leq$1 Hz), the effects of the ionosphere and troposphere can be neglected. The observation equation is as follows:

$$\lambda * \Delta\overset{\sim adj}{\Phi} = \lambda * \Delta\tilde{\Phi} - \Delta D + \Delta g = -[e * \Delta r_u] + c * \Delta\delta t_u + \Delta\varepsilon \tag{5}$$

where $\Delta\overset{\sim adj}{\Phi}$ represents the adjusted TDCP observations; Δr_u represents the receiver position change; $\Delta\delta t_u$ represents the receiver clock drift; and $\Delta\varepsilon$ includes multipath errors and noise. The unknowns include the receiver's position change in three dimensions and the receiver clock drift, totaling four unknowns, which can be used to calculate the average velocity between epochs t_k and t_{k-1} using the following equation:

$$\overline{v_{TDCP}} = \Delta r_u / (t_k - t_{k-1}) \tag{6}$$

where $\overline{v_{TDCP}}$ represents the average velocity between epochs t_k and t_{k-1}.

From Equations (4) and (5), it can be observed that at epoch t_k, the velocity calculated using Doppler observations is the instantaneous velocity $v_{Doppler}$ at epoch t_k, while the velocity calculated using TDCP observations is the average velocity $\overline{v_{TDCP}}$ between epochs t_k and t_{k-1}. In the case of small sampling intervals, it is generally assumed that the average velocity between epochs t_k and t_{k-1} is equivalent to the instantaneous velocity at epoch t_k. However, in complex real-world vehicle scenarios, the average velocity cannot always be fully equated to the instantaneous velocity. Therefore, based on the use of velocity and acceleration in the constant acceleration model within dynamic Kalman filtering, we modified the Doppler and TDCP observation equations. When the constant acceleration model is employed for position state estimation in the Kalman filter, the position at epoch t_k is predicted using the position result at epoch t_{k-1} and the velocity result at epoch t_{k-1}:

$$x_k = x_{k-1} + v_{k-1} * \Delta t + \frac{1}{2} * a_{k-1} * \Delta t^2 \tag{7}$$

where x_k is the predicted position at the current epoch; x_{k-1} is the filtered position result at the previous epoch; v_{k-1} and a_{k-1} are the instantaneous velocity and instantaneous acceleration at the previous epoch; and Δt is the time interval between epochs t_k and t_{k-1}. The average velocity between epochs t_k and t_{k-1} can be expressed as follows:

$$\overline{v} = (v_k + v_{k-1})/2 \tag{8}$$

where $\overline{v}$ represents the average velocity between epochs t_k and t_{k-1}; v_k represents the instantaneous velocity at epoch t_k; and v_{k-1} represents the instantaneous velocity at epoch t_{k-1}. According to the constant acceleration model

$$v_k = v_{k-1} + a_{k-1} * \Delta t \tag{9}$$

where a_{k-1} represents the instantaneous acceleration at epoch t_{k-1}. According to Equation (6)

$$\overline{v} = \overline{v_{TDCP}} \tag{10}$$

From Equations (8)–(10)

$$\overline{v_{TDCP}} = v_{k-1} + \frac{1}{2} * a_{k-1} * \Delta t \tag{11}$$

Similarly, the instantaneous velocity $v_{Doppler}$ at epoch t_k can be calculated from the Doppler observations at epoch t_k. According to the acceleration model

$$v_{Doppler} = v_{k-1} + a_{k-1} * \Delta t \tag{12}$$

Through Equations (11) and (12), the acceleration is introduced as an estimated parameter into both the Doppler and TDCP observation equations.

In the TDCP observation equation

$$\frac{\Delta r_u}{\Delta t} = \overline{v_{TDCP}} = v_{k-1} + \frac{1}{2} * a_{k-1} * \Delta t \tag{13}$$

In the Doppler observation equation

$$v_R = v_{Doppler} = v_{k-1} + a_{k-1} * \Delta t \tag{14}$$

To refine the estimation approach, we conducted a comparative analysis of the accuracy of velocity estimates obtained from different methodologies—least squares and Kalman filter—using various observational data, including Doppler and TDCP observations. Further details on the velocity estimation strategies are presented in Table 1. An outlier detection approach was applied across all five estimation methods.

Table 1. Five velocity processing strategies and descriptions.

Strategy	Observation	Description
LS-D	Doppler	Velocity for each epoch estimated using least squares method
LS-T	TDCP	Velocity for each epoch estimated using least squares method
LS-DT	TDCP + Doppler	Velocity for each epoch estimated using least squares method
KF-DT1	TDCP + Doppler	Velocity for each epoch estimated using Kalman filer method with traditional constant acceleration model
KF-DT2	TDCP + Doppler	Velocity for each epoch estimated using Kalman filer method with enhanced constant acceleration model

2.3. Improved Kalman Filter Position Estimation Model (KFSPP-P)

In the Kalman filter, it is assumed that data between epochs are uncorrelated. Therefore, using the average velocity of two epochs as an estimator violates the independence assumption [34]. In Section 2.2, the instantaneous velocity and acceleration of the previous epoch are solved through the Doppler and TDCP observations of the current epoch, ensuring the independence assumption is not violated. Additionally, in the Kalman filter, the diagonal elements of the covariance matrix P describe the variance of each state, while the off-diagonal elements represent the covariances between different states. In Section 2.2, within the velocity estimation KF-DT2 strategy, in addition to obtaining the current epoch's state vector x_v, the covariance matrix P, which is associated with the instantaneous velocity and acceleration, is also derived:

$$x_v = \begin{bmatrix} v \\ a \\ d \end{bmatrix} \; P = \begin{bmatrix} P_{vv} & P_{va} & P_{vd} \\ P_{va} & P_{aa} & P_{ad} \\ P_{vd} & P_{ad} & P_{dd} \end{bmatrix} \tag{15}$$

where v, a, and d represent the velocity, acceleration, and clock drift state variables, respectively; P_{xx} is the diagonal covariance matrix (where x can be v, a, or d); and P_{xy} is the off-diagonal covariance matrix (where x and y can be v, a, or d, and $x \neq y$).

The covariance matrix P reflects the error characteristics of the current state estimate. A larger value in the P matrix indicates greater uncertainty in the Kalman filter's estimation of the current state, while a smaller P suggests that the system's state estimate is relatively reliable. For smartphones in dynamic scenarios, changes in velocity and acceleration are typically gradual within short sampling intervals (1 Hz), with no sudden jumps exceeding 5 m/s. Therefore, when using the Kalman filter for position prediction, employing a constant acceleration model is more reliable than a constant velocity model, as shown in Equation (7).

In the position filter, the state vector x_p and the covariance matrix $\overline{P}$ obtained after the filter update from the previous epoch are given as follows:

$$x_p = \begin{bmatrix} \overline{p} \\ \overline{v} \\ \overline{a} \\ \overline{d} \end{bmatrix} \; \overline{P} = \begin{bmatrix} \overline{P_{\overline{pp}}} & \overline{P_{\overline{pv}}} & \overline{P_{\overline{pa}}} & \overline{P_{\overline{p\bar{d}}}} \\ \overline{P_{\overline{pv}}} & \overline{P_{\overline{vv}}} & \overline{P_{\overline{va}}} & \overline{P_{\overline{v\bar{d}}}} \\ \overline{P_{\overline{pa}}} & \overline{P_{\overline{va}}} & \overline{P_{\overline{aa}}} & \overline{P_{\overline{a\bar{d}}}} \\ \overline{P_{\overline{p\bar{d}}}} & \overline{P_{\overline{v\bar{d}}}} & \overline{P_{\overline{a\bar{d}}}} & \overline{P_{\overline{dd}}} \end{bmatrix} \tag{16}$$

where $\overline{p}$, $\overline{v}$, $\overline{a}$, and $\overline{d}$ represent the position, velocity, acceleration, and clock bias state variables, respectively; $\overline{P_{\overline{xx}}}$ is the diagonal covariance matrix (where x can be p, v, a, or d); and $\overline{P_{\overline{xy}}}$ is the off-diagonal covariance matrix (where x and y can be p, v, a, or d, and $x \neq y$).

In Equation (15), the velocity v and acceleration a, solved through the velocity Kalman filter, as well as the covariance matrix between the velocity and acceleration, P_{vv}, P_{va}, and P_{aa}, are included. This velocity-related information is incorporated into the state vector and covariance matrix of the position Kalman filter in Equation (16). The resulting formulation is as follows:

$$x_p = \begin{bmatrix} \overline{p} \\ v \\ a \\ \overline{d} \end{bmatrix} \ \overline{P} = \begin{bmatrix} \overline{P_{pp}} & 0 & 0 & \overline{P_{pd}} \\ 0 & P_{vv} & P_{va} & \overline{P_{vd}} \\ 0 & P_{va} & P_{aa} & \overline{P_{ad}} \\ \overline{P_{pd}} & \overline{P_{vd}} & \overline{P_{ad}} & \overline{P_{dd}} \end{bmatrix} \tag{17}$$

The covariance matrix $\acute{P}$ describing the state of the current epoch is obtained as follows:

$$\acute{P} = FPF^T + Q \tag{18}$$

where

$$F = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \ Q = \sigma_a^2 \begin{bmatrix} \frac{1}{20}\Delta t^4 & \frac{1}{8}\Delta t^3 & \frac{1}{6}\Delta t^2 \\ \frac{1}{8}\Delta t^3 & \frac{1}{3}\Delta t^2 & \frac{1}{2}\Delta \mathrm{t} \\ \frac{1}{6}\Delta t^2 & \frac{1}{2}\Delta \mathrm{t} & 1 \end{bmatrix} \tag{19}$$

where F is the state transition matrix; Q is the process noise matrix [12]; and σ_a^2 is the a priori variance.

In the constant acceleration model of the Kalman filter, process noise is added to the acceleration and propagated through the state transition matrix, thereby introducing white noise to both velocity and position estimates. During the velocity estimation in the Kalman filter, the uncertainty of velocity and acceleration is already well characterized by the covariance matrix P. When the instantaneous velocity and acceleration, as computed in Section 2.2, are used for state estimation, the true velocity covariance is incorporated into the position estimate through the state transition in Equation (17), followed by the measurement update to output the final position. It is important to note that the velocity estimation process in Section 2.2 and the position estimation process here are treated as the same process. The acceleration's white noise is transferred to the velocity and position covariance matrices via the state transition matrix, and the acceleration variance σ_a^2 used in both processes remains consistent.

Accurate velocity estimation is a key element of the KFSPP-P strategy. To analyze the improved velocity-aided KFSPP-P strategy, we conducted a comparative analysis of position results obtained through different methods, including the SPP strategy and KFSPP-V strategy. The SPP solution employs a least squares model weighted by C/N0. The detailed method for using the C/N0 stochastic model can be found in reference [26,35]. The KFSPP-V strategy represents a common velocity-aided smartphone positioning approach [3,26,34], relying on the velocity information derived from the KF-DT2 strategy for state estimation. For the KFSPP-V strategy, the covariance matrix P is updated by adding process noise Q and performing measurement updates, as described in Equation (14). In contrast, the KFSPP-P strategy incorporates the velocity covariance matrix P when using the velocity derived from KF-DT2 for state estimation, as shown in Equation (15). Apart from this, the KFSPP-P and KFSPP-V strategies use the same data processing methods, including ionospheric and tropospheric correction models and ephemeris data, as summarized in Table 2. As discussed in Section 4.1, we assigned different weights to each GNSS during the solution process, considering the varying impacts of multipath interference across different systems.

Table 2. Processing strategies for positioning.

Item	Model
GNSS measurements	GPS (L1), BDS (B1I), Galileo (E1), GLONASS (G1)
System weighting	GPS:BDS:Galileo:GLONASS = 1:1:1:1.5
Sampling interval	1 s
C/N0	20 dB-Hz
Functional model	SPP model
Elevation cut-off angle	10°
Ionospheric delay	BRDC model
Tropospheric delay	Saastamoinen model
Kalman filter	Kinematic
Stochastic model	C/N0 and elevation weighting
Ephemeris	Broadcast ephemeris
Satellite and Receiver antenna Phase center	PCO and PCV values from igs14.atx

3. Data Collection and Experimental Design

To evaluate the proposed Doppler and TDCP Kalman filter-based velocity estimation model and the improved Kalman filter position estimation model, this study employed several smart devices, including the Samsung S21 (Seoul, Republic of Korea), Tecno CL8 (Shenzhen, China), and Tecno AD11 (Shenzhen, China), all of which support single-frequency tracking of GPS, BDS, Galileo, and GLONASS constellations. As shown in Figure 2 the smartphones were placed facing the user to simulate the real behavior of a driver, while a dashcam was mounted on the left side to record changes in the driving environment. Gnsslogger 3.0.6.3 version, which was developed by Google in 2016 and provides users with observations in Comma-Separated Values (CSV), RINEX, or National Marine Electronics Association (NMEA) format, was used as the GNSS raw data collection software with a sampling frequency of 1 Hz. Although Gnsslogger can directly convert raw GNSS observation data into RINEX format, it generates inconsistent code and carrier phase observations, which can affect outlier detection and the final positioning performance. Therefore, we utilized the RINEX data conversion software, UofC-CSV2RINEX tool (https://github.com/FarzanehZangeneh/csv2rinex, accessed on 6 April 2024), developed by the Farzaneh's team [30]. Given that an increasing number of smartphones have eliminated the duty cycle mechanism [26], we disabled the duty cycle function on all smartphones used in the experiment.

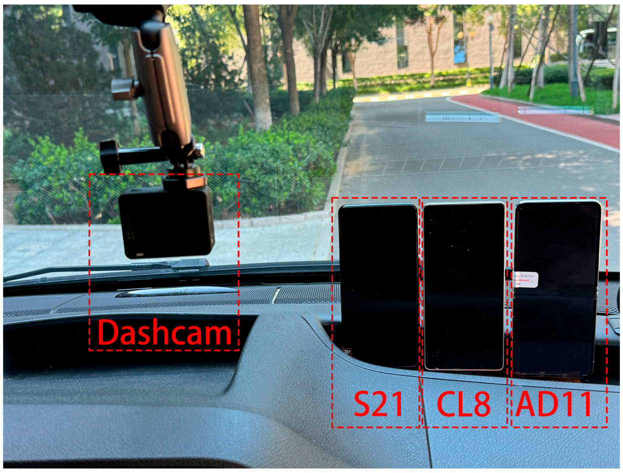

Figure 2. Installation method of smartphones in the vehicle. From left to right are S21, CL8, and AD11. The dashcam is shown in the upper left corner.

As shown on the left side of Figure 3, the experimental route is located in Haidian District, Beijing, with a total length of 20.6 km, taking approximately 41 min to complete.

Based on the actual driving conditions and the classification in the literature [36], we categorized the driving environment into four types: open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments, represented on the map in blue, red, green, and yellow, respectively. In our complex scenarios, the roads are narrower, and there are more obstructions. The definitions are shown in Table 3:

Table 3. The Types and Definitions of the Four Scenarios.

Type	Definition
(A) open-sky road	Main roads with relatively open views, with obstructions from trees on one side in most cases
(B) suburban	Side roads with obstructions from buildings or trees on one side
(C) tree-lined road	Main roads with significant obstructions from trees on both sides
(D) urban	Main roads with obstructions from buildings on both sides, and continuous passage through five urban overpasses, each 5–10 m wide

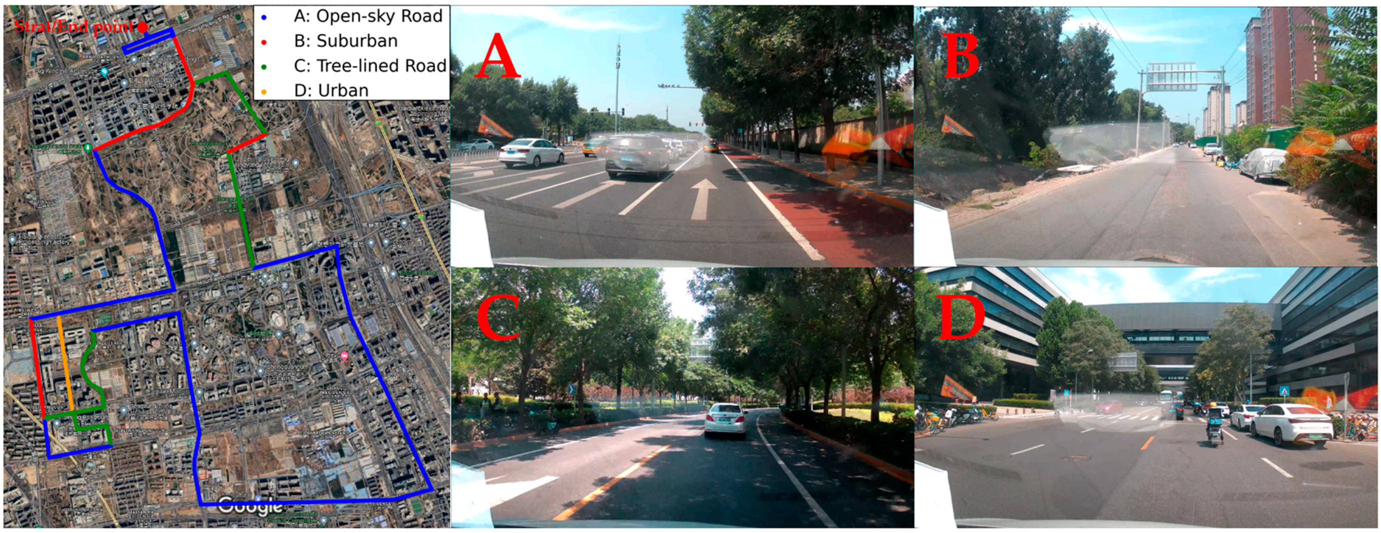

Figure 3. The experimental route trajectory is shown on the left, with blue, red, green, and yellow corresponding to open-sky road, suburban, tree-lined road, and urban environments, respectively. The right image depicts the actual environments corresponding to open-sky road (**A**), suburban (**B**), tree-lined road (**C**), and urban (**D**).

A tactical-grade Inertial Measurement Unit (IMU) device, the ISA100C, was installed at the rear of the vehicle in conjunction with a NovAtel SPAN geodetic receiver. The setup had a sampling frequency of 200 Hz, a gyro bias of 0.5°/h, an accelerometer linearity/scale factor of 100 ppm, and angular and velocity random walks of $0.03^\circ/\sqrt{\mathrm{h}}$ and $0.1\ \mathrm{m/s}/\sqrt{\mathrm{h}}$, respectively. Figure 4 shows the schematic diagram of the in-vehicle experimental setup: smartphones were mounted at the front, the GNSS receiver antenna was placed on the roof, and the ISA100C was located in the rear trunk. Additionally, a reference station equipped with a Septentrio PolaRx5 GNSS receiver (Belgium, Europe) was set up on the rooftop of Building D at the New Technology Base of the Chinese Academy of Sciences. The reference station used Inertial Explorer version 8.9 software for postprocessing to compute a tightly coupled RTK-IMU solution as the reference coordinates for the vehicular experiment, providing centimeter-level accuracy [37]. To facilitate comparison with the GNSS results from the smartphones, the timestamps of each IMU sensor were aligned with GPS time to achieve time synchronization. The geometric offset between the center of the smartphone's GNSS antenna and the center of the ISA100C device was accurately measured to calibrate lever-arm offset, achieving spatial synchronization between the smartphone and the IMU.

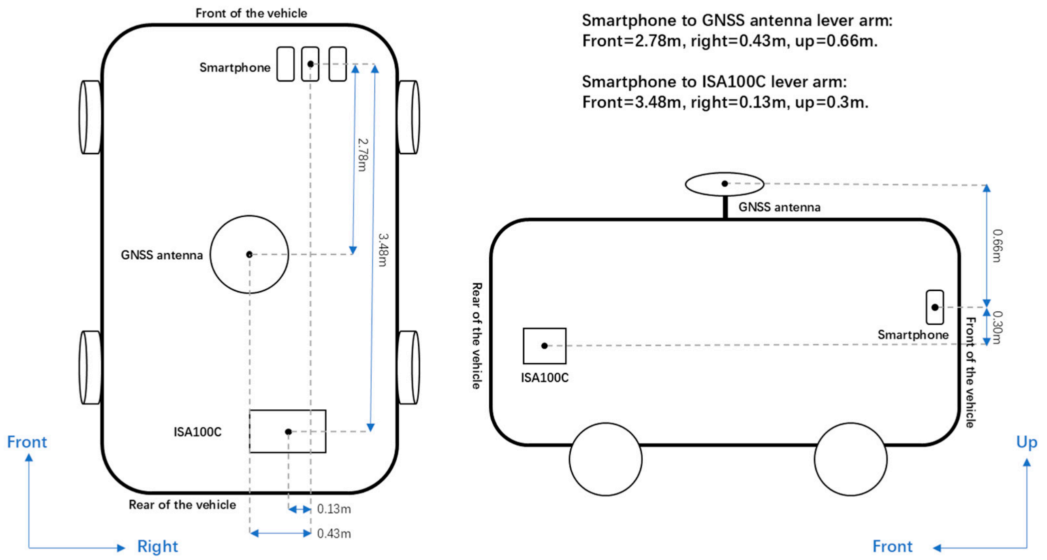

Figure 4. The schematic diagram of the in-vehicle experimental setup. The lever arm of the smartphone to GNSS antenna is front = 2.78 m, right = 0.43 m, and up = 0.66 m; the lever arm of the smartphone to ISA100C is front = 3.48 m, right = 0.13 m, and up = 0.3 m.

4. Results

This section investigates the observation quality of smartphones in complex environments, focusing on the number of visible satellites and code, Doppler, and TDCP observations; Time-Differenced Code-Minus-Carrier (TDCMC) values; and outlier detection rates. The observation quality of the smartphones used in the experiment was analyzed across different environments, including open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments. Subsequently, five velocity estimation strategies, LS-D, LS-T, LS-DT, KF-DT1, and KF-DT2, were compared in terms of their accuracy. Finally, three positioning processing modes, SPP, a common velocity-aided smartphone positioning approach (KFSPP-V), and KFSPP-P, were evaluated, with their performance assessed through root mean square (RMS) analysis in different environments.

4.1. Analysis of the Quality of Raw Smartphone Observations in Different Scenarios

Examining observation quality during the transition from open-sky road (A) to urban (D) environments is essential, as this shift typically reduces satellite visibility, degrades geometry, and exacerbates noise and multipath effects. This section explores the influence of different environments on observation quality, emphasizing satellite availability, geometry, noise, and multipath, and their impact on positioning accuracy. Figure 5 illustrates variations in the number of visible satellites and the position dilution of precision (PDOP) for the S21 smartphone throughout the entire route. Under open-sky observation environments, approximately 23 satellites are visible, with PDOP values remaining below 1.5. However, in high-multipath environments, such as suburban/urban environments or areas with dense tree coverage, the number of visible satellites frequently decreases to around 15 due to the signal blockage from buildings or trees, with PDOP values often exceeding 2. This indicates that, in high-multipath environments, smartphones experience a loss of 5–7 satellites on average, resulting in an increase in PDOP values, highlighting the difficulties in maintaining stable satellite tracking under multipath conditions.

Figure 6 further compares the variations in the number of code, Doppler, and carrier phase observations under different environments for the S21 along the test route. In the left panel, the red and blue lines representing code and Doppler observations nearly overlap, with deviations of only 2–3 observations in some cases, while the green line for carrier phase observations shows significant divergence, often falling below 10. The average number of code observations was 22.72, slightly higher than Doppler observations at 21.73, while carrier phase observations averaged only 16.31, significantly lower than both. As shown in the right panel of Figure 6, carrier phase observations are highly sensitive to environmental conditions, dropping from approximately 21 under open-sky road (A) to just 5 in urban (D) environments. By contrast, code and Doppler observations exhibit minimal variation across environments, maintaining approximately 20 observations even in urban (D) environments. This characteristic is also observed in the CL8 and AD11 devices, as shown in Table 4. This phenomenon further underscores the importance of utilizing code and Doppler observations for smartphone positioning and reveals the vulnerability of carrier phase-based positioning under high-multipath environments, such as in urban areas.

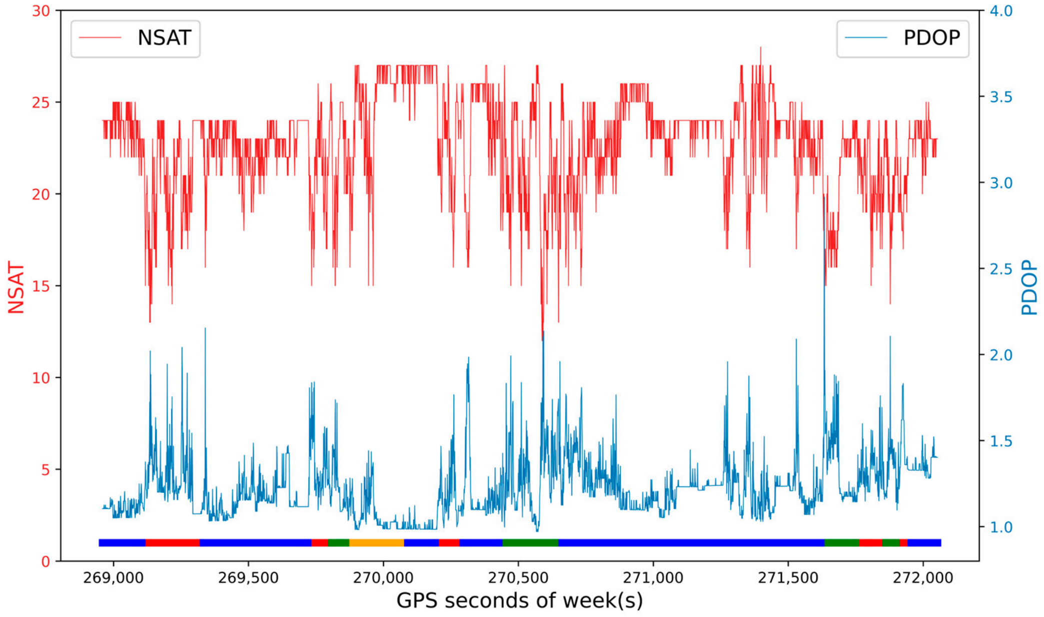

Figure 5. Number of satellites and PDOP values for the S21 on the test route. The color blocks located at the bottom of the image represent changes in environmental scenes: blue, red, green, and yellow correspond to open-sky road (A), suburban (B), tree-lined road (C), and urban (D), respectively.

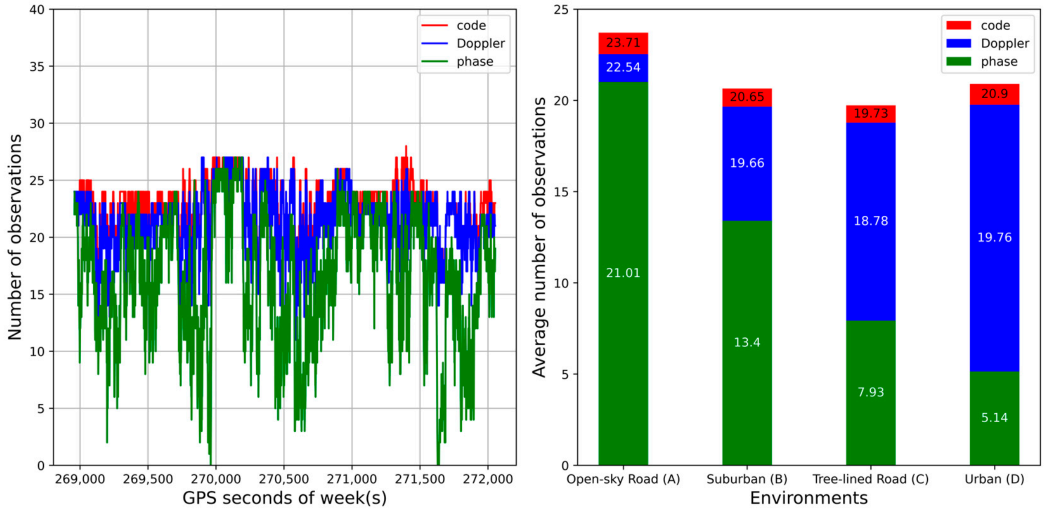

Figure 6. The number of code (red), Doppler (blue), and phase (green) observations recorded by the S21 smartphone along the experimental trajectory (**left**); the average number of each observation type in open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments (**right**).

Table 4. The number of code, Doppler, and phase observations for the S21, CL8, and AD11 in open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments.

Scenario	S21 (%)				CL8 (%)				AD11 (%)			
	A	B	C	D	A	B	C	D	A	B	C	D
Code	23.71	20.65	19.73	20.9	39.45	39.22	37.57	34.58	38.46	36.81	36.03	29.66
Dopler	22.54	19.66	18.78	19.76	32.12	33.58	30.37	27.28	33.08	33.62	30.01	25.19
Phase	21.01	13.40	7.93	5.14	24.15	22.61	18.12	15.04	19.80	10.72	6.48	2.66

While the number of code observations remains relatively consistent across varying conditions, their quality can be substantially degraded in high-multipath environments. To further investigate the impact of multipath along the test route, TDCMC values were calculated for GPS, GLONASS, Galileo, and BDS. TDCMC is an effective metric for describing the impact of multipath effects, cycle slips, and noise on observation quality, as outlined in [38]. As shown in Figure 7, the TDCMC results for individual satellites are represented by different colors, offering insights into the multipath dynamics across different environments. The code observations are clearly susceptible to multipath effects, with TDCMC values reaching 40–60 m in urban environments, indicating degraded code quality due to multipath interference. Among the four GNSSs, GLONASS exhibits the highest TDCMC values, reflecting the poorest code quality compared to the others. The average TDCMC values for GPS, Galileo, and BDS are 1.44 m, 2.05 m, and 3.41 m, respectively, while GLONASS has an average TDCMC of 7.00 m. This demonstrates that GLONASS is significantly noisier than the other systems, highlighting the need for careful system weighting in multi-GNSS combinations to optimize performance.

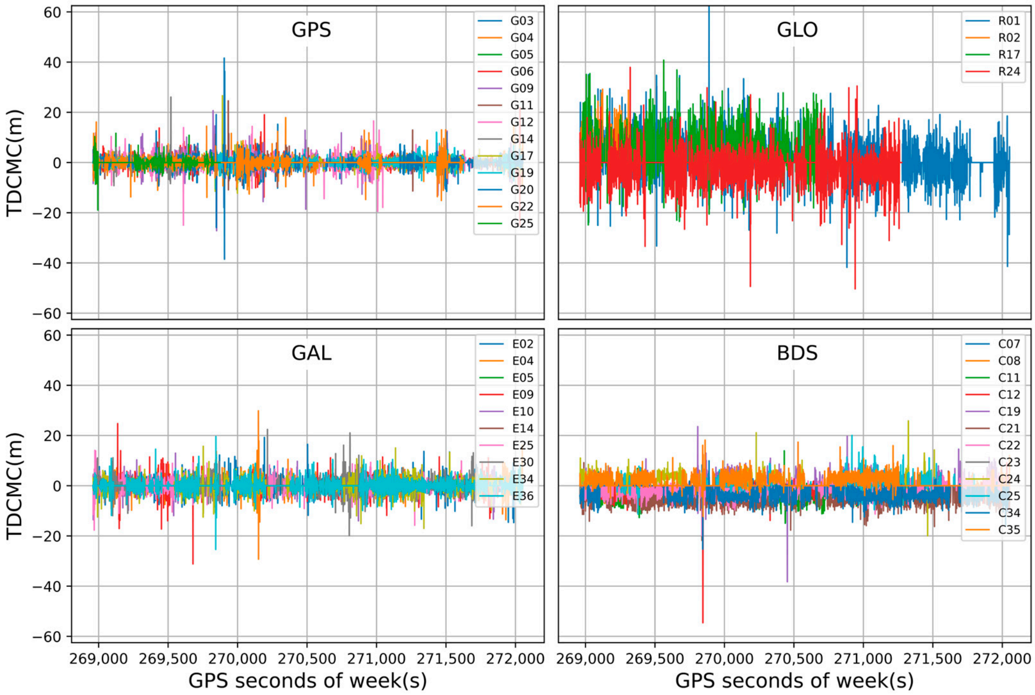

Figure 7. TDCMC statistics for GPS, Galileo, BDS, and GLONASS systems recorded by the S21 smartphone along the experimental trajectory, with different colors representing individual satellites.

Given the high sensitivity of observations in smartphones to multipath effects, employing effective and robust outlier detection strategies during positioning or velocity estimation becomes crucial. Figure 8 shows the distribution of post-fit residuals during Doppler-based velocity estimation using S21, with and without the use of a robust algorithm. The upper data reveal that in multipath environments without outlier detection, Doppler is highly susceptible to interference. By applying a robust algorithm, post-fit residuals in dynamic high-multipath environments can be controlled within 0.4 m/s, and within 0.05 m/s in low-multipath environments. Additionally, the use of robust processing results in cleaner data, playing a significant role in improving estimation accuracy, particularly in high-multipath environments. Further details on the benefits of adopting robust outlier detection strategies will be explained in Section 4.2.

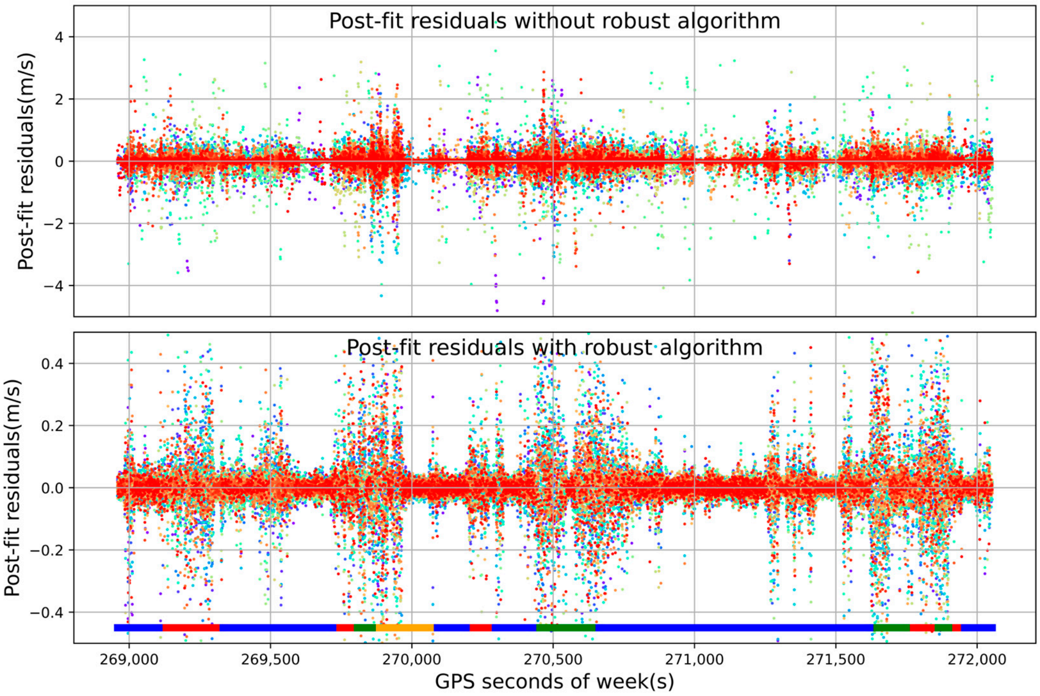

Figure 8. Distribution of post-fit residuals during Doppler-based velocity estimation using the S21, showing results without (**top**) and with (**bottom**) robust estimation algorithms applied, with different colors representing different satellites. Notably, the y-axis scale range is −4 to 4 m/s (**top**) and −0.4 to 0.4 m/s (**bottom**). The color blocks located at the bottom of the image represent changes in environmental scenes: blue, red, green, and yellow correspond to open-sky road (A), suburban (B), tree-lined road (C), and urban (D), respectively.

Table 5 shows the percentage of outliers detected in Doppler and TDCP observations across different environments for the three devices, using the methods we propose. Overall, both the Doppler and TDCP observations exhibit a clear increase in the percentage of outliers as the environment transitions from open-sky road (A) to urban (D) environments. Additionally, Doppler consistently shows a higher percentage of outliers compared to TDCP across all environments. However, despite the higher percentage of outliers in Doppler observations, the total number of Doppler observations significantly exceeds that of TDCP, meaning we still have more Doppler observations overall.

Table 5. Outliers detected in Doppler and TDCP observations for the S21, CL8, and AD11 devices after applying the robust estimation algorithm in open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments.

Scenario	S21 (%)		CL8 (%)		AD11 (%)	
	Doppler	TDCP	Doppler	TDCP	Doppler	TDCP
A	11.93	7.48	27.66	16.08	22.94	2.25
B	21.12	8.25	30.10	18.06	27.39	6.45
C	19.48	14.3	34.86	17.28	30.63	7.45
D	29.90	15.5	36.69	23.97	38.38	8.97

4.2. Velocity Performance Evaluation

As discussed in Section 4.1, GNSS observations are highly sensitive to high-multipath environments, leading to a significant number of erroneous observations. If these errors are not properly identified and removed, they can severely compromise the accuracy of velocity estimates. To quantitatively evaluate the impact of these outliers, velocity estimates for the S21 smartphone were estimated both with and without adopting the outlier detection procedure proposed in Section 2.1. The velocity errors calculated using the velocity from ISA100C as a reference are presented in Figure 9. The results clearly demonstrate that, without applying the outlier detection procedure to the Doppler observations, the velocity estimates contain frequent large errors, highlighting the strong susceptibility of Doppler observations to multipath interference. As shown in Table 6, the velocity estimates achieve the highest accuracy in open-sky road (A) environments but experience a significant decrease in accuracy in tree-lined road (C) or urban (D) environments for both methods. However, the adoption of the outlier detection procedure consistently yields superior results compared to the method without outlier detection across all scenarios, including open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments. After removing outliers from Doppler observations, centimeter-level accuracy is achievable in open-sky road (A) environments, with decimeter-level accuracy maintained in high-multipath environments. This underscores the critical importance of implementing effective error detection and removal in GNSS observations to preserve the accuracy of velocity estimates, particularly in high-multipath environments.

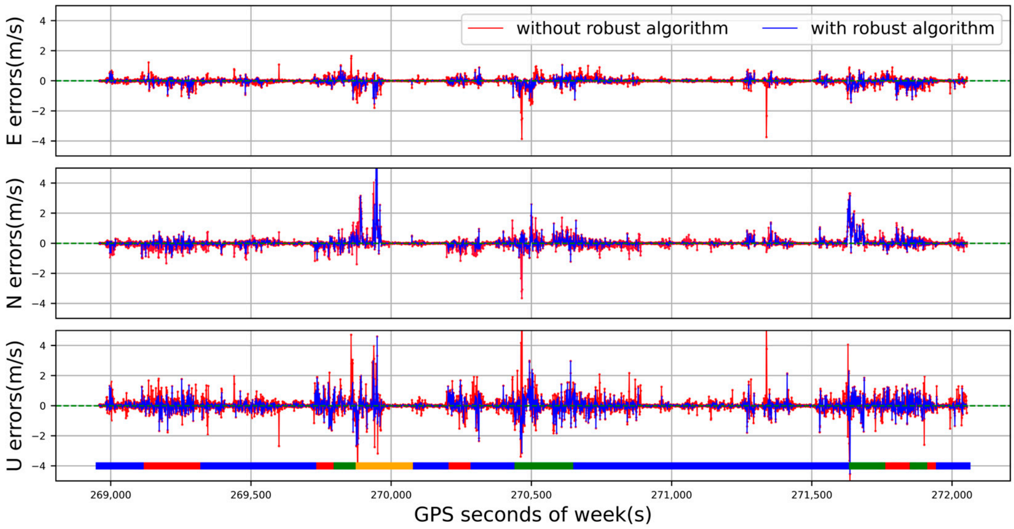

Figure 9. Doppler velocity estimation errors for the S21, without (red) and with (blue) the robust estimation algorithm applied. The color blocks located at the bottom of the image represent changes in environmental scenes: blue, red, green, and yellow correspond to open-sky road (A), suburban (B), tree-lined road (C), and urban (D), respectively.

Table 6. RMS values of Doppler velocity accuracy for the S21 without and with the robust estimation algorithm in open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments, as well as for the entire route.

S21	Direction	A (cm/s)	B (cm/s)	C (cm/s)	D (cm/s)	ALL (cm/s)
Without robust algorithm	E	5.03	27.47	55.01	48.80	27.01
	N	12.26	33.56	62.16	125.61	41.34
	U	17.95	56.73	118.34	137.02	61.98
With robust algorithm	E	2.58	18.82	30.18	32.16	17.38
	N	4.45	22.91	38.84	92.74	30.59
	U	8.87	38.86	72.11	82.44	40.70

Figure 10 presents the accuracy of velocity estimates using different strategies across three devices: the S21, CL8, and AD11 in East (red), North (blue) and Up (green) directions. The figure shows that velocity estimates from both LS-D and LS-T strategies experience decreased accuracy as multipath effects increase, particularly in tree-lined road (C) or urban (D) environments. In high-multipath environments, the number of GNSS observations may drop below four due to signal blockage by buildings or the removal of outliers, resulting in no solution being obtained for those epochs. As shown in Table 7, the LS-D strategy maintains 100% solution integrity across all environments, whereas the LS-T strategy is more sensitive to environmental conditions, with the S21 smartphone's integrity decreasing from 100% in open-sky road (A) to 75.32% in urban (D) environments. This finding, alongside the analysis of observation quantities in Section 4.1, further highlights the sensitivity of TDCP to the observation environment, particularly in urban environments. As shown in Table 8, both LS-D and LS-T methods achieve decimeter-level accuracy, with LS-D outperforming LS-T for CL8 and AD11 in horizontal directions due to fewer available TDCP observations compared to Doppler in multipath environments (see Section 4.1).

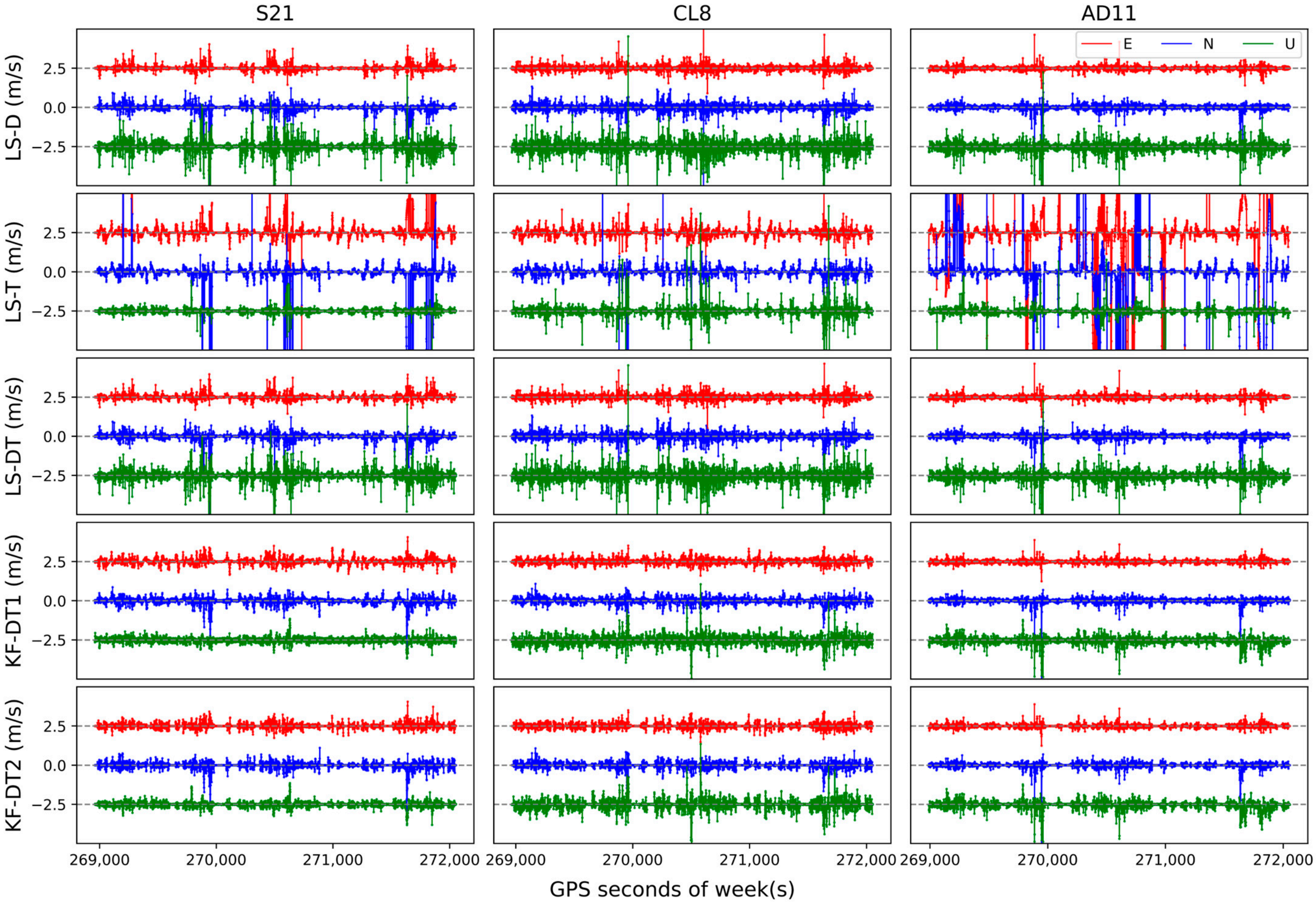

Figure 10. Time series of velocity errors for the S21, CL8, and AD11 using LS-D, LS-T, LS-DT, KF-DT1, and KF-DT2 solutions. The red, blue, and green lines represent the velocity errors in the E, N, and U directions, respectively. Here, the velocity errors in the E and U directions are presented with y = 2.5 and y = −2.5 as the respective reference baselines for the vertical axis.

Table 7. The solution integrity of LS-D and LS-T strategies in open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments.

Scenario	S21 (%)		CL8 (%)		AD11 (%)	
	Doppler	TDCP	Doppler	TDCP	Doppler	TDCP
A	100	100	100	100	100	99.18
B	100	97.79	100	100	100	64.31
C	100	88.65	100	100	100	57.14
D	100	75.32	100	97.40	100	41.42

Table 8. RMS values of velocity accuracy for the S21, CL8, and AD11 using five different velocity processing strategies.

Strategy	S21 (cm/s)			CL8 (cm/s)			AD11 (cm/s)		
	E	N	U	E	N	U	E	N	U
LS-D	17.3	30.5	40.7	17.3	24.0	43.5	12.7	24.8	33.7
LS-T	23.3	22.9	16.5	26.6	27.5	44.9	30.1	28.6	32.6
LS-DT	17.2	30.3	40.6	17.5	24.0	43.6	12.6	25.0	33.8
KF-DT1	16.7	24.6	17.3	16.8	19.4	32.5	11.6	24.1	26.6
KF-DT2	15.8	22.7	16.8	15.6	18.5	33.9	11.4	23.9	26.0

To further evaluate the performance of different estimation strategies, including LS-DT, KF-DT1 and KF-DT2 strategies, we assessed the velocity accuracy obtained from these three strategies using a combination of Doppler and TDCP observations. The Kalman filter provides an optimal estimate of the state variables by minimizing the root mean square of the estimation error, resulting in a more refined model compared to the least squares approach. As shown in Table 8, the KF-DT1 and KF-DT2 strategies outperform the LS-DT strategy across all devices. Compared to the LS-DT strategy, the average performance in the horizontal direction improved by approximately 15%, while the vertical direction saw an enhancement of about 30%. Therefore, an effective parameter estimation method is critical to enhancing the accuracy of velocity estimation. The KF-DT1 strategy treats velocity and clock drift as parameters to be estimated, with acceleration recursively calculated through the state transition matrix in the dynamic Kalman filter. The KF-DT2 strategy refines the observation equation by simultaneously estimating acceleration, velocity, and clock drift, leading to a more detailed model, as shown in Equations (11) and (12). As the data in Table 8 indicate, the KF-DT2 strategy demonstrates slight improvements across all three devices compared to the KF-DT1 strategy, with an average performance gain of about 6%.

In conclusion, Doppler and TDCP observations in smartphones equipped with low-cost linearly polarized GNSS chips demonstrate high sensitivity in high-multipath environments, where poor observations occur frequently. Implementing an effective outlier detection method is crucial for identifying and removing outliers, significantly improving the accuracy of velocity estimates. Additionally, the use of reasonable parameter estimation methods proves to be an encouraging approach to improve accuracy. The KF-DT2 strategy, which refines the observation equation by incorporating acceleration as an estimated parameter, results in a more detailed observation model and achieves a slight improvement in accuracy.

4.3. Positioning Performance Evaluation

In the previous section, we discussed smartphone velocity estimation from different perspectives of observation data, parameter estimation methods, and observation models. In this section, we will analyze the positioning performance of three smartphones using SPP, KFSPP-V, and KFSPP-P strategies under different environments. Figure 11 illustrates the positioning errors for the S21, CL8, and AD11 smartphones under different strategies, with red, blue, and green lines representing positioning error from the SPP, KFSPP-V, and

KFSPP-P strategies, respectively. The results reveal that when vehicles are driven in open-sky road (A) and suburban (B) environments, the SPP errors are generally within 10 m. However, as the vehicles move into tree-lined road (C) and urban (D) environments, where frequent high-multipath effects and signal blockages occur, the SPP errors exhibit significant fluctuations, with horizontal errors reaching up to 20–40 m, as shown in Table 9. The U direction experiences even more pronounced fluctuations, with RMS values exceeding 10 m across all three devices, confirming the challenges of using code-only positioning in high-multipath environments for smartphones.

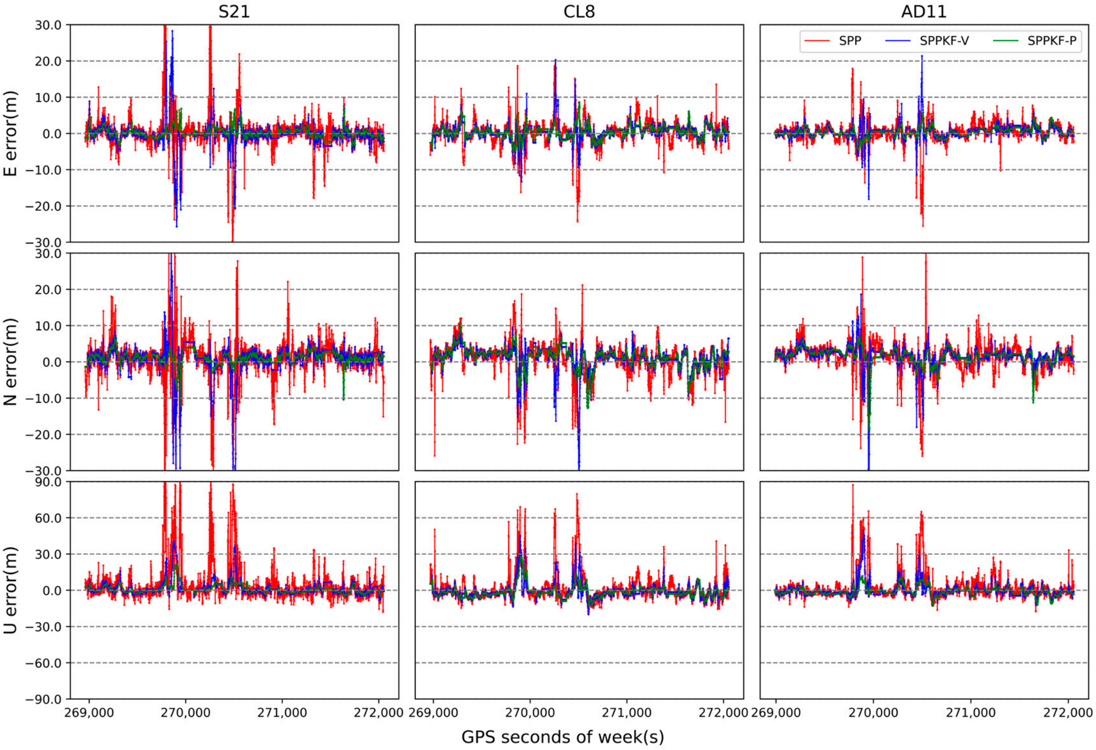

Figure 11. Error plots for the S21, CL8, and AD11 in the E, N, and U directions using the SPP (red), KFSPP-V (blue), and KFSPP-P (green) solutions.

Table 9. Maximum error and RMS accuracy values for the S21, CL8, and AD11 in the E, N, and U directions over the entire route using the SPP, KFSPP-V, and KFSPP-P solutions.

Smartphone		S21 (m)			CL8 (m)			AD11 (m)		
		E	N	U	E	N	U	E	N	U
SPP	RMS	4.99	5.62	16.93	3.52	4.81	11.63	3.03	4.48	11.35
	Max	41.14	46.69	115.74	24.31	25.90	79.73	25.51	31.57	87.15
KFSPP-V	RMS	3.69	5.57	6.65	2.38	3.77	6.84	2.00	3.53	6.17
	Max	28.27	49.90	41.14	20.31	39.08	45.31	21.39	39.82	50.34
KFSPP-P	RMS	1.37	2.03	3.58	1.76	3.02	5.34	1.30	2.82	3.55
	Max	5.56	19.04	12.86	8.65	12.76	27.95	5.08	18.05	16.19

In open-sky road (A) and suburban (B) environments, the positioning errors of both the KFSPP-V and KFSPP-P strategies remain within 5 m, indicating that velocity-aided positioning methods effectively reduce positioning errors caused by poor code observations in low-multipath environments. However, as the vehicles move into tree-lined road (C) and urban (D) environments, the positioning errors with the KFSPP-V strategy show significant fluctuations, frequently leading to large errors, with horizontal errors reaching 20–50 m. In contrast, the KFSPP-P strategy manages to keep horizontal errors within 10 m, with the horizontal RMS values of all three devices consistently below 3 m, demonstrating good robustness. These results show that velocity-aided positioning schemes are a reliable method for smartphone positioning in low-multipath environments. However, in high-multipath environments, poor velocity information can lead to incorrect state estimation, resulting in larger positioning errors. Therefore, employing an appropriate velocity-aided positioning approach, such as the KFSPP-P strategy, is crucial for ensuring accurate and robust positioning in such challenging environments.

To further analyze the performance of the KFSPP-P strategy in different observation environments, we selected representative sections of the route that correspond to four typical environments: open-sky road (A), suburban (B), tree-lined road (C), and urban (D) environments, as shown in Figure 12. Route (a) represents a relatively open main road with light tree coverage on both sides with a total length of 1.5 km; route (b) is a suburban road with typical suburban characteristics with a total length of 2.8 km; route (c) is a main road heavily covered by trees on both sides with a total length of 1.7 km; and route (d) is a typical urban environment with 5–10 m wide urban overpasses and tall buildings on both sides, with a total length of 0.8 km. Using the S21 smartphone as an example, Figure 13 illustrates the positioning errors of the SPP, KFSPP-V, and KFSPP-P strategies in the four routes (a, b, c, and d), with red, blue, and green lines representing the SPP, KFSPP-V, and KFSPP-P strategies, respectively. From the figure, two key conclusions can be drawn. Firstly, as the observation environment shifts from open-sky to urban environments, the positioning accuracy of all strategies decreases. For instance, the horizontal accuracy of the KFSPP-P strategy remains within 5 m in route (a), but decreases to within 10 m in route (d). Secondly, the KFSPP-V and KFSPP-P strategies perform similarly in route (a) and route (b). However, as the vehicle enters route (c) and route (d), the KFSPP-V strategy experiences significant divergence, while the KFSPP-P strategy maintains robust positioning accuracy.

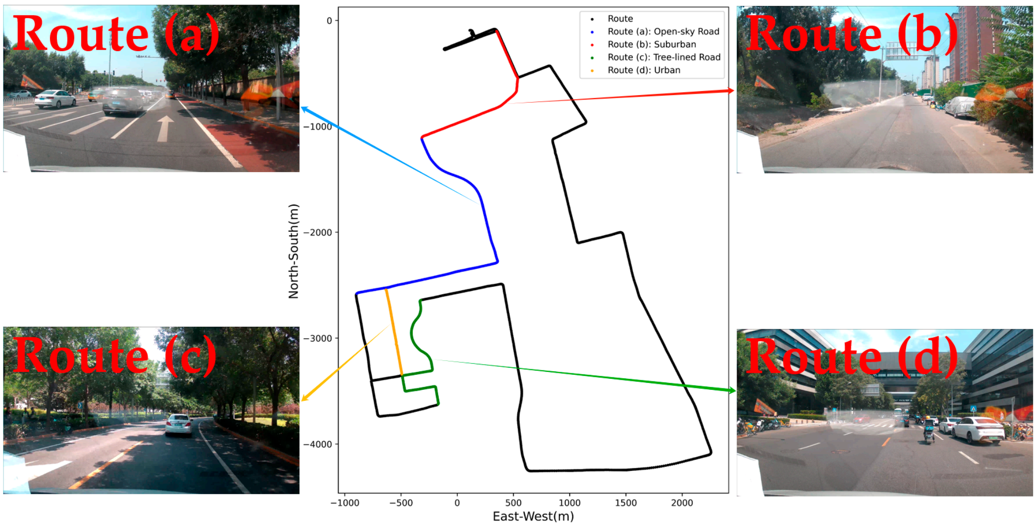

Figure 12. Four typical environments, with real routes (**a**–**d**) corresponding to open-sky road (A—blue), suburban (B—red), tree-lined road (C—green), and urban (D—yellow).

Table 10 summarizes the RMS positioning accuracy of the S21, CL8, and AD11 smartphones across the four routes (a, b, c, and d). The results show that in route (a) and route (b), the positioning accuracy of both the KFSPP-V and KFSPP-P strategies outperforms the SPP strategy for all devices. Compared to the KFSPP-V strategy, the KFSPP-P strategy achieved an average improvement of 8% in horizontal accuracy, while, overall, the two strategies showed comparable performance. This indicates that using velocity-aided positioning methods can significantly improve positioning accuracy in low-multipath environments. However, when the vehicle enters route (c) and route (d), the performance of the KFSPP-V strategy deteriorates significantly. For instance, in route (c) and route (d), the RMS positioning error of the S21 smartphone under the KFSPP-V strategy increases to over 10 m. In contrast, the KFSPP-P strategy maintains high positioning accuracy. Therefore, it is crucial to implement a robust estimation algorithm that can perform well across all environments. The KFSPP-P strategy consistently demonstrates enhanced robustness in all routes, including open-sky roads (A), suburban (B), tree-lined roads (C), and urban (D) environments. It effectively controls the RMS horizontal error within 5 m in high-multipath environments and improves performance by an average of 50% compared to the KFSPP-V strategy.

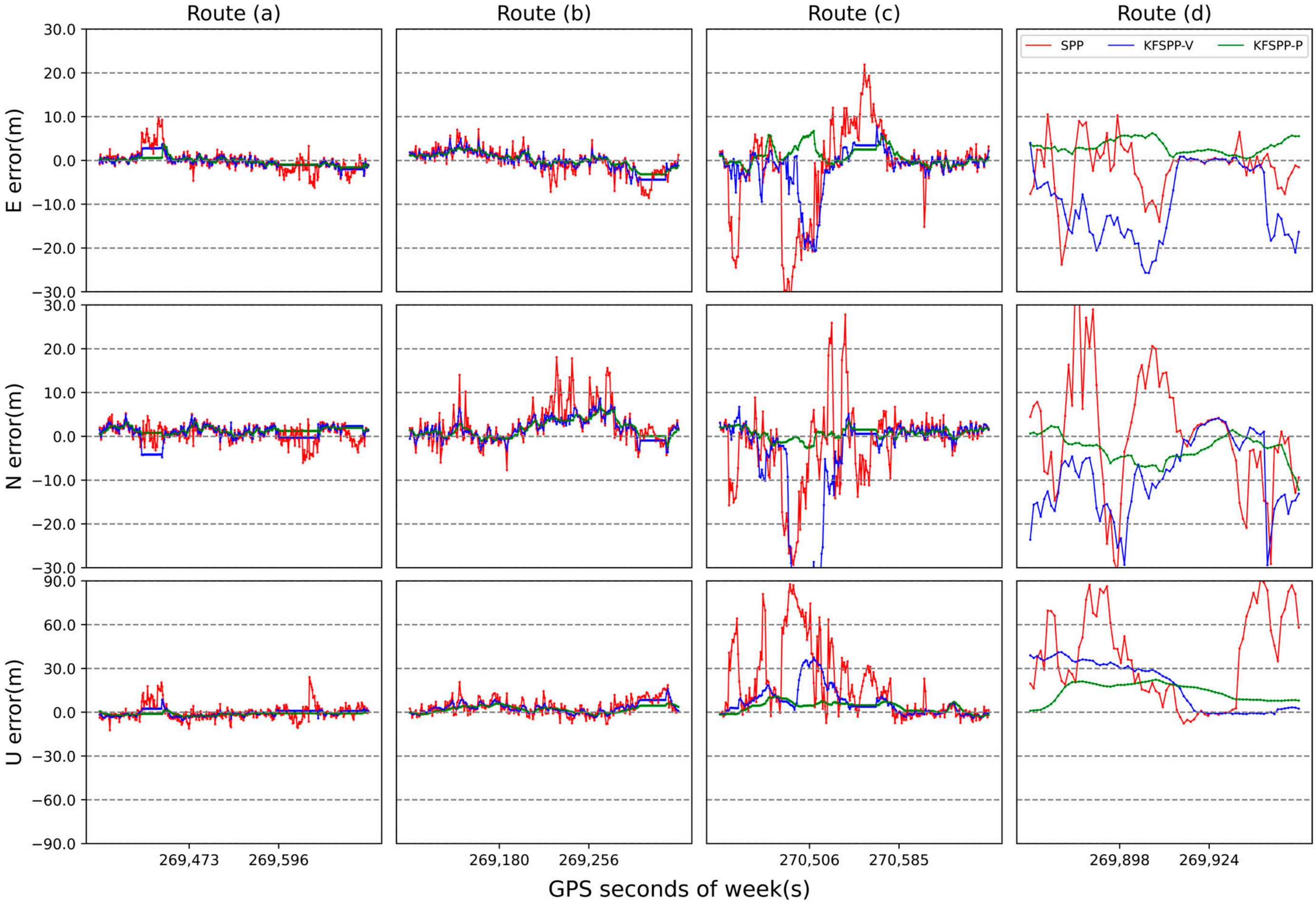

Figure 13. Positioning errors in the E, N, and U directions for the S21 smartphone across four routes (**a**–**d**). The red, blue, and green lines represent the SPP, KFSPP-V, and KFSPP-P solutions, respectively.

Table 10. RMS values of positioning accuracy in the E, N, and U directions for the S21, CL8, and AD11 using the SPP, KFSPP-V, and KFSPP-P solutions in the four typical scenarios (a, b, c, d).

Route	Strategy	S21 (m)			CL8 (m)			AD11 (m)		
		E	N	U	E	N	U	E	N	U
a	SPP	2.32	2.02	5.22	2.27	2.39	4.40	1.43	2.59	2.94
	KF-V	1.29	2.02	2.12	0.96	2.43	3.86	0.90	2.48	2.22
	KF-P	0.90	1.54	1.44	0.77	2.42	4.29	0.87	2.66	2.56
b	SPP	2.85	4.96	6.43	3.48	5.49	5.14	1.94	4.66	3.37
	KF-V	2.18	3.00	4.61	2.37	4.44	3.39	1.22	3.50	2.64
	KF-P	1.65	2.67	3.07	2.40	4.97	3.62	0.99	3.48	1.95
c	SPP	10.35	9.04	30.70	6.99	9.03	24.72	7.25	9.58	24.23
	KF-V	5.51	13.65	12.18	3.46	8.57	12.22	4.26	4.50	9.10
	KF-P	2.24	1.39	4.79	2.74	4.64	7.19	1.97	2.15	6.31
d	SPP	6.35	13.41	47.73	5.86	10.59	33.70	4.99	8.25	35.27
	KF-V	12.68	12.31	24.57	4.82	6.06	22.98	4.71	8.62	26.86
	KF-P	3.51	4.14	15.27	2.32	3.95	19.98	1.91	3.99	6.57

5. Discussion

This study demonstrates that accurate positioning can be achieved using smartphones in complex urban environments, marking a significant step toward replacing geodetic-grade receivers with smartphones. However, the path to replacing geodetic receivers with smartphones equipped with low-cost GNSS chips remains challenging. In real-world scenarios, factors such as temperature fluctuations, weather conditions, and the movement of people and objects can affect smartphone positioning in addition to environmental changes impacting GNSS observations. Temperature variations may influence GNSS receiver hardware parameters, thereby affecting positioning performance, and high-temperature environments may cause GNSS loggers in smartphones to stop functioning, making it difficult to collect raw GNSS observation data [30,39]. Extreme space weather conditions, such as geomagnetic storms, ionospheric irregularities, and solar radio bursts, may lead to a degradation in GNSS performance [40]. Studies have shown that compared to being placed outside the vehicle, mounting a P40 smartphone inside the vehicle is more susceptible to multipath interference, with C/N0 values decreasing by an average of 2–7 dB-Hz [41]. A smartphone placed flat experiences degraded RTK performance, with double-difference carrier phase residuals showing non-zero mean and linear trends, whereas a vertically placed smartphone does not exhibit this trend [42]. Additionally, high traffic near vehicle-mounted smartphones increases signal interference from passing vehicles, leading to higher GNSS observation noise [36]. In summary, various factors impact smartphone positioning in urban areas, making high-precision urban positioning with smartphones highly challenging.

In this experiment, traditional Kalman filtering methods were used to address kinematic measurement problems, but more robust and reliable methods like extended Kalman filters, unscented Kalman filters, or particle filters could be more suitable for handling nonlinear models. Moreover, Artificial Intelligence, factor graph optimization, and graph convolutional neural network techniques have been widely applied to smartphone positioning [43,44]. Therefore, future studies will involve analysis and comparison of these different methods.

Currently, there are over 290 types of smartphones on the market that support both single-frequency and dual-frequency observations, with single-frequency phones dominating the market [45]. Additionally, variations in GNSS chips and antennas used by different manufacturers lead to significant differences in GNSS observation quality and single/dual-frequency capabilities across smartphones. As a result, smartphone positioning accuracy is also influenced by hardware, contributing to the differences in positioning performance between various smartphones. Overall, there are still many areas of GNSS positioning on smartphones that warrant further exploration: the impact of external conditions on GNSS quality, the application of different filtering algorithms, and hardware improvements in smartphones. These advancements make the future of smartphones replacing geodetic receivers for high-precision positioning increasingly promising.

6. Conclusions

In this study, we conducted vehicle-mounted experiments using three smartphones, namely the S21, CL8, and AD11, across various environments, including open-sky roads, suburban, tree-lined roads, and urban environments, to explore the performance of an improved velocity-aided positioning strategy. The results show that as smartphones transition from low- to high-multipath environments, the number of tracked satellites, carrier phase observations, and TDCMC values exhibit high sensitivity to environmental factors. The use of robust outlier detection algorithms effectively identifies erroneous observations, proving to be a reliable quality control method.

By improving the Doppler and TDCP observation equations and applying appropriate parameter estimation methods, the velocity performance of smartphones under vehicle-mounted conditions was enhanced. The results indicate that Kalman filtering outperforms the least squares method, with velocity performance improving by approximately 15% and 30% in horizontal and vertical directions, respectively. The use of more refined observation equations led to a slight 6% improvement in speed performance across all devices, achieving centimeter-level accuracy in low-multipath environments and decimeter-level accuracy in high-multipath environments.

In open-sky roads and suburban environments, both the KFSPP-V and KFSPP-P strategies outperform the SPP strategy, with similar positioning accuracy between the two. However, in tree-lined roads and urban environments, the positioning performance of the KFSPP-V strategy significantly degrades. In contrast, the KFSPP-P strategy demonstrates strong robustness in extreme environments, keeping the horizontal positioning RMS error within 5 m and improving average performance by approximately 50% compared to the KFSPP-V strategy.

In summary, this study highlights the advantages of the improved velocity-aided positioning algorithm KFSPP-P in smartphone applications, particularly in complex urban environments, further enhancing positioning accuracy and offering a robust positioning solution for smartphones in challenging conditions. However, the KFSPP-P strategy relies heavily on sufficient GNSS raw observations. In environments with complete signal blockage, such as tunnels or caves, the performance of the KFSPP-P strategy may be significantly limited, resulting in a sharp decline in positioning accuracy. In contrast, the integration of inertial navigation, visual sensors, and other sensor data allows for updating the position information even when GNSS observations are unavailable. Therefore, future work will focus on combining multi-sensor data to further enhance the KFSPP-P strategy.

Author Contributions: Conceptualization, Z.H. and X.W.; methodology, Z.H.; software, Z.H.; validation, Z.H., X.W. and J.Z.; formal analysis, Z.H.; investigation, Z.H. and X.W.; resources, X.W, S.X. and J.Z.; data curation, Z.H. and X.W.; writing—original draft preparation, Z.H.; writing—review and editing, X.W., Q.H., S.X. and S.S.; visualization, Z.H.; supervision, X.W.; project administration, X.W.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 42474015.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors would like to acknowledge Google for providing the Gnsslogger 3.0.6.3 version raw data collection software and Zhang Xiaohong's team for their research, "Quality assessment of GNSS observations from an Android N smartphone and positioning performance analysis using time-differenced filtering approach", which inspired the idea of this paper.

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

References

1. Li, X.; Ge, M.; Dai, X.; Ren, X.; Fritsche, M.; Wickert, J.; Schuh, H. Accuracy and Reliability of Multi-GNSS Real-Time Precise Positioning: GPS, GLONASS, BeiDou, and Galileo. *J. Geod.* **2015**, *89*, 607–635. [CrossRef]
2. Paziewski, J.; Fortunato, M.; Mazzoni, A.; Odolinski, R. An Analysis of Multi-GNSS Observations Tracked by Recent Android Smartphones and Smartphone-Only Relative Positioning Results. *Measurement* **2021**, *175*, 109162. [CrossRef]
3. Zhang, X.; Tao, X.; Zhu, F.; Shi, X.; Wang, F. Quality Assessment of GNSS Observations from an Android N Smartphone and Positioning Performance Analysis Using Time-Differenced Filtering Approach. *GPS Solut.* **2018**, *22*, 70. [CrossRef]
4. Paziewski, J.; Sieradzki, R.; Baryla, R. Signal Characterization and Assessment of Code GNSS Positioning with Low-Power Consumption Smartphones. *GPS Solut.* **2019**, *23*, 98. [CrossRef]
5. Håkansson, M. Characterization of GNSS Observations from a Nexus 9 Android Tablet. *GPS Solut.* **2018**, *23*, 21. [CrossRef]
6. Chen, B.; Gao, C.; Liu, Y.; Sun, P. Real-Time Precise Point Positioning with a Xiaomi MI 8 Android Smartphone. *Sensors* **2019**, *19*, 2835. [CrossRef]
7. Realini, E.; Caldera, S.; Pertusini, L.; Sampietro, D. Precise GNSS Positioning Using Smart Devices. *Sensors* **2017**, *17*, 2434. [CrossRef]
8. Dabove, P.; Di Pietra, V. Towards High Accuracy GNSS Real-Time Positioning with Smartphones. *Adv. Space Res.* **2019**, *63*, 94–102. [CrossRef]

9. Laurichesse, D.; Rouch, C.; Marmet, F.-X.; Pascaud, M. Smartphone Applications for Precise Point Positioning. In Proceedings of the 30th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2017), Portland, OR, USA, 25–29 September 2017; pp. 171–187.
10. Wu, Q.; Zhang, P.; Sun, M.; Liu, S.; Wang, H.; Chen, S. Performance Evaluation of GIMs Released by Different IGS Ionosphere Associate Analysis Centers in Ionospheric Constrained Single-Frequency Precise Point Positioning. *Adv. Space Res.* **2021**, *68*, 4834–4856. [CrossRef]
11. Li, M.; Huang, T.; Li, W.; Zhao, Q.; Jiang, K. Precise Point Positioning with Mixed Single- and Dual-Frequency GNSS Observations from Android Smartphones Considering Code-Carrier Inconsistency. *Adv. Space Res.* **2024**, *74*, 2664–2679. [CrossRef]
12. Zhang, K.; Jiao, W.; Wang, L.; Li, Z.; Li, J.; Zhou, K. Smart-RTK: Multi-GNSS Kinematic Positioning Approach on Android Smart Devices with Doppler-Smoothed-Code Filter and Constant Acceleration Model. *Adv. Space Res.* **2019**, *64*, 1662–1674. [CrossRef]
13. Shinghal, G.; Bisnath, S. Conditioning and PPP Processing of Smartphone GNSS Measurements in Realistic Environments. *Satell. Navig.* **2021**, *2*, 10. [CrossRef] [PubMed]
14. Yi, D.; Bisnath, S.; Naciri, N.; Vana, S. Effects of Ionospheric Constraints in Precise Point Positioning Processing of Geodetic, Low-Cost and Smartphone GNSS Measurements. *Measurement* **2021**, *183*, 109887. [CrossRef]
15. Fortunato, M.; Critchley-Marrows, J.; Siutkowska, M.; Ivanovici, M.L.; Benedetti, E.; Roberts, W. Enabling High Accuracy Dynamic Applications in Urban Environments Using PPP and RTK on Android Multi-Frequency and Multi-GNSS Smartphones. In Proceedings of the 2019 European Navigation Conference (ENC), Warsaw, Poland, 9–12 April 2019; pp. 1–9.
16. Aggrey, J.; Bisnath, S.; Naciri, N.; Shinghal, G.; Yang, S. Multi-GNSS Precise Point Positioning with next-Generation Smartphone Measurements. *J. Spat. Sci.* **2020**, *65*, 79–98. [CrossRef]
17. Zhang, Z.; Li, X.; Yuan, H. Best Integer Equivariant Estimation Based on Unsupervised Machine Learning for GNSS Precise Positioning and Navigation in Complex Environments. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *60*, 2672–2682. [CrossRef]
18. Bahadur, B.; Schön, S. Improving the Stochastic Model for Code Pseudorange Observations from Android Smartphones. *GPS Solut.* **2024**, *28*, 148. [CrossRef]
19. Bahadur, B. A Study on the Real-Time Code-Based GNSS Positioning with Android Smartphones. *Measurement* **2022**, *194*, 111078. [CrossRef]
20. Li, L.; Zhong, J.; Zhao, M. Doppler-Aided GNSS Position Estimation with Weighted Least Squares. *IEEE Trans. Veh. Technol.* **2011**, *60*, 3615–3624. [CrossRef]
21. Hatch, R. The Synergism of GPS Code and Carrier Measurements. 1983, Volume 2, pp. 1213–1231. Available online: https://ui.adsabs.harvard.edu/abs/1983igss.conf.1213H//abstract (accessed on 23 September 2024).
22. Wang, R.; Hu, C.; Wang, Z.; Yuan, F.; Wang, Y. Performance of Smartphone BDS-3/GPS/Galileo Multi-Frequency Ionosphere-Free Precise Code Positioning. *Remote Sens.* **2023**, *15*, 5371. [CrossRef]
23. Geng, J.; Jiang, E.; Li, G.; Xin, S.; Wei, N. An Improved Hatch Filter Algorithm towards Sub-Meter Positioning Using Only Android Raw GNSS Measurements without External Augmentation Corrections. *Remote Sens.* **2019**, *11*, 1679. [CrossRef]
24. Mazher, K.; Tahir, M.; Ali, K. GNSS Pseudorange Smoothing: Linear vs Non-Linear Filtering Paradigm. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–10.
25. Bai, S.; Lai, J.; Lyu, P.; Cen, Y.; Sun, X.; Wang, B. Performance Enhancement of Tightly Coupled GNSS/IMU Integration Based on Factor Graph with Robust TDCP Loop Closure. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 2437–2449. [CrossRef]
26. Liu, W.; Shi, X.; Zhu, F.; Tao, X.; Wang, F. Quality Analysis of Multi-GNSS Raw Observations and a Velocity-Aided Positioning Approach Based on Smartphones. *Adv. Space Res.* **2019**, *63*, 2358–2377. [CrossRef]
27. Robustelli, U.; Paziewski, J.; Pugliano, G. Observation Quality Assessment and Performance of GNSS Standalone Positioning with Code Pseudoranges of Dual-Frequency Android Smartphones. *Sensors* **2021**, *21*, 2125. [CrossRef] [PubMed]
28. Liu, Q.; Gao, C.; Peng, Z.; Zhang, R.; Shang, R. Smartphone Positioning and Accuracy Analysis Based on Real-Time Regional Ionospheric Correction Model. *Sensors* **2021**, *21*, 3879. [CrossRef]
29. Li, G.; Geng, J.; Chu, B. High-Precision Velocity Determination Using Mass-Market Android GNSS Measurements in the Case of Anomalous Clock Variations. *GPS Solut.* **2023**, *27*, 98. [CrossRef]
30. Zangenehnejad, F.; Jiang, Y.; Gao, Y. GNSS Observation Generation from Smartphone Android Location API: Performance of Existing Apps, Issues and Improvement. *Sensors* **2023**, *23*, 777. [CrossRef]
31. Xie, W.; Huang, G.; Fu, W.; Shu, B.; Cui, B.; Li, M.; Yue, F. A Quality Control Method Based on Improved IQR for Estimating Multi-GNSS Real-Time Satellite Clock Offset. *Measurement* **2022**, *201*, 111695. [CrossRef]
32. Vinutha, H.P.; Poornima, B.; Sagar, B.M. Detection of Outliers Using Interquartile Range Technique from Intrusion Dataset. In *Information and Decision Sciences*; Satapathy, S.C., Tavares, J.M.R.S., Bhateja, V., Mohanty, J.R., Eds.; Springer: Singapore, 2018; pp. 511–518.
33. Gioia, C.; Angrisano, A.; Gaglione, S. Galileo-Based Doppler Shifts and Time Difference Carrier Phase: A Static Case Demonstration. *Sensors* **2023**, *23*, 6828. [CrossRef]
34. Agarwal, N.; O'Keefe, K. Use of GNSS Doppler for Prediction in Kalman Filtering for Smartphone Positioning. *IEEE J. Indoor Seamless Position. Navig.* **2023**, *1*, 151–160. [CrossRef]
35. Wang, L.; Li, Z.; Wang, N.; Wang, Z. Real-Time GNSS Precise Point Positioning for Low-Cost Smart Devices. *GPS Solut.* **2021**, *25*, 69. [CrossRef]

36. Yi, D.; Hu, J.; Bisnath, S. Improving PPP Smartphone Processing with Adaptive Quality Control Method in Obstructed Environments When Carrier-Phase Measurements Are Missing. *GPS Solut.* **2024**, *28*, 56. [CrossRef]
37. 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]
38. Caamano, M.; Crespillo, O.G.; Gerbeth, D.; Grosch, A. Detection of GNSS Multipath with Time-Differenced Code-Minus-Carrier for Land-Based Applications. In Proceedings of the 2020 European Navigation Conference (ENC), Dresden, Germany, 23–24 November 2020; pp. 1–12.
39. Mi, X.; Zhang, B.; Odolinski, R.; Yuan, Y. On the Temperature Sensitivity of Multi-GNSS Intra- and Inter-System Biases and the Impact on RTK Positioning. *GPS Solut.* **2020**, *24*, 112. [CrossRef]
40. Demyanov, V.V.; Yasyukevich, Y.V. Space weather: Risk factors for global navigation satellite systems. *Sol.-Terr. Phys.* **2021**, *7*, 28–47. [CrossRef]
41. Li, Z.; Wang, L.; Wang, N.; Li, R.; Liu, A. Real-Time GNSS Precise Point Positioning with Smartphones for Vehicle Navigation. *Satell. Navig.* **2022**, *3*, 19. [CrossRef]
42. Yong, C.Z.; Odolinski, R.; Zaminpardaz, S.; Moore, M.; Rubinov, E.; Er, J.; Denham, M. Instantaneous, Dual-Frequency, Multi-GNSS Precise RTK Positioning Using Google Pixel 4 and Samsung Galaxy S20 Smartphones for Zero and Short Baselines. *Sensors* **2021**, *21*, 8318. [CrossRef]
43. Mohanty, A.; Gao, G. Learning GNSS Positioning Corrections for Smartphones Using Graph Convolution Neural Networks. *Navig. J. Inst. Navig.* **2023**, *70*, navi.622. [CrossRef]
44. Jiang, Y.; Gao, Y.; Ding, W.; Gao, Y. GNSS Precise Positioning for Smartphones Based on the Integration of Factor Graph Optimization and Solution Separation. *Measurement* **2022**, *203*, 111924. [CrossRef]
45. Bisnath, S.; Aggrey, J. Current Limitations and Prospects for Smartphone GNSS Precise Positioning. In Proceedings of the 2024 International Technical Meeting of The Institute of Navigation, Long Beach, CA, USA, 23–25 January 2024; pp. 1–24.

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.