

Smartphone RTK for C-ITS: Architecture and PPK-Based Field Evaluation

Steve Zerafa^{1,2}, Kenneth Scerri³

¹PIXAM Ltd, O3 Orchid Court, Valley Road, Msida MSD 9027, Malta

²Malta College of Arts, Science and Technology (MCAST), Main Campus, Triq Kordin, Paola PLA 9032, Malta

³University of Malta, Msida MSD 2080, Malta

Abstract

Reliable, low-latency positioning using commodity smartphones is relevant to Cooperative Intelligent Transport Systems (C-ITS) Day 1 and Day 1.5 Smart Road services. This industry/work-in-progress paper presents a PIXAM smartphone Global Navigation Satellite System (GNSS) positioning architecture supporting both real-time kinematic (RTK) operation and offline post-processed kinematic (PPK) evaluation. In the intended real-time architecture, raw Android GNSS observations are transmitted using Message Queuing Telemetry Transport (MQTT) and combined with Radio Technical Commission for Maritime Services (RTCM) reference-station data and broadcast navigation data.

Two processing approaches are evaluated. The first is an operational baseline based on National Marine Electronics Association (NMEA) positions recorded by Google GnsLogger, followed by filtering and smoothing. The second operates directly on raw smartphone GNSS observations using a causal sliding-window PPK procedure. At each output epoch, only the current and preceding observations are processed and only the solution for the newest epoch is retained. The reported configuration uses a 300-epoch window. Carrier-phase measurements are used when valid accumulated delta range observations are available; otherwise, the same processing pipeline falls back to pseudorange and Doppler without integer ambiguity resolution.

Field data were collected on road routes in Rome during an evaluation campaign supported by ANAS S.p.A. and the European Space Agency (ESA). Smartphone trajectories were evaluated against an Applanix POS LV reference system installed on the Cartesio mobile laboratory. Datasets with usable carrier-phase observations provided the strongest raw-observation results, whereas datasets without usable carrier phase showed substantially larger error tails, particularly in the proportion of epochs exceeding 3 m. A four-device trajectory-fusion experiment achieved 63.8% of epochs within 1 m and 81.5% within 2 m, slightly exceeding the strongest individual raw-observation result at these thresholds. The fusion result is treated as a proof of concept rather than a controlled ablation study. The results demonstrate the potential of the architecture using real field data, while matched-epoch statistics and full end-to-end real-time validation remain part of the next development stage.

Keywords

GNSS, RTK, PPK, smartphone positioning, C-ITS, Smart Road, sliding-window estimation, trajectory fusion, Android raw GNSS measurements

1. Introduction

Cooperative Intelligent Transport Systems (C-ITS) combine information from vehicles, infrastructure and road users to support road-safety and traffic-management services. Smartphones are relevant to this ecosystem as user-facing C-ITS terminals and as low-cost sensing and positioning platforms; prior research has considered smartphone-supported vulnerable-road-user safety and mobile C-ITS hazard-warning applications [2, 3].

In European C-ITS terminology, *Day 1* denotes the initial set of mature services prioritised for early deployment, including hazardous-location notifications and signage applications. *Day 1.5* denotes a subsequent set including vulnerable-road-user protection, parking information, connected and cooperative navigation, and traffic information and smart routing [1]. These terms are used here to motivate the positioning use case; the present experiments do not constitute certification of any specific C-ITS service.

WIPHAL 2026: Work-in-Progress in Hardware and Software for Location Computation, co-located with ICL-GNSS 2026, October 6–8, 2026, Rome, Italy

✉ steve.zerafa@mcast.edu.mt (S. Zerafa); kenneth.scerri@um.edu.mt (K. Scerri)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Achieving robust metre- or sub-metre-level positioning using mass-market smartphones remains challenging, particularly in urban canyons where multipath and non-line-of-sight (NLOS) reception can dominate the error budget. Smartphone GNSS observations also vary strongly between handset models because of antenna, chipset, firmware and measurement-availability differences. Paziewski et al. [4] reported substantially higher pseudorange noise than is typical of geodetic receivers and identified carrier-phase anomalies including long-term drifts and receiver-dependent biases. A 2024 multi-sensor Android dataset released by Grenier et al. [11] explicitly includes open-sky, urban-canyon and indoor conditions, illustrating the breadth of environments in which current smartphone positioning is being studied.

Recent work also shows that the achievable accuracy is not limited to the metre level when higher-quality observations and robust estimation are available. Geng et al. [5] developed a robust Android RTK method using sliding-window factor-graph optimisation for open-sky and complex urban environments. Pourmina et al. [6] investigated smartphone RTK with robust filtering and inertial integration in complex urban conditions. The 2024 Google Smartphone Decimeter Challenge further demonstrated trajectory-level methods based on raw GNSS, Doppler, carrier phase and, in some entries, inertial measurements: Yun et al. [7], Suzuki [8] and Motooka [9] reported the three leading approaches. These challenge results were obtained in a post-processing setting using complete recorded trajectories, whereas the architecture proposed in this work is designed for causal, low-latency operation in which future observations are not available. Hu et al. [10] subsequently investigated partial wide-lane ambiguity resolution as a path towards real-time, decimetre-level smartphone relative positioning. Together, these studies motivate robust multi-epoch processing and careful handling of device-dependent observation quality.

The work presented here focuses on smartphone positioning for future Smart Road and C-ITS applications. The PIXAM architecture is designed for real-time operation: raw smartphone GNSS observations are transmitted using MQTT and combined on the server with RTCM reference-station data and broadcast navigation information from a PIXAM GNSS base station. The same differential positioning core can also process stored observations in PPK mode. In the present study, the causal sliding-window branch is evaluated offline using recorded measurements to provide repeatable comparison with the reference trajectory. Importantly, the estimator uses only the current and preceding epochs and does not use future observations. The offline evaluation therefore tests the same causal processing principle intended for subsequent real-time operation, while full end-to-end live validation remains part of the next development stage.

The evaluation uses road-survey datasets collected in Rome as part of the ANAS S.p.A. and European Space Agency Grand Challenge. The broader ANAS-ESA Grand Challenge explicitly included real-time kinematic smartphone positioning among its objectives; however, the material available for the present evaluation consisted of recorded field datasets. The causal processing adopted here was therefore evaluated offline under repeatable conditions, while retaining the no-future-observation constraint required for subsequent real-time operation.

The comparisons reported in this paper are descriptive because the currently available Algorithm 1 and Algorithm 2 solution sets do not contain identical sets of valid epochs. Consequently, the reported percentage differences should not be interpreted as statistically validated paired improvements. A matched-epoch analysis using the common intersection of the available solution and reference timestamps is planned as part of the subsequent evaluation, together with root mean square error (RMSE), median, minimum, maximum and upper-tail percentile statistics.

The main contributions of this work are threefold. First, it describes a single smartphone GNSS architecture that can operate on live Android raw measurements in RTK mode or on recorded measurements using causal sliding-window PPK without changing the core observation model. Second, it reports a multi-device road evaluation against an Applanix reference trajectory and separates the observed behaviour of datasets with and without usable carrier-phase observations. Third, it investigates a co-located multi-smartphone fusion concept and interprets the resulting accuracy and error-tail behaviour in the context of future C-ITS operation. The emphasis is therefore on an implementable architecture and field evidence rather than on claiming a fully validated production C-ITS service.

The remainder of this paper is organised as follows. Section 2 describes the real-time and post-

processed system architecture, the NMEA baseline, the causal sliding-window PPK method and the trajectory-fusion method. Section 3 presents the experimental setup and field-data collection campaign. Section 4 reports the preliminary field-evaluation results for both road datasets. Section 5 discusses the observed error-tail behaviour, the multi-device fusion experiment and the implications for C-ITS-oriented operation, together with the principal limitations of the present evaluation. Section 6 concludes the paper and outlines the next development stage.

2. System Architecture and Positioning Method

2.1. Real-Time Reference Architecture

In the intended real-time configuration, a PIXAM GNSS base station operates from a surveyed location and continuously tracks multiple GNSS constellations. The base station generates Radio Technical Commission for Maritime Services (RTCM) reference-station messages and provides broadcast ephemeris/navigation data, which are transmitted to a central Message Queuing Telemetry Transport (MQTT) broker [13, 15]. At the same time, the PIXAM Android GnsLogger application collects raw smartphone observations through the Android GNSS Measurement API and publishes rover-specific measurement messages to MQTT.

On the server, the smartphone messages are decoded and reconstructed into rover GNSS observables, while the RTCM and navigation streams are decoded into the reference-station observations and ephemeris required by the differential engine. Internal bench tests have shown MQTT transfer times below 100 ms over Wi-Fi and mobile networks. This value refers to communication latency only and does not include measurement reconstruction, RTK/PPK estimation or hybrid solution selection. Total end-to-end latency will be characterised during future live vehicle trials.

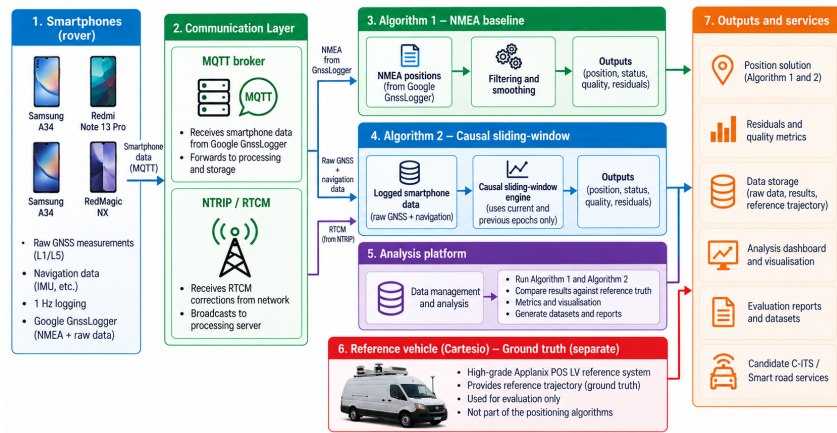


Figure 1: PIXAM real-time smartphone positioning architecture. Smartphone measurements and base-station data are transmitted through MQTT to the server, where decoding, reconstruction and estimation produce candidate solutions for hybrid selection. The results in Section 4 use the offline evaluation mode described in Section 2.3.

2.2. Software Implementation and Measurement Reconstruction

The real-time system is implemented as three cooperating software layers: the PIXAM *GnsLog* Android application, the MQTT communication layer and the server-side C++ positioning software. The smartphone is responsible for raw measurement acquisition and transmission, whereas reconstruction of GNSS observables, synchronisation with reference-station data and final RTK/PPK estimation are performed on the server. This separation keeps the handset-side processing lightweight and allows the same server-side engine to support multiple smartphone models and both live and recorded datasets.

The PIXAM *GnssLog* Android application uses the Android GNSS Measurement API to access the receiver-clock parameters and per-satellite measurement fields exposed by the handset GNSS chipset. At each measurement epoch, the application packages the available timing information, accumulated delta range (ADR), pseudorange rate, carrier frequency, signal-quality indicators and measurement-state flags into a rover-specific message and publishes it to an MQTT topic over Wi-Fi or mobile data. The Android application does not calculate the final RTK/PPK position; its role is to provide a consistent stream of raw measurement information to the positioning server.

The MQTT layer carries three logical data streams. The first is the smartphone rover-measurement stream. The second is the RTCM stream generated by the surveyed PIXAM reference station. The third contains broadcast navigation and ephemeris information originating from the base-station side. Processing services subscribe to the corresponding topics and receive the streams independently as data become available. For reproducibility, the streams can also be persisted: RTCM is retained as the reference-station correction record, navigation data can be stored in Receiver Independent Exchange Format (RINEX) navigation files, and reconstructed rover observations can be stored as RINEX observation files [14].

A dedicated C++ processing layer on the server converts the received data into the observation structures required by the positioning engine. For the smartphone stream, receiver and satellite timing fields are used to reconstruct code pseudorange; valid ADR measurements are converted to carrier-phase observations; and pseudorange-rate measurements are converted to Doppler using the corresponding carrier wavelength. The converter also identifies constellation and signal frequency, assembles measurements into common epochs, checks measurement-state information and handles receiver-clock or tracking discontinuities where indicated by the Android data.

In parallel, the base-station RTCM stream is decoded to recover the reference observations required for differential processing, while the navigation stream provides the broadcast ephemerides used to determine satellite orbit and clock information. The reconstructed rover observations, decoded base observations, broadcast ephemerides and surveyed base-station coordinates are then aligned in GNSS time. The resulting rover–base epoch pair is passed to the differential positioning core, which forms the code and carrier-phase models described in the observation model and estimators described in the following subsections.

The server-side positioning core is also implemented in C++. It contains the causal sliding-window estimator, the recursive real-time branch, integer ambiguity handling and the hybrid solution-selection logic. The same internal observation model is used for live RTK and offline PPK operation. In real time, the input is supplied continuously by MQTT; in post-processing, the input is read from stored rover, reference-station and navigation files. The difference between the two modes is therefore primarily the source and timing of the input data rather than a separate positioning algorithm.

2.3. Post-Processed Evaluation Mode

The field evaluation was performed offline using recorded rover observations, reference-station data and navigation data. The same core positioning engine used by the real-time architecture was adapted to process the stored observations in PPK mode. Changes were required for file parsing, epoch reconstruction, lock-time and cycle-slip handling, quality control and rover-to-base time alignment.

For the reported Algorithm 2 results, only the causal sliding-window PPK branch was exercised. The recursive branch forms part of the complete real-time architecture but was not used during the offline evaluation.

2.4. Algorithm 1: NMEA-Based Operational Baseline

Algorithm 1 provides the comparison baseline. It uses National Marine Electronics Association (NMEA) position output recorded by Google *GnssLogger* rather than processing raw GNSS observations directly. The algorithm reads the \$GPRMC and \$GPGGA messages, reconstructs GNSS time, converts the reported

coordinates into the required internal representation and applies a constant-velocity Kalman filter for temporal smoothing and outlier reduction.

Because Algorithm 1 begins with an already calculated smartphone position, it does not independently resolve carrier-phase ambiguities or perform RTK estimation.

2.5. GNSS Observation Model

For receiver r , satellite s and epoch k , the pseudorange observation is represented as

$$P_{r,k}^s = \rho_{r,k}^s + c(\delta t_{r,k} - \delta t_k^s) + I_{r,k}^s + T_{r,k}^s + \epsilon_{P,k}^s, \quad (1)$$

where $\rho_{r,k}^s$ is the geometric range, c is the speed of light, $\delta t_{r,k}$ and δt_k^s are the receiver and satellite clock offsets, $I_{r,k}^s$ and $T_{r,k}^s$ are the ionospheric and tropospheric delays, and $\epsilon_{P,k}^s$ contains measurement noise and multipath.

For smartphones providing usable carrier-phase observations, the carrier measurement expressed in metres is

$$L_{r,k}^s = \rho_{r,k}^s + c(\delta t_{r,k} - \delta t_k^s) - I_{r,k}^s + T_{r,k}^s + \lambda N_r^s + \epsilon_{L,k}^s, \quad (2)$$

where λ is the carrier wavelength and N_r^s is the integer ambiguity.

When base-station observations are available, a rover-to-base single difference is formed as

$$\Delta P_{rb,k}^s = P_{r,k}^s - P_{b,k}^s, \quad (3)$$

followed by a double difference between satellites s and q ,

$$\nabla \Delta P_{rb,k}^{s,q} = \Delta P_{rb,k}^s - \Delta P_{rb,k}^q. \quad (4)$$

The corresponding carrier-phase differences are formed in the same way. Differencing removes receiver and satellite clock terms and reduces errors common to rover and base-station observations.

2.6. Algorithm 2: Causal Sliding-Window PPK Estimation

Algorithm 2 is the raw-observation method evaluated in this paper. Its inputs are the reconstructed smartphone rover observations, time-aligned reference-station observations, broadcast navigation data and the surveyed base-station coordinates. Its output at epoch k is the rover position associated with the newest epoch in a finite causal window. The processing therefore uses recent history in a PPK-style differential solution, but it does not use observations from future epochs.

For each output epoch k , Algorithm 2 performs the following sequence:

1. reconstruct rover pseudorange, Doppler and, when valid accumulated delta range (ADR) is available, carrier-phase observations from the Android raw measurements;
2. decode the reference-station observations and broadcast ephemerides, then align rover and base observations in GNSS time;
3. form a causal window containing the current epoch and the preceding epochs; the experiments reported here use $W = 300$ epochs once the window is fully populated;
4. form the differential code and, when available, carrier-phase residuals over the window, applying measurement-validity and cycle-slip checks;
5. estimate the windowed states and carrier-phase ambiguities when phase observations are available, including integer ambiguity resolution using the LAMBDA method [12]; and
6. retain only the position solution associated with the newest epoch k , advance the window by one epoch and repeat.

For a window containing W epochs and ending at epoch k , the state is estimated in the generic weighted least-squares form

$$\hat{\mathbf{X}}_k = \arg \min_{\mathbf{X}} \sum_{i=k-W+1}^k \mathbf{r}_i^T \mathbf{W}_i \mathbf{r}_i, \quad (5)$$

with

$$\mathbf{r}_i = \mathbf{z}_i - h(\mathbf{x}_i), \quad (6)$$

where $\mathbf{z}_i$ contains the rover/base differential observations, $h(\mathbf{x}_i)$ represents their predicted values and $\mathbf{W}_i$ defines their relative weighting.

During start-up, the available history is shorter than W . Once the window is full, a new epoch is appended and the oldest epoch is removed. The estimator remains causal because only measurements at epochs $i \leq k$ are used. The 300-epoch history conditions the current estimate; it does not impose a 300-epoch output delay because only the newest state is emitted after each window update.

When usable carrier-phase observations are present, Algorithm 2 performs carrier-phase differential PPK and ambiguity estimation. When carrier phase is not available or fails the validity checks, the same pipeline operates using pseudorange and Doppler observations without estimating integer ambiguities. This fallback maintains one processing architecture across heterogeneous handsets, but the results in Section 4 show that its error-tail behaviour is currently weaker than the carrier-phase mode.

2.7. Recursive Estimator and Hybrid Solution Selection

The intended real-time engine contains a second, epoch-by-epoch recursive branch in parallel with the sliding-window estimator. This recursive branch and the subsequent hybrid selector are architectural extensions and are not part of Algorithm 2 as evaluated in this paper. The two branches serve complementary roles. The windowed solution exploits a longer recent history to provide stronger temporal smoothing and ambiguity conditioning, while the recursive solution is designed to react more rapidly to newly arriving observations and changes in vehicle dynamics.

In live operation, both candidate solutions are assessed using solution status, estimated covariance, residual behaviour and mutual consistency. A stable fixed windowed solution can therefore be favoured when its quality indicators remain consistent, while the recursive branch provides a responsive candidate when the recent window is disturbed by events such as cycle slips, satellite geometry changes or abrupt dynamics. The arbitration stage is part of the real-time architecture shown in Fig. 1; it was not exercised in the offline results reported here, for which only the causal sliding-window branch was executed. Consequently, no end-to-end performance claim is made for the hybrid selector in this paper.

2.8. Multi-Device Trajectory Fusion

A preliminary trajectory-fusion method investigates whether several nearby consumer smartphones can be combined to reduce device-specific positioning errors. Let the horizontal position produced by smartphone j at epoch k be

$$\mathbf{p}_{j,k} = \mathbf{p}_k + \mathbf{e}_{j,k}, \quad (7)$$

where $\mathbf{p}_k$ represents the common vehicle position and $\mathbf{e}_{j,k}$ represents the error associated with device j .

The smartphone solutions are aligned to a common 1 Hz time grid. When M_k valid devices are available, the present proof-of-concept uses

$$\hat{\mathbf{p}}_k = \frac{1}{M_k} \sum_{j=1}^{M_k} \mathbf{p}_{j,k}. \quad (8)$$

Under the idealised assumption that the device-specific errors are zero-mean, independent and have equal covariance Σ , averaging M_k co-located solutions reduces the covariance of the fused estimate to

$$\text{Cov}(\hat{\mathbf{p}}_k) = \frac{\Sigma}{M_k}. \quad (9)$$

This provides the basic statistical motivation for the fusion approach: uncorrelated receiver-specific errors are progressively attenuated as more independent devices contribute to the estimate. In practice, however, smartphone positioning errors are not fully independent. Co-located devices observe nearly

the same satellite geometry and propagation environment, and common multipath, NLOS effects and other shared error sources can introduce correlation between their solutions. Consequently, the theoretical $1/M_k$ covariance reduction should be regarded as an ideal limiting case rather than an expected scaling law for the present experiment.

The present implementation uses equal weighting and therefore does not account for differences in device quality or solution uncertainty. A future extension will use uncertainty-aware fusion, for example by weighting each smartphone solution according to its estimated covariance or integrity indicators. Such an approach should reduce the influence of degraded devices while retaining the benefits of cooperative positioning.

3. Experimental Setup and Data Collection

The evaluation uses road-survey datasets collected in Rome as part of the ANAS S.p.A. and European Space Agency Grand Challenge. Although the broader ANAS–ESA Grand Challenge targeted real-time smartphone positioning, the material available for the present evaluation consisted of recorded field datasets. The causal processing adopted here was therefore evaluated offline under repeatable conditions, while retaining the no-future-observation constraint required for subsequent real-time operation.

Raw smartphone measurements were recorded using Google GnsLogger, while the reference vehicle trajectory was provided by the Cartesio mobile laboratory.

For one of the two analysed acquisitions, the available field documentation records measurements collected on 27 October 2025 between 10:30 and 13:00 local time (UTC+1). Four smartphones were positioned along the lower part of the Cartesio windscreen in the following order from left to right: Samsung A34 5G, Redmi Note 13 Pro, Samsung A34 5G and RedMagic NX769J. The vehicle was equipped with an Applanix POS LV V5 positioning system [16, 17].

RTCM correction data for the Rome evaluation were obtained from the ROMA3 (ROM3) reference station, equipped with a TPS NET-G5 receiver and a TPSG5_A1 antenna, located at latitude 41.8553° N, longitude 12.4660° E and altitude 63.45 m [18].

The documented smartphone coordinates relative to the top of the POS LV antenna were (4.10, -1.365 , -0.11) m for Samsung A34 #1, (4.10, -1.365 , 0.00) m for the Redmi Note 13 Pro, (4.10, -1.365 , 0.11) m for Samsung A34 #2 and (4.10, -1.365 , 0.22) m for the RedMagic NX769J. The field documentation notes that the stated reference does not include the antenna phase-centre correction [17].

Five smartphone devices appear across the evaluated datasets. Four of these (Samsung A34 #1, Samsung A34 #2, Redmi Note 13 Pro and RedMagic NX769J) form the windscreen-mounted array documented in the supplied field sheet. The Oppo Find X5 appears in the evaluation results but is not included in that four-phone field sheet, so an equivalent lever-arm value is not reported here. The Algorithm 2 processing modes are summarised in Table 1.

Table 1

Smartphone processing modes used by Algorithm 2. “Redmi Note 13” refers to the Redmi Note 13 Pro unit described in Section 3; the short form is used throughout the tables for brevity.

Device	Processing mode	Observations
Samsung A34 #1	Carrier phase	Code, Doppler, phase
Samsung A34 #2	Carrier phase	Code, Doppler, phase
Redmi Note 13	Carrier phase	Code, Doppler, phase
Oppo Find X5	Code-based	Code, Doppler
RedMagic NX	Code-based	Code, Doppler

Table 1 summarises the observation types available to Algorithm 2 for each smartphone. Carrier-phase observations were available in the recorded datasets for the two Samsung A34 units and the

Redmi Note 13 Pro. For the Oppo Find X5 and RedMagic NX769J, valid accumulated delta range (ADR) measurements corresponding to carrier phase were not available in the recorded Android raw GNSS observations. These datasets were therefore processed using pseudorange and Doppler observations and are referred to as *code-based* datasets throughout the remainder of the paper. This classification reflects the measurements available in the evaluated datasets and should not be interpreted as a general statement that the corresponding handset models are incapable of providing carrier-phase observations under all firmware or software configurations.

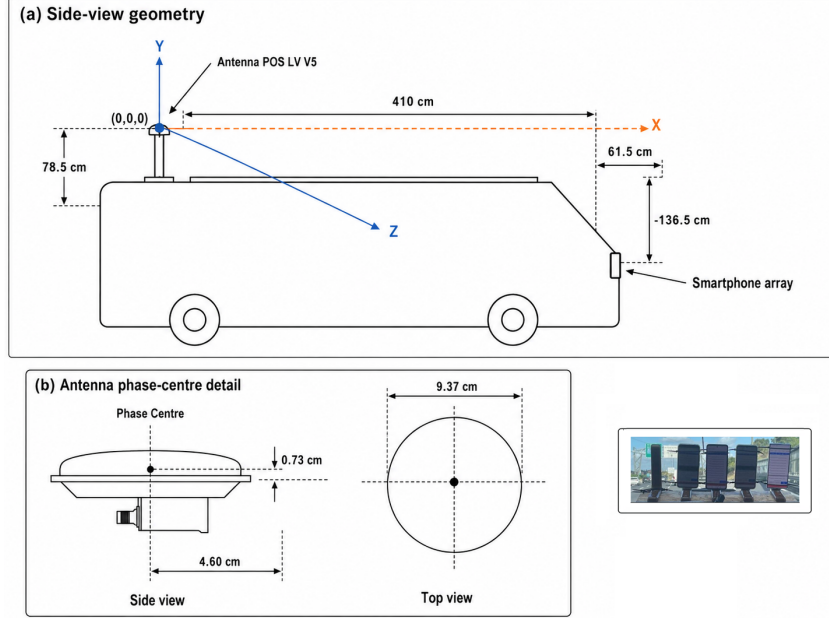


Figure 2: Cartesio mobile laboratory measurement geometry. The POS LV V5 antenna defines the reference origin, while the smartphone array was positioned at the base of the windscreen. The dimensions indicate the measured lever-arm geometry used during the field campaign.

The Applanix POS LV system installed on Cartesio provides the reference trajectory. Smartphone positions are produced at approximately 1 Hz, while the reference trajectory contains more frequent samples. Each smartphone epoch is associated with the nearest reference epoch in time before horizontal error is calculated. For smartphone position $\mathbf{p}_k$ and the associated reference position $\mathbf{p}_k^{\text{ref}}$, horizontal error in a local horizontal coordinate frame is

$$e_k = \left\| \mathbf{p}_k - \mathbf{p}_k^{\text{ref}} \right\|_2. \quad (10)$$

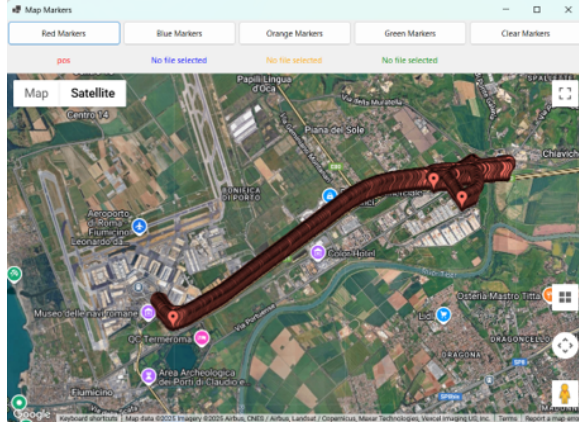
The exact timestamp-association tolerance and the detailed POS LV processing mode are not available in the supplied field sheet. The survey documentation also states that the reported lever-arm coordinates are referenced to the top of the POS LV antenna rather than its phase centre. These points are treated as limitations of the present work-in-progress evaluation.

4. Preliminary Results

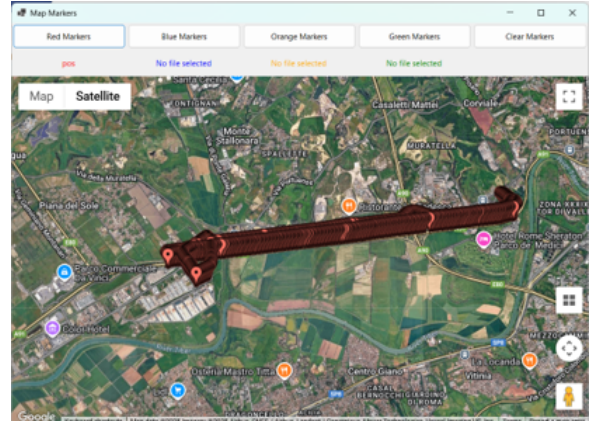
The current results compare Algorithm 2 with the Algorithm 1 NMEA baseline. They are reported as preliminary because the number of valid epochs differs between the two processing approaches. Consequently, the percentages below describe each algorithm’s available solution set and should not be interpreted as a strictly matched statistical comparison. A final matched analysis will use the common intersection of Algorithm 1, Algorithm 2 and Cartesio timestamps. The 0.5 m, 1 m and 2 m thresholds are used here as descriptive accuracy bands, while the > 3 m category is used to expose the large-error tail; these bands are not intended to represent formal C-ITS service-acceptance limits.

For clarity, the two road datasets analysed in this paper are referred to simply as Route 1 and Route 2.

4.1. Route 1



(a) Route 1



(b) Route 2

Figure 3: Road corridors used for the field evaluation: (a) Route 1 and (b) Route 2. The plotted points correspond to the Cartesio reference trajectory used as ground truth for the smartphone evaluation.

Table 2

Preliminary Route 1 cumulative horizontal-error coverage.

Device	Alg.	N	≤ 0.5 m	≤ 1 m	≤ 2 m	> 3 m
Samsung A34 #1 CP	A2	1083	27.9%	52.5%	76.9%	12.7%
Samsung A34 #1 CP	A1	1783	31.6%	66.2%	80.4%	7.3%
Samsung A34 #2 CP	A2	1172	36.4%	59.8%	78.6%	10.3%
Samsung A34 #2 CP	A1	1787	33.5%	67.0%	83.2%	4.0%
Redmi Note 13 CP	A2	1689	26.8%	46.8%	64.3%	13.1%
Redmi Note 13 CP	A1	1787	27.8%	46.1%	68.2%	9.2%
Oppo Find X5 CB	A2	1781	13.6%	26.0%	45.0%	36.7%
Oppo Find X5 CB	A1	1790	20.8%	49.3%	81.6%	1.8%
RedMagic NX CB	A2	1777	16.6%	31.3%	51.4%	24.4%
RedMagic NX CB	A1	1788	24.9%	49.5%	65.1%	4.5%

CP denotes processing with usable carrier-phase observations; CB denotes code-based processing using pseudorange and Doppler.

Within the currently available Route 1 solution sets, the datasets providing usable carrier-phase observations provide the strongest Algorithm 2 results. Algorithm 1 shows higher coverage at the 1 m and 2 m thresholds for all devices, although the Redmi Note 13 is close to the baseline at the 1 m threshold. The code-based Algorithm 2 solutions show a substantially larger error tail, most clearly for the Oppo Find X5, where 36.7% of valid epochs exceed 3 m compared with 1.8% for Algorithm 1. Because the two algorithms currently contain different epoch sets, these differences are descriptive rather than a matched comparison. The code-based processing mode remains a principal target for the next development cycle.

4.2. Route 2

Two Samsung A34 units were mounted during Route 2, as in Route 1. The Samsung A34 #2 recording was excluded because of data-quality problems in the recorded observations; therefore, only the Samsung A34 #1 result is reported for Route 2.

Within the currently available Route 2 solution sets, the datasets providing usable carrier-phase observations provide the strongest Algorithm 2 performance. For the Samsung A34 #1 and Redmi Note 13, more than half of the valid Algorithm 2 epochs fall within 1 m. The code-based Oppo Find X5 and

Table 3

Preliminary Route 2 cumulative horizontal-error coverage.

Device	Alg.	N	≤ 0.5 m	≤ 1 m	≤ 2 m	> 3 m
Samsung A34 #1 CP	A2	1154	26.9%	52.4%	78.7%	6.3%
Samsung A34 #1 CP	A1	1604	33.9%	49.0%	76.4%	5.3%
Redmi Note 13 CP	A2	1508	37.0%	60.0%	76.8%	6.2%
Redmi Note 13 CP	A1	1602	24.7%	62.5%	82.1%	6.9%
Oppo Find X5 CB	A2	1586	15.8%	30.9%	51.9%	27.2%
Oppo Find X5 CB	A1	1595	29.0%	68.5%	88.1%	1.6%
RedMagic NX CB	A2	1590	20.1%	39.8%	61.1%	14.2%
RedMagic NX CB	A1	1604	48.4%	67.6%	90.8%	1.8%
Four-device fusion	–	1600	35.9%	63.8%	81.5%	7.6%

CP denotes processing with usable carrier-phase observations; CB denotes code-based processing using pseudorange and Doppler.

RedMagic NX datasets show a larger error tail. The four-device fused trajectory achieves 63.8% of epochs within 1 m, compared with 39.8% for the individual RedMagic NX solution, an increase of 24.0 percentage points (60.3% relative). As for Route 1, the Algorithm 1 and Algorithm 2 percentages are not yet based on identical epoch sets.

4.3. Combined Graphical Comparison Across Both Routes

Tables 2 and 3 provide the complete cumulative threshold results. To avoid repeating the same information across several figures, Fig. 4 combines the two quantities most relevant to the present discussion for both road datasets: the proportion of valid epochs within 1 m and the proportion with horizontal error greater than 3 m. Algorithm 1 and Algorithm 2 are shown side by side for each available smartphone dataset. The Samsung A34 #2 result appears only for Route 1 because the corresponding Route 2 recording was excluded because of data-quality problems. The four-device fusion result is retained in Table 3 and discussed separately in Section 5, but is omitted from the figure to keep the algorithm comparison visually compact.

5. Discussion

5.1. Observation Capability and Error-Tail Behaviour

The clearest pattern across the two routes is the separation between datasets that expose usable carrier-phase observations and those processed using code and Doppler only. The carrier-phase devices allow Algorithm 2 to form carrier-phase differential observations and estimate integer ambiguities, whereas the code-based processing mode cannot exploit ambiguity fixing. On Route 2, the Algorithm 2 proportion of epochs exceeding 3 m is 6.3% for Samsung A34 #1 and 6.2% for the Redmi Note 13, compared with 27.2% for the Oppo Find X5 and 14.2% for the RedMagic NX. Route 1 shows the same qualitative tendency, with the largest tails again occurring on the code-based Oppo and RedMagic datasets.

This distinction is important because it suggests that the present weakness of the code-based processing mode is not adequately described by a single nominal accuracy figure. The large-error tail is a major part of the observed degradation. This is especially visible in Fig. 4. For C-ITS-oriented positioning, such rare but large errors can be more important than a modest change in central accuracy because they can produce incorrect road-link, geofence or relative-position decisions.

Algorithm 1 should also be interpreted carefully. It starts from Google’s already calculated NMEA position output and subsequently applies filtering and smoothing, while Algorithm 2 operates from the raw GNSS observables available through the Android measurement interface. Algorithm 1 can therefore benefit from processing internal to the handset or Google location stack that is not available

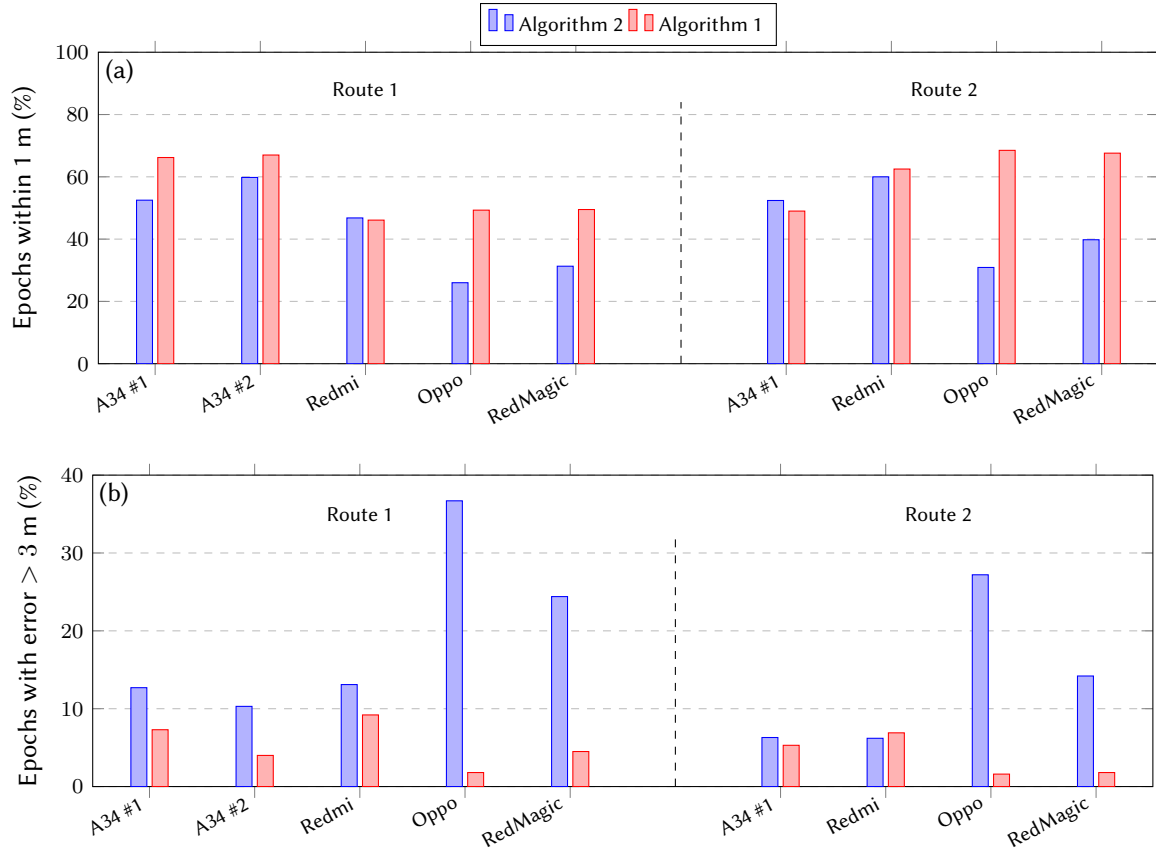


Figure 4: Combined graphical comparison of the two road datasets. Panel (a) shows cumulative coverage within 1 m and panel (b) shows the proportion of valid epochs with horizontal error greater than 3 m. Algorithm 1 is the NMEA-based operational baseline and Algorithm 2 is the raw-GNSS-observation processing method evaluated in this work. Values refer to each algorithm’s available solution set and are therefore descriptive rather than a matched-epoch comparison.

to the transparent raw-observation branch. The comparison in this paper is consequently useful as an operational baseline, but it is not a comparison of two equivalent raw-observable estimators.

The present results point to a focused optimisation path for Algorithm 2. In particular, the code-based processing mode can be extended with lightweight innovation or residual gating, signal-quality and elevation-dependent weighting, short-window Doppler-assisted code smoothing, and stronger rejection or down-weighting of measurements affected by multipath and NLOS reception. These methods are intended to reduce the tail while preserving the causal, low-latency character of the real-time engine; they have not yet been applied to the results reported here.

5.2. Multi-Device Fusion as Cooperative Positioning

The trajectory-fusion experiment provides a second observation from the field data. For Route 2, the individual RedMagic NX Algorithm 2 solution provides 20.1%, 39.8% and 61.1% cumulative coverage at 0.5 m, 1 m and 2 m, respectively. The four-device fused trajectory increases these values to 35.9%, 63.8% and 81.5%, corresponding to gains of 15.8, 24.0 and 20.4 percentage points. The proportion of epochs above 3 m is also reduced from 14.2% for the individual RedMagic trajectory to 7.6% for the fused result.

The proof-of-concept model in Eq. 8 assumes that at least part of the smartphone error is device-specific rather than perfectly correlated across all handsets. Under this assumption, averaging co-located trajectories can suppress some receiver-specific noise and short-duration excursions while retaining the common vehicle motion. The experiment is relevant to a server-based C-ITS architecture because the cooperative operation can be performed centrally without increasing the GNSS processing load on

each smartphone.

The result should nevertheless not be interpreted as evidence of fleet-scale performance. Only four co-located devices were used, the degree of correlation between their GNSS errors has not been independently estimated, and the comparison with one individual RedMagic solution is illustrative rather than a controlled ablation study. A future implementation should replace equal weighting with uncertainty-aware fusion and should evaluate larger numbers of independently configured handsets and vehicles.

5.3. Implications for C-ITS-Oriented Operation

The field results should be interpreted as evidence of positioning capability rather than as certification of any specific C-ITS service requirement. The current performance is most relevant to applications in which positioning supports road-segment awareness, direction of travel, vehicle behaviour and coarse proximity, where small residual position offsets are less critical. Examples include road-hazard notification, slow or stationary vehicle detection, traffic-state reporting and other services that primarily require reliable identification of the road segment and direction of movement.

Table 4

Indicative interpretation of the observed positioning performance for C-ITS-oriented application classes. The positioning scales are used as engineering reference levels rather than formal service-certification requirements.

Application class	Indicative positioning scale	Readiness from present evaluation
Road-hazard and traffic-state awareness	Road-segment level	Promising. Metre-scale positioning is achieved for a substantial proportion of epochs, although the code-based datasets show a heavier large-error tail.
Cooperative vehicle-level positioning	1–2 m	Supported experimentally. On Route 2, the carrier-phase Algorithm 2 solutions achieve 76.8–78.7% coverage within 2 m, while the four-device fused trajectory reaches 81.5% within 2 m.
Lane-level or fine-geofence positioning	≤ 0.5 m	Work in progress. Across the evaluated carrier-phase Algorithm 2 datasets, approximately 27–37% of valid epochs fall within 0.5 m. Further improvement and stronger tail suppression are required for reliable lane-level operation.

Table 4 summarises the application-oriented interpretation of the present results. It is intended as a concise engineering reading of the field evidence rather than a definition of formal C-ITS performance requirements.

More demanding cooperative safety applications place greater emphasis on positioning integrity and on maintaining a low probability of large errors. In these cases, the heavier error tails observed for some devices, particularly those processed without carrier-phase observations, remain an important limitation. Such applications would benefit from stronger outlier suppression, measurement-integrity monitoring and explicit uncertainty information. Applications requiring reliable lane assignment, fine geofencing or decimetre-level localisation under multipath and NLOS conditions are not demonstrated by the present work-in-progress evaluation. The results therefore support the proposed architecture as a basis for C-ITS-oriented positioning experiments, while further validation is required before extending the conclusions to lane-level or safety-critical positioning functions.

This distinction is also relevant to the architecture adopted in this work. Raw GNSS measurements are acquired by the smartphone, while the more computationally demanding estimation and cooperative processing are performed on the server. This separation provides a natural path for introducing future integrity monitoring, adaptive measurement weighting and multi-device cooperative positioning without substantially increasing the computational burden on individual handsets. The present field evaluation, however, primarily validates positioning accuracy using recorded observations. Complete

end-to-end latency, communication delay, C-ITS message timing, service-level logic and operational safety integrity remain to be evaluated in live trials.

5.4. Limitations

The principal limitation of the present evaluation is that Algorithm 1 and Algorithm 2 have not yet been recomputed using an identical set of valid epochs. The percentages reported in Tables 2 and 3, therefore describe the available solution sets produced by each processing approach and should not be interpreted as strict paired statistical comparisons. A subsequent evaluation should use the common intersection of Algorithm 1, Algorithm 2 and Cartesio timestamps and report additional distribution statistics, including median horizontal error, root mean square error (RMSE), minimum and maximum error, and upper-tail percentiles, on the matched epoch set.

Full end-to-end positioning latency has also not yet been characterised under live vehicle operation. Although the complete architecture includes a recursive real-time estimator and hybrid solution-selection stage, only the causal sliding-window branch was exercised for the offline results reported in this paper. Consequently, the present experiments validate the core positioning approach using recorded field observations but do not constitute complete validation of the real-time architecture.

Additional limitations arise from the available field documentation. The same level of mounting and reference-processing metadata is not available for every smartphone and acquisition. The reported smartphone lever arms are referenced to the top of the POS LV antenna rather than to its antenna phase centre, while the exact tolerance used for associating smartphone epochs with the higher-rate reference trajectory is not available. These uncertainties are retained explicitly in the analysis because they define important next steps for a more rigorous matched-epoch and live real-time validation campaign.

6. Conclusion and Future Work

This paper presented a smartphone GNSS positioning architecture developed by PIXAM for Smart Road and C-ITS-oriented applications. The same server-side positioning core can support live RTK operation from streamed Android raw measurements and offline PPK processing from recorded observations. Algorithm 1 provides an NMEA-based operational baseline, while Algorithm 2 works directly from raw smartphone GNSS observables using a 300-epoch causal sliding-window PPK estimator. The complete real-time architecture additionally contains a recursive estimator and hybrid solution-selection stage.

The Rome field evaluation against the Cartesio POS LV reference system shows a clear dependence on smartphone observation capability. Algorithm 2 performs most consistently on datasets providing usable carrier-phase measurements, while the present code-based processing mode exhibits a substantially heavier large-error tail. The combined graphical analysis reinforces this distinction across both road datasets. On Route 1, the Algorithm 2 proportions of valid epochs with horizontal error above 3 m are 36.7% and 24.4% for the Oppo Find X5 and RedMagic NX datasets, respectively. On Route 2, the corresponding values are 27.2% and 14.2%, while the carrier-phase Samsung A34 #1 and Redmi Note 13 datasets are substantially lower at 6.3% and 6.2%, respectively.

The Algorithm 1 and Algorithm 2 results are not yet based on identical sets of valid epochs. Consequently, the reported percentage differences between the two processing approaches should be interpreted as descriptive trends rather than as statistically validated paired improvements. A subsequent evaluation will use matched epochs from the common intersection of the smartphone solutions and Cartesio reference timestamps.

A co-located four-device trajectory-fusion experiment also demonstrates the potential of cooperative processing. The fused result achieves 63.8% of positions within 1 m and 81.5% within 2 m, slightly exceeding the strongest individual Algorithm 2 result at these thresholds. Relative to the individual RedMagic NX trajectory, the proportion of epochs above 3 m is reduced from 14.2% to 7.6%. This result is treated as a proof of concept rather than as evidence of fleet-scale performance.

The next development stage will focus on four tasks: matched-epoch re-evaluation against Cartesio; improved detection and down-weighting of multipath- and NLOS-affected measurements; short-window

Doppler-assisted code smoothing and uncertainty-aware multi-device fusion; and live vehicle validation of the recursive estimator, hybrid solution selection and complete end-to-end latency. A further objective is to integrate GNSS measurement integrity indicators so that the final position is accompanied by information about observation stress and solution reliability. These steps are required before moving from the present architecture and field evidence to a fully validated real-time C-ITS positioning service.

Acknowledgments

The authors thank ANAS S.p.A. and the European Space Agency for the field campaign and for providing smartphone and Cartesio datasets used in this evaluation.

Declaration on Generative AI

During the preparation of this manuscript, the authors used ChatGPT solely to support language editing, including grammar, spelling, and clarity. All generated suggestions were carefully reviewed and revised by the authors, who retain full responsibility for the accuracy, integrity, and final content of the publication.

References

- [1] European Commission, *A European Strategy on Cooperative Intelligent Transport Systems, a Milestone Towards Cooperative, Connected and Automated Mobility*, COM(2016) 766 final, Brussels, 2016.
- [2] I. Vourgidis, L. Maglaras, A. S. Alfakeeh, A. H. Al-Bayatti, and M. A. Ferrag, “Use of Smartphones for Ensuring Vulnerable Road User Safety through Path Prediction and Early Warning: An In-Depth Review of Capabilities, Limitations and Their Applications in Cooperative Intelligent Transport Systems,” *Sensors*, vol. 20, no. 4, 997, 2020.
- [3] G. Burger and J. Guna, “Enhancing Driving Safety through User Experience Evaluation of the C-ITS Mobile Application: A Case Study of the DARS Traffic Plus App in a Driving Simulator Environment,” *Sensors*, vol. 24, no. 15, 4948, 2024.
- [4] J. Paziewski, M. Fortunato, A. Mazzoni, and R. Odolinski, “An Analysis of Multi-GNSS Observations Tracked by Recent Android Smartphones and Smartphone-Only Relative Positioning Results,” *Measurement*, vol. 175, 109162, 2021.
- [5] J. Geng, C. Long, and G. Li, “A Robust Android GNSS RTK Positioning Scheme Using Factor Graph Optimization,” *IEEE Sensors Journal*, vol. 23, no. 12, pp. 13280–13291, 2023.
- [6] A. H. Pourmina, M. M. Alizadeh, and H. Schuh, “Decimeter-Level Accuracy for Smartphone Real-Time Kinematic Positioning Implementing a Robust Kalman Filter Approach and Inertial Navigation System Infusion in Complex Urban Environments,” *Sensors*, vol. 24, no. 18, 5907, 2024.
- [7] J. Yun, S. Kim, T. Youn, G. Kim, W. Park, and B. Park, “Third Place Winner of the Smartphone Decimeter Challenge: Improving Smartphone Accuracy with a Two-Step Accurate Velocity Estimation,” in *Proceedings of the 37th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2024)*, Baltimore, MD, USA, 2024, pp. 2691–2702.
- [8] T. Suzuki, “Second Place Winner of the Smartphone Decimeter Challenge: An Open-Source Factor Graph Optimization Package for GNSS and IMU Integration in Smartphones,” in *Proceedings of the 37th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2024)*, Baltimore, MD, USA, 2024, pp. 2703–2713.
- [9] N. Motooka, “First Place Winner of the Smartphone Decimeter Challenge: Optimized GNSS/INS Integration Approach for Smartphone Positioning,” in *Proceedings of the 37th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2024)*, Baltimore, MD, USA, 2024, pp. 2714–2728.

- [10] J. Hu, P. Li, J. Feng, F. Zhou, D. Yi, and S. Bisnath, "Enhancing Smartphone Relative Positioning With Partial Wide-Lane Ambiguity Resolution: Path to Real-Time, Decimeter-Level Positioning in User Environments," *IEEE Internet of Things Journal*, vol. 12, no. 3, pp. 3177–3187, 2025.
- [11] A. Grenier, Z. Li, N. Zhu, A. Ometov, E. S. Lohan, V. Renaudin, and J. Nurmi, *Multi-Sensor Dataset in Outdoor and Indoor Environment from Android Smart Devices and ULISS*, Zenodo, 2024, doi:10.5281/zenodo.12566912.
- [12] P. J. G. Teunissen, "The Least-Squares Ambiguity Decorrelation Adjustment: A Method for Fast GPS Integer Ambiguity Estimation," *Journal of Geodesy*, vol. 70, no. 1, pp. 65–82, 1995.
- [13] RTCM Special Committee 104, *RTCM Standard 10403.3: Differential GNSS Services, Version 3*, Radio Technical Commission for Maritime Services, 2016.
- [14] International GNSS Service, *RINEX: The Receiver Independent Exchange Format, Version 3.05*, 2023.
- [15] OASIS, *MQTT Version 5.0*, OASIS Standard, 2019.
- [16] Applanix Corporation, *POS LV: Designed for Integration, Built for Performance*, Trimble Inc., PN 022520-034A, 2023.
- [17] ANAS, *ANAS-ESA Grand Challenge: Road Surveys 2.0 with POS LV and Smartphones*, field campaign documentation, 2025.
- [18] Topnet Live, *Site Information Protocol: ROMA3 (ROM3)*, reference-station site information, 2025.