



Proceeding Paper

# Treatment and Analysis of the GNSS Signal from Smartphones and Its Applicability to Urban Mobility<sup>†</sup>

Jorge Hernández Olcina \* , Ana B. Anquela Julián and Ángel E. Martín Furones

Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría, Universidad Politécnica de València, 46021 Valencia, Spain; anquela@upv.edu.es (A.B.A.J.); aemartin@upv.edu.es (Á.E.M.F.)

\* Correspondence: jorgeho1995@gmail.com

<sup>†</sup> Presented at the IV Conference on Geomatics Engineering, Madrid, Spain, 6–7 July 2023.

**Abstract:** High-precision GNSS algorithms have a complexity that requires a high computational load, so not all mobile devices will be capable of calculating position through their use, and without compromising their efficiency and wasting battery. The development of an app will provide the opportunity to research the possibility of controlling 100% of the raw data from the GNSS sensor from smartphones. Oriented to “cloud computing”, since the main objective is to provide a real-time computing tool to achieve a navigation solution, we use GNSS algorithms for this purpose, and thus avoid the computational load on the smartphone.

**Keywords:** GNSS; smartphone; cloud computing



**Citation:** Olcina, J.H.; Julián, A.B.A.; Furones, Á.E.M. Treatment and Analysis of the GNSS Signal from Smartphones and Its Applicability to Urban Mobility. *Environ. Sci. Proc.* **2023**, *28*, 1. <https://doi.org/10.3390/environsciproc2023028001>

Academic Editors: María Belén Benito Oterino, José Fernández Torres, Rosa María García Blanco, Jorge Miguel Gaspar Escribano, Miguel Ángel Manso Callejo and Antonio Vázquez Hoehne

Published: 18 December 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The research focuses on the development of an app that allows for studying, complementing, expanding and introducing different analyses of the GNSS signal received on smartphones, in terms of precision and accuracy, with special emphasis on the signal/noise ratio of the signal itself.

To do this, the development of our own app allows us to control 100% of the raw data from the GNSS sensor (raw data) from smartphones. The application is oriented to “cloud computing”, since the main objective is not only to be able to capture the data for its subsequent post-processing, but also to provide a real-time computing tool for achieving a navigation solution; we use GNSS algorithms for this purpose, thus avoiding the computational load on the smartphone, since it will be carried out on a server that is more prepared for it.

Regarding the applicability of the research, referring to urban mobility and smart cities, GNSS data from mobile devices allow mobility studies to be carried out either in situations in which behaviour is common, or in atypical situations, performing continuous and precise spatial monitoring, for example, to analyse the traffic situation in terms of driving speeds, the load on the road network, the distribution of trips, public transport networks, traffic accidents, etc. The implementation of such a solution would mean that citizens agree to upload their raw GNSS data to the cloud and share it, which today, and for the purposes of this doctoral thesis, is unfeasible.

To summarise, the objectives of the research are the analysis of the GNSS signal received on a smartphone in terms of precision and accuracy, with special emphasis on the signal/noise ratio and the generation of a GNSS data storage and computing platform in the cloud.

## 2. Background and Current Status

The development of low-cost GNSS chips prompted a revolution in positioning and navigation devices. The accuracies provided by GNSS receivers in smartphones enabled

a wide range of location-based services including social media, vehicle tracking, weather services, etc. [1].

During the “I/O 2016” conference held in May 2016, Google announced that raw GNSS measurements from smartphones and tablets running the Android N operating system (“Nougat” = version 7) will be made available to developers [1,2]. On the other hand, in May 2018, the first multi-constellation and double-frequency mobile phone (Xiaomi MI 8, Xiaomi Corporation, Beijing, China) [3] was launched on the market. These two milestones have marked a new era in positioning and navigation. The implications of these releases are important for the community, as they will allow us to move away from the GNSS receiver black box concept that provides metric-level accuracies, and will open up the possibilities of using pseudorange, Doppler, and carrier-phase measurements to obtain more accurate positions [1].

All of these factors are pushing today’s receiver technology to its limits, due to the high demand for perfect, precise, and safe positioning. In this context, the arrival of new applications is of great importance, where their size must be reduced, their energy consumption must be low, and their computational capacities are limited by the users’ devices, which generates serious concerns when implementing GNSS data processing tasks [4,5].

For this reason, the need to solve this problem in an efficient way arises, and with it, the idea of using cloud computing [4–7]. Through cloud computing, the problems of GNSS data treatment and processing in the receivers themselves are solved, as well as a reduction in energy consumption in them, since they are usually associated with batteries that run out when handling a greater computational load [4].

### 3. Material and Methods

For the complete analysis of the GNSS signal in mobile devices, our own mobile application is being developed, which allows testing and analysis in different scenarios. For the development of the application, the Google SDK known as Flutter (using Dart programming language) is being used, which has many plugins. The cloud computing part, in terms of data management and communication between application and server, is being carried out using the Node.js framework; regarding the computation of the GNSS data, the open-source library RTKLib is being used.

With all this, different new scenarios will be considered, where new executions will be carried out, either in post-processing or in real time, thus being able to determine new parameters that allow for improvements in the precision and accuracy of GNSS positioning in mobile devices.

In addition, these scenarios will be oriented towards urban mobility and smart cities, seeking their applicability, especially in urban services that can improve the quality of mobility in cities.

#### 3.1. Material

##### 3.1.1. Hardware

Google Pixel 7 Pro (Google, Mountain View, CA, USA): A smartphone selected for app development and testing. The Google device offers the possibility and ease of accessing all the GNSS sensor data without any type of hardware restriction.

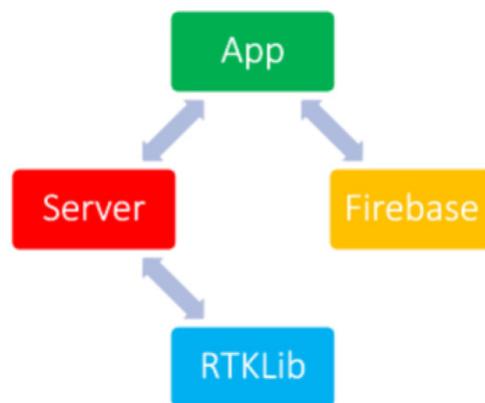
##### 3.1.2. Software

- Flutter: An open-source mobile app created by Google SDK. It is often used to develop user interfaces for Android, iOS, and Web applications, as well as being the primary method for creating applications for Google Fuchsia.
- NodeJS: A cross-platform, open-source, server-layer runtime environment based on the JavaScript programming language; it is asynchronous, with data I/O in an event-driven architecture and based on Google’s V8 engine.

- RTKLib: Open-source software package for standard and accurate positioning with GNSS (global navigation satellite system). RTKLIB consists of a portable program library and various APs (application programs) that use the library.

### 3.2. Methods

As depicted in Figure 1, the app (green) takes the data from the GNSS sensor and sends them through a WebSocket connection to the server (red), where the data are captured and converted to the RTCM3 format for subsequent delivery to the RTKLib software (blue). The server oversees launching the execution of RTKLib, sending the raw data and the navigations, and when RTKLib generates the solution, it is sent back to the server, which collects it and sends it to the smartphone. The smartphone stores the received data in plain text files and stores them in the Firebase database (orange). Within the different positioning modes that RTKLib offers, the possibility of using single, kinematic and PPP is raised; this may allow us to analyze the performance of the different modes, and identify if it is possible to obtain said solutions in RT from GNSS data from smartphones.

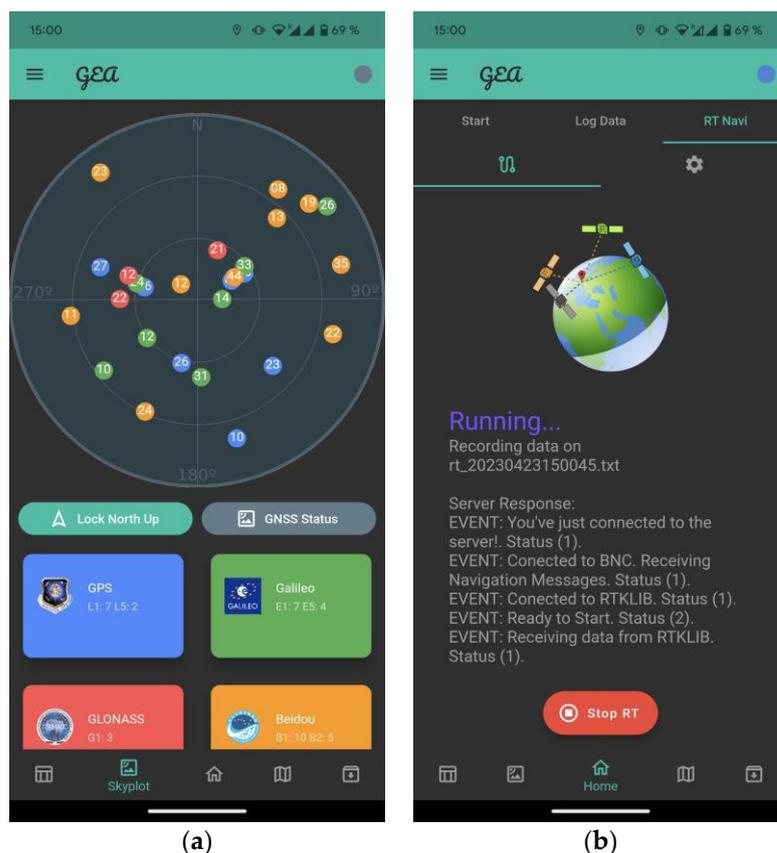


**Figure 1.** App/server structure. The figure shows the different parts that make up the structure of the project, as well as the relationships between them.

## 4. Results

Currently, the mobile application is in the development phase, where two parts will be differentiated:

1. Data capture and dumping data into plain text files: Currently, the application already has the possibility of capturing and storing GNSS data for their subsequent post-processing as shown in Figure 2. In addition, the user interface has been improved so that all the executions that are carried out are stored in the database; thus, in the case of phone data loss, said executions can be recovered, thus also generating an information store that can be used in the future to establish clear parameters on the applicability of the solution.
2. Raw data processing in real-time: The development of this part of the application is being finished. Currently, on the client side, a first version of the interface for sending data to the computing server has already been defined and implemented. For this, a connection has been implemented through websockets, creating an information tunnel between the user and the server, which will remain open until the server and/or the client close the connection to request information in real time. The raw data from the sensor are converted to an internal NMEA-type format, thus generating a message for each of the observables (satellites) received by the sensor. On the server side, the implementation of the data reception part through websockets is being completed, as well as the connection between the web server and the GNSS data processing application (RTKLIB). For this, the transformation of raw GNSS data to the standard RTCM3 format has been defined as an interface.



**Figure 2.** App interface examples: (a) GNSS status tab, showing the sky status; (b) Real-time launcher tab, showing the status of the connection with the server when an RT execution is running.

**Author Contributions:** J.H.O. conceptualized and designed the study, outlining the framework for the Android app. J.H.O. led the programming efforts, developing the core functionalities of the app and focused on the theoretical underpinnings of the raw data conversion process. A.B.A.J. and Á.E.M.F. conducted extensive literature reviews and contributed to the methodology section. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Informed Consent Statement:** Informed consent was obtained from all subjects involved in the study.

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Banville, S.; Diggelen, F.V. Precision GNSS for everyone: Precise positioning using raw GPS measurements from Android smartphones. *GPS World* **2016**, *27*, 43–48.
2. European Global Navigation Satellite System (GSA). *Using GNSS Raw Measurements on Android Devices: Towards Better Location Performance in Mass Market Applications*; Publications Office of the European Union: Luxembourg, 2017. [[CrossRef](#)]
3. Robustelli, U.; Baiocchi, V.; Pugliano, G. Assessment of Dual Frequency GNSS Observations from a Xiaomi Mi 8 Android Smartphone and Positioning Performance Analysis. *Electronics* **2019**, *8*, 91. [[CrossRef](#)]
4. Lucas-Sabola, V.; Seco-Granados, G.; Lopez-Salcedo, J.A.; Garcia-Molina, J.A.; Crisci, M. Cloud GNSS receivers: New advanced applications made possible. In Proceedings of the 2016 International Conference on Localization and GNSS (ICL-GNSS), Barcelona, Spain, 28–30 June 2016; IEEE: New York City, NY, USA, 2016. [[CrossRef](#)]
5. García-Molina, J.A.; Parro, J.M. Cloud-based GNSS Processing of Distributed Receivers of Opportunity: Techniques, Applications and Data-collection Strategies. In Proceedings of the 6th International Colloquium on Scientific and Fundamental Aspects of GNSS/Galileo, Valencia, Spain, 25–27 October 2017.

6. Favenza, A.; Rossi, C.; Pasin, M.; Dominici, F. A Cloud-Based Approach to GNSS Augmentation for Navigation Services. Proceedings of 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC), London, UK, 8–11 December 2014; IEEE: New York City, NY, USA, 2015; pp. 489–490. [[CrossRef](#)]
7. Lucas-Sabola, V.; Seco-Granados, G.; López-Salcedo, J.A.; García-Molina, J.A.; Hein, G.W. GNSS IoT Positioning: From Conventional Sensors to a Cloud-Based Solution. Inside GNSS, 15 June 2018. Available online: [http://spcomnav.uab.es/docs/journals/Lucas\\_InsideGNSS2018.pdf](http://spcomnav.uab.es/docs/journals/Lucas_InsideGNSS2018.pdf) (accessed on 5 June 2023).

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