Precise Cooperative Positioning of Low-Cost Mobiles in an Urban Environment

Thomas VERHEYDE

2023 February, 10th

Thesis Jury:

MARAIS Juliette	*Reviewer*	*Université Gustave Eiffel*
DOVIS Fabio	*Reviewer*	*Politecnico di Torino*
MACABIAU Christophe	*Thesis director*	*École Nationale d'Aviation Civile*
BLAIS Antoine	*Thesis co-director*	*École Nationale d'Aviation Civile*
MARMET François-Xavier	*Invitee*	*Centre National d'Etudes Spatiales*
SERANT Damien	*Invitee*	*Thalès Alenia Space*

Presentation Outline

Presentation Outline

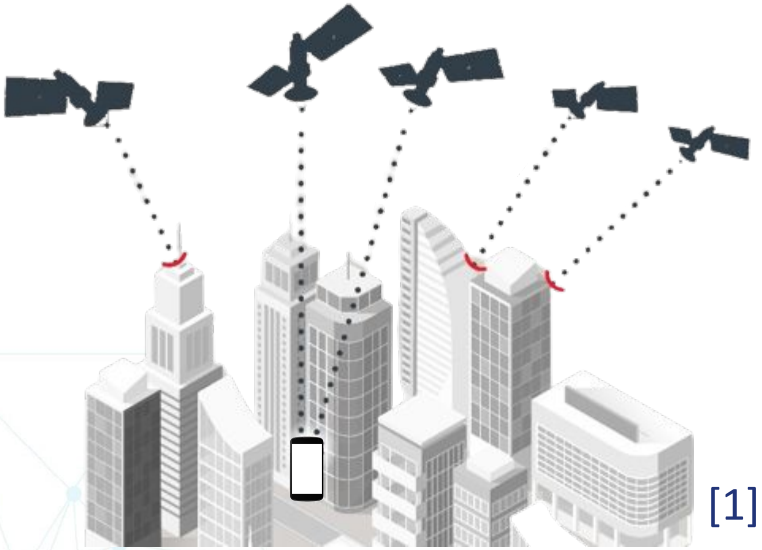
I. Introduction - Context & Motivation

Urban Positioning Constraints

&

Hyper-connectivity of urban areas

EUSPA Key Figures [2]



[1]

- Multipath
- Non-line of Sight (NLOS)
- Signal Blockage
- Reduced availability

2.5 Billions

Consumer devices are smartphones (90%)

~ ***25 apps / person***

regularly rely on user's position

---

[1] Pictures modified from: https://www.spirent.com/blogs/reliable-gnss-positioning-in-urban-areas-a-key-technical-challenge-for-drones-and-self-driving-cars

[2] EUSPA – 2022 GNSS User Technology Report https://www.euspa.europa.eu/sites/default/files/uploads/euspa_market_report_2022.pdf

I. Introduction - Context & Motivation

Chronological timeline of Smartphone Positioning Technical Improvement

2006 → **2007** → **2016** → **2017** → **2018** → **2020** - - - → **2024**

2006: Every mobile phones equipped with GPS

2007: Launch 1st Smartphone: iPhone

2016: **Android Unlocked GNSS Raw Measurements**

2017: Creation of the EUSPA Android Measurements Task Force

Samsung S8

2018: 1st Multi-Constellation & Multi-Frequency GNSS Chip

Xiaomi Mi 8

2020: Integration of *GnssAntennaInfo* class in Android 11

2024: * Implementation of Galileo OS-NMA on Smartphones

*expected

I. Introduction - Objectives

Aim at creating a collaborative network of smartphone users sharing data for mitigating urban GNSS positioning constraints.

Research Objectives

- **Mastering and Characterizing Android GNSS Raw Data Measurements**
  - Characterizing smartphone positioning capabilities
  - Evaluating Android GNSS raw data measurements
- **Developing a Smartphone-based Collaborative Network:**
  - Estimating a vectorial link between users using smartphone-based GNSS raw measurements
  - Setting up and solving a constrained nonlinear optimization problem for estimating cooperative positions.

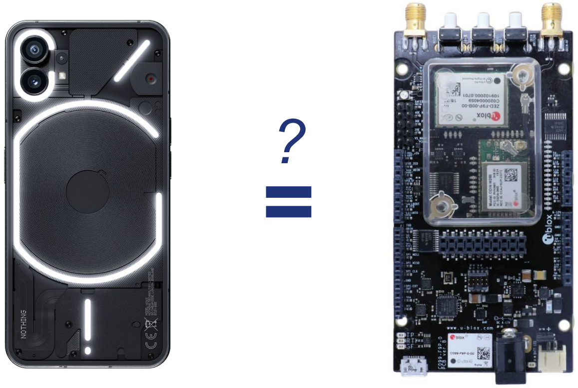
Presentation Outline

II. What kind of GNSS Receiver in a Smartphone ? - Android GNSS Raw Data Measurements

Advantages

- Smartphones are equipped with low-cost GNSS chipsets
- Chipset with dual-frequency and multi-constellation
- Availability of Code and Phase measurements
- Easy data recording through GNSS Raw data Measurement class within Android API
- Connectivity
- Exponential development of IOT applications in smartphone makes this platform the most used positioning device

→ **Accuracy matters!**



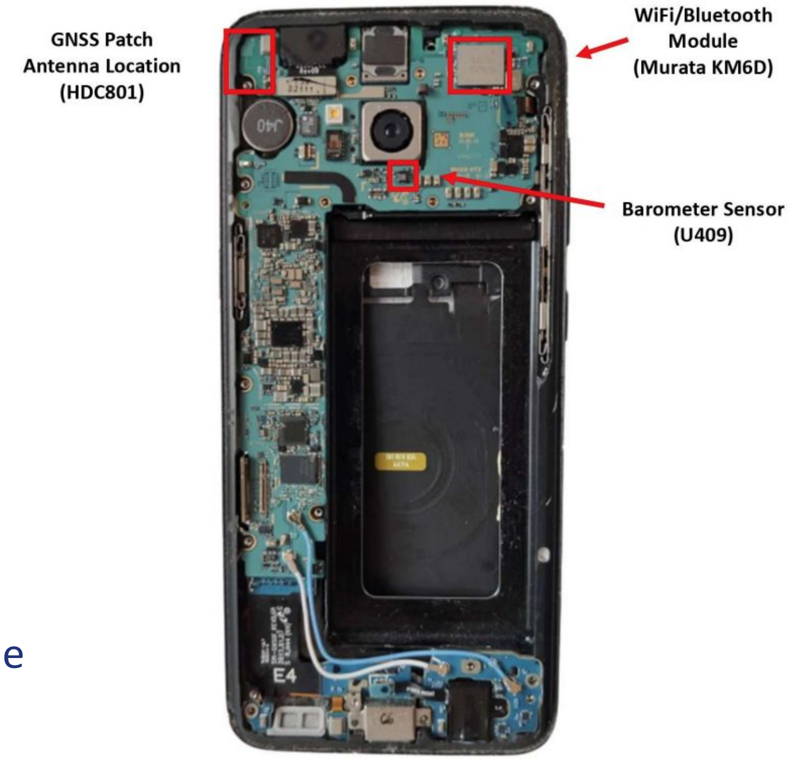
Receiver Characteristics

- Dual-frequency
- Five constellations
  (GPS+GAL+GLO+BEI+QZSS)
- Augmentation Systems

II. What kind of GNSS Receiver in a Smartphone ? - Android GNSS Raw Data Measurements

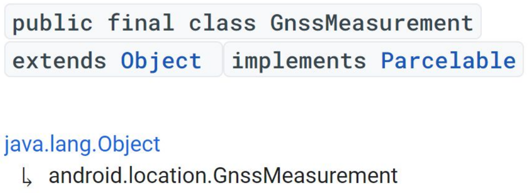
Disadvantages

- Tight integration of low-cost components unoptimized for processing GNSS signals
  - Dedicated processor
  - Linearly polarized antenna
- Duty cycle
- Unreliable phase measurements
- Frequent loss of lock and cycle slips
- Inconsistent characteristics throughout smartphone brands and models.



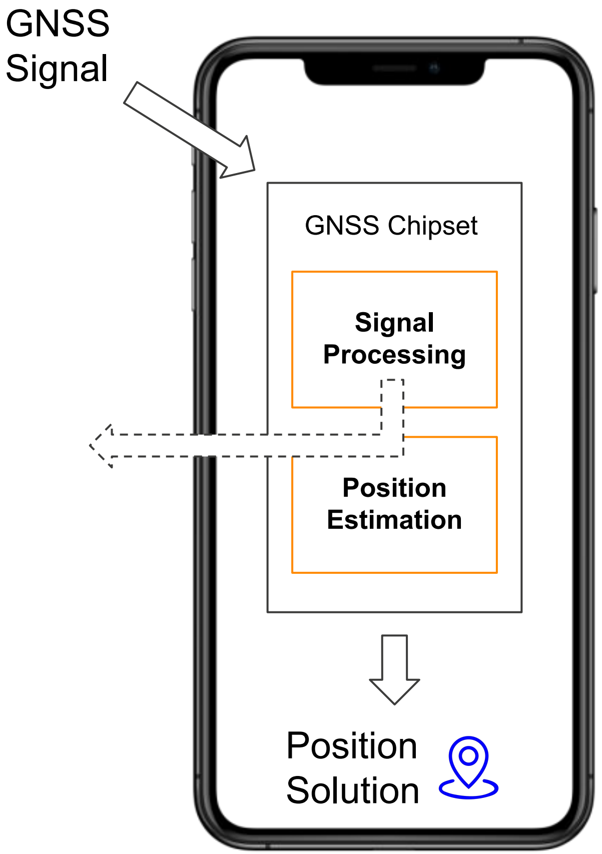
Samsung S8 Hardware Architecture

II. What kind of GNSS Receiver in a Smartphone ? - Android GNSS Raw Data Measurements



```
public final class GnssMeasurement
extends Object  implements Parcelable
```

java.lang.Object
↳ android.location.GnssMeasurement



Android GNSS Raw data Measurements [3]

- Code
- Phase
- Doppler
- C/N0 Signal Power
- Flags

Recorded for Post-processing

[3] Android API Documentation - 2022
https://developer.android.com/reference/android/location/GnssMeasurement

II. What kind of GNSS Receiver in a Smartphone ? - Android GNSS Raw Data Measurements

Logging Android GNSS raw data measurements via Smartphone Applications (*2 main options*):

1. Geo++® RINEX Logger

2. GnssLogger

(Extra) SMART LOGGER GNSS

(+)

- GNSS measurements recorded in RINEX format
- Widely used

(-)

- Loss of information and data during RINEX format conversion

II. What kind of GNSS Receiver in a Smartphone ? - Android GNSS Raw Data Measurements

Logging Android GNSS raw data measurements via Smartphone Applications (*2 main options*):

1. Geo++® RINEX Logger

2. GnssLogger

(Extra) SMART LOGGER GNSS

(+)

- Android native logging application
- Basic functionalities
- Record measurements in a .log format

(-)

- Rare updates

→ **Application used!**

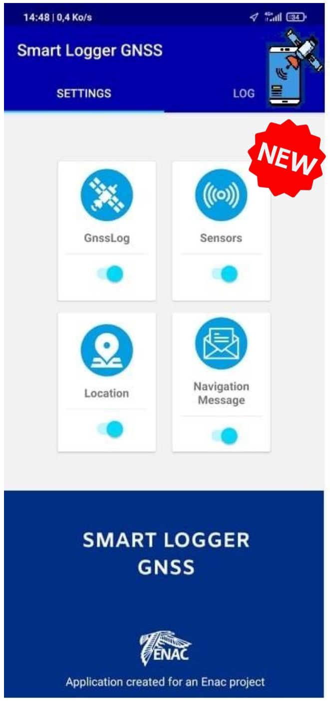
II. What kind of GNSS Receiver in a Smartphone ? - Android GNSS Raw Data Measurements

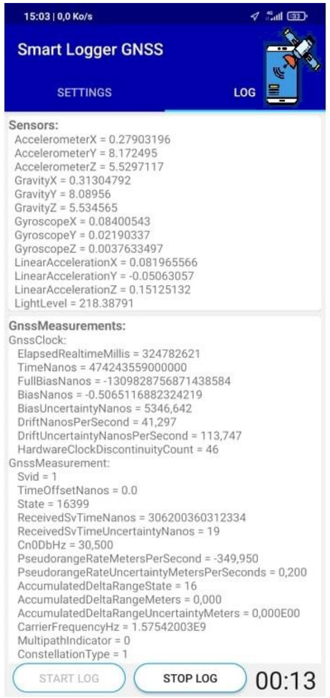
Logging Android GNSS raw data measurements via Smartphone Applications (*2 main options*):

1. GnssLogger

2. Geo++® RINEX Logger

(Extra) SMART LOGGER GNSS

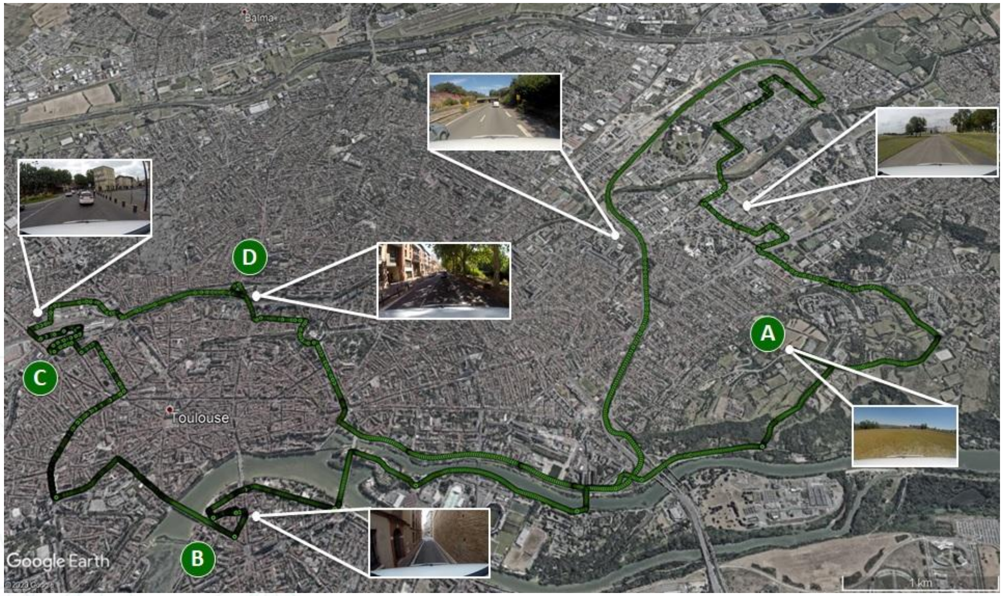




```
Sensors:
  AccelerometerX = 0.27903196
  AccelerometerY = 8.172495
  AccelerometerZ = 5.5297117
  GravityX = 0.31304792
  GravityY = 8.08956
  GravityZ = 5.534565
  GyroscopeX = 0.08400543
  GyroscopeY = 0.02190337
  GyroscopeZ = 0.0037633497
  LinearAccelerationX = 0.081965566
  LinearAccelerationY = -0.05063057
  LinearAccelerationZ = 0.15125132
  LightLevel = 218.38791
GnssMeasurements:
GnssClock:
   ElapsedRealtimeMillis = 324782621
   TimeNanos = 474243559000000
   FullBiasNanos = -1309828756871438584
   BiasNanos = -0.5065116882324219
   BiasUncertaintyNanos = 5346,642
   DriftNanosPerSecond = 41,297
   DriftUncertaintyNanosPerSecond = 113,747
   HardwareClockDiscontinuityCount = 46
GnssMeasurement:
   Svid = 1
   TimeOffsetNanos = 0.0
   State = 16399
   ReceivedSvTimeNanos = 306200360312334
   ReceivedSvTimeUncertaintyNanos = 19
   Cn0DbHz = 30,500
   PseudorangeRateMetersPerSecond = -349,950
   PseudorangeRateUncertaintyMetersPerSeconds = 0,200
   AccumulatedDeltaRangeState = 16
   AccumulatedDeltaRangeMeters = 0,000
   AccumulatedDeltaRangeUncertaintyMeters = 0,000E00
   CarrierFrequencyHz = 1.57542003E9
   MultipathIndicator = 0
   ConstellationType = 1
```

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Data Collection Campaign Trajectory in Toulouse, FRANCE ***August 6th, 2019***



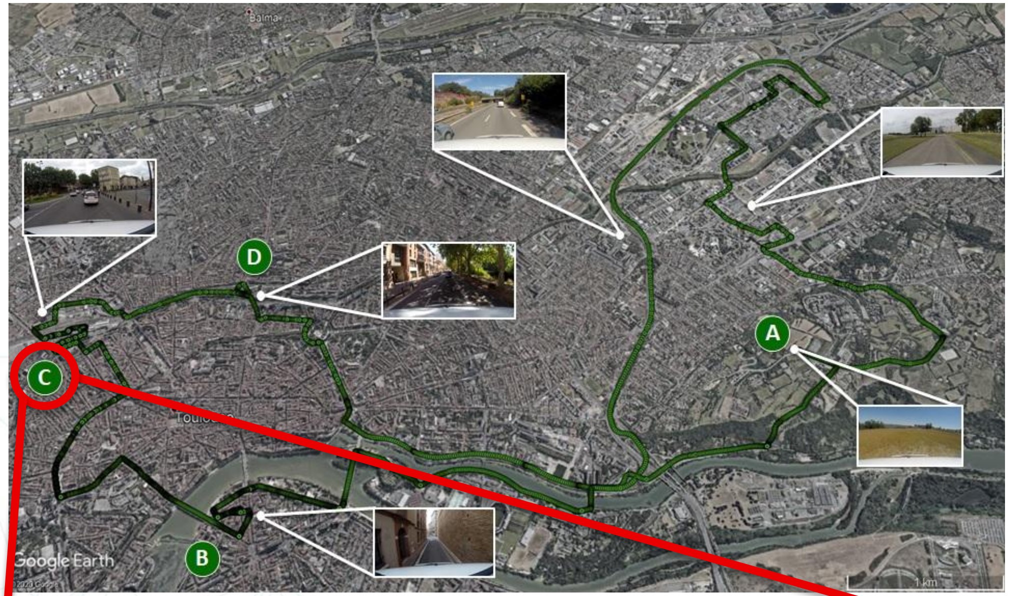
Smartphone rooftop configuration

Data Collection Campaign Methodology:

- 1 Trajectory (downtown + urban + open-sky)
- Static and dynamic scenarios
- 4 Collaborative Scenarios

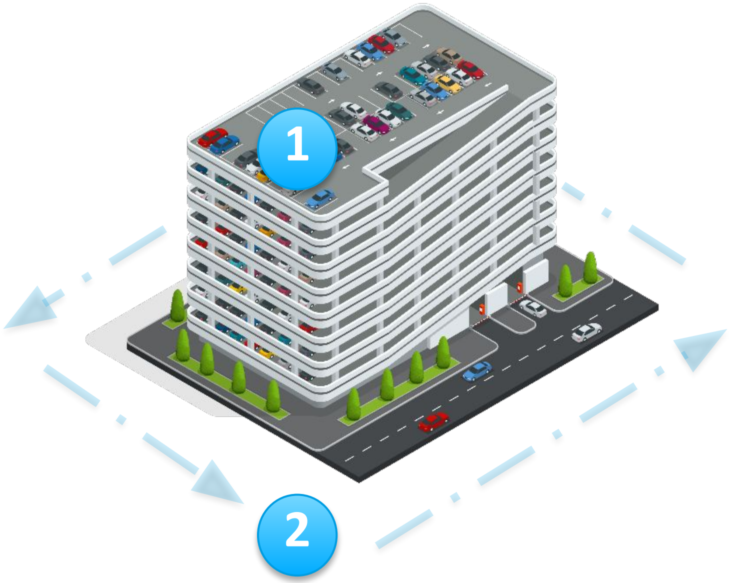
II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Data Collection Campaign Trajectory in Toulouse, FRANCE ***August 6th, 2019***



*Collaborative Scenario **C** : Open-sky Environment in an Urban City Center*

Collaborative Scenario **C**



- **Helping a user in degraded conditions while in an open-sky environment.**
- **Related to real-life scenario**
- **GNSS reception conditions differs from vehicle 1 and 2.**

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

11/2019

Google Pixel 3

Qualcomm
Snapdragon 845

Multi-Constellation
Single Frequency

x2

01/2019

Honor View 20

HiSilicon
Kirin 980

Multi-Constellation
Multi-Frequency

04/2019

Xiaomi Mi 9

Qualcomm
Snapdragon 855

Multi-Constellation
Multi-Frequency

x2

06/2018

Xiaomi Mi 8

Qualcomm
Snapdragon 845

Broadcom
BCM47755

Multi-Constellation
Multi-Frequency

11/2018

Huawei Mate 20X

HiSilicon
Kirin 980

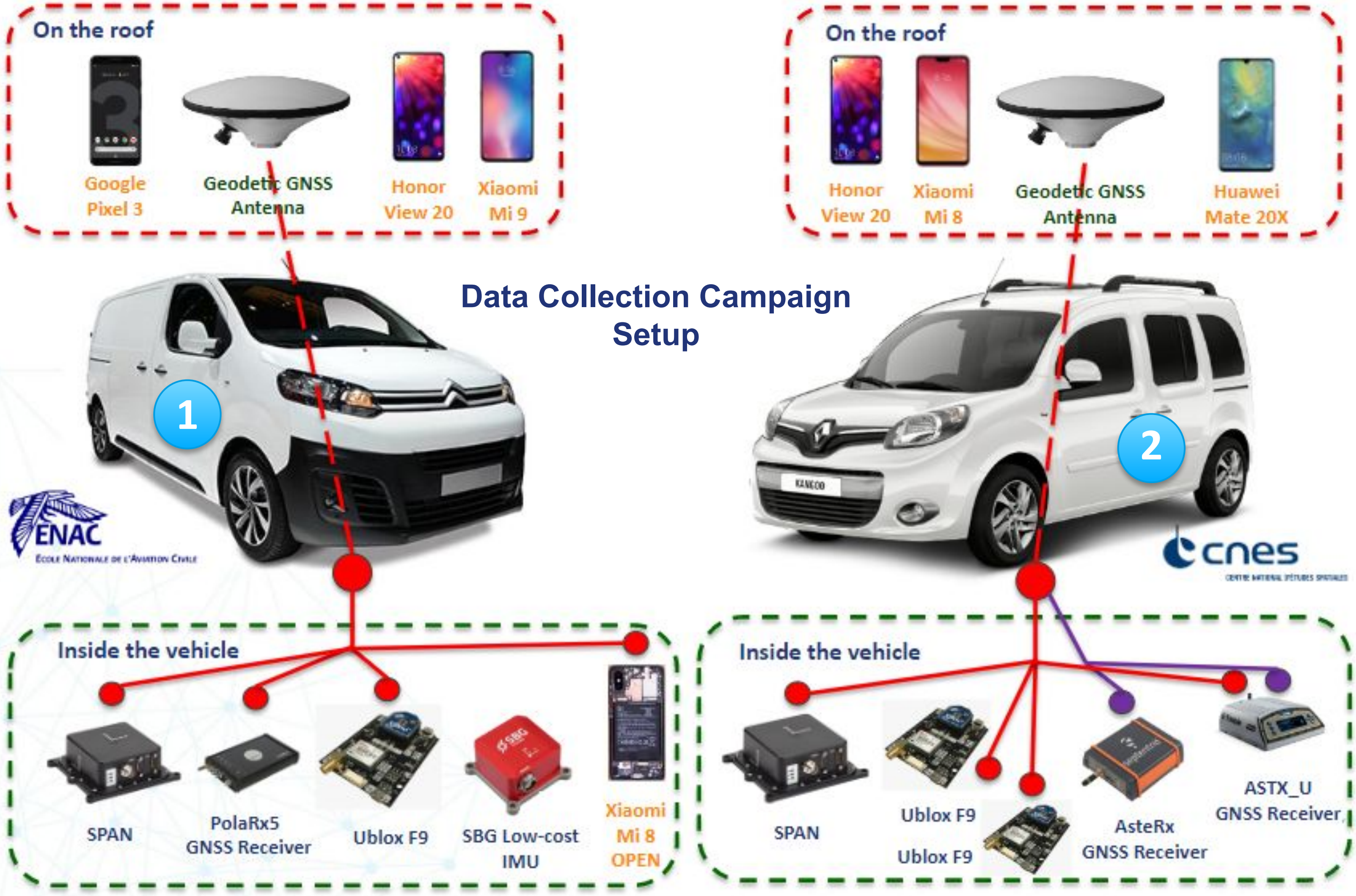
Multi-Constellation
Multi-Frequency

Telecommunications for Space and Aeronautics

CENTRE NATIONAL D'ÉTUDES SPATIALES

a Thales / Leonardo company

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements



Data Collection Campaign Setup

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Static/Nominal Conditions

Static & Open-Sky Conditions

Scenario A

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Static/Nominal Conditions

Receiver Clock Drift Estimation

Pseudorange rate model: $\dot{\rho}_i^{sv} = (v_{x_i} - v_{ux}) \cdot a_{x_i} + (v_{y_i} - v_{uy}) \cdot a_{y_i} + (v_{z_i} - v_{uz}) \cdot a_{z_i} + c\dot{\delta t} + \varepsilon_{\dot{\rho}_i^s}$

```
public double getPseudorangeRateMetersPerSecond ()
```

```
pseudorange rate = -k * doppler shift   (where k is a constant)
```

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Static/Nominal Conditions

Receiver Clock Drift Estimation

Pseudorange rate model: $\dot{\rho}_i^{sv} = (v_{x_i} - v_{ux}) \cdot a_{x_i} + (v_{y_i} - v_{uy}) \cdot a_{y_i} + (v_{z_i} - v_{uz}) \cdot a_{z_i} + c\dot{\delta t} + \varepsilon_{\dot{\rho}_i^s}$

`public double getPseudorangeRateMetersPerSecond ()`

Pseudorange rate residual: $d_i^{sv} = \dot{\rho}_i^{sv} - \boxed{\boldsymbol{v}_i^{sv} \, . \, \boldsymbol{a}_i^{sv}} \longrightarrow$ *Predicted Radial Satellite Movement*

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Static/Nominal Conditions

Receiver Clock Drift Estimation

Pseudorange rate model: $\dot{\rho}_i^{sv} = (v_{x_i} - v_{ux}) \cdot a_{x_i} + (v_{y_i} - v_{uy}) \cdot a_{y_i} + (v_{z_i} - v_{uz}) \cdot a_{z_i} + c\dot{\delta t} + \varepsilon_{\dot{\rho}_i^s}$

`public double getPseudorangeRateMetersPerSecond ()`

Pseudorange rate residual: $d_i^{sv} = \dot{\rho}_i^{sv} - \boxed{\boldsymbol{v}_i^{sv} \, . \, \boldsymbol{a}_i^{sv}} \longrightarrow$ *Predicted Radial Satellite Movement*

$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_i \end{pmatrix} = \begin{pmatrix} -a_{x_1} & -a_{y_1} & -a_{z_1} & 1 \\ -a_{x_2} & -a_{y_2} & -a_{z_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -a_{x_i} & -a_{y_i} & -a_{z_i} & 1 \end{pmatrix} \cdot \begin{pmatrix} v_{ux} \\ v_{uy} \\ v_{uz} \\ c\dot{\delta t} \end{pmatrix} + \begin{pmatrix} \varepsilon_{\dot{\rho}_1} \\ \varepsilon_{\dot{\rho}_2} \\ \vdots \\ \varepsilon_{\dot{\rho}_i} \end{pmatrix}$$

Solving this equation with a WLSE, we obtain $c\dot{\delta t}$.

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Static/Nominal Conditions

Receiver Clock Drift Estimation

Pseudorange rate model: $\dot{\rho}_i^{sv} = (v_{x_i} - \overset{\text{Static}}{\cancel{v_{ux}}}) \cdot a_{x_i} + (v_{y_i} - \overset{\text{Static}}{\cancel{v_{uy}}}) \cdot a_{y_i} + (v_{z_i} - \overset{\text{Static}}{\cancel{v_{uz}}}) \cdot a_{z_i} + c\dot{\delta t} + \varepsilon_{\dot{\rho}_i^s}$

```
public double getPseudorangeRateMetersPerSecond ()
```

Pseudorange rate residual: $d_i^{sv} = \dot{\rho}_i^{sv} - \boxed{v_i^{sv} \;.\; a_i^{sv}}$ → *Predicted Radial Satellite Movement*

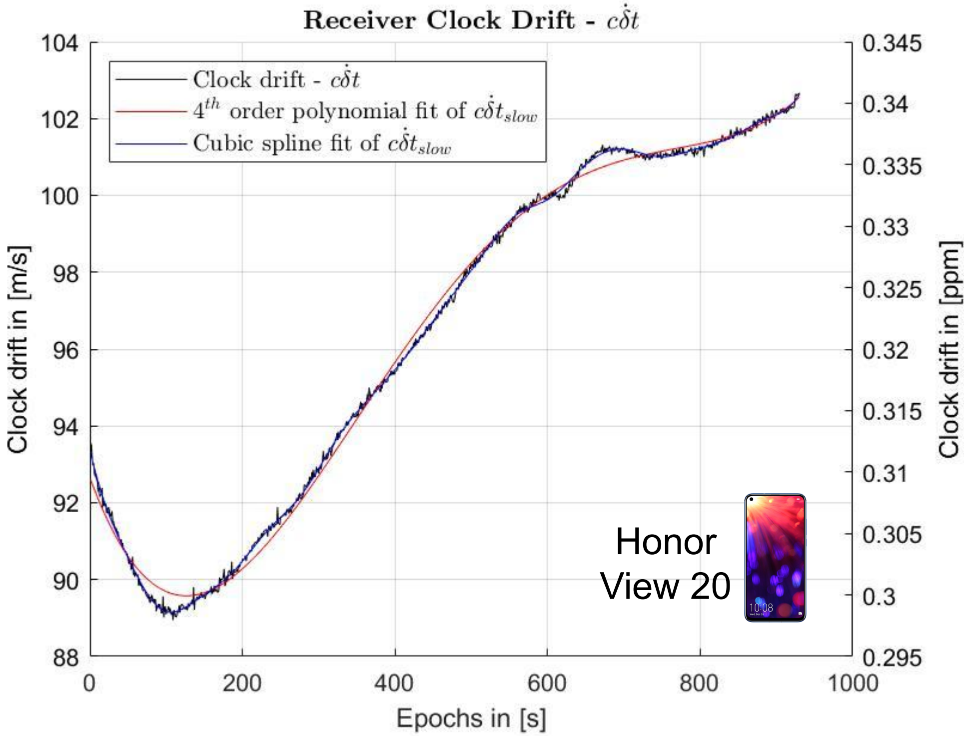
$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_i \end{pmatrix} = \begin{pmatrix} -a_{x_1} & -a_{y_1} & -a_{z_1} & 1 \\ -a_{x_2} & -a_{y_2} & -a_{z_2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ -a_{x_i} & -a_{y_i} & -a_{z_i} & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ c\dot{\delta t} \end{pmatrix} + \begin{pmatrix} \varepsilon_{\dot{\rho}_1} \\ \varepsilon_{\dot{\rho}_2} \\ \vdots \\ \varepsilon_{\dot{\rho}_i} \end{pmatrix}$$

Static Scenario

Solving this equation with a WLSE, we obtain $c\dot{\delta t}$.

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

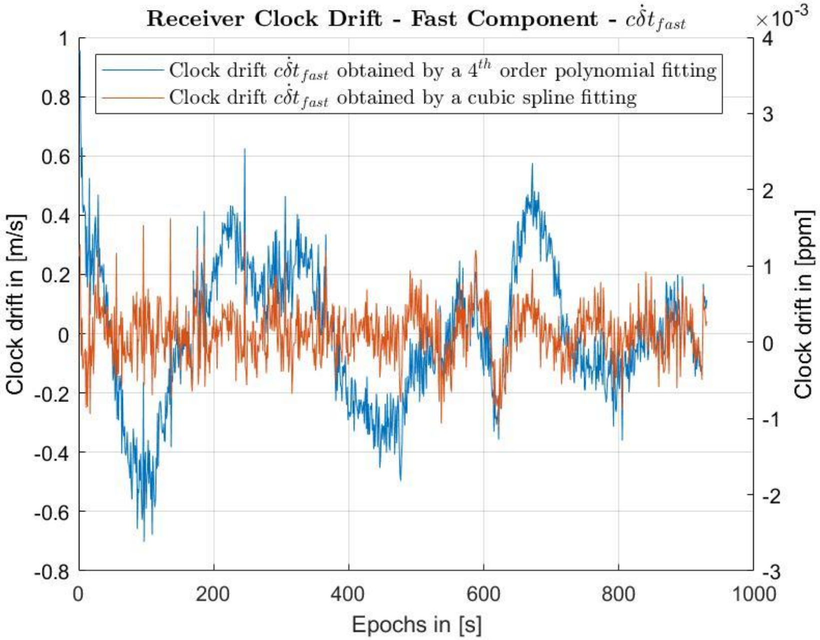
Measurements Evaluation: Nominal Conditions



Receiver Clock Drift Estimation

$c\dot{\delta t}$ modeling : $c\dot{\delta t} = c\dot{\delta t}^{fast} + c\dot{\delta t}^{slow}$

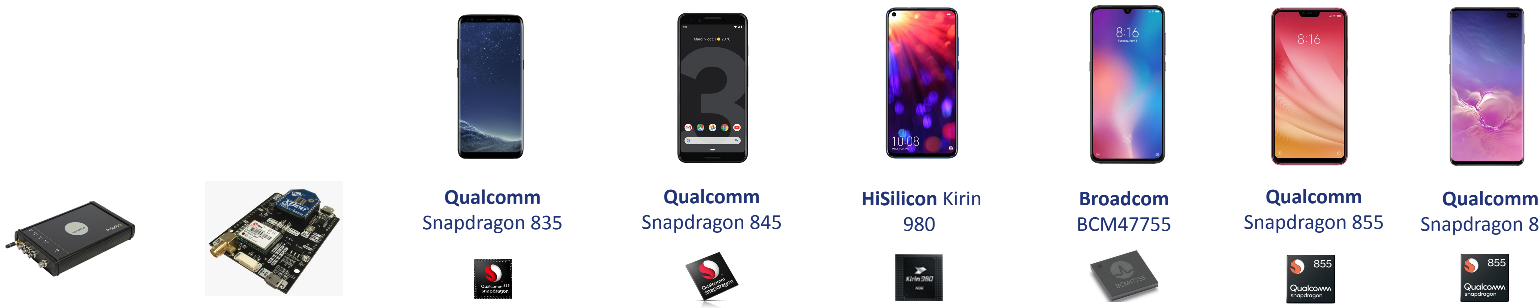
$c\dot{\delta t}^{slow}$ *is modeled by a curve fitting*



II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Nominal Conditions

Receiver Clock Drift Estimation



Receiver	*Septentrio PolaRX5S**	*u-blox M8T**	*Samsung S8**	Google Pixel 3	Honor View 20	Xiaomi Mi 8	Xiaomi Mi 9	Samsung S10+
$\sigma_{c\dot{\delta t}_{fast}}$ $[m/s]$	< 0.03	0.14	0.18	4.72	0.61	0.16	3.93	0.44
					0.42	0.25	2nd Phone	
						0.26	3rd Phone	

** Results obtained by Lehtola. et al [4]*

→ **Similar clock performance compare to COTS low-cost GNSS receiver**

[4] Lehtola VV, Söderholm S, Koivisto M, Montloin L. "*Exploring GNSS Crowdsourcing Feasibility: Combinations of Measurements for Modeling Smartphone and Higher End GNSS Receiver Performance*". Sensors. 2019; 19(13):3018.

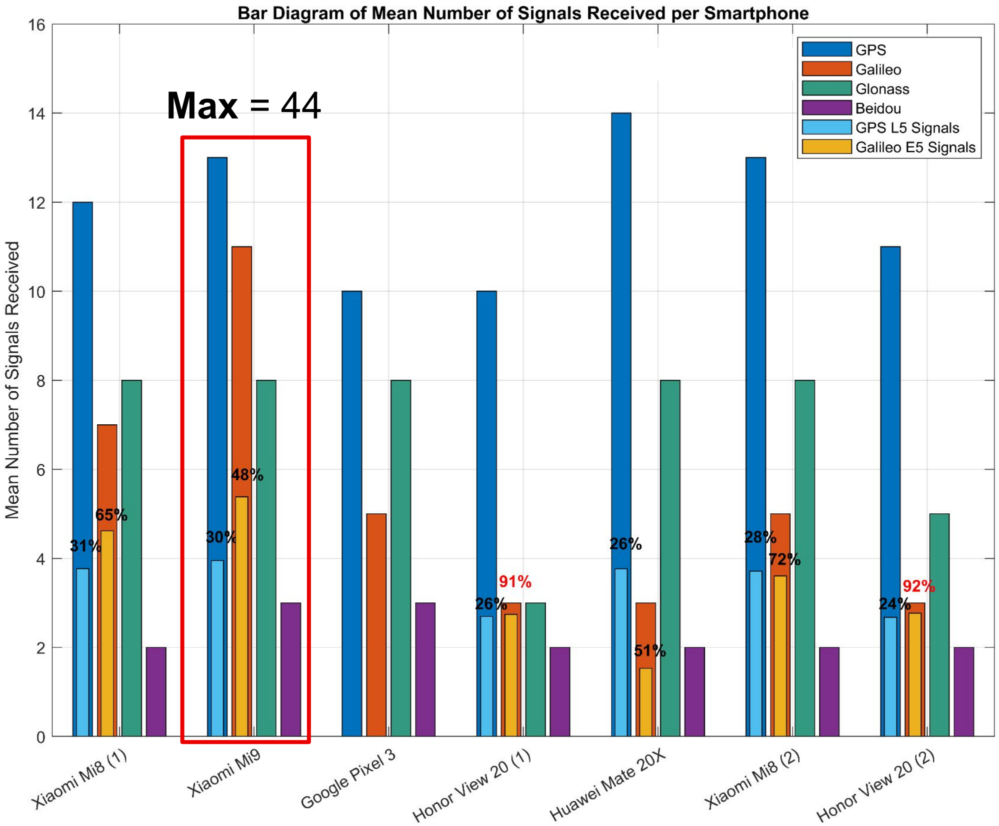
II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Urban Conditions

Urban Conditions

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Urban Conditions



Signal Availability

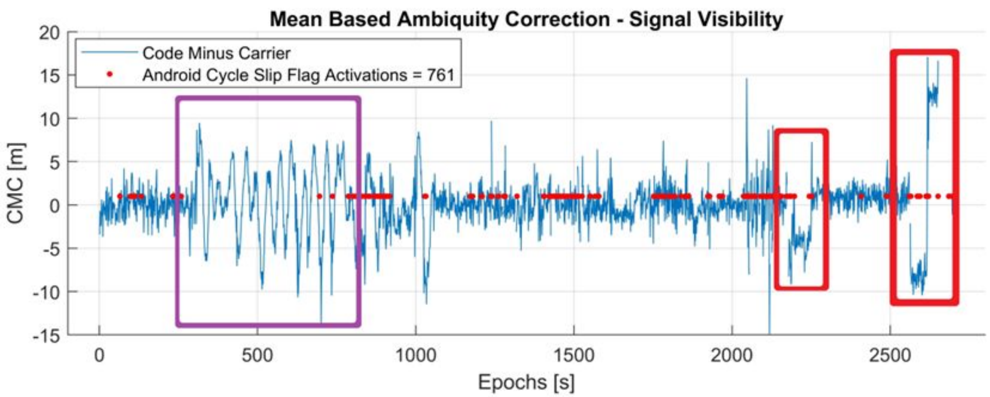
- Average number of signals tracked by smartphones = 36 (compared to 38 for Ublox F9P)
- Disparity between smartphones
- Weak tracking of E5a signals

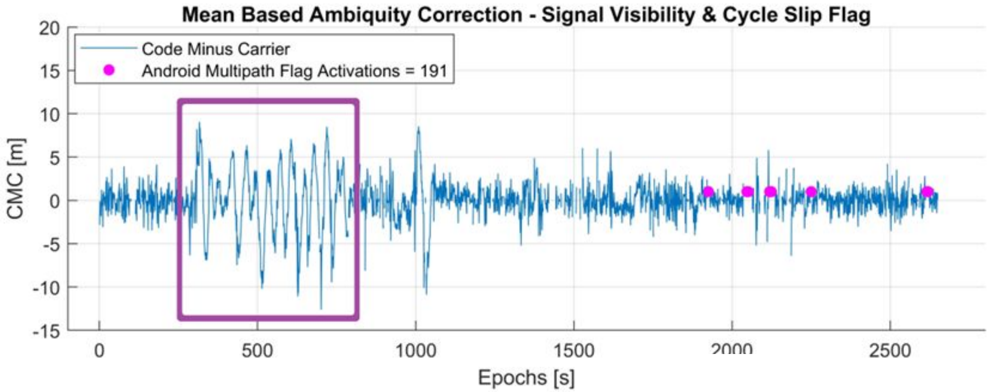
→ **High signal availability in urban conditions**

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Urban Conditions

Code Minus Carrier Analysis - GALILEO 12 E1 - Xiaomi Mi 8 (2)





Signal Analysis & Flag Detection Mechanisms

Cycle Slip Flags

- Available only if phase measurements are present
- Over-activation (761 over 2800 epochs)

Multipath Flags

- Unreliable in urban environments

→ **Android Flags are NOT suitable for collaborative purposes.**

II. What kind of GNSS Receiver in a Smartphone ? - Other Studies Summary

Throughout this research, multiple analyses have been performed in both open-sky and urban conditions:

- Signal Tracking
    - High-sensitivity chipset (signal tracking between 10 and 15 dBHz) coupled with great signal availability in dense urban areas.
- Measurements Error Statistical Analysis
    - Characterization of PLL and DLL error shows similar characteristics as for low-cost receivers.
- Standalone Positioning Performance (benchmark)
    - In nominal conditions (static and open-sky), smartphones show a horizontal positioning error of 2.5m (68%)
    - However, in urban conditions (low-dynamic) the horizontal error reaches 8.7m (68%). Degradation by a factor of 3 compared to dedicated low-cost GNSS receiver

II. What kind of GNSS Receiver in a Smartphone ? - Conclusions

- Multiple analyses allow for the assessment and characterization Android GNSS raw data measurements enabling the mathematical description for multiple smartphones

- Our urban study allowed us to experiment smartphone positioning in constraint environment.
    - Poor standalone horizontal positioning performance (below GNSS low-cost receiver standards)
    - Unreliable phase measurements and Android flag mechanisms
    - High signal availability in constrained environments

Smartphones can be exploited as GNSS receivers through the use of Android GNSS raw data measurements

Presentation Outline

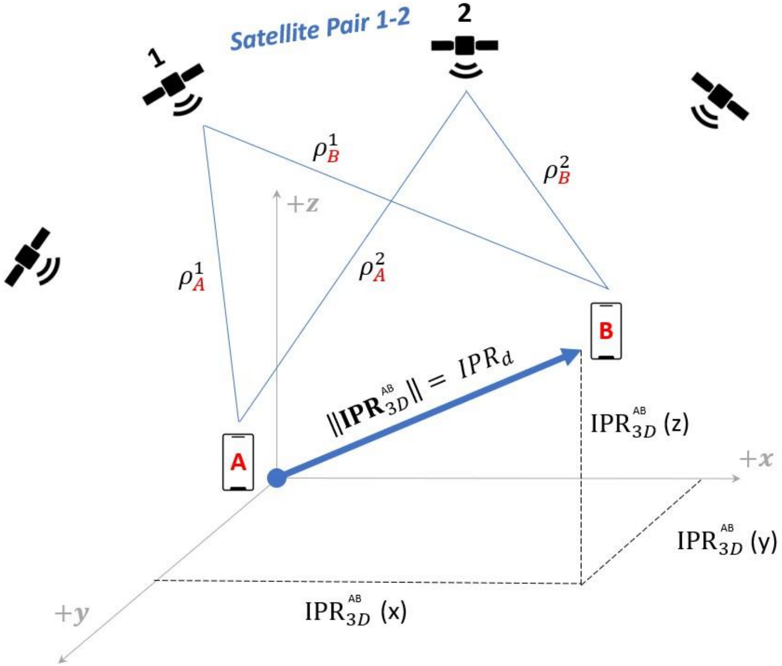
III. *Inter-Phone Ranging* (IPR)

Ranging estimation techniques called
Inter-**P**hone **R**anging (IPR)

- Creation of a 3D vector linking network users
- Estimated by a weighted least square of double code difference (WLS-DD)
- Algorithm adapted from [5] & [6] in order to fit smartphone specificities

→ **Removes common errors**

→ **Creates auxiliary information between two users**



[5] 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.

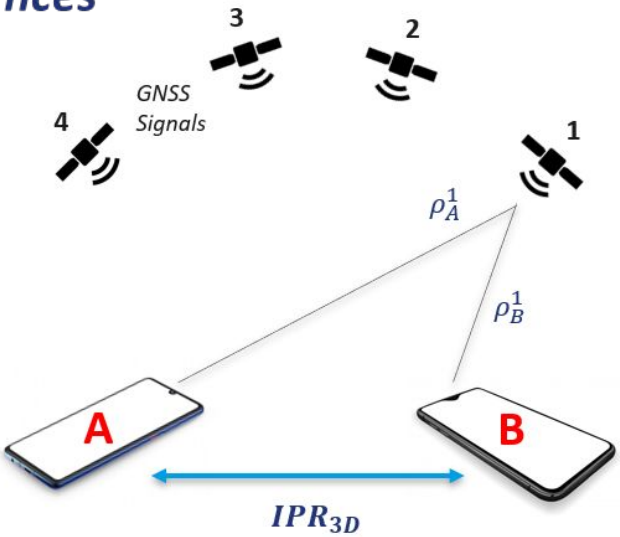
[6] N. Gogoi, A. Minetto, and F. Dovis, “On the cooperative ranging between android smartphones sharing raw GNSS measurements,” in 2019 IEEE 90th Vehicular Technology Conference (VTC 2019), 2019, pp. 1–5

III. *Inter-Phone Ranging* (IPR) - Methodology

Inter-Phone Ranging (IPR): Pseudoranges Double Differences

Methodology

- Formating data
- Establishing a list of common received signals
- Hatch filter
- Doppler synchronization of measurements
- **Double code difference**
- WLS Estimation

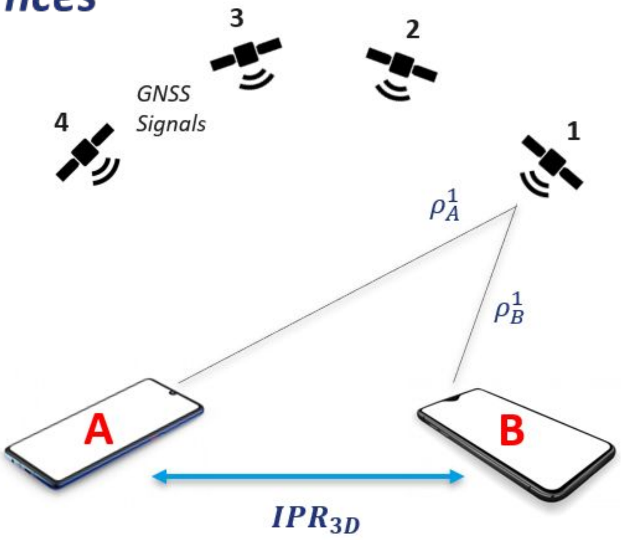


Code model (Phone A) : $\rho_A^1 = r + c\,(t_{rx}^A - t_{tx}) + \varepsilon_{Iono} + \varepsilon_{Tropo} + \varepsilon_{code}^1 + \varepsilon_{Multipath}^{\rho_A}$

Code model (Phone B) : $\rho_B^1 = r + c\,(t_{rx}^B - t_{tx}) + \varepsilon_{Iono} + \varepsilon_{Tropo} + \varepsilon_{code}^1 + \varepsilon_{Multipath}^{\rho_B}$

III. *Inter-Phone Ranging* (IPR) - Methodology

Inter-Phone Ranging (IPR): Pseudoranges Double Differences



Single Difference:

- Removes common propagation errors
- Mitigates satellite clock errors

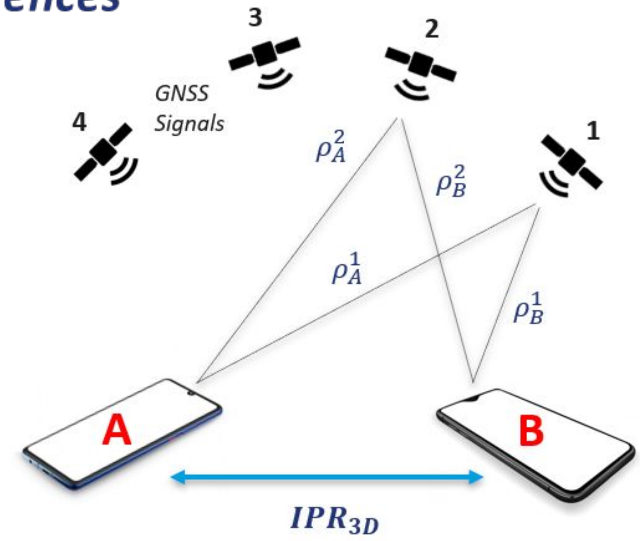
***Code** model (Phone A) :* $\rho_A^1 = r + c\,(t_{rx}^A - \cancel{t_{tx}}) + \cancel{\varepsilon_{Iono}} + \cancel{\varepsilon_{Tropo}} + \varepsilon_{code}^1 + \varepsilon_{Multipath}^{\rho_A}$

***Code** model (Phone B) :* $\rho_B^1 = r + c\,(t_{rx}^B - \cancel{t_{tx}}) + \cancel{\varepsilon_{Iono}} + \cancel{\varepsilon_{Tropo}} + \varepsilon_{code}^1 + \varepsilon_{Multipath}^{\rho_B}$ $-$

***Single Difference** (Sat 1):* $D_{AB}^1 = \rho_B^1 - \rho_A^1 = \Delta r_{AB}^1 + c\,.\Delta t_{rx} + \Delta\varepsilon_{code}{}_{AB}^{1} + \Delta\varepsilon_{Multipath_1}^{\rho_{AB}}$ $=$

III. *Inter-Phone Ranging* (IPR) - Methodology

Inter-Phone Ranging (IPR): Pseudoranges Double Differences



Double Difference:

- Removes receivers clock biases

***Single Difference** (Sat **1**):* $$D_{AB}^1 = \rho_B^1 - \rho_A^1 = \Delta r_{AB}^1 + c\,.\,\cancel{\Delta t_{rx}^{AB}} + \Delta\varepsilon_{code}{}_{AB}^{1} + \Delta\varepsilon_{Multipath_1}^{\rho_{AB}}$$

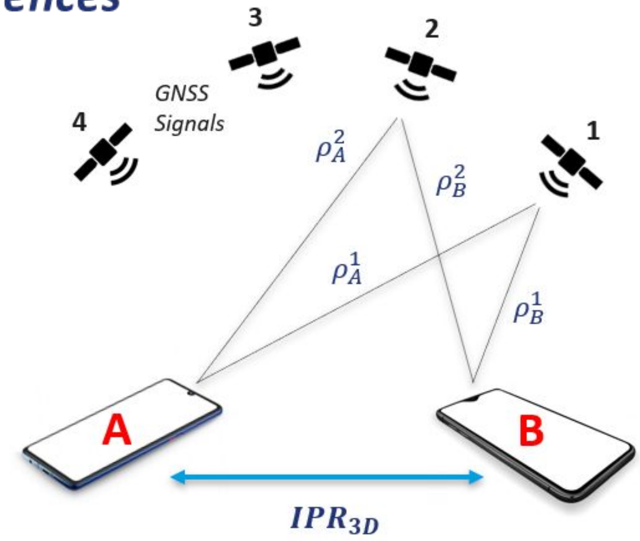
$-$

***Single Difference** (Sat **2**):* $$D_{AB}^2 = \rho_B^2 - \rho_A^2 = \Delta r_{AB}^2 + c\,.\,\cancel{\Delta t_{rx}^{AB}} + \Delta\varepsilon_{code}{}_{AB}^{2} + \Delta\varepsilon_{Multipath_2}^{\rho_{AB}}$$

Double Difference: $$dD_{AB}^{12} = D_{AB}^2 - D_{AB}^1 = [\boldsymbol{IPR_{3D}}.(\vec{e}^{\,2} - \vec{e}^{\,1})] + \Delta\varepsilon_{AB}^{12} \quad =$$

III. *Inter-Phone Ranging* (IPR) - Methodology

Inter-Phone Ranging (IPR): Pseudoranges Double Differences



→ Solves for IPR_{3D} with a WLS

$$IPR_{3D} = (H^T W H)^{-1} H^T W \, dD_{AB}$$

Single Difference *(Sat 1):* $D_{AB}^1 = \rho_B^1 - \rho_A^1 = \Delta r_{AB}^1 + \cancel{c\,.\,\Delta t_{rx}^{AB}} + \Delta\varepsilon_{code}{}_{AB}^{1} + \Delta\varepsilon_{Multipath_1}^{\rho_{AB}}$

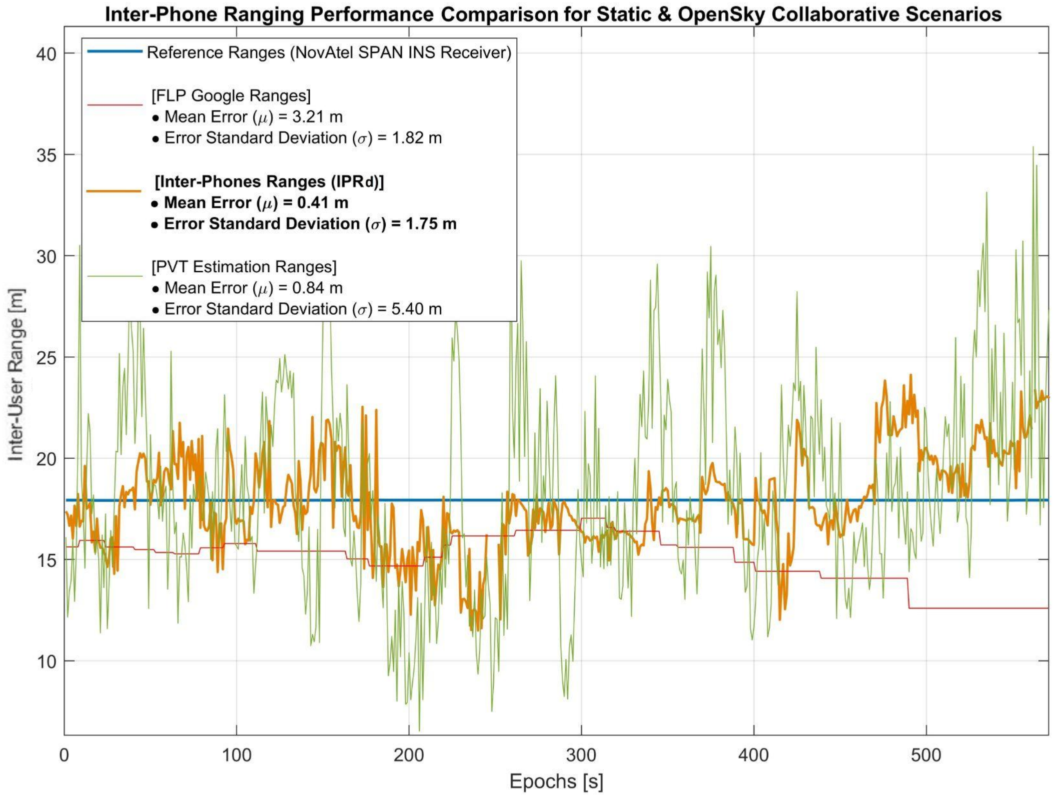
−

Single Difference *(Sat 2):* $D_{AB}^2 = \rho_B^2 - \rho_A^2 = \Delta r_{AB}^2 + \cancel{c\,.\,\Delta t_{rx}^{AB}} + \Delta\varepsilon_{code}{}_{AB}^{2} + \Delta\varepsilon_{Multipath_2}^{\rho_{AB}}$

Double Difference: $dD_{AB}^{12} = D_{AB}^2 - D_{AB}^1 = [IPR_{3D}.(\vec{e}^{\,2} - \vec{e}^{\,1})] + \Delta\varepsilon_{AB}^{12}$ =

III. *Inter-Phone Ranging* (IPR) - Results Analysis

Result Analysis: Static & Open-Sky



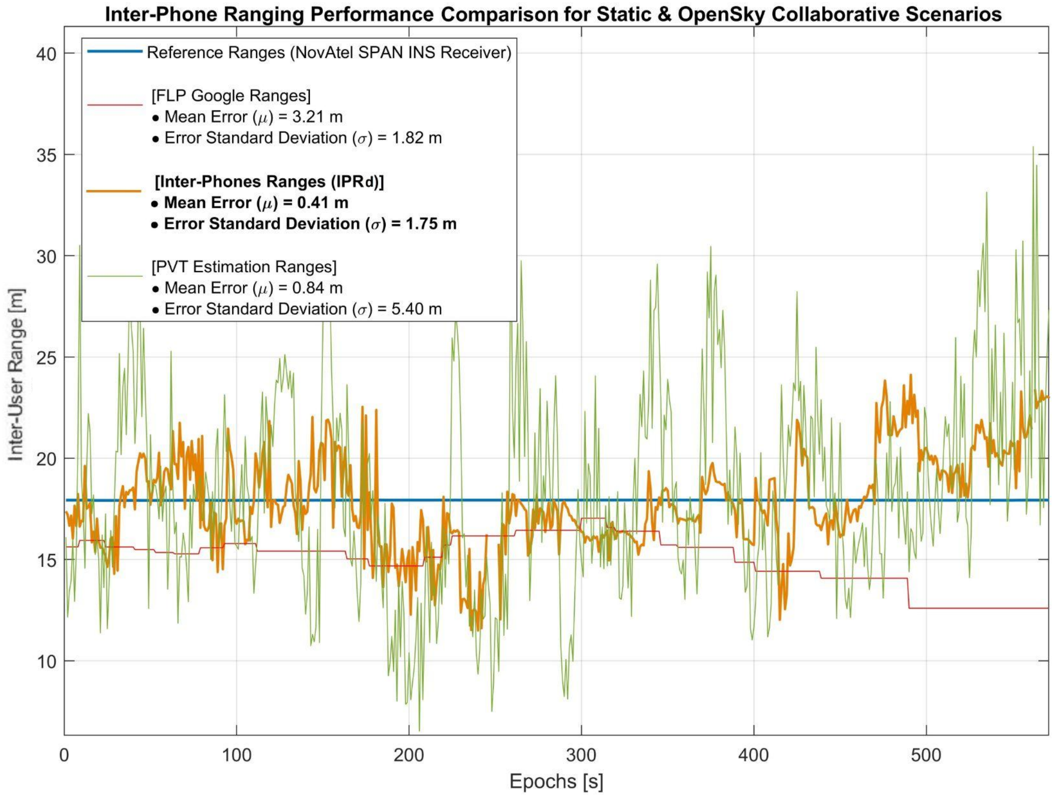
Comparison Analysis

- 🟥 **FLP** (Fused Location Provider)
- 🟩 **PVT** (Standalone)
- 🟦 **Reference** (NovAtel SPAN INS)
- 🟧 IPRd (Inter-Phone Ranging Distance)

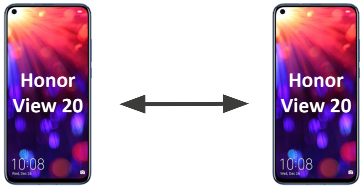
$$\|IPR_{3D}\| = IPRd$$

III. *Inter-Phone Ranging* (IPR) - Results Analysis

Result Analysis: Static & Open-Sky



2 static smartphones separated by 17m

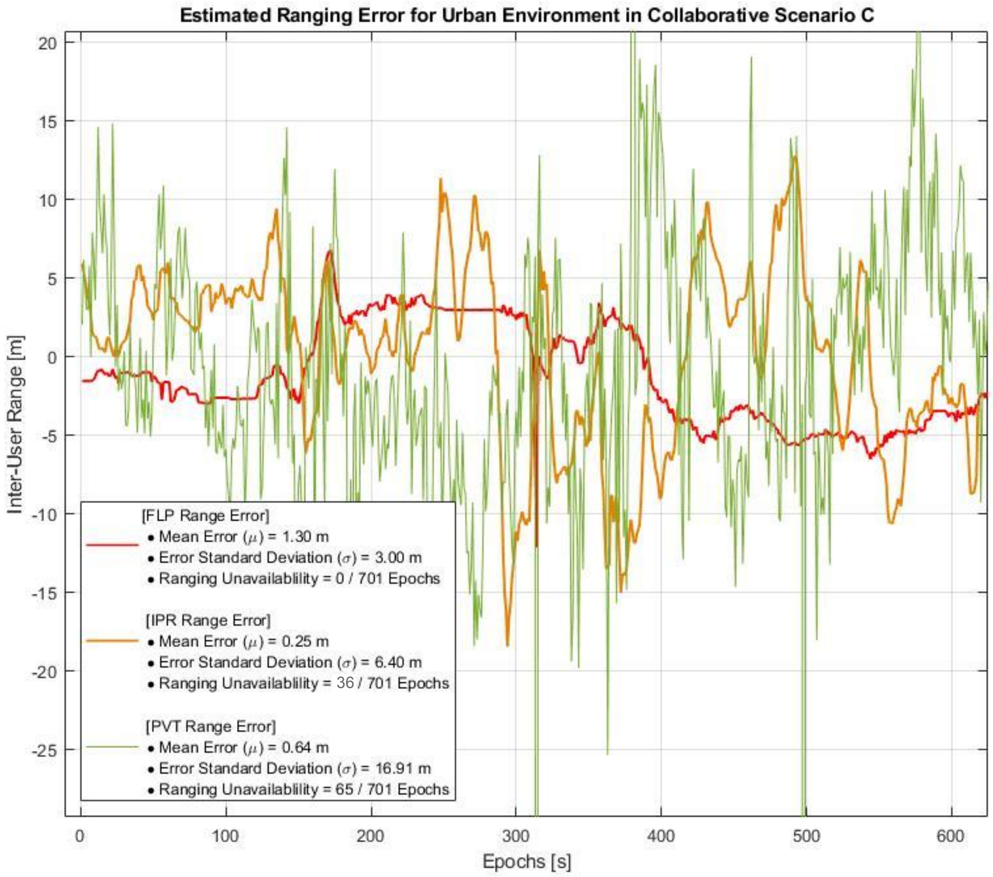


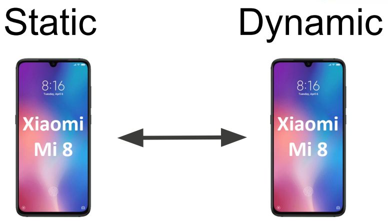
Baseline Error	IPRd	FLP	PVT
Mean [m]	0.41	3.21	0.84
Standard deviation [m]	1.75	1.82	5.40

- Accurate IPRd estimates
- Coherent estimates across smartphones brands and models
- Bias observed in the FLP estimation

III. *Inter-Phone Ranging* (IPR) - Results Analysis

Result Analysis: Dynamic & Urban - Collaborative Scenario C



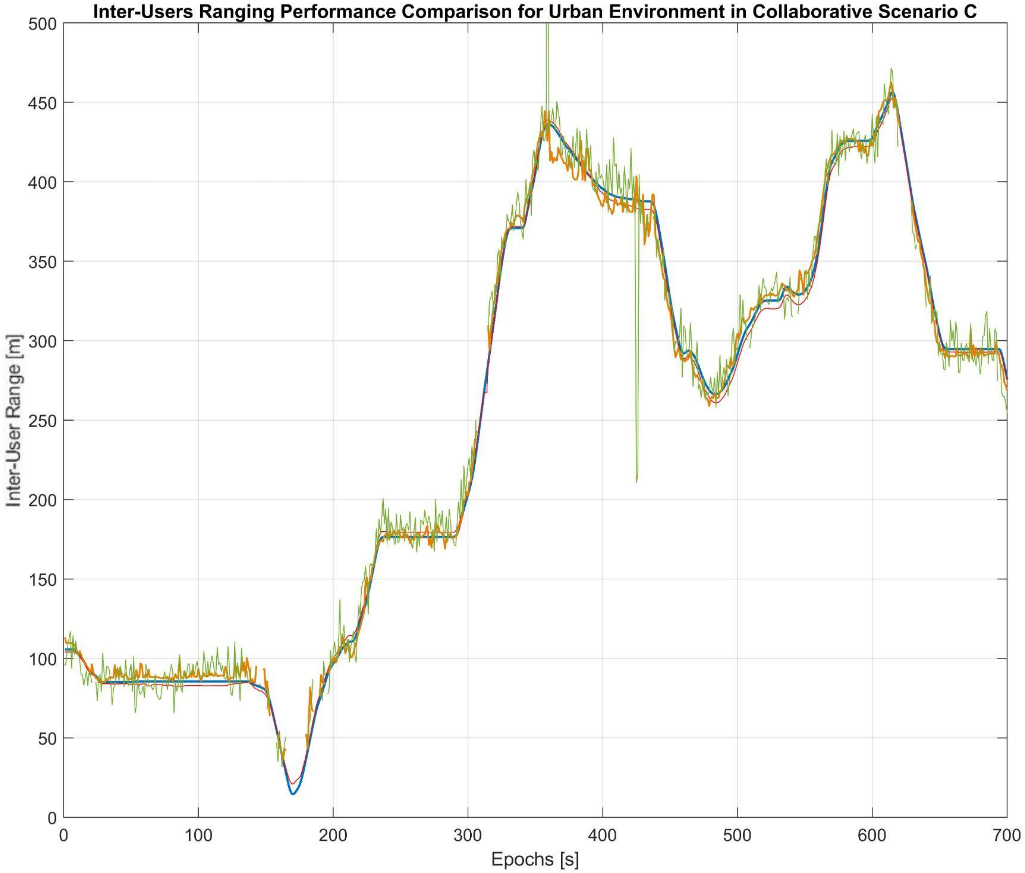


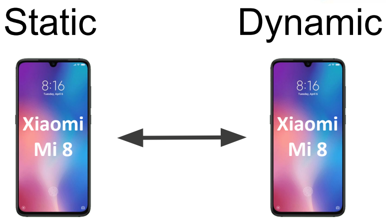
Baseline Error	IPRd	FLP	PVT
Mean [m]	0.25	1.3	0.64
Standard deviation [m]	6.4	3.0	16.91

- **94.8 %** ranging availability in urban (unavailability is x2 for PVT)
- Average of 10 common signals

III. *Inter-Phone Ranging* (IPR) - Results Analysis

Result Analysis: Dynamic & Urban - Collaborative Scenario **C**





Baseline Error	IPRd	FLP	PVT
Mean [m]	0.25	1.3	0.64
Standard deviation [m]	6.4	3.0	16.91

- High dynamic scenario for the second smartphones
- FLP positions favored by the hybridization of IMU and extra sensors in this scenario

III. *Inter-Phone Ranging* (IPR) - Conclusions

- Introduction to an innovative DGNSS 3D ranging techniques specifically designed towards smartphone measurements: Inter-Phone Ranging (**IPR**)
- Performance analysis for both nominal and urban environment:
    - Reliable and accurate IPR estimation
    - Creation of an additional measurement between two users
    - In future work, the effect of aided IPR ranging method by FLP positions shall be studied for improving urban estimations

IPR technique represents the first stepping stone towards Smartphone collaborative positioning.

Presentation Outline

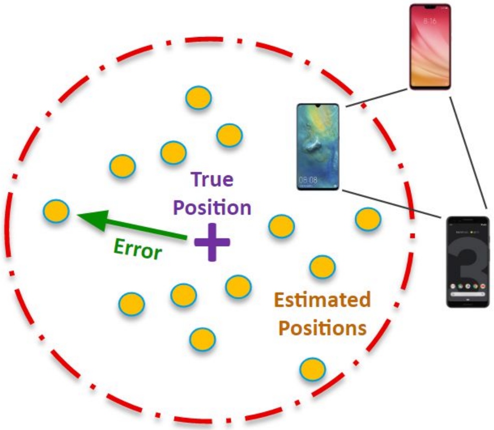
IV. Smartphone Collaborative Positioning

Collaborative Network: Establish a collaborative method based on simple assumptions taking advantages of smartphones' capabilities and volume in today's dense city centers.

Cooperative engine based on a *constrained optimization problem.*

Specifications

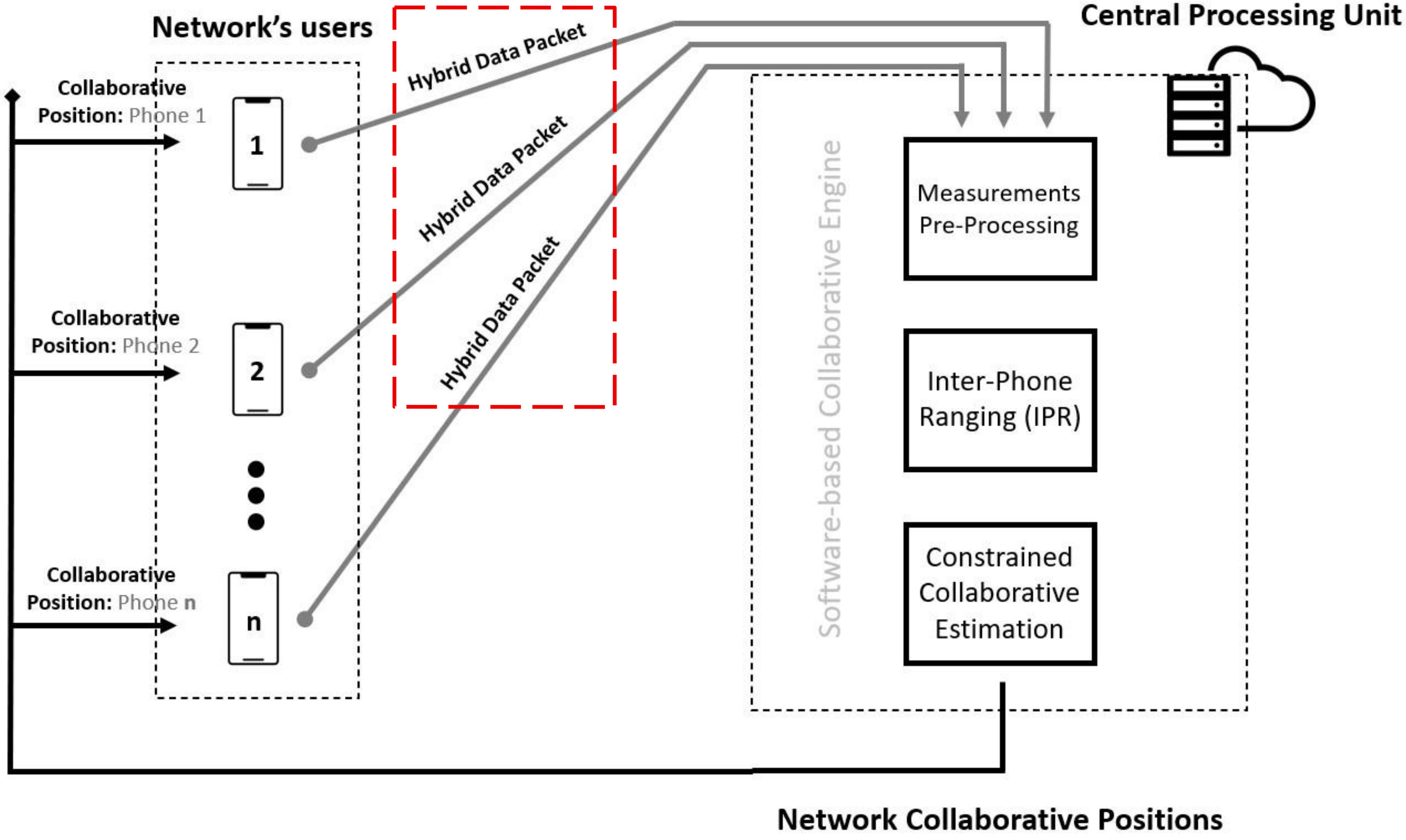
- *Low-cost cooperative network*
- *Scalable to the size of a city center*
- *Multiple users service provider*
- *Easy network implementation for users*



The efficiency of the proposed cooperative approach will be evaluated against the main key performance indicator: Accuracy

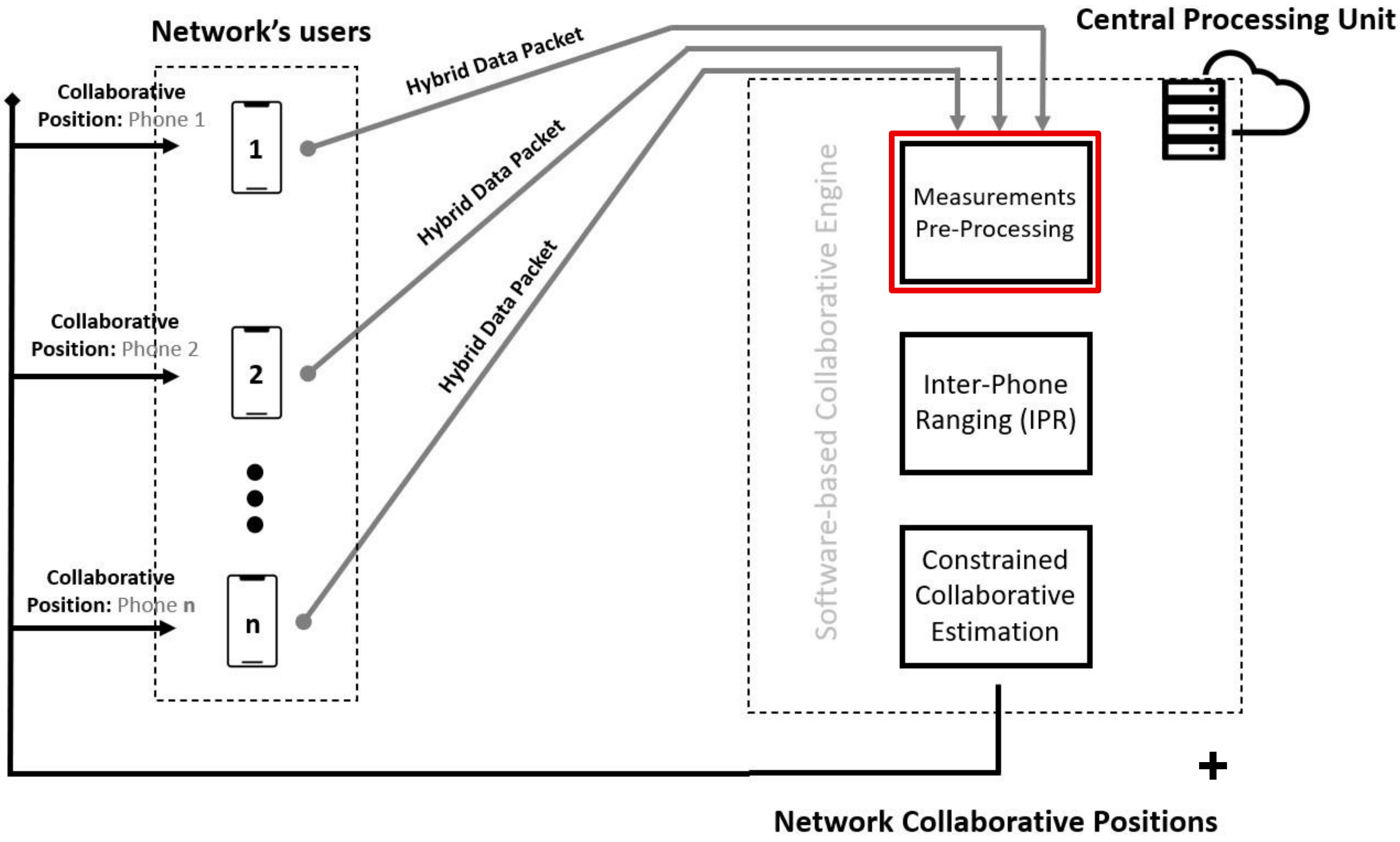
IV. Smartphone Collaborative Positioning

Smartphone-based Collaborative Network Structure Block Diagram



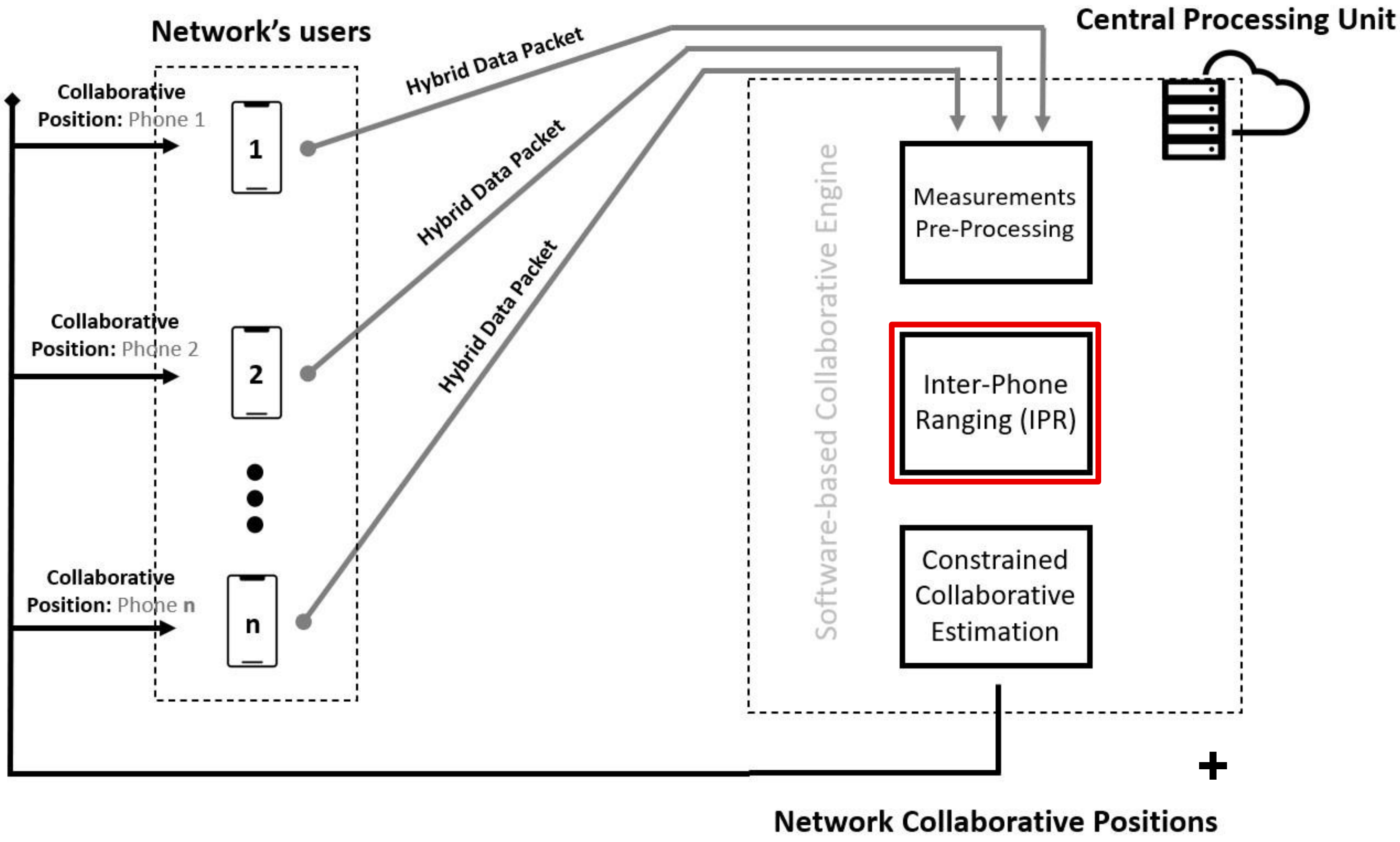
IV. Smartphone Collaborative Positioning

Smartphone-based Collaborative Network Structure Block Diagram



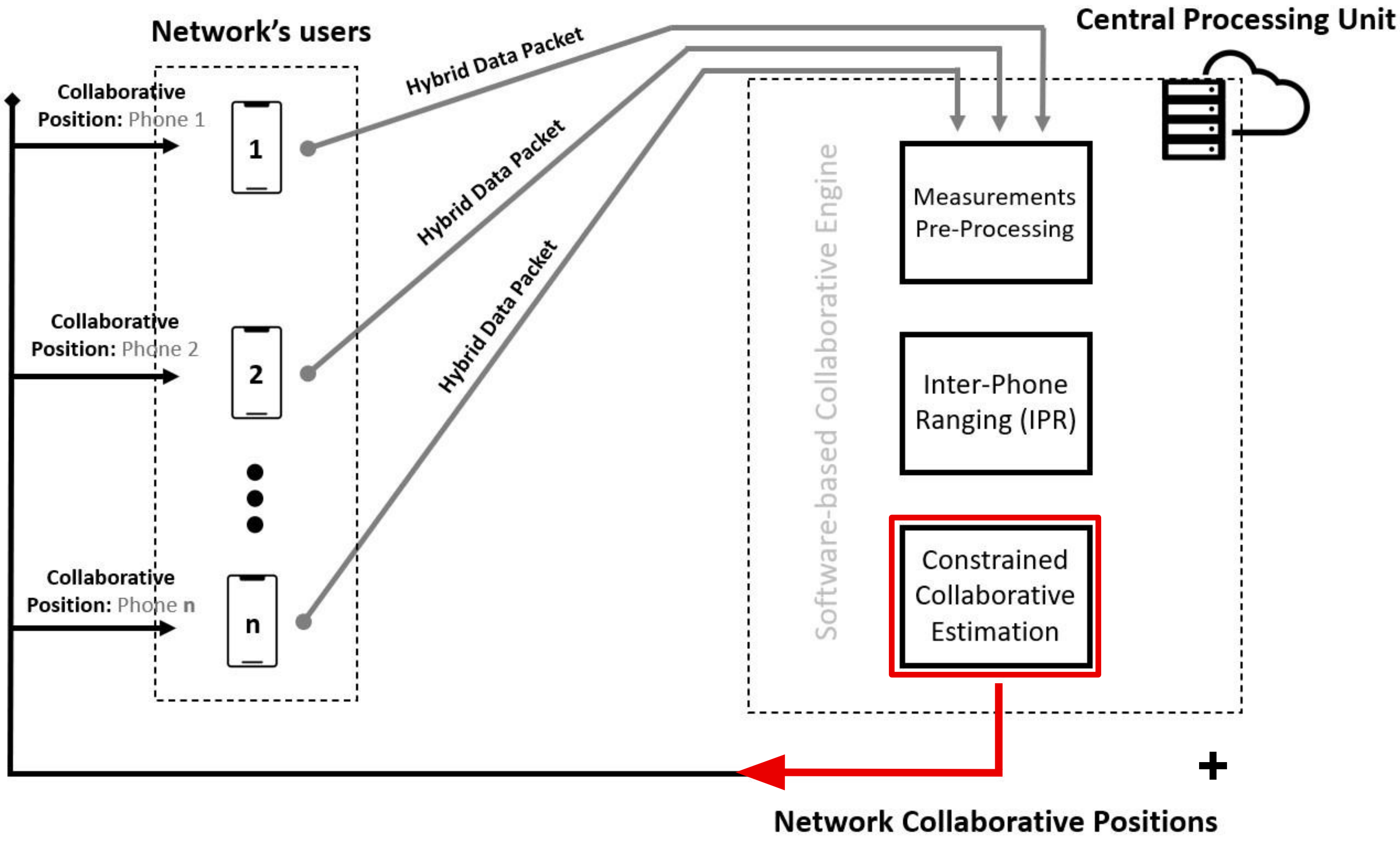
IV. Smartphone Collaborative Positioning

Smartphone-based Collaborative Network Structure Block Diagram



IV. Smartphone Collaborative Positioning

Smartphone-based Collaborative Network Structure Block Diagram

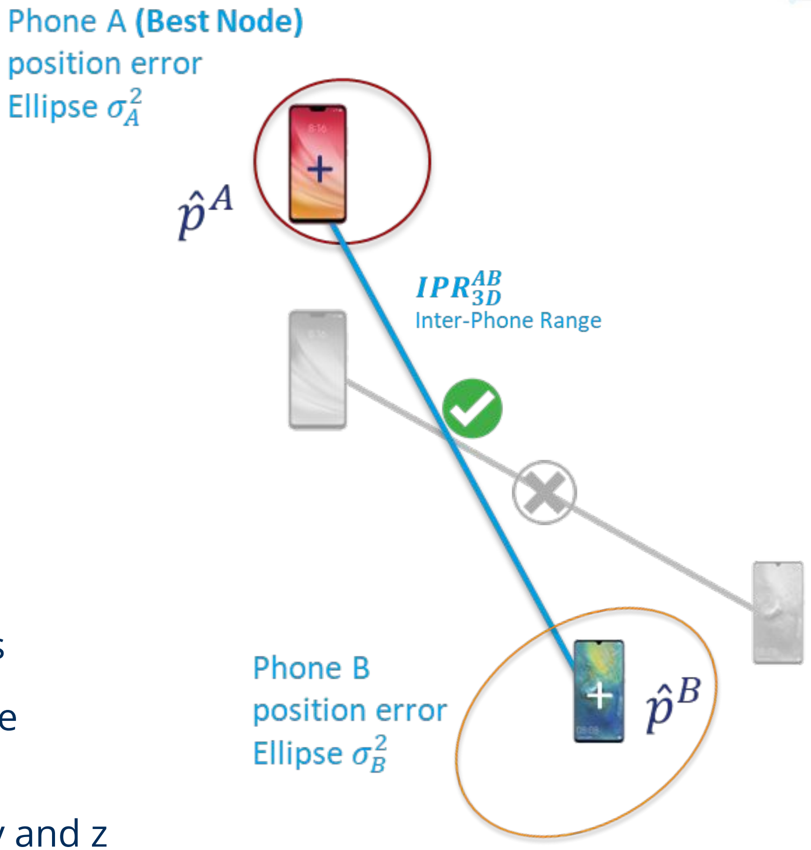


IV. Smartphone Collaborative Positioning

- Minimization of the sum of 3D position discrepancies between the newly estimated collaborative positions and the original positioning solutions.
- Minimization process **constrained** by a set of equations satisfying the equality between the estimated collaborative positions and the IPR

Hypotheses

- A secure and reliable communication link exists
- Independence between standalone smartphone positioning errors
- Independence of GNSS positioning error on x, y and z
- Positioning error follows a Gaussian distribution (known $\boldsymbol{\sigma}^2$)



IV. Smartphone Collaborative Positioning

$$\widehat{\boldsymbol{P}} = arg\ min \sum_{u=1}^{n} \frac{\left(\hat{p}_{u,x} - \tilde{p}_{u,x} - \mu_{u,x}\right)^2}{2\sigma_{u,x}^2} + \frac{\left(\hat{p}_{u,y} - \tilde{p}_{u,y} - \mu_{u,y}\right)^2}{2\sigma_{u,y}^2} + \frac{\left(\hat{p}_{u,z} - \tilde{p}_{u,z} - \mu_{u,z}\right)^2}{2\sigma_{u,z}^2}$$

Satisfying the constraining 3D equations given by:

$$\begin{cases} \left(\hat{p}_{v,x} - \hat{p}_{u,x}\right) = \mathbf{IPR}_{3D,x}^{uv} \\ \left(\hat{p}_{v,y} - \hat{p}_{u,y}\right) = \mathbf{IPR}_{3D,y}^{uv} \\ \left(\hat{p}_{v,z} - \hat{p}_{u,z}\right) = \mathbf{IPR}_{3D,z}^{uv} \end{cases}$$

with:

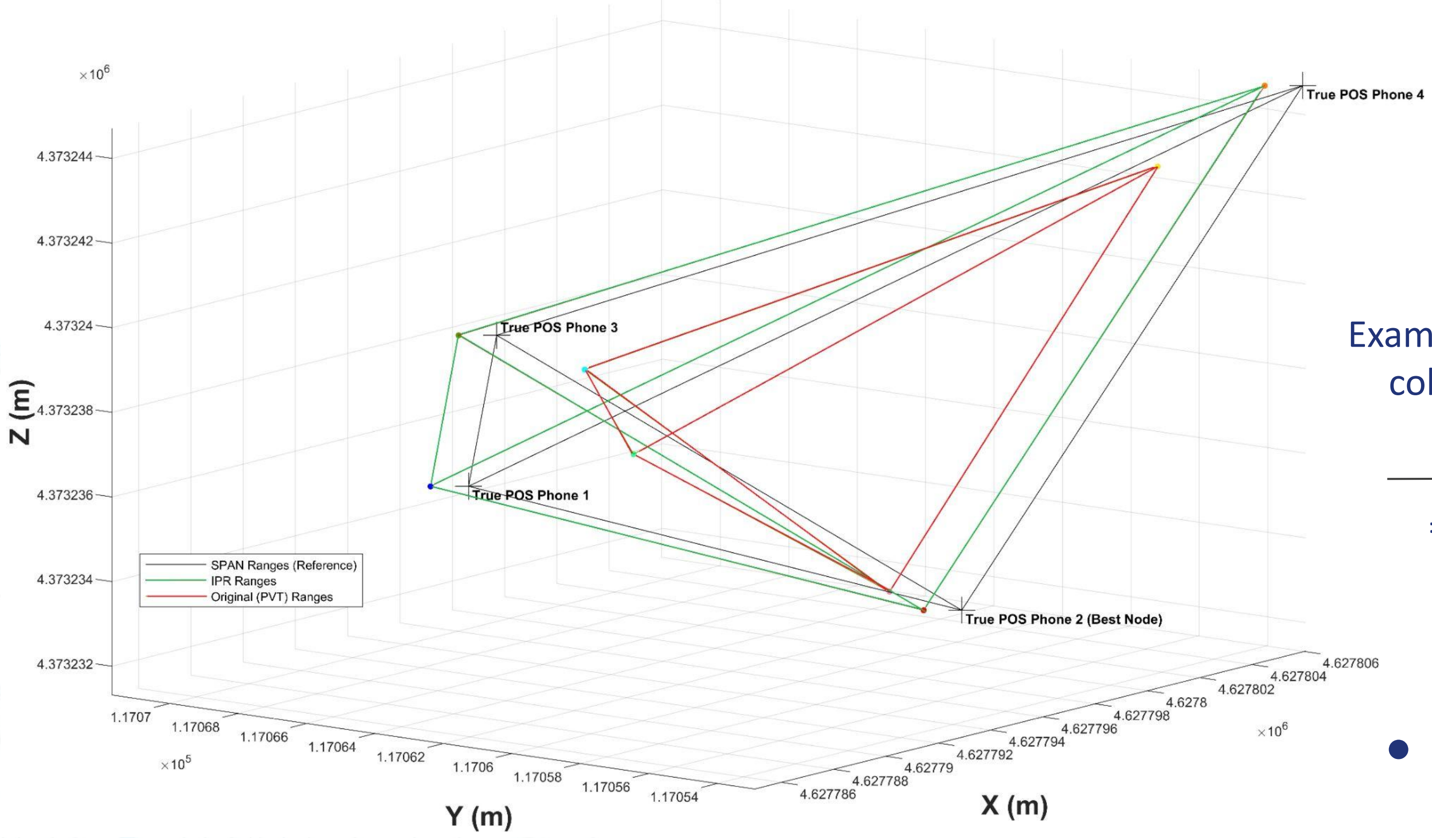
$\hat{p}$: Collaborative Position

$\tilde{p}$: Initial Position

μ and σ^2: Distribution Parameters

- Optimization method implemented in Matlab using *fmincon*
- Minimizing the objective function through iteration **while** satisfying constraints
- Selected solver: SQP (Sequential Quadratic Programming)

IV. Smartphone Collaborative Positioning



Number of constraints' set

$$\frac{M(M-1)}{2}$$

Example for a network of **M = 10** collaborative network users
= 45 sets of constraint

= **135 total constraints**

- **Selection of a Best Node phone (2)**

IV. Smartphone Collaborative Positioning

Static & Open-sky Scenario

Simulating a Collaborative Network of 10 smartphone users

- Simulating Random Standalone Positions

$$\tilde{P} \sim \mathcal{N}(0,\sigma^2)$$

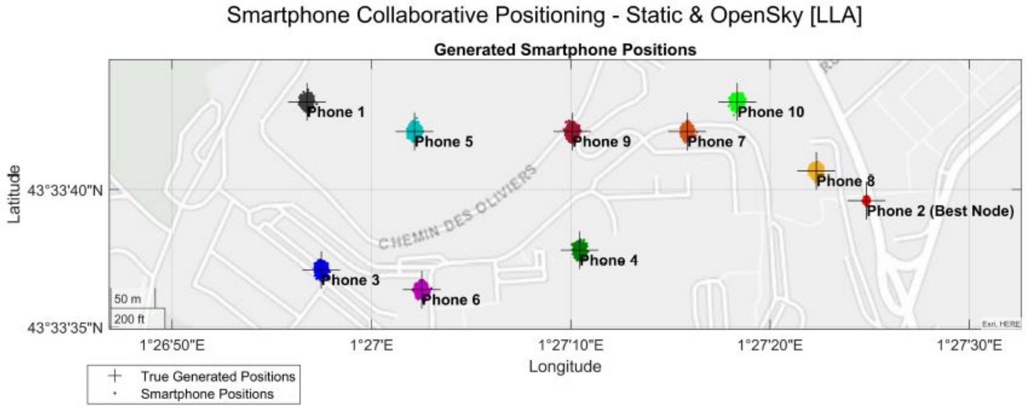
- $\sigma_{\tilde{p}} = [2.5\,,2.5\,,3.8]$ m
- $\sigma_{\tilde{p}_{Best\ Node}} = [1\,,1\,,2]$ m

- Simulating **IPR** Vector Ranges

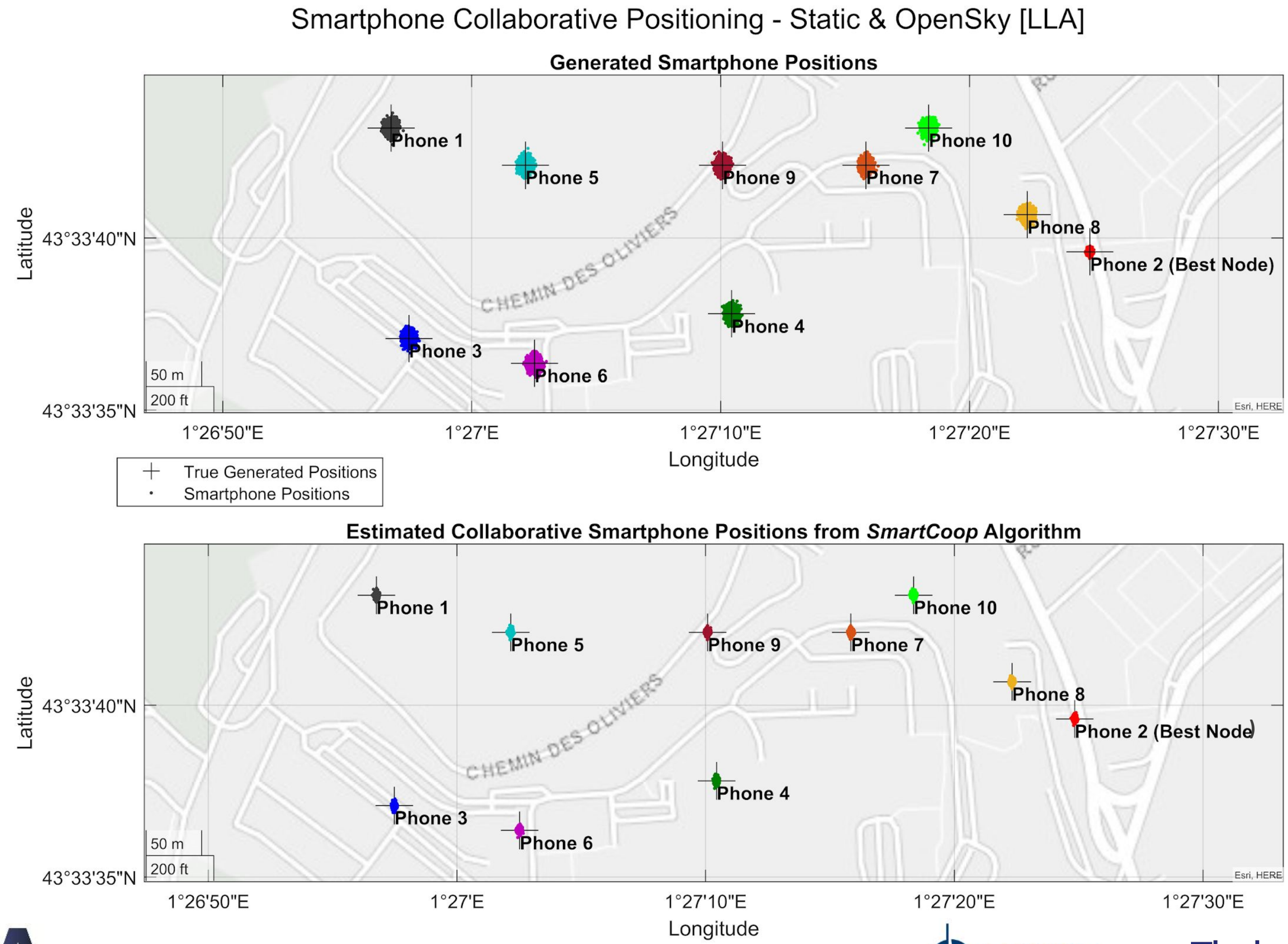
$$\boldsymbol{IPR} \sim \mathcal{N}(0,\sigma^2)$$

- $\sigma_{IPR} = [1.75\,,1.75\,,2.1]$ m

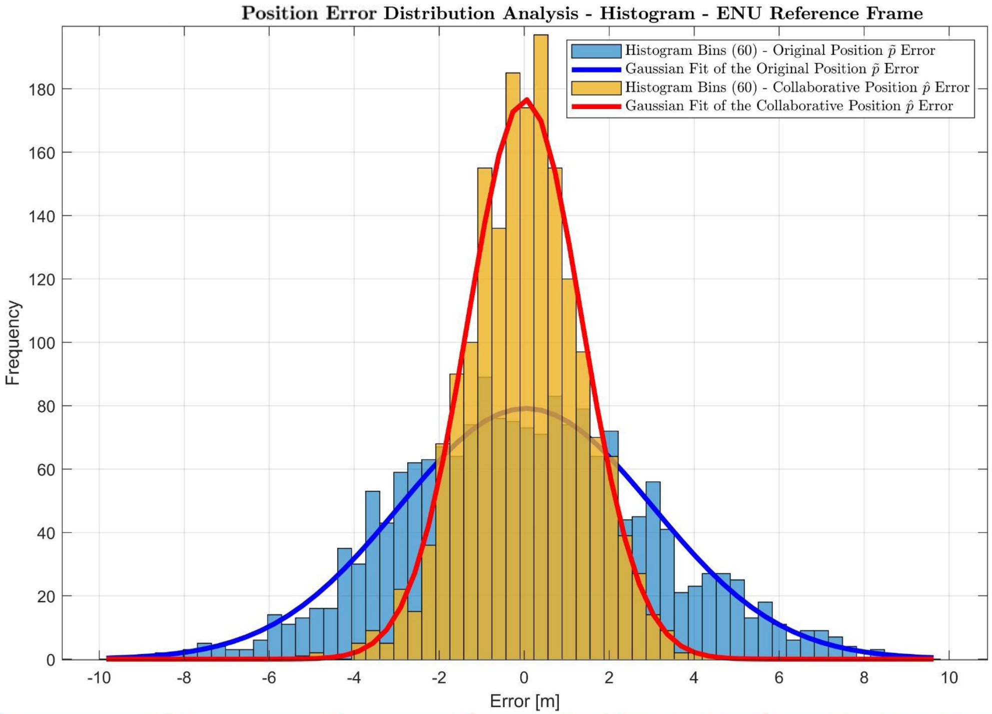
- Simulation Length = 3600 epochs @ 1Hz
- Phone 2 is defined as the best node of the network



IV. Smartphone Collaborative Positioning



IV. Smartphone Collaborative Positioning

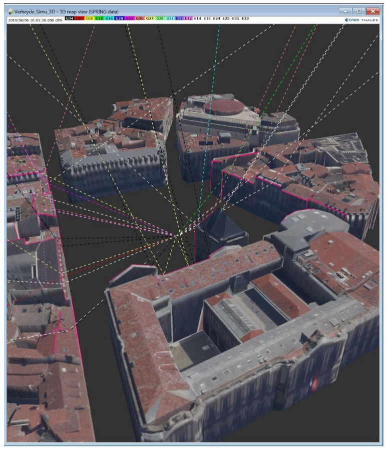


Static & Open-sky Scenario

- Collaborative positioning improved 9/10 smartphones
- Position accuracy was improved for 92% of epochs
- Average accuracy gain (68th percentile) of 3m

Network's Smartphones	68% (σ) [m]	95% (2σ) [m]	Max Error [m]
Most Improved (Phone 8)	**1.61** *(-3.2)*	**2.9** *(-3.5)*	**3.9** < 8.94
Least Improved (Phone 4)	**1.82** *(-3.0)*	**3.3** *(-2.9)*	**4.4** < 6.8
Best Node (Phone 2)	**1.34** *(+0.11)*	**2.5** *(+0.4)*	**3.4** > 3.0

IV. Smartphone Collaborative Positioning



a)

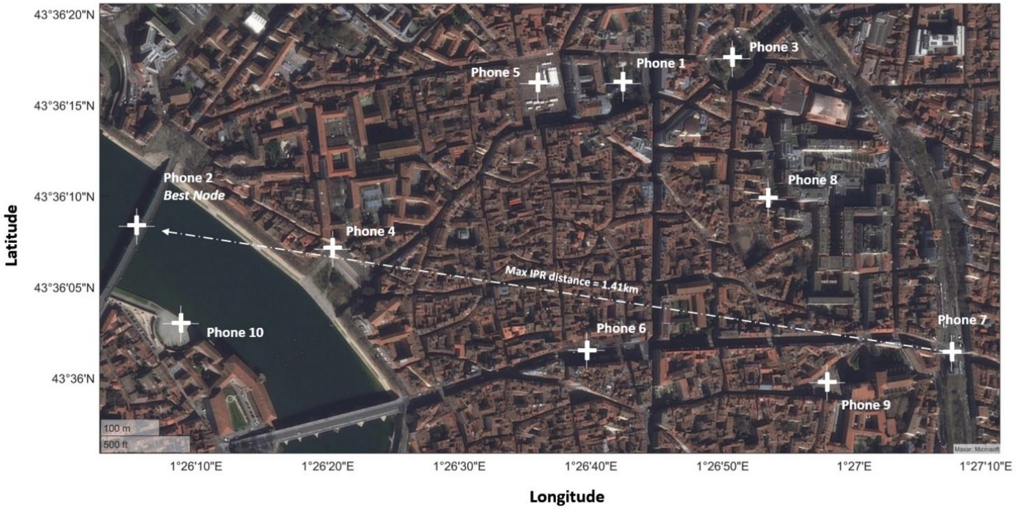
b)

Static & Urban Scenario

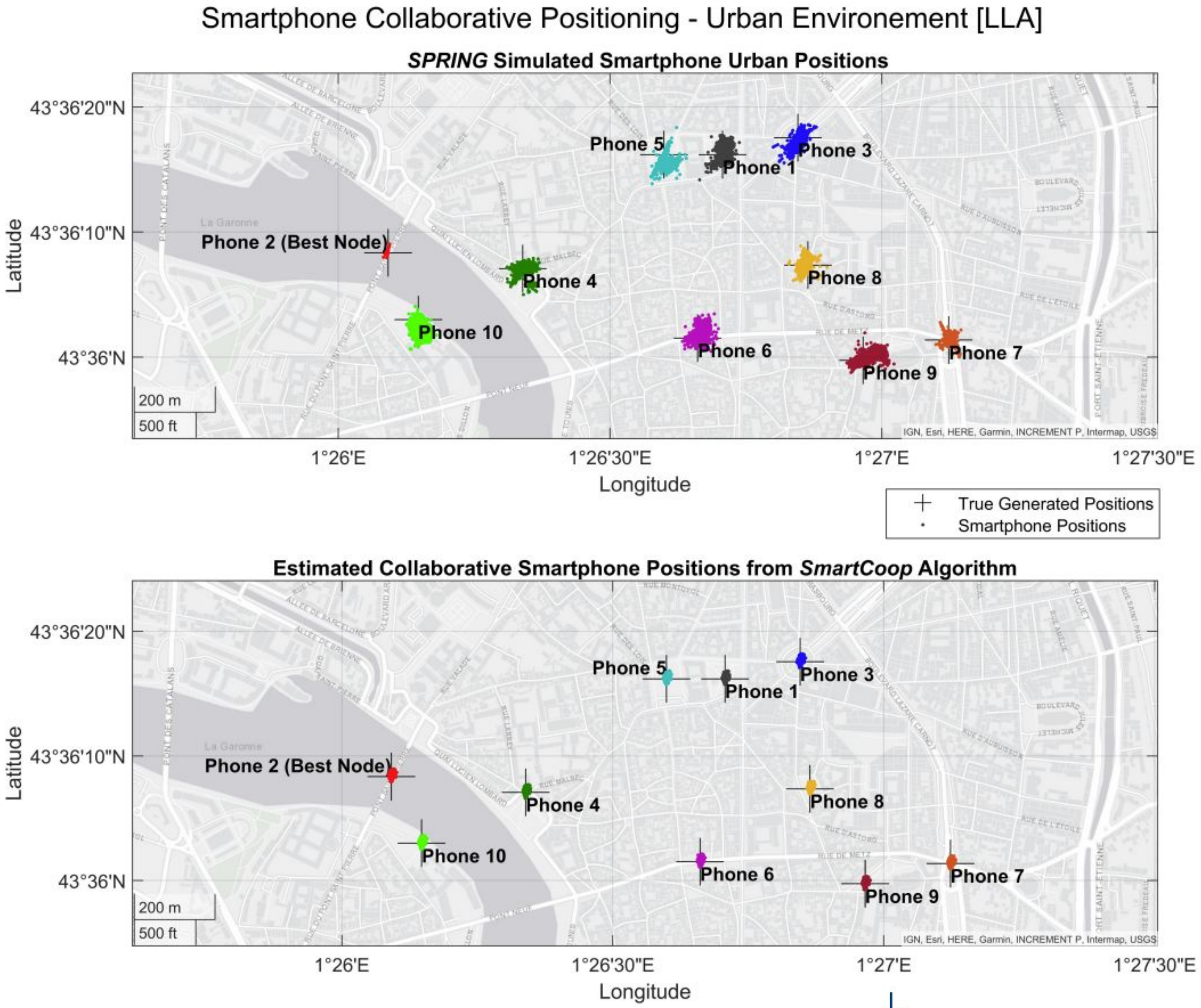
Simulating Smartphone urban positions with **SPRING©**, a CNES software:

- Simulated code measurements (including degraded signals)

- A network of **10 users** within Toulouse
- Various environments
- Phone 2 is defined as the best node of the network
- Simulation Length = 3600 epochs @ 1Hz

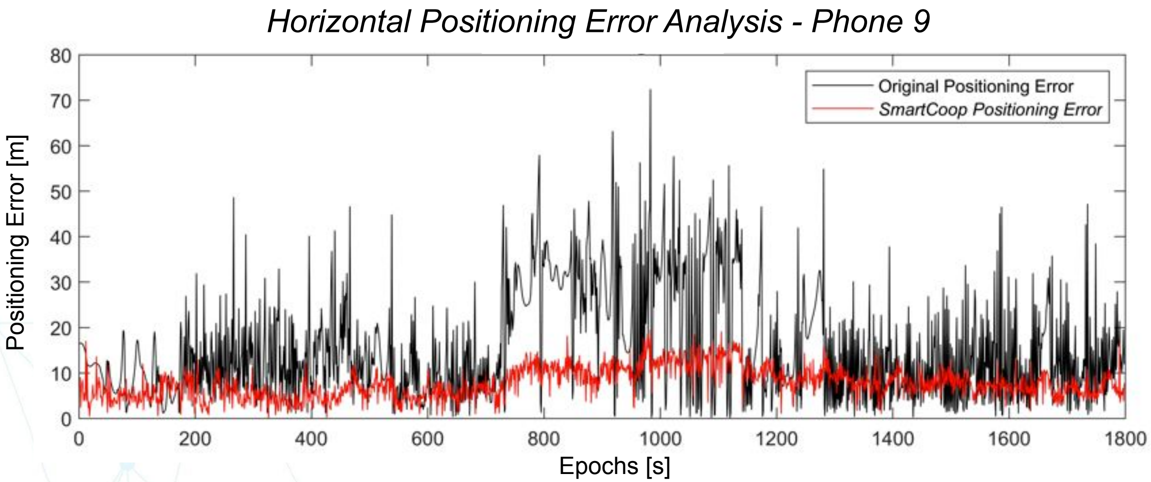


IV. Smartphone Collaborative Positioning



IV. Smartphone Collaborative Positioning

Static & Urban Scenario



- Collaborative positioning improved **9/10** smartphones
- Position accuracy was improved for **75%** of epochs
- Average accuracy gain (68th percentile) of **10m**

Network's Smartphones	68% (σ) [m]	95% (2σ) [m]	Max Error [m]
Most Improved (Phone 9)	**3.26** *(-14.7)*	**9.7** *(-13.7)*	**17.8** < 68.3
Least Improved (Phone 7)	**3.49** *(-2.43)*	**12.6** *(-9.9)*	**18.7** < 42.5
Best Node (Phone 2)	**3.04** *(+1.4)*	**2.5** *(+0.4)*	**11.2** > 8.0

IV. Smartphone Collaborative Positioning

- Conclusions & Discussion

- Development of a collaborative network exclusively based on Android smartphone's GNSS measurements
- Simulation results demonstrate the proof-of-concept for collaborative positioning in both urban and nominal conditions

- Impact of the best node phone: Smartphone used to aid the connected set of users through the computation of IPR ranges.
- Impact of the user geometry: The geometrical repartition of users on a 2D plane might influence the estimation process.

***SmartCoop* cooperative engine shows promising results with consistent positioning improvement in urban environment**

Presentation Outline

Conclusions

- **Characterizing Smartphone GNSS Raw Measurements**
  - ○ Development of an Android application made for recording Android GNSS raw data measurements and additional sensors (SmartLogger)
  - ○ Coordination of a data collection campaign that includes multiple receivers and vehicles designed for testing collaborative scenarios in various situations targeting urban applications
  - ○ Analysis revealed the strengths and weaknesses of the Android GNSS measurements. Great availability (> 30 tracked signals) despite unadapted hardware components. Study highlighted the suitability of Android GNSS raw data measurements for Collaborative purposes
  - ○ Successful characterization and evaluation of Android GNSS raw data measurements

Conclusions

- Smartphone Ranging Techniques
    - Inter-Phone Ranging (IPR) is a technique generating 3D ranges between mobiles that specifically designed for smartphone-based measurements.
    - A methodology have been reported describing the protocol and estimation of a classical DGNSS technique using Android GNSS raw data measurements
    - A detailed analysis was shown depicting the performance of IPR estimates compared to other ranging means. This analysis highlighted the benefits in nominal conditions and the estimation availability in constrained environments (urban)
    - The computation of a reliable inter-user range granted the realization of a smartphone-based collaborative network.

Conclusions

- **Collaborative Positioning Network**
  - Our envisioned network structure was presented outlining a central processing unit system where users could exchange hybrid data packets against newly estimated collaborative positions
  - An optimization algorithm was developed in Matlab for estimating collaborative positions using a set of constraining equations given by IPR estimates for the entire network
  - Simulations (urban and nominal conditions) were conducted for evaluating smartphone-based collaborative performance.
  - SmartCoop allows for a significant positioning improvement for static datasets in both open-sky and urban environment. Position accuracy is globally improved for network users (9/10) for 84% of the computed epochs for both conditions.

Perspectives

- **Android GNSS raw measurement characterization method**
    - Validate measurement quality for newest smartphone brands & models.
- **Inter-Phone Ranging Constraints**
    - Additional sensor information (i.e: barometer) could be added as new constraints.
    - Computation of aided-IPR based on FLP positions in urban environment
- **Smartphone Collaborative Network**
    - The collaborative engine shall now be tested in real conditions. A dedicated data collection campaign should take place (example: users could be placed on geo-referenced points).
    - Creation of a live collaborative network shall overcome the correct data transmission of measurement and ensure the time-consistency among the exchanged data.

Publications

[1] T. Verheyde, A. Blais, C. Macabiau, and F.-X. Marmet, "***Statistical Analysis of Android GNSS Raw Data Measurements in an Urban Environment for Smartphone Collaborative Positioning Methods***," Presentation at INC 2019, 2019.

[2] T. Verheyde, A. Blais, C. Macabiau, and F.-X. Marmet, "***Analyzing Android GNSS Raw Measurements Flags Detection Mechanisms for Collaborative Positioning in Urban Environment***", *2020 International Conference on Localization and GNSS (ICL-GNSS)*, Tampere, Finland, 2020, pp. 1-6, doi: 10.1109/ICL-GNSS49876.2020.9115564.

[3] 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.

[4] T. Verheyde, A. Blais, C. Macabiau, and F.-X. Marmet, "***Smartcoop Algorithm: Improving Smartphones Position Accuracy and Reliability Through Collaborative Positioning***", *2021 International Conference on Localization and GNSS (ICL-GNSS)*, Tampere, Finland, 2021, pp. 1-6, doi: 10.1109/ICL-GNSS49876.2020.9115564,2021.

Thomas VERHEYDE

2023 February, 10^{th}

Questions ?

Precise Cooperative Positioning of Low-Cost Mobiles in an Urban Environment

MARAIS Juliette	*Reviewer*	*Université Gustave Eiffel*
DOVIS Fabio	*Reviewer*	*Politecnico di Torino*
MACABIAU Christophe	*Thesis director*	*École Nationale d'Aviation Civile*
BLAIS Antoine	*Thesis co-director*	*École Nationale d'Aviation Civile*
MARMET François-Xavier	*Invitee*	*Centre National d'Etudes Spatiales*
SERANT Damien	*Invitee*	*Thalès Alenia Space*

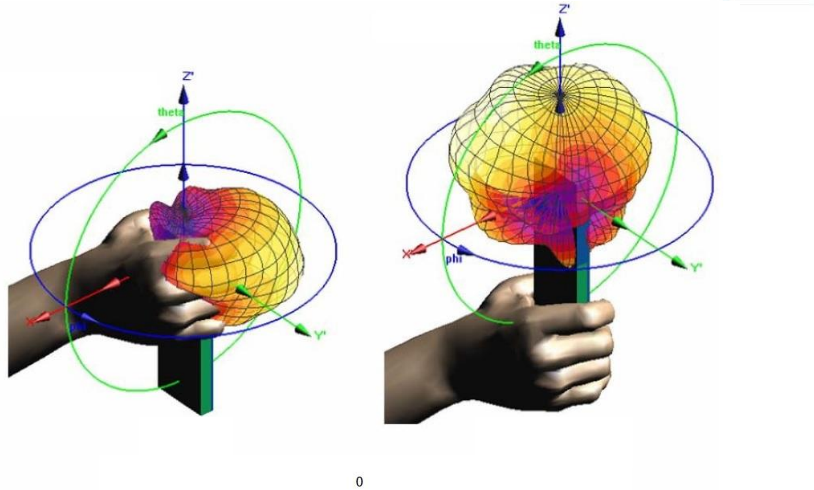
PhD Thesis has been made in collaboration with:

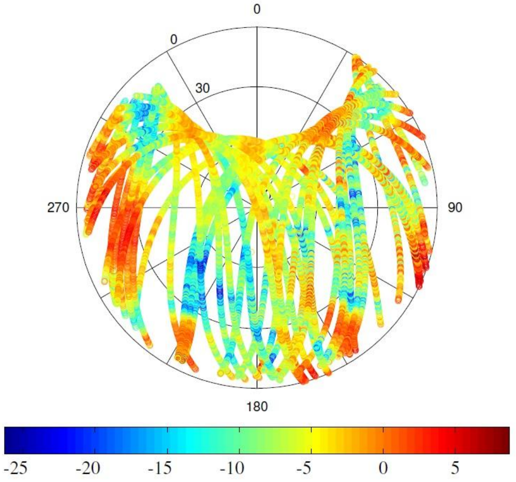
Backup Slides

II. Can Smartphones be considered as a GNSS receiver? - Android GNSS Raw Data Measurements

Disadvantages

- Tight integration of low-cost components unoptimized for processing GNSS signals
  - Low-cost IMU
  - Linearly polarized antenna
- Duty cycle
- Unreliable phase measurements
- Frequent loss of lock and cycle slips
- Inconsistent characteristics throughout smartphone brands and models.





II. Can Smartphones be considered as a GNSS receiver? - Android GNSS Raw Data Measurements

android.location: API for Android Location-based services (Android 12.0 | API Level 31) – Updated 09/2021

ANDROID CLASS	Field	Public Method	Description
android.location.**GnssClock**	*LeapSecond*	`public int getLeapSecond ()`	Leap second associated with the clock's time. [s]
android.location.**GnssClock**	*TimeNanos*	`public long getTimeNanos ()`	Embedded GNSS Receiver clock value. [ns]
android.location.**GnssClock**	*TimeUncertaintyNanos*	`public double getTimeUncertaintyNanos ()`	Clock's time uncertainty (1-Sigma). [ns]
android.location.**GnssClock**	*BiasNanos*	`public double getBiasNanos ()`	Clock's sub-nanosecond bias. [ns]
android.location.**GnssClock**	*UncertaintyNanos*	`public double getBiasUncertaintyNanos ()`	Clock's bias uncertainty. (1 Sigma) [ns]
android.location.**GnssClock**	*DriftNanosPerSecond*	`public double getDriftNanosPerSecond ()`	Clock's Drift. A positive value indicates that the frequency is higher than GPS master clock. [ns/s]
android.location.**GnssClock**	*DriftUncertaintyNanos PerSecond*	`public double getDriftUncertaintyNanosPerSecond ()`	Clock's Drift uncertainty. (1-Sigma) [ns/s]
android.location.**GnssClock**	*ElapsedRealTimeNanos*	`public long getElapsedRealtimeNanos ()`	Elapsed real-time of the clock since system boot. [ns]
android.location.**GnssClock**	*FullBiasNanos*	`public long getFullBiasNanos ()`	Difference between the hardware clock (*TimeNanos*) and the true GPS time since 0000Z January 6th, 1980. [ns]
android.location.**GnssClock**	*HardwareClock DiscontinuityCount*	`public long getHardwareClockDiscontinuityCount ()`	Count of hardware counts discontinuity. When value is similar between epochs, clock is continuous and can be modelled from classic clock & drift models.

GnssClock

II. Can Smartphones be considered as a GNSS receiver? - Android GNSS Raw Data Measurements

android.location: API for Android Location-based services (Android 12.0 | API Level 31) – Updated 09/2021

ANDROID CLASS	Field	Public Method	Description
android.location.**GnssClock**	*LeapSecond*	public int getLeapSecond ()	Leap second associated with the clock’s time. [s]
android.location.**GnssClock**	*TimeNanos*	public long getTimeNanos ()	Embedded GNSS Receiver clock value. [ns]
android.location.**GnssClock**	*TimeUncertaintyNanos*	public double getTimeUncertaintyNanos ()	Clock’s time uncertainty (1-Sigma). [ns]
android.location.**GnssClock**	*BiasNanos*	public double getBiasNanos ()	Clock’s sub-nanosecond bias. [ns]
android.location.**GnssClock**	*UncertaintyNanos*	public double getBiasUncertaintyNanos ()	Clock’s bias uncertainty. (1 Sigma) [ns]
android.location.**GnssClock**	*DriftNanosPerSecond*	public double getDriftNanosPerSecond ()	Clock’s Drift. A positive value indicates that the frequency is higher than GPS master clock. [ns/s]
android.location.**GnssClock**	*DriftUncertaintyNanos PerSecond*	public double getDriftUncertaintyNanosPerSecond ()	Clock’s Drift uncertainty. (1-Sigma) [ns/s]
android.location.**GnssClock**	*ElapsedRealTimeNanos*	public long getElapsedRealtimeNanos ()	Elapsed real-time of the clock since system boot. [ns]
android.location.**GnssClock**	*FullBiasNanos*	public long getFullBiasNanos()	Difference between the hardware clock (*TimeNanos*) and the true GPS time since 0000Z January 6th, 1980. [ns]
android.location.**GnssClock**	*HardwareClock DiscontinuityCount*	public long getHardwareClockDiscontinuityCount ()	Count of hardware counts discontinuity. When value is similar between epochs, clock is continuous and can be modelled from classic clock & drift models.
android.location.**GNSSAntennaInfo**	*PhaseCenterOffset*	public object getPhaseCenterOffset ()	Return object containing the phase center offset and the associated uncertainties. [mm]
android.location.**GNSSAntennaInfo**	*PhaseCenterVariation Corrections*	public object getPhaseCenterVariation Correction ()	Return object with phase center variation correction and associated uncertainties. [mm]
android.location.**GNSSAntennaInfo**	*SignalGainCorrections*	public object getSignalGainCorrections()	Return a spherical correction object containing the signal gain corrections and associated uncertainties. [dBi]
android.location.**GNSSAntennaInfo**	*CarrierFrequencyMHz*	public double getCarrierFrequencyMHz ()	Tracked signal carrier frequency. [MHz]

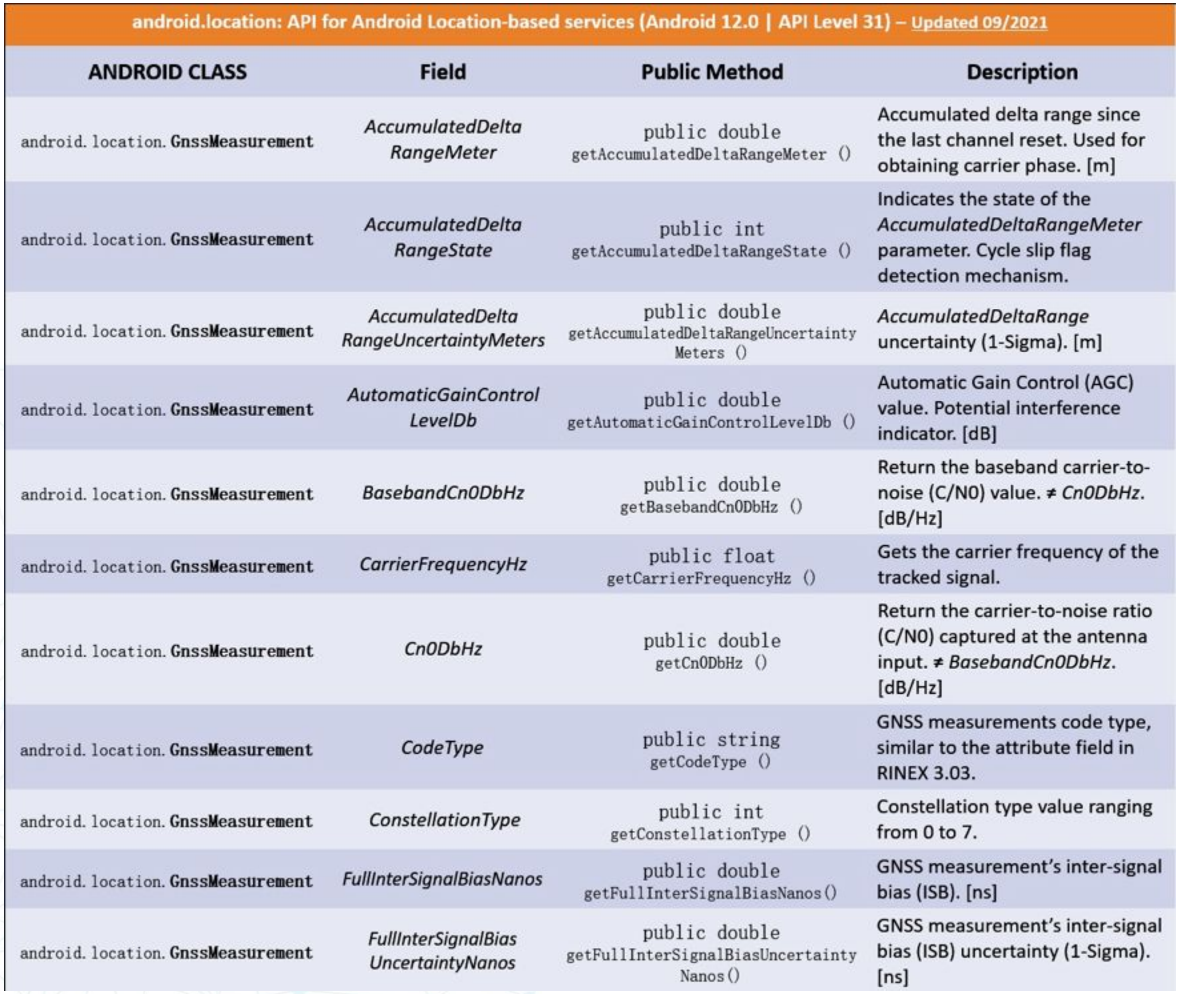
GnssClock & AntennaInfo

AccumulatedDeltaRangeState: Constant State Indicator List

State Indicators Name	Value
ADR_STATE_CYCLE_SLIP	**4**
ADR_STATE_HALF_CYCLE_REPORTED	**16**
ADR_STATE_HALF_CYCLE_RESOLVED	**8**
ADR_STATE_RESET	**2**
ADR_STATE_UNKNOWN	**0**
ADR_STATE_VALID	**1**

AccumulatedDeltaRangeState

II. Can Smartphones be considered as a GNSS receiver? - Android GNSS Raw Data Measurements

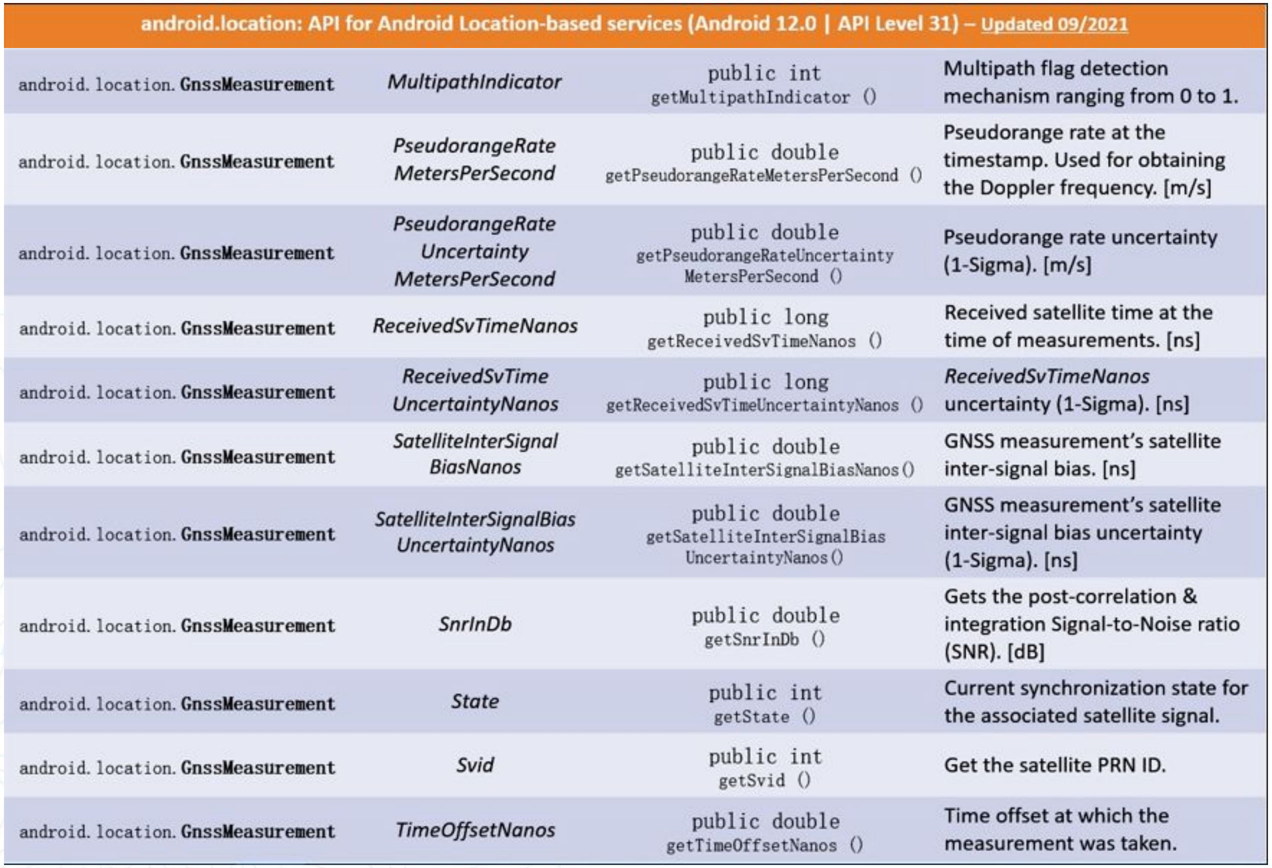


android.location: API for Android Location-based services (Android 12.0 | API Level 31) – Updated 09/2021

ANDROID CLASS	Field	Public Method	Description
android.location.**GnssMeasurement**	*AccumulatedDeltaRangeMeter*	public double getAccumulatedDeltaRangeMeter ()	Accumulated delta range since the last channel reset. Used for obtaining carrier phase. [m]
android.location.**GnssMeasurement**	*AccumulatedDeltaRangeState*	public int getAccumulatedDeltaRangeState ()	Indicates the state of the *AccumulatedDeltaRangeMeter* parameter. Cycle slip flag detection mechanism.
android.location.**GnssMeasurement**	*AccumulatedDeltaRangeUncertaintyMeters*	public double getAccumulatedDeltaRangeUncertaintyMeters ()	*AccumulatedDeltaRange* uncertainty (1-Sigma). [m]
android.location.**GnssMeasurement**	*AutomaticGainControlLevelDb*	public double getAutomaticGainControlLevelDb ()	Automatic Gain Control (AGC) value. Potential interference indicator. [dB]
android.location.**GnssMeasurement**	*BasebandCn0DbHz*	public double getBasebandCn0DbHz ()	Return the baseband carrier-to-noise (C/N0) value. ≠ *Cn0DbHz*. [dB/Hz]
android.location.**GnssMeasurement**	*CarrierFrequencyHz*	public float getCarrierFrequencyHz ()	Gets the carrier frequency of the tracked signal.
android.location.**GnssMeasurement**	*Cn0DbHz*	public double getCn0DbHz ()	Return the carrier-to-noise ratio (C/N0) captured at the antenna input. ≠ *BasebandCn0DbHz*. [dB/Hz]
android.location.**GnssMeasurement**	*CodeType*	public string getCodeType ()	GNSS measurements code type, similar to the attribute field in RINEX 3.03.
android.location.**GnssMeasurement**	*ConstellationType*	public int getConstellationType ()	Constellation type value ranging from 0 to 7.
android.location.**GnssMeasurement**	*FullInterSignalBiasNanos*	public double getFullInterSignalBiasNanos()	GNSS measurement's inter-signal bias (ISB). [ns]
android.location.**GnssMeasurement**	*FullInterSignalBiasUncertaintyNanos*	public double getFullInterSignalBiasUncertaintyNanos()	GNSS measurement's inter-signal bias (ISB) uncertainty (1-Sigma). [ns]

GnssMeasurement

II. Can Smartphones be considered as a GNSS receiver? - Android GNSS Raw Data Measurements



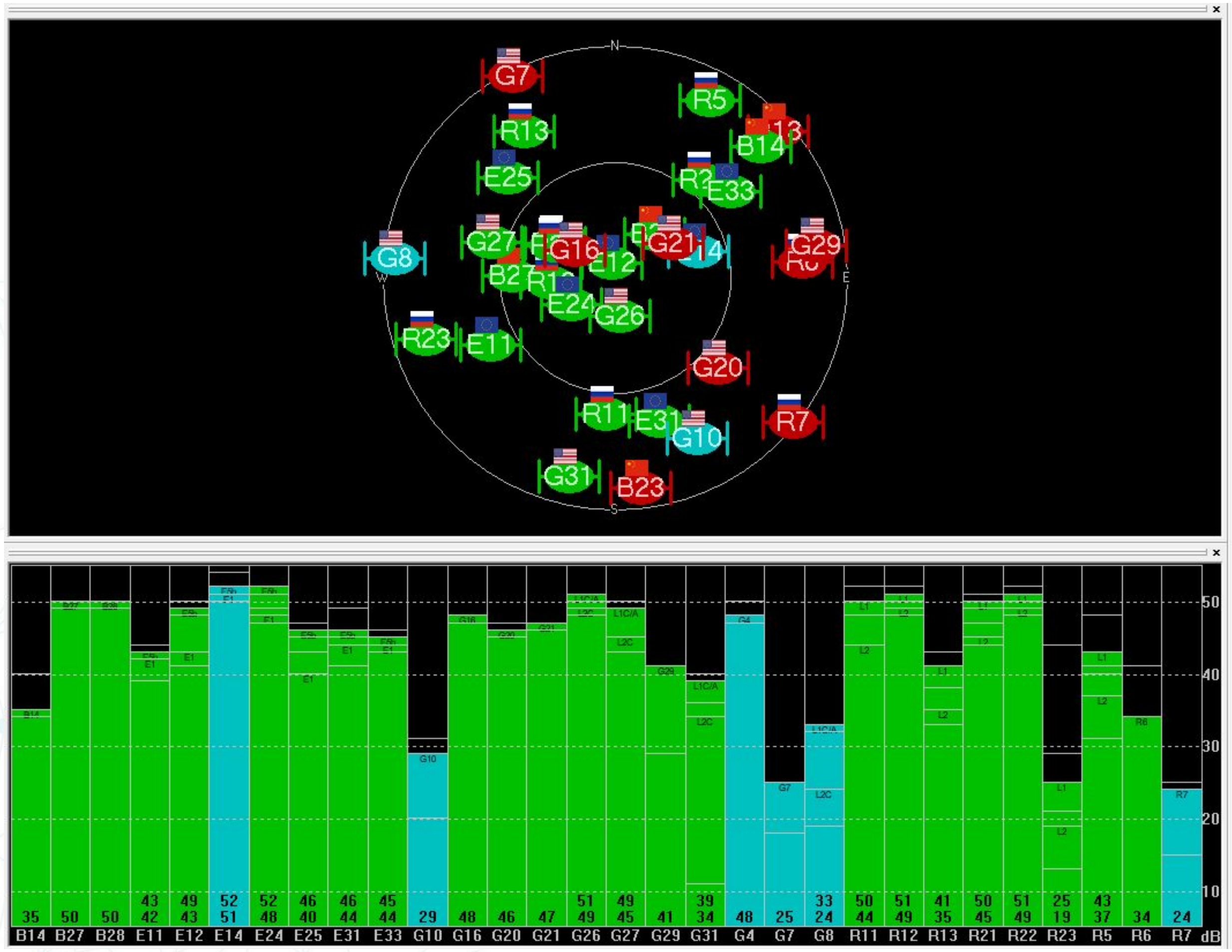
android.location: API for Android Location-based services (Android 12.0 | API Level 31) – Updated 09/2021

android.location.**GnssMeasurement**	*MultipathIndicator*	public int getMultipathIndicator ()	Multipath flag detection mechanism ranging from 0 to 1.
android.location.**GnssMeasurement**	*PseudorangeRate MetersPerSecond*	public double getPseudorangeRateMetersPerSecond ()	Pseudorange rate at the timestamp. Used for obtaining the Doppler frequency. [m/s]
android.location.**GnssMeasurement**	*PseudorangeRate Uncertainty MetersPerSecond*	public double getPseudorangeRateUncertainty MetersPerSecond ()	Pseudorange rate uncertainty (1-Sigma). [m/s]
android.location.**GnssMeasurement**	*ReceivedSvTimeNanos*	public long getReceivedSvTimeNanos ()	Received satellite time at the time of measurements. [ns]
android.location.**GnssMeasurement**	*ReceivedSvTime UncertaintyNanos*	public long getReceivedSvTimeUncertaintyNanos ()	*ReceivedSvTimeNanos* uncertainty (1-Sigma). [ns]
android.location.**GnssMeasurement**	*SatelliteInterSignal BiasNanos*	public double getSatelliteInterSignalBiasNanos()	GNSS measurement's satellite inter-signal bias. [ns]
android.location.**GnssMeasurement**	*SatelliteInterSignalBias UncertaintyNanos*	public double getSatelliteInterSignalBias UncertaintyNanos()	GNSS measurement's satellite inter-signal bias uncertainty (1-Sigma). [ns]
android.location.**GnssMeasurement**	*SnrInDb*	public double getSnrInDb ()	Gets the post-correlation & integration Signal-to-Noise ratio (SNR). [dB]
android.location.**GnssMeasurement**	*State*	public int getState ()	Current synchronization state for the associated satellite signal.
android.location.**GnssMeasurement**	*Svid*	public int getSvid ()	Get the satellite PRN ID.
android.location.**GnssMeasurement**	*TimeOffsetNanos*	public double getTimeOffsetNanos ()	Time offset at which the measurement was taken.

GnssMeasurement (Cont.)

II. Can Smartphones be considered as a GNSS receiver? - Evaluating Smartphones Measurements

Measurements Evaluation: Urban Conditions



Brief Overlook :

Average # of tracked signals* = 26

Minimum # of tracked signal = 15

(*30 tracked by Ublox F9P)

II. Can Smartphones be considered as a GNSS receiver?

Measurements Evaluation: Urban Conditions

Urban Positioning Performance

Smartphone A

Horizontal positioning error
(1Hz - 1200 epochs)

FLP
(Fused Location Provider)

1σ (68%)	Max
3.7 m	9.3 m

PVT
(Standalone)

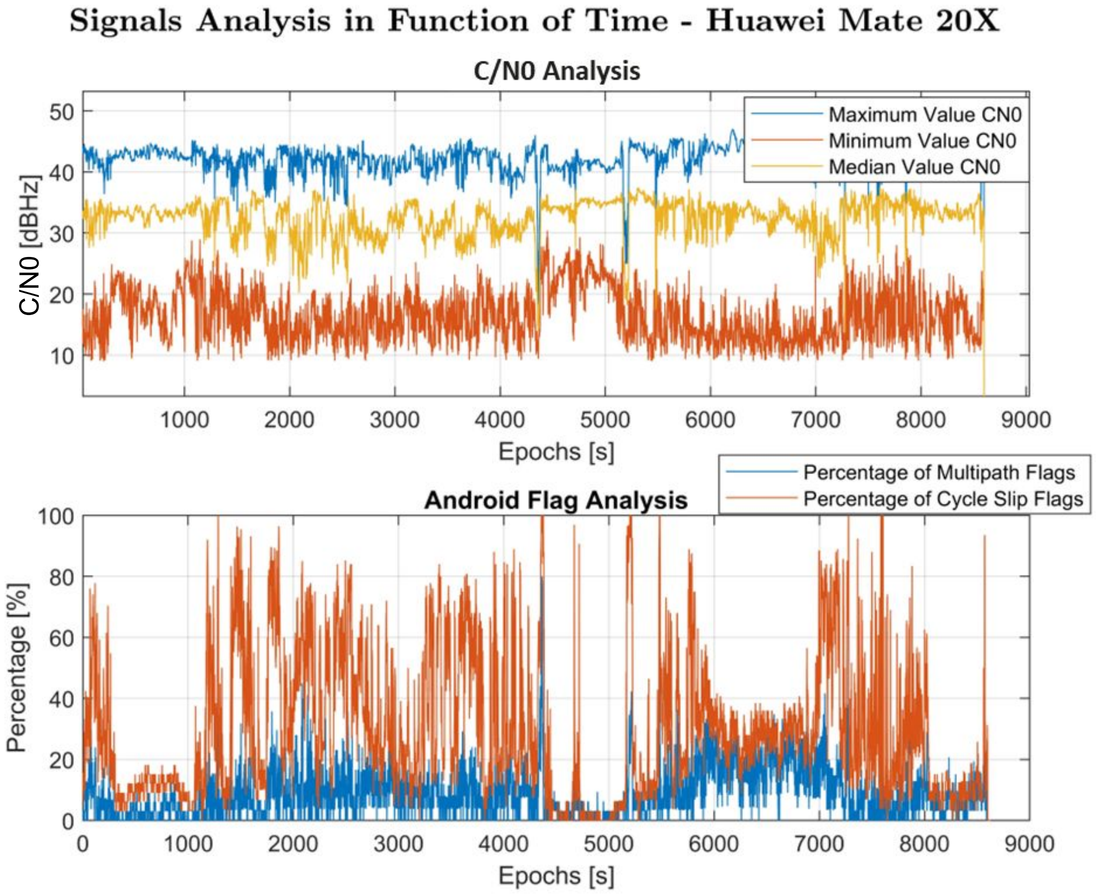
1σ (68%)	Max
8.7 m	62 m

II. Can Smartphones be considered as a GNSS receiver?

Measurements Evaluation: Urban Conditions

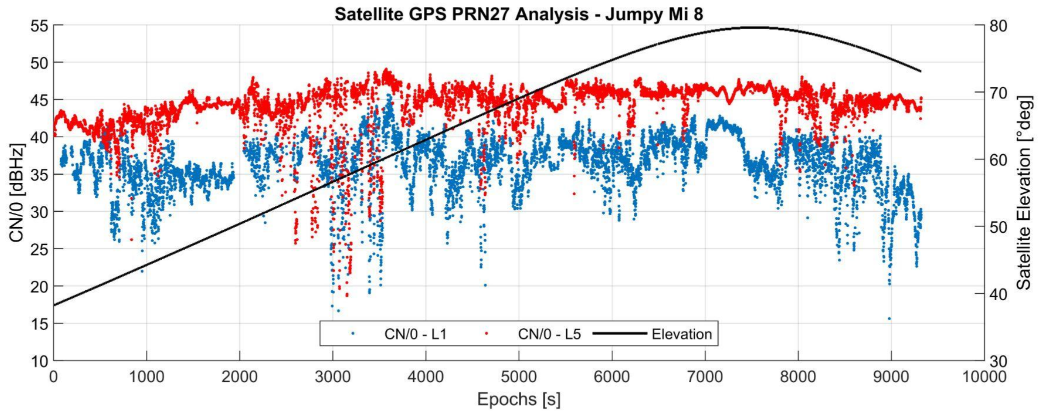
Urban Positioning Performance

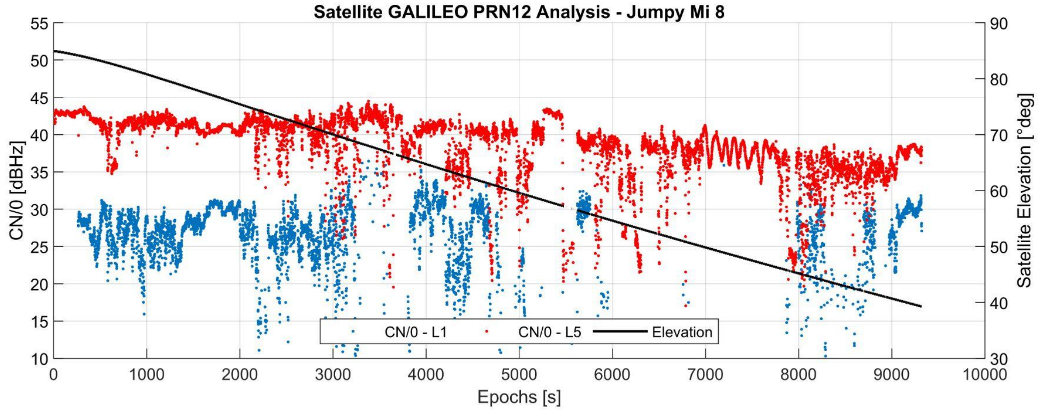
HiSilicon
Kirin 980



II. Can Smartphones be considered as a GNSS receiver?

Measurements Evaluation: Urban Conditions



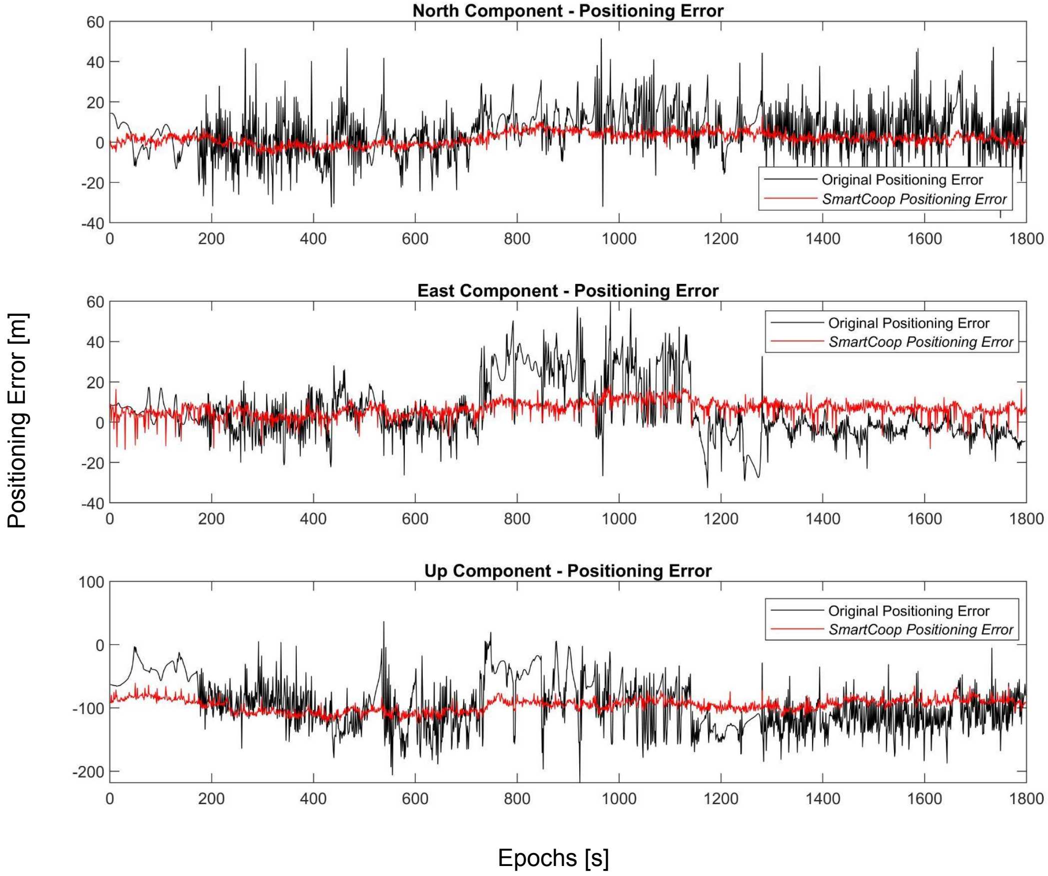


Urban Positioning Performance

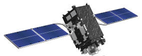
IV. Smartphone Collaborative Positioning

Static & Urban Scenario

Horizontal Positioning Error Analysis - Phone 9



II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements



Best available satellite signal

Receiver Code Noise Estimation (DLL)

Measurements Evaluation: Nominal Conditions

***Code** model :* $\rho_i^{sv} = r + c\,(t_{rx} - t_{tx}) + \varepsilon_{Iono} + \varepsilon_{Tropo} + \varepsilon_{Code} + \varepsilon_{Multipath}^{\rho}$

***Phase** model :* $\varphi_i^{sv} = r + c\,(t_{rx} - t_{tx}) - \varepsilon_{Iono} + \varepsilon_{Tropo} + \Delta\varphi + \varepsilon_{Phase} + \varepsilon_{Multipath}^{\varphi}$

***Code minus carrier** (CMC) model :* $\rho_i^{sv} - \varphi_i^{sv} = 2\varepsilon_{Iono} + \varepsilon_{Code} - \cancel{\varepsilon_{Phase}} + \cancel{\Delta\varepsilon_{Multipath}} + \cancel{\Delta\varphi}$

***CMC Time differenced** model :*

$$\frac{d}{dt}(\rho_i^{sv} - \varphi_i^{sv}) = \frac{d}{dt}(\cancel{2\varepsilon_{Iono}} + \varepsilon_{Code})$$

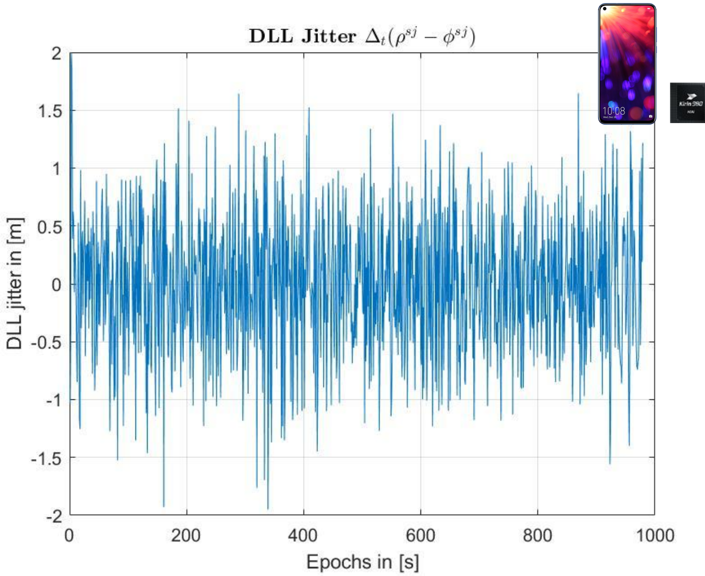
Hypothesis

Cycle slip detection: Android flag detection

Open sky scenario : $\Delta\varepsilon_{Multipath} \approx 0$

Error model : $\varepsilon_{Code} \gg \varepsilon_{Phase}$

Ionospheric variation: $\frac{d}{dt}(\varepsilon_{Iono}) \approx 0$

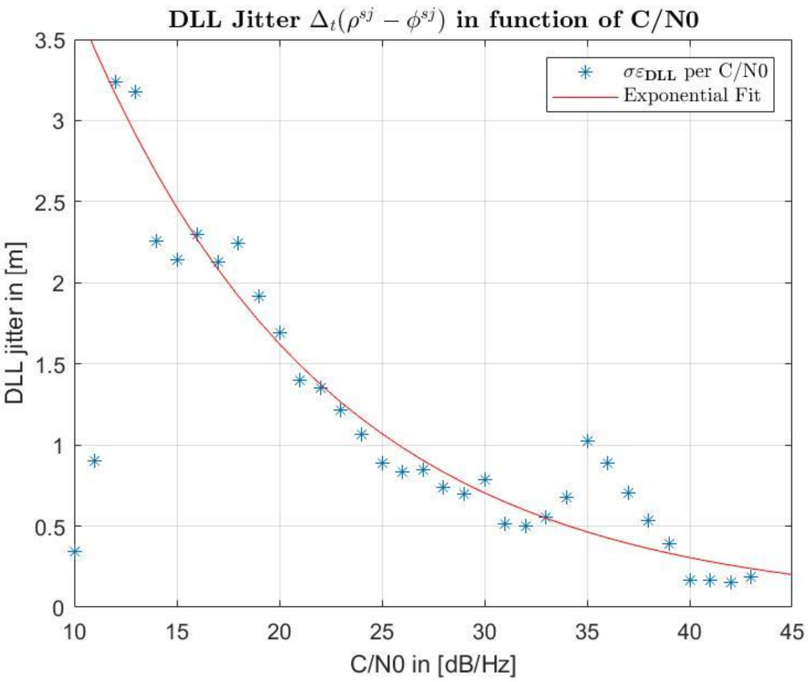


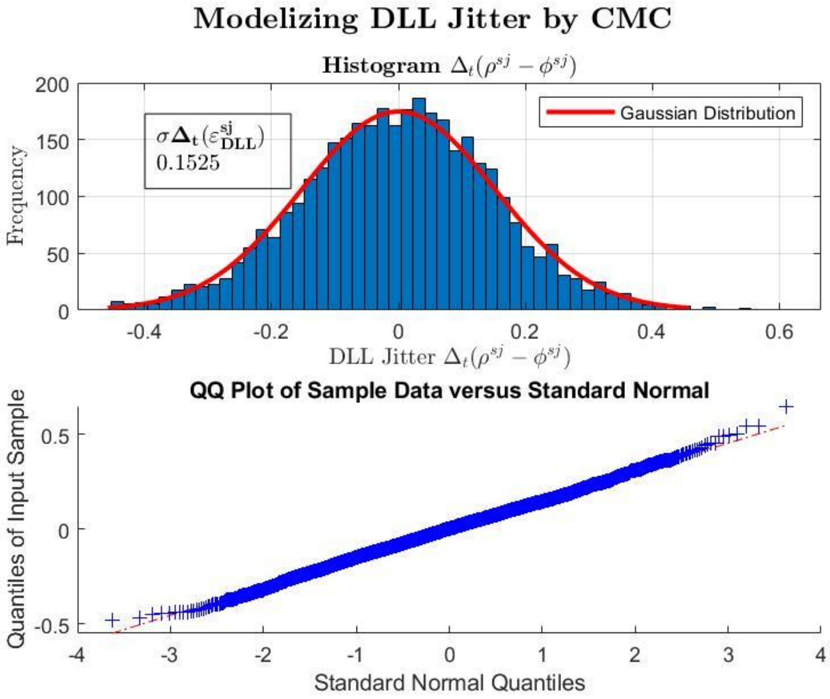
DLL Jitter for a Honor View 20

Galileo – PRN 21 – E1

II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Nominal Conditions

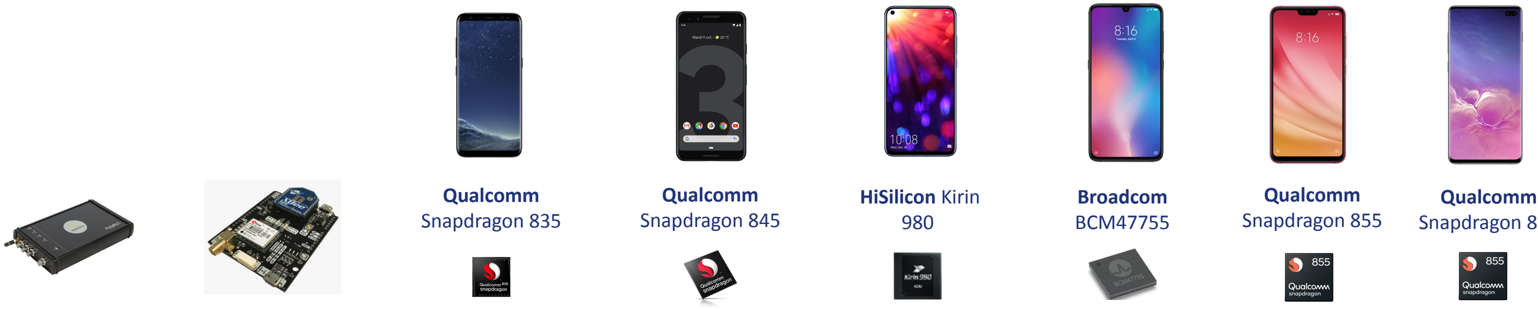




II. What kind of GNSS Receiver in a Smartphone ? - Evaluating Smartphones Measurements

Measurements Evaluation: Nominal Conditions

Receiver Clock Drift Estimation



Receiver	*Septentrio PolaRX5S**	*u-blox M8T**	*Samsung S8**	Google Pixel 3	Honor View 20	Xiaomi Mi 8	Xiaomi Mi 9	Samsung S10+
$\sigma_{DLL}[m/s]$	0.049	0.22	6.42	1.88	0.152	0.68	0.24	0.97
					0.135	1.06	2nd Phone	
						0.71	3rd Phone	

** Results obtained by Lehtola. et al [4]*

→ **Similar clock performance compare to COTS low-cost GNSS receiver**

[4] Lehtola VV, Söderholm S, Koivisto M, Montloin L. "*Exploring GNSS Crowdsourcing Feasibility: Combinations of Measurements for Modeling Smartphone and Higher End GNSS Receiver Performance*". Sensors. 2019; 19(13):3018.