SmartCoop Algorithm: Improving Smartphones Position Accuracy and Reliability through Collaborative Positioning

Thomas VERHEYDE
TéSA Research Laboratory
Toulouse, FRANCE
thomas.verheyde@recherche.enac.fr

Antoine BLAIS
ENAC, Université de Toulouse
Toulouse, FRANCE
antoine.blais@enac.fr

Christophe MACABIAU
ENAC, Université de Toulouse
Toulouse, FRANCE
christophe.macabiau@enac.fr

François-Xavier MARMET
Centre National d'Études Spatiales
Toulouse, FRANCE
francois-xavier.marmet@cnes.fr

Abstract— **In recent years, our society has been preparing for a paradigm shift toward the hyper-connectivity of urban areas. This highly anticipated rise of connected smart city centers is led by the development of low-cost connected smartphone devices owned by each one of us. In this context, the demand for low-cost, high-precision localization solutions is driven by the development of novel autonomous systems. The creation of a collaborative network will take advantage of the large number of connected devices in today's city center. This paper validates the positioning performance increase of Android low-cost smartphones device present in a collaborative network. The assessment will be made on both simulated and collected smartphone's GNSS raw data measurements. We propose a collaborative method based on the estimation of distances between network mobile users contributing to a *SMARTphone COOPerative Positioning* algorithm (SmartCoop). Previous analysis made on smartphone data allow us to generate simulated data for experimenting our cooperative engine in nominal conditions. The evaluation and analysis of this innovative method shows a significant increase of accuracy and reliability of smartphones positioning capabilities. Position accuracy improves by more than 3m, in average, for all smartphones within the collaborative network.**

Index Terms— **Android GNSS Raw Data Measurements, Smartphones, Collaborative Positioning, Non-linear Constrained Optimization, *SmartCoop* Cooperation Engine, Inter-Phone Ranging (*IPR*).**

I. Introduction

In recent years, we observed an exponential increase of wireless signals being used in today's busy urban areas. This change is supported by technological innovations made available for everyone. The device that comes quickly to mind is everyone's favorite, our smartphone. The increasing need of Location Based Services (LBS) provoked the rapid evolution of smartphones' embedded low-cost Global Navigation Satellite System (GNSS) chipsets within the last few years. Most Android devices are now equipped with multi-constellation and multi-frequency positioning units.

After Google announced the release of Android GNSS raw data measurements on mobile devices, the enthusiasm around those low-cost positioning device quickly grew in the scientific community. In the wake of this revolution, multiple papers were published characterizing Smartphone based GNSS raw data measurements [1] [2]. Preliminary results exposed smartphones weakness to multipath in degraded conditions mainly due to their multi-purposes antenna [3]. On the other hand, the embedded GNSS chipsets proved to be reliable, accurate and efficient compared to higher-end GNSS receivers. The implementation of advanced algorithms such as Precise Point Positioning (PPP) and Real-Time Kinematic (RTK) have also been experimented [4] [5]. However, it was shown that real-world applications were difficult to achieve due to smartphone hardware flaws.

In order to overcome those difficulties, we ambition to develop a collaborative network positioning system between smartphone devices. Cooperative positioning localization have been recently studied for an application in the autonomous vehicular transportation and in the robotic domain [6]. Only few studies have been made on smartphone-based networking. However, research works introduced preliminary groundwork concerning cooperative ranging between smartphones showcased in [7] as well as a collaborative positioning technique in [8].

Our collaborative network will take advantage of the tremendous number of connected devices present in highly frequented city centers. In the context of smartphone hyper-connectivity, we assume that a reliable, secure and efficient communication link is made available to network's users. We developed a *SMARTphone COOPerative Positioning* (SmartCoop) algorithm. Our collaborative positioning engine improve smartphones' positions accuracy and reliability. Its implementation relies on the minimization of users' position errors in an optimization problem constrained by estimated inter-phone distances.

The aforementioned distances are computed by our algorithm based on a double differencing technique. Our approach combine smartphones GNSS raw data measurements, cooperatively, in order to estimate a 3D ranging vector, referred as *Inter-Phone Ranging*, between each user in the network. This paper will be articulated around two main sections. First, our smartphone ranging techniques will be introduce by describing the computation of the 3D Inter-Phone Ranges (IPR) and presenting the results of a static open-sky reference test. Thereafter, our SmartCoop engine will be presented. A thorough analysis will be provided, using simulated data, demonstrating the algorithm positioning accuracy performance for a static open-sky collaborative scenario.

II. Smartphone Ranging Technique

Inter-Phone Ranging (IPR)

Our collaborative system is dependent on the range computation between network's users. We propose a double difference method based on Android raw data measurements. This algorithm is based on the preliminary work made by Gogoi et al [7]. We implemented a new approach for processing efficiently multi-constellation and multi-frequency measurements from peers' smartphones. Our innovative ranging technique allows to compute a 3D inter-user range vector referred as *Inter-Phone Ranges* (IPR).

A. Methodology

This subsection describes the methodology used by our double differentiating algorithm, underlining the good practices regarding the use of Android raw data measurements in a collaborative context. We assume that a communication link is available between network's users for sharing their raw GNSS data. A detailed description of available Android measurements can be found in [9] and [10].

1) Measurements Synchronization:

Previous works demonstrated that a time-synchronization between measurements is mandatory in inter-user ranging methods [7]. The developed time-synchronization technique was built around Doppler-based compensation algorithms, well known from the GNSS community. This method rely on the assumption that relative movement between the satellite and the receiver's clock frequency bias are constant over a short interval of time. This hypothesis holds in the case of smartphone use in low-dynamic and static scenario. Equation 1 shows the time-synchronization of a pseudorange ρ at a time t. Time synchronization is achieved when $\rho_A(t)$ has been interpolated at a time $t + \Delta t$ corresponding to the time of $\rho_B(t + \Delta t)$ between two pseudoranges measurements A and B. Equation 1 shows the interpolation method.

$$\rho(t + \Delta t) = \rho(t) + (\Delta t \;.\; \lambda \;.\; \dot{\phi(t)}) \tag{1}$$

Equation 1 presented above has been adapted for the use of Android GNSS raw data measurements. The term $\lambda \;.\; \dot{\phi(t)}$ representing the Doppler shift can be approximated by the pseudorange rate measurements. This measurement can be easily retrieved from recorded Android data as `PseudorangeRateMetersPerSecond` Secondly, the generation of the term Δt is straightforward since the Android raw measurement, `FullBiasNanos`, is already synchronized to the true GPS time [10]. Δt is simply approximated by a difference of the Android measurements cited above between pseudoranges measurements A and B.

2) Pseudoranges Double Differences:

After measurements synchronization has been achieved, double differences on code measurements are being processed. This operation allows to mitigate common errors shared between users and satellites measurements. At first, a single difference is being computed between two smartphones users and one common satellite. Equation 2 shows this single pseudorange (ρ) difference between users a and b to satellite i.

$$D^i_{ab} = \rho^i_a - \rho^i_b = \Delta r^i_{ab} + \Delta\epsilon^i_{ab} + c \;.\; \Delta b_{ab} \tag{2}$$

At this stage, common propagation errors (atmospheric and ionospheric) and the satellite clock bias are being removed. Terms remaining in the equation are the difference in true range (Δr^i_{ab}) , receivers clock biases (Δb_{ab}), and residual noise ($\Delta\epsilon$). Next step results in taking a second difference between two single differences of two satellites signals shared by both users. This phase is referred as a double difference. Equation 3 presents a double difference between two satellites i and j in view of users a and b. At this step, the difference of two single differences mitigates the impact of both receivers clock biases.

$$\mathrm{DD}^{ij}_{ab} = D^i_{ab} - D^j_{ab} = [\vec{r}_{ab} \;.\; (\vec{e}^{\,i} - \vec{e}^{\,j})] + [\Delta\epsilon^i_{ab} - \Delta\epsilon^j_{ab}] \tag{3}$$

where $\vec{e}^{\,i}$ is the steering vector from the receiver to the satellite i and $\vec{r}_{ab}$ define the range vector between two network's users. We set a list of n common satellites signals between receivers a and b. After classifying common satellites signals by the value of `Cn0DbHz` (ndlr: Signal to Noise Ratio: SNR) and their elevation, we select the best available signal as the reference for our double differences. In equation 3, satellite j will be our reference signal 0. Thus, equation 3 simplifies as:

$$\mathbf{DD}_{ab} = \mathbf{H}\,\vec{r}_{ab} + \epsilon \tag{4}$$

where $\mathbf{DD}_{ab} = [DD^{10}_{ab}, DD^{20}_{ab}, ..., DD^{n0}_{ab}]^T$ is a $[n \times 1]$ matrix of double differences for all common signals between both users. $\mathbf{H} = [(\vec{e}^{\,1} - \vec{e}^{\,0}), (\vec{e}^{\,2} - \vec{e}^{\,0}), ..., (\vec{e}^{\,n} - \vec{e}^{\,0})]^T$ is a $[n \times 3]$ matrix of steering vector differences.

Finally, the $\vec{r}_{ab}$ vector can be estimated via a weighted least square and will be named **IPR**.

$$\mathbf{IPR} = (\mathbf{H}^T\mathbf{W}\,\mathbf{H})^{-1}\,\mathbf{H}^T\mathbf{W}\,\mathbf{DD}_{ab} \tag{5}$$

with $\mathbf{W}$ being the covariance matrix, taking into account correlated noise emerging from the double differences. This error covariance matrix is populated with values computed by error estimates given by the Android API and explicitly shown by the equation below:

$$\sigma^n_\epsilon = \texttt{ReceivedSvTimeUncertaintyNanos}(t) \;.\; 1^{-9} \;.\; c$$

This method allows to efficiently compute **IPR** ranges, however a few drawbacks remains. At least 3 sets of satellite pairs (or 4 common signals) are needed to estimate our range. We also have to keep in mind that the application of a double difference increase significantly background noise initially present on retrieved code measurements.

3) Application to the smartphone domain:

Due to limitations and constraints of our application domain, we recommend to adopt the following guidelines for processing Android GNSS raw data measurements accurately for ranging computation.

For simplification purposes, our methodology presents n satellites to be used as if only one signal was received per satellites. However, today's smartphone embedded positioning chipset are multi-constellation and multi-frequency. We advise to report about double differences as difference between common received signals instead of common received satellites. Moreover, the multiplication of signals availability on smartphone favors the constructions of common received signals pairing. During our data collection campaign, [2], we observed that on average, 18 pairs of common received signals were seen by our tested Android device in static open-sky scenarios. On the other hand, on average, 10 pairs of common received signals were observed in urban environment, which easily allow for the computation of IPR ranging. Finally, after a careful selection of GNSS raw measurements by following mechanisms described in [11] and before time synchronization, we suggest to set up a Hatch filter. This filter is used for smoothing smartphones' estimated pseudoranges and accounts for non-continuous signals segments. Our implementation of the Hatch filter is presented by equation 6.

$$\tilde{\rho}(t) = \frac{1}{k}\,\rho(t) + \frac{k-1}{k}\,[\rho(t-1) + \lambda\ .\ \phi(t-1)] \qquad (6)$$

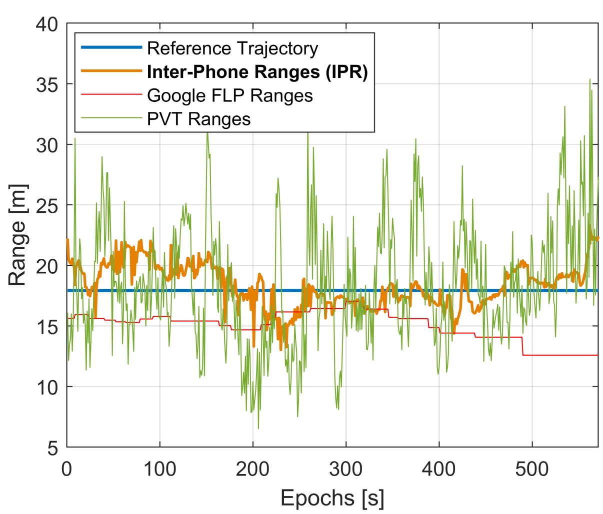

Fig. 1. Inter-Phone Ranging Comparison Performance for Static & Open-Sky Environment Scenario

TABLE I
RANGES ESTIMATION PERFORMANCE

Ranging techniques	Static & Open-Sky $[600sec]$ $\mu[m]$	$\sigma[m]$	$RMS[m]$
IPR	0.41	1.75	1.79
FLP	3.21	1.82	3.68
PVT	0.84	5.40	5.46

where k is characterized by the length of a continuous time segment (in epochs) where a specific signal has been received and correctly retrieved by our embedded smartphone GNSS receiver. Each time a signal loss of track is detected, a reinitialization of our parameter k is made.

B. Results Analysis

In this section, we are presenting a performance analysis of our Inter-Phone Ranging (IPR) algorithm. In order to draw a fair comparison analysis of our estimated inter-users ranges, we take into consideration other means of range computation available to smartphone users. The first process is straightforward, it consists of estimating smartphones position by a Weighted Least Square (WLS) algorithm and using the obtained positions to compute a range between two users. The second technique is taking advantages of a unique feature available to Android mobiles' users, the Google Fused Location Provider (FLP). Those retrieved positions will allow us to evaluate ranges between two smartphone peers. FLP positions are intended to be the ultimate positioning solution obtainable by a given Android device. However, the black-box processes used by FLP measurements while combining GNSS, cellular network and sensors informations, makes them ambiguous and unreliable for scientific analysis. On the other hand, they provide a good reference for intrinsic quality of smartphones positioning capabilities. This reason justify their use in our performance analysis for ranging purposes.

1) Experimental Protocol:

Our results analysis will be based on a static scenario in open-sky conditions. This preliminary analysis will allow us to assert IPR estimation performance for a nominal case. The data used were retrieved during a data collection campaign in Toulouse. During this campaign, multiple collaborative scenario were put in place, including nominal static open-sky cases. Our data collection campaign featured 7 modern Android smartphones (2× Honor View 20, Google Pixel 3, Xiaomi Mi 9, 2× Xiaomi Mi 8 and a Huawei Mate 20X). The combination of Android devices tested represent a large variety of brands and models equipped with a diverse array of positioning chipsets (Qualcomm, HiSilicon and Broadcom). Most of them, have multi-constellation and multi-frequency capabilities (except Google Pixel 3). Finally, all smartphones were running the Android Q (10.0) Operating System (OS) version.

The following results analysis is based on two Xiaomi Mi8, placed on the rooftop of two static vehicles under open-sky conditions. The reference trajectory, in this context referred to as our baseline, was recorded by two high-end receivers (NovAtel SPAN) on the two vehicles.

2) Static & Open-Sky Scenario:
Our static, open-sky scenario lasted for 10min. The reference baseline computed during this test was equal to 17.65m. Figure 1 shows the ranging performances of 3 ranging strategies (IPR, FLP and PVT) compared to the reference baseline. Ranges are here plotted as the norm of the estimated vector. The first observation that can be drawn is the excellent behavior displayed by our double-differenced ranging techniques (IPR) compared to the extracted PVT ranges. The FLP ranges seems to be impacted by a bias all along the test. The overall performance of the IPR estimated ranges exceeds the one of PVT and FLP computed ranges.
This conclusion is supported by the error characterization provided in table I resulting from our experiment. We observe a standard deviation of 1.75m and a mean error of 0.41m for the IPR estimated ranges. Those results comfort our idea that Inter-Phone Ranging (IPR) estimation technique is suitable to be used for collaborative positioning. Furthermore, similar analysis were made for urban canyons showing comparable results and will be presented in future work. The controlled conditions of our experiment setup, and the repetition of previously shown results with other Android devices, demonstrate the reliability of our analysis in nominal conditions.

III. Collaborative Network Positioning *SmartCoop Algorithm*

This section will present our proposed smartphone collaborative algorithm, referred as SmartCoop. Our innovative cooperative engine aims at improving positioning performances of smartphone users. Collaborative positions from network's members will be computed based on a first raw position estimate (an approximate PVT position) while being restrained by previously computed Inter-Phones Ranges (IPR) between all smartphones users.
The following methodology will characterize the constitution of the nonlinear constrained optimization problem that we need to solve. Then, it will be followed by the presentation of SmartCoop algorithm performance. A simulation of a 10-smartphones network will be made demonstrating the efficiency of our collaborative algorithm.

A. Methodology

This cooperative algorithm aims at improving network's users positions. A similar method, developed for the automotive industry has been studied in [6].

1) Definition:
We aim at characterizing our nonlinear constrained optimization problem in order to estimate new cooperative positions. The objective is to minimize the 3D positions error between

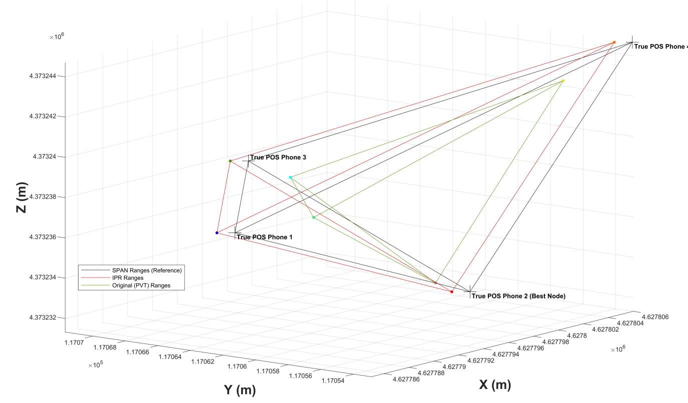

Fig. 2. IPR Constraints Visualization - 3D [ECEF]

the newly estimated cooperative positions $\hat{p}$ and their reciprocal true positions p_{true}. This minimization is leveraged by 3D Inter-Phones Ranges (**IPR**) vectors computed between each individual forming the collaborative network. The array of network's member positions is denoted by a capital letter and is approximated by equation 7.

$$\hat{P} = \min \sum_{k=1}^{M} ||\, \hat{p}^k - p_{true}^k \,|| \tag{7}$$

with M being the total number of peers forming the cooperative network. If we consider that the positioning errors follows a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, then equation 7 can be derived as:

$$\hat{P} = \min \sum_{k=1}^{M} \frac{(\hat{p}_x^k - \tilde{p}_x^k - \mu_x)^2}{2\sigma_x^2} + \frac{(\hat{p}_y^k - \tilde{p}_y^k - \mu_y)^2}{2\sigma_y^2} + \frac{(\hat{p}_z^k - \tilde{p}_z^k - \mu_z)^2}{2\sigma_z^2} \tag{8}$$

Detailed derivation of the equation above has been performed by Liu et al in [6] [12]. Equation 8 will be utilize as our objective function for our optimization problem. We solve the above mentioned nonlinear constrained optimization problem by setting a solver-based Matlab algorithm employing the `fmincon` function.

2) Constraints & Hypothesis:
The objective function that we are minimizing is binded by nonlinear constraints. A constraint is defined by a norm difference between two estimated cooperative positions. It is then declined on x, y and z axes, in order to bind our system in the 3D positioning domain. For our collaborative engine, two independent users, i and j, are constrained by a set of equations, shown in 9.

$$\text{Set of Constraints} = \begin{cases} \text{IPR}_x^{ij} = \sqrt{(\hat{p}_x^i - \hat{p}_x^j\,)^2} \\ \text{IPR}_y^{ij} = \sqrt{(\hat{p}_y^i - \hat{p}_y^j\,)^2} \\ \text{IPR}_z^{ij} = \sqrt{(\hat{p}_z^i - \hat{p}_z^j\,)^2} \end{cases} \tag{9}$$

The number of constraints sets needed for bidding each smartphone users in the network vary. The number of sets needed can be easily computed by:

$$\text{Number of constraints' set} = \frac{M(M-1)}{2} \tag{10}$$

Figure 2 represents a 3D representation of constraints bidding an array of positions. This example involves a network of four unique smartphones, thus generating 6 set of constraints totaling 18 equations restraining our minimization process. For analysis purposes, the true positions (represented by crosses) and true range (black lines) are pictured on this figure. Obviously, none of those parameters were used in the collaborative engine and are defined here only for evaluation purposes. The green lines and light colored spheres represent initial approximate PVT positions and the resulting unsatisfied constraints. Whereas, the red lines represent the satisfied constraints (defined by **IPR**). Newly estimated collaborative positions are drawn in darker color and are linked by the red lines. This figure clearly demonstrates the importance of satisfying the previously defined constraints and shows how they properly impact the new estimation of collaborative positions. The main hypothesis made by our algorithm is that one smartphone, from the established cooperative network, will be defined as the best performer in standalone positioning. This implies that the initial positions given to our algorithm ($\tilde{p}_{BestNode}$) will be more accurate. This phone is referred as the *Best Node* of the network. Consequently, this hypothesis helps setup the initial constraints structure. We consider this hypothesis as credible, since one user of the collaborative network could be using a more recent device and/or be in better reception conditions than anyone else in the network justifying the higher intrinsic positioning performance of our best node device. The straight application of this hypothesis is illustrated by phone 2 on both figure 2 and 3.

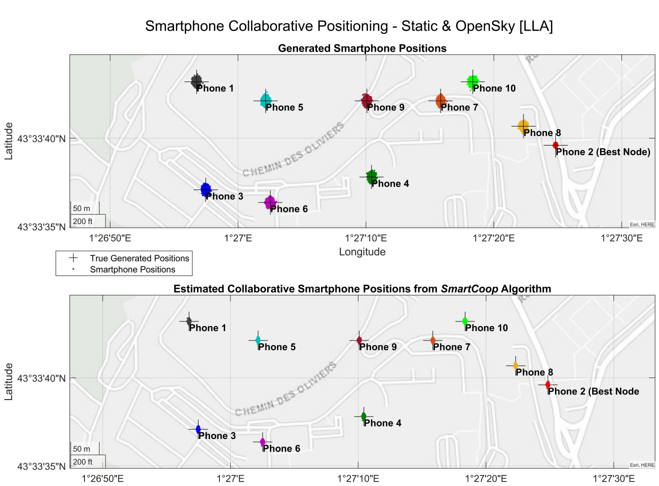

Fig. 3. Smartphone Collaborative Positioning Analysis [LLA]

TABLE II
SMARTCOOP POSITIONING PERFORMANCE

Network's Smartphones	**SmartCoop** Positions			
	$\mu[m]$	$\sigma[m]$	$\Delta\sigma[m]$	ΔRMSE$[m]$
Phone 1	0.06	1.61	-3.27	-4.60
Phone 2 (*Best Node*)	0.03	1.34	+0.11	+0.16
Phone 3	0.06	1.63	-3.26	-4.67
Phone 4	0.05	1.63	-3.23	-4.62
Phone 5	0.03	1.63	-3.31	-4.52
Phone 6	0.04	1.61	-3.22	-4.53
Phone 7	0.01	1.75	-3.27	-4.51
Phone 8	0.02	1.82	-3.29	-4.68
Phone 9	0.08	1.60	-3.21	-4.40
Phone 10	0.02	1.65	-3.22	-4.63

B. *Results Analysis*

This final section will present the results obtained by our SmartCoop algorithm using simulated smartphone positions and simulated inter-phone ranges (**IPR**) vector.

1) *Simulated Data:*

In order to analysis the positioning performance of our SmartCoop collaborative engine, we have simulated smartphone data measurements based on previous studies [13]. Initial smartphones positions are randomly generated following a centered Gaussian distribution. Standard error deviation used to generate those positions have been selected based on previous observation made on smartphones devices for static open-sky scenarios. The error position standard deviation has been selected as: $\sigma_{\tilde{p}} = [2.5, 2.5, 3.8]$ in meters on x, y and z for all simulated smartphone except *Phone 2*. Indeed, the second generated user (*Phone 2*), has been randomly selected to be the best node of our system. The error distribution for this phone has been set by: $\sigma_{\tilde{p}_{BestNode}} = [1, 1, 2]$ on x, y and z in meters.
Based on our analysis made in section II-B, we can also generate Inter-Phone Ranges (IPR) with confidence. We simulated 3D IPR vectors based on the statistical analysis provided above. Furthermore, for this analysis, we constituted a network of 10 collaborative smartphone users. Their initially generated positions are shown on figure 3, on the top graph. Each dot represents an estimated position at 1Hz frequency and their associated cross describes their reference position. Smartphone data and IPR ranges have been simulated for 3600 epochs.

2) *Collaborative Positioning:*

Figure 3 demonstrate the efficiency of the SmartCoop algorithm. The top graph, shows the smartphone positions before collaboration. The comparison can be directly made with the plot below, displaying the newly estimated collaborative positions. Collaborative positions have significantly improved users position accuracy. Moreover, the positions dispersion have been reduced, thus increasing smartphone position reliability. If we analyze the performance of our SmartCoop algorithm epoch by epoch, we notice that our algorithm increase position accuracy more than 92% of the time for all smartphones within the network.

Table II presents a statistical characterization of positioning error of the newly estimated cooperative positions. The first two columns shows the statistical distribution of the newly estimated collaborative position error. Those parameters are to be compared with the simulated standalone position errors exposed in III-B1 as $\sigma_{\tilde{p}}$. Thus, the two last columns of the table, display measures variation between initial positions ($\tilde{p}$) and collaborative positions from SmartCoop ($\hat{p}$). Negative values indicate an improvement observed with our collaborative method. Essentially, this table validate the improving positioning performances induced by SmartCoop. All smartphones (except *Phone 2*) positions performances improve. In particular, the drastic improvement of the Root Mean Square Error (RMSE) parameter confirms the positioning performance enhancement. The *Best Node* phone represents an exception to the previous analysis, undeniably no improvement has been made on the global estimation of this user's position on its already (assumed) optimal position performance.

IV. Conclusion & Future Work

In this paper, we proposed a collaborative positioning engine called SmartCoop. This cooperative algorithm has been specifically designed for smartphone-based network using Android GNSS raw data measurements. The development of this method was supported by the progress made on ranging estimation techniques. The smartphone adaptation to a double difference algorithm, estimating Inter-Phones Ranges as a 3D vector, has been successfully tested and analyzed. The conclusion drawn by our research was that the IPR estimation technique is the preferred method to be used for ranging in the context of Android smartphone positioning. Thus, IPR ranging can be adopted for implementing a collaborative system. A exhaustive description of our proposed cooperative engine has been conducted. A first analysis confirmed that our proposed solution, *SmartCoop*, allows for a significant positioning quality increase by improving smartphones positions accuracy and reliability.

Future work will be devoted to the improvement of our collaborative engine, by implementing a cooperative scenario between users in an urban environment. Preliminary results, concerning the estimation of IPR ranges in urban canyon, shows promising outcomes. Future Studies of our collaborative positioning engine in urban environment will consider the impact of the *Best Node* phone on the estimated collaborative positions but also will define the *SmartCoop* algorithm limitations due to specific cases and network's users geometry.

The generation of Android GNSS raw measurements for urban scenarios will be conducted in association with simulated data concerning flags detection events previsouly studied in [11]. Furthermore, we ambition to utilize smartphone's sensors for constraining our collaborative system. Previous literature studies demonstrated that low-grade IMU embedded smartphone units are reliable [14] [15]. Another example of smartphone sensors' measurements usage in the context of collaborative positioning could also be to use relative barometric measurements for constraining the z axis variations of network's peers for urban environment.

V. Acknowledgment

This study has been conducted as part of a PhD research project on *Precise cooperative positioning of low-cost mobiles in an urban environment*. The authors would like to acknowledge Thalès Alénia Space and CNES for sponsoring this work and ENAC Telecom Laboratory for providing access to research equipments.

References

[1] L. Massarweh, F. Darugna, D. Psychas, and J. Bruno, "Statistical Investigation of Android GNSS Data: Case Study Using Xiaomi Mi 8 Dual-Frequency Raw Measurements," *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, pp. pp. 3847–3861., Sep. 2019.

[2] T. Verheyde, A. Blais, C. Macabiau, and F.-X. Marmet, "An Assessment Methodology of Smartphones Positioning Performance for Collaborative Scenarios in Urban Environment," *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. pp. 1893–1901., Sep. 2020.

[3] N. Gogoi, A. Minetto, N. Linty, and F. Dovis, "A Controlled-Environment Quality Assessment of Android GNSS Raw Measurements," *Electronics*, vol. 8, no. 1, 2019. [Online]. Available: https://www.mdpi.com/2079-9292/8/1/5

[4] U. Robustelli, V. Baiocchi, and G. Pugliano, "Assessment of Dual Frequency GNSS Observations from a Xiaomi Mi 8 Android Smartphone and Positioning Performance Analysis," *Communications Smart City*, 2019.

[5] B. Chen, G. Chengfa, L. Yongsheng, and S. Puyu, "Real-time Precise Point Positioning with a Xiaomi MI 8 Android Smartphone," *Remote Sensors, Control, and Telemetry.*, 2019.

[6] K. Liu, H. B. Lim, E. Frazzoli, H. Ji, and V. C. S. Lee, "Improving Positioning Accuracy Using GPS Pseudorange Measurements for Cooperative Vehicular Localization," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 6, pp. 2544–2556, 2014.

[7] N. Gogoi, A. Minetto, and F. Dovis, "On the Cooperative Ranging between Android Smartphones Sharing Raw GNSS Measurements," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019)*, 2019, pp. 1–5.

[8] A. Minetto, A. Gurrieri, and F. Dovis, "A Cognitive Particle Filter for Collaborative DGNSS Positioning," *IEEE Access*, vol. 8, pp. 194 765–194 779, 2020.

[9] Google, "Guides to Raw GNSS Measurements - GNSSMeasurement," *https://developer.android.com/reference/android/location/GnssMeasurement*.

[10] GNSS.Raw.Measurements.Task.Force, "Using gnss raw measurements on android device," *White Paper - European GNSS Agency (GSA)*, 2017.

[11] T. Verheyde, A. Blais, C. Macabiau, and F. X. Marmet, "Analyzing Android GNSS Raw Measurements Flags Detection Mechanisms for Collaborative Positioning in Urban Environment," *IEEEXplore*, vol. 2020 International Conference on Localization and GNSS (ICL-GNSS), pp. pp. 1–6, doi: 10.1109/ICL–GNSS49 876.2020.9 115 564, 2020.

[12] K. Liu and H. B. Lim, "Positioning accuracy improvement via distributed location estimate in cooperative vehicular networks," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*, 2012, pp. 1549–1554.

[13] T. Verheyde, A. Blais, C. Macabiau, and F.-X. Marmet, "Staticical Analysis of Android GNSS Raw Data Measurements in an Urban Environment for Smartphone Collaborative Positioning Methods," *International Navigation Conference (INC)*, 2019.

[14] Z. Niu, P. Nie, L. Tao, and B. Sun, J. ansd Zhu, "RTK with the Assistance of an IMU-Based Pedestrian Navigation Algorithm for Smartphones," *Sensors*, pp. 19, 3228. https://doi.org/10.3390/s19 143 228, 2019.

[15] A. Sheta, A. Mohsen, B. Sheta, and M. Hassan, "Improved Localization for Android Smartphones Based on Integration of Raw GNSS Measurements and IMU Sensors," in *2018 International Conference on Computer and Applications (ICCA)*, 2018, pp. 297–302.