



CroCoord v1.0—An Android Application for Nearly Real-Time 3D PPP Multi-frequency Multi-GNSS Static Surveying Using Smartphones

Marta Pokupić¹ (✉) , Matej Varga² , Zvonimir Nevistić² , Marijan Grgić² ,
and Tomislav Bašić² 

¹ Bilogorska 20, Vukosavljevica, 33404 Špišić Bukovica, Croatia
marta.pokupic95@gmail.com

² Faculty of Geodesy, University of Zagreb, Zagreb, Croatia
{matej.varga, zvonimir.nevistic, marijan.grgic,
tomislav.basic}@geof.unizg.hr

Abstract. The hardware of new-generation smartphones enables navigation, positioning and surveying (NPS) using Global Navigation Satellite Systems (GNSS). Moreover, unprecedented progress has been made in the last few years related to the accuracy and reliability of smartphone NPS giving three significant changes: (1) accessing raw code and carrier GNSS measurements; (2) collecting multi-frequency measurements and; (3) collecting measurements from multiple global satellite navigation constellations (GPS, GLONASS, Galileo, BeiDou-3). We developed CroCoord v1.0—an Android application for nearly real time PPP multi-GNSS static positioning and surveying. An application offers increased accuracy compared to other solutions given by smartphone applications, as it uses PPP GNSS post-processing method to correct collected multi-frequency multi-GNSS measurements for several sources of errors, including ephemeris, ionosphere, troposphere, satellite clocks, etc. The positional accuracy of static surveying using smartphones (including also our application) was tested on one control test site with three GNSS processing methods (SPP, PPP, and DGNSS). The positional accuracy of CroCoord in PPP mode is below 1 m and may even reach 0.5 m for an observation period longer than 1 h. The positional accuracy of CroCoord is higher for 65% compared to the accuracy obtained by default processing routines of android smartphone. Furtherly, the combination of smartphone and permanent GNSS station data (a network-DGNSS mode) reached the possibility to obtain the accuracy of 9 cm for 2-h static session. Our results confirm the enormous potential of smartphone multi-frequency multi-GNSS measurements in all civilian and professional applications which have benefits from increased positional accuracy.

Keywords: CroCoord android application · Multi-frequency · Multi-GNSS · Positioning · Post-processing · Raw GNSS measurements

1 Introduction

The Global Navigation Satellite System (GNSS) provides the position, velocity, and time (PVT) for any point in a global reference 3D coordinate system by using radio signals transmitted from satellites orbiting around the Earth [1]. Currently, there are four full constellations of global navigation satellites: GPS (NAVigational Satellite Time and Ranking Global Positioning System), Russian GLONASS (rus. GLObal'naya NAVigatsionnaya Sputnikovaya Sistema), Galileo and BeiDou-3. These four global navigation systems, supplemented by one fully operational Regional Navigation Satellite System (RNSS), the Japanese Quasi-Zenith Satellite System (QZSS), and augmentation systems (such as EGNOS), form the so called multi global navigation satellite system multi-GNSS. In the past few years, thanks to the technological advancement of hardware, newer generations of smartphones have been shown to achieve a positioning accuracy of several decimetres in nearly real time applications. Moreover, a number of studies have been performed to verify the positioning accuracy with these smartphones for different purposes [2–11].

However, until 2017 all Android smartphones have used the signal on one frequency band only (L1, code pseudoranges), and most often have received signals only from GPS satellites. The accuracy of such absolute positioning is in the range of 5–15 m. The first dual-frequency smartphone to receive signals within the L1/E1 and L5/E5 bands is the Xiaomi Mi 8 that was released in May 2018. Since then, new dual-frequency smartphones have been constantly appearing on the market and may now receive L5 band of the GPS system with a median frequency of 1176 MHz, or on the E1 and E5 band of the Galileo system with a median frequency of 1575.42 and 1191.795 MHz. To date, there are 41 dual-frequency smartphone models on the market, from 10 manufacturers [12]. The first commercial chip to have this ability is the BCM47755, which can simultaneously receive the following signals: GPS (L1-C/A and L5), GLONASS (L1), BeiDou-2 (B1), QZSS (L1 and L5) and Galileo (E1 and E5a). The second chip Quectel LC79D integrated in smartphones with the ability to obtain measurements on multiple frequencies was designed in the late 2019 [13]. Chip and smartphone manufacturers are very aware of the capability that dual-frequency offers in terms of positioning accuracy for location based services. The possibility to obtain measurements on second frequency has several advantages, for example: increased reliability to users—if one of the frequency bands fails, the other can be used as backup; reduced signal acquisition time and improved accuracy of positioning and timing; reduced problems caused by obstructions such as buildings and other obstacles; reduced problem to multipath errors; the possibility to eliminate ionospheric refraction [12].

Another improvement in smartphone positioning is related to the open access of users to raw GNSS measurements, instead of getting “black-box” coordinates calculated by internal hardware and software of smartphones. Currently, the user may obtain actual measurements in the form of clock values and can use them for computation of code pseudoranges or carrier-phases by itself. Furthermore, one has access to additional data, such as SNR (Signal to Noise Ratio) estimates, Doppler and navigation messages. By making raw GNSS measurements available, users have full control over processing and obtaining coordinates, and may use advanced processing techniques such as optimizing multiple GNSS solutions, selecting satellites, frequencies, etc. Developers are able to leverage the access to raw GNSS measurements and the enhanced accuracy offered by

this dual-frequency capability to create new applications requiring high accuracy robust positioning.

In a broader sense, satellite positioning is performed by using a large number of surveying methods and techniques, which depend on the user's needs for precision and speed. Positioning can be divided into absolute and relative. Absolute positioning is a method of determining the coordinates of a single receiver, while at relative positioning a minimum of two or more receivers, are used at the same time. Furthermore, positioning can be static or kinematic, depending on whether the receiver is moving or not. The collected measurements can be based on the Code modulation or Phase modulation of the carrier wave and processed in real time or in post-processing [1]. In this paper, the emphasis will be on static measurements with three methods: Single Point Positioning (SPP), Precise Point Positioning (PPP) and differential GNSS (DGNSS). SPP uses broadcast satellite orbits and clocks, and no other corrections except for ionosphere in case measurements contain multiple frequencies. PPP is a method of absolute static positioning of a single receiver in real time, where various satellite corrections and ephemeris are applied. It uses precise orbits and clocks instead of the broadcast message. The precondition for PPP method is to have a GNSS receiver (in this case a smartphone) which can collect multi-frequency (MF) measurements. PPP method is used in everyday practice by the number of users, civilians and geoscientists in applications such as smart cities, augmented reality applications, construction, sports, transportation, etc. DGNSS/RTK technique uses a base or reference receiver at a known location (such as from another GNSS device in surrounding or permanent GNSS station) collecting GNSS data at the same time as a receiver at an unknown surveyed location.

2 Raw GNSS Measurements

Raw GNSS measurements which user gets from the chip of Android smartphones are accessed by the Application Program Interface (API) which is a set of routines, protocols, and tools for building software programs and applications. Raw measurements include: (i) clock values of the receiver (smartphone) and their deviation, (ii) clock values of satellites, accumulated delta ranges (number of whole waves of the signal carrier), (iii) bits of navigation messages, and (iv) other metadata [2]. Raw GNSS measurements can be accessed through Android 7.0 and later versions with an API Level 24, which contains *GnssClock*, *GnssNavigationMessage*, and *GnssMeasurement* Java classes. The *GnssClock* and *GnssMeasurements* classes contain methods that provide the data necessary to calculate code pseudoranges and carrier phases from clock values of the smartphone and the satellite. Methods from the *GnssNavigationMessage* class provide a navigation message for all satellite constellations. API also includes Android classes *OnNmeaMessageListener* and *Location*. *OnNmeaMessageListener* class, using *onNmeaMessage* method, retrieves NMEA information about clock values of GNSS satellites. NMEA stands for National Marine Electronics Association, which defines and controls the NMEA 0183 standard for communicating with marine electronic devices such as transmitters, sonar, gyroscope, autopilot, GPS receiver, etc. [14]. The *Location* class contains methods which provide the location and associated metadata.

Java code for all methods is available freely and developers can use them in building Android applications [15]. We used them in developing CroCoord application in order to build the function which performs collection (surveying) of GNSS measurements [2].

2.1 Smartphone Hardware Design

The quality of GNSS data, obtained by a smartphone depends on several hardware components, mainly the GNSS chip. Smartphone hardware is designed in the way that it saves battery power, minimizes Time To First Fix (TTFF), and improves accessibility and position continuity. Smartphone design influences the selection of hardware components and software algorithms. Hardware consists of low-cost components; for positioning and navigation the most important are the antenna and oscillator. An oscillator is located outside GNSS chip and compensates temperature variations which affect the stability of the frequency. An alternative, low-energy crystal oscillator controls the time when GNSS is off with an accuracy degradation of 6 s per week. The antenna used in smartphones is called PIFA (Planar Inverted-F Antenna) which is a laminated copper plate on which all components are soldered, mechanically fixed and electrically connected. Because of its linear (instead of circular) polarization and directionality of the radiation diagram, it results in a decrease of radio signal strength by several dB. The position of the antenna depends more on the design of the smartphone than on the signal constraints, and for this reason, the interaction with the user's hand further weakens the reception of the GNSS signal. The smartphone antenna is subject to multiple reflections and its signal-to-noise ratio is reduced by about 10 dB/Hz compared to the geodetic receiver [4, 5, 16, 17].

3 CroCoord Android Application

Several Android GNSS data processing applications are available for static positioning and geodetic surveying using smartphones. They all may provide coordinates from collecting GNSS measurements using SPP. However, CroCoord Android application was developed with an idea of improving the accuracy by using advanced post-processing of raw multi-frequency multi-GNSS measurements compared to other applications. Positioning with CroCoord is based on the PPP method. The main result of processing collected raw GNSS measurements are coordinates of surveyed points in WGS84 reference coordinate system, and in official reference coordinate systems of the Republic of Croatia (HTRS96, HDKS, HVRS71). Results are available in nearly real time, because collected raw measurements take few seconds to process on the external server. Our application was developed in Android studio, its entire functionality is written in Java programming language, whereas the design is in XML format. Application logo (see Fig. 1) and the name itself, "Cro" from "Croatian" and "Coord" from "Coordinates", refer to the suitability of the application for positioning and surveying in Croatia. The CroCoord methodology opens up the possibility of simply extending the application to other countries, but with the need for additional data such as geoid models (to obtain orthometric height of surveyed point) and transformation parameters for transformation from the epoch of measurement to the epoch of the country-specific coordinate reference frame.



Fig. 1. CroCoord application logo

For full functionality of our application only requirements are internet access and permission to collect satellite positioning signals. Currently, source codes of the application are not available due to development. However, Android application executable files can be obtained at any time after the request of the corresponding author, followed by instructions for use. Also, the server is not active all the time, it must be started by author when planning to use the application.

3.1 Application Flowchart

CroCoord application runs as follows:

1. User starts application and collects single or multi-frequency multi-GNSS measurements via GUI for any selected point and session duration. The application uses all Java classes and methods from Sect. 2.
2. Collected measurements are pushed (uploaded) to the synchronization server via mobile internet.
3. The client downloads collected measurements from the server.
4. RTKLIB GNSS post-processing software processes data (calculates coordinates and other metadata).
5. Client uploads processed results back to the synchronization server.
6. User pulls (downloads) results from the synchronization server.
7. GUI displays downloaded results in the form of point coordinates and position of users on the map.
8. Results and all metadata are saved in the internal memory of a mobile phone as a CSV file.

The CroCoord GUI consists of three tabs: *Points*, *Map*, and *Info* (see Fig. 2). On the *Points* tab, the user enters the name of the point and its description (such as a “polygon point”, “tree”, “curb”, etc.). The user selects the duration of the session using *Timer* button. The surveying of a point starts by pressing the *Start* button and is stopped either after expiration of time set by timer, or after pressing the *Stop* button. After the user stops the collection of measurements, collected data are sent to the synchronization server, as a file under its own ID, using the *sendPostRequest* method.

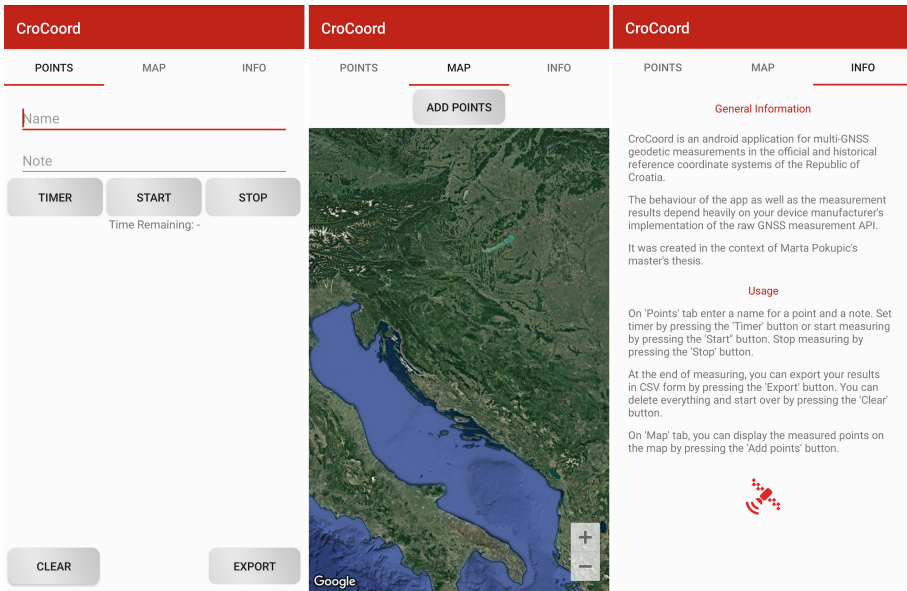


Fig. 2. CroCoord application interface

The server then returns the processing results (response time under 10 s) and the application retrieves them by using the *sendGetRequest* method based on the measurement ID.

Some processing results are displayed on the *Points* tab in the form of coordinates (Easting, Northing) in HTRS96/TM (Transverse Mercator) projection and orthometric height of point in the HVR571 vertical reference system. On the *Map* tab user can add the position of a measured point to a map view by clicking the *Add points* button, which calls the *addPointsToMap* method (see Fig. 3). All measured points, as well as those that are not displayed in the interface, can be stored on internal memory in CSV format and shared by pressing the *Export* button. The entire process of using a CroCoord application, i.e. how to use it and position with it, is described in the third *Info* tab.

3.2 Raw GNSS Measurements of Android Application

By pressing the *Start* button, the application starts simultaneously collecting raw multi-GNSS measurements by using the *Location*, *GnssNavigationMessage*, *OnNmeaMessageListener*, *GnssClock* and *GnssMeasurement* Java classes methods, and writing raw GNSS measurements into a new “rawData” text string using the *startNewLog* method. When completed, the application sends the file with measurements to the synchronization server in *.txt format. Raw GNSS data are just numeric values that cannot be directly processed in GNSS measurement software packages, but this problem is solved by the client that converts the measurement file to RINEX v2 format.

After CroCoord application sends collected measurements to the *synchronization server*, the measurements are being processed on the desktop computer, so in addition

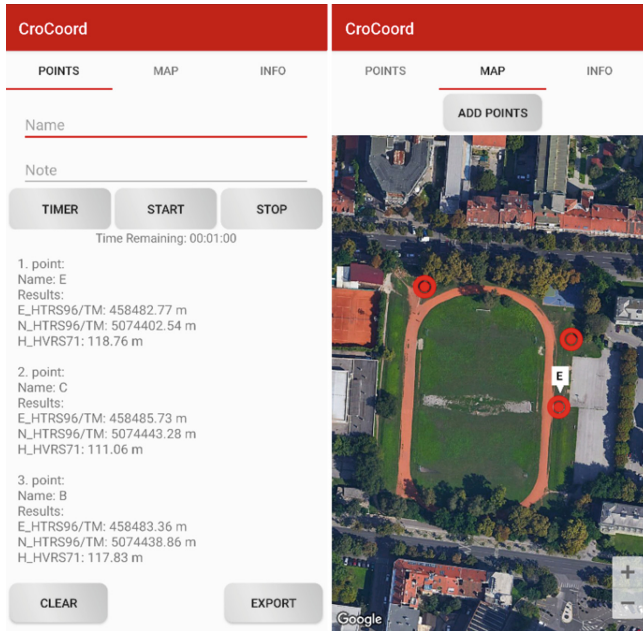


Fig. 3. Surveying results displayed

to the Android application, there have been created program scripts that serve both as the sync server and program scripts to process the collected raw GNSS measurements. The sync server serves only as the bridge between Android GUI and the measurement processing script called *client*, which is located on a desktop computer. The computer performs post-processing of the collected measurements and calculates coordinates of measured point using RTKLIB [18]. The sync server and the client are collectively referred to as the *server* and the whole concept of application \leftrightarrow server work is illustrated in Fig. 4.

3.3 Synchronization Server

The synchronization server is a script written in the Python programming language that establishes a connection between Android application and the client-computer that performs post-processing of collected GNSS measurements. The scripts are located in a cloud through an online Heroku platform that supports several programming languages.

The sync server can be accessed from multiple devices, at any time and from different places with an available internet connection. The sync server and the client communicate through directories. The sync server stores raw GNSS measurements in measurements directory from which the client retrieves measurements, and after processing them, saves their results to the results directory. The client converts the result *.txt file into *.JSON format, adding the finished state to the file for the sync server to recognize the right file and respond to an Android application that then uses the necessary data to display it.

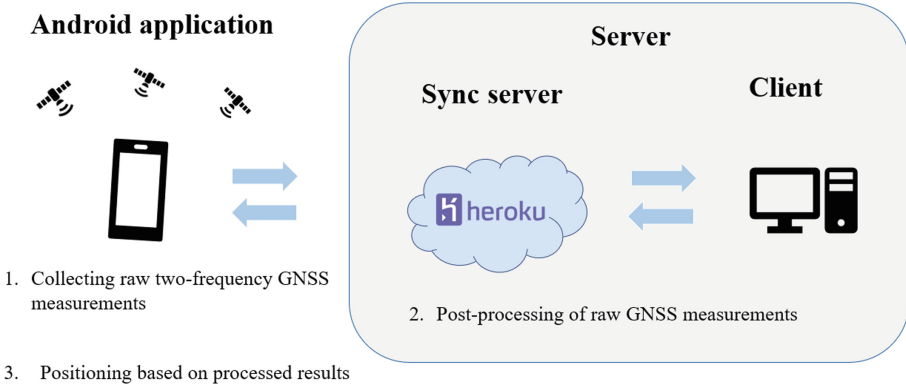


Fig. 4. Concept of application ↔ server

3.4 Client

The client is the central element of the entire process that enables CroCoord application positioning. The part of the client that communicates with the sync server is written in the Python programming language, and the scripts that process the measurement data are written in the MatLab programming language and they are located on the computer. The main MatLab post-processing script defines the names of the input files and result files, as well as directories of their downloads or storage. It also runs other scripts and packages via batch files. The whole work of the client can be shown in the following:

1. Retrieval of the *.txt file of raw GNSS measurements from the measurement directory from the sync server.
2. Conversion of *.txt file of raw GNSS measurements to RINEX v 2.11 format.
3. Registering the time of the last measurement from the RINEX file by which GPS week, GPS day of the year, GPS seconds of the week and Julian Day are calculated.
4. Broadcast ephemeris from NASA FTP are retrieved.
5. Processing RINEX measurement file by using RTKLIB software in PPP batching mode, which as a result provides coordinate series (latitude, longitude, height) for each measurement epoch.
6. Filtering outliers from the coordinate series using reliability factors (99%).
7. Calculation of definite coordinates of points and their standard deviations using the statistical function of the median set.
8. Transformation of coordinates from WGS84 (measurement epoch) to desired coordinate system (HTRS96, HDKS and HVRS71).
9. Saves the results (orthometric heights in HVRS71 and HVRS1875 vertical coordinate systems, terrestrial coordinates, \bar{E} and \bar{N} in HTRS96/TM and \bar{y} i \bar{x} in Gauss-Krüger (HDKS), and their metadata) in a *.json file to the *results* directory from where the results are loaded to the sync server.

Obviously, the central part of computing coordinate of the surveyed point is using RTKLIB in PPP mode. RTKLIB is an open source software for GNSS positioning and

analysis, which supports GPS, GLONASS, Galileo, QZSS, BeiDou and SBAS systems [18, 19]. It is known to be used in some other GNSS post-processing services, such as Rokubun Jason [20]. The main MatLab script starts running RTKLIB using a batch file in which it defines all the required parameters and settings, specifying the processing method and type of output results. Some of the settings which are defined prior to processing are: positioning method, used frequencies, elevation mask, SNR mask, Earth tidal wave corrections, dynamic models, ionospheric correction, tropospheric correction, satellite ephemeris, ambiguity solution method for different systems, ambiguity determination settings, maximum age differential, GDOP threshold, result format (coordinates), time format, height type, carrier-based measurement settings, antenna settings, etc. After processing of measurements, RTKLIB gives results in the form of the coordinate series (latitude, longitude and height) in the WGS84 reference coordinate system for each measurement epoch together with the associated standard deviations and other metadata such as a number of satellites, DOP, etc.

4 Accuracy of Static GNSS Surveying Using Smartphones

One of the biggest advantages of CroCoord, compared to the limits in accuracy of smartphone-based GNSS data processing, is its accuracy, since it uses RTKLIB for PPP processing of collected data. In order to verify it, we performed one static session on December 31st, 2019 which lasted 2 h from 09:00 to 11:00 UTC on one selected control point of the University of Zagreb Faculty of Geodesy. Collected data (from smartphone and ZAGR CROPOS (CROatian POSitioning System) permanent station), as well as additional data used in post-processing (clocks, ephemeris, EOP), may be downloaded on following permanent <https://doi.org/10.5281/zenodo.3633374>.

Reference (true) point is part of the well-stabilized geodetic network of six points used by the Faculty of Geodesy. The network is reobserved and readjusted each year with six GPS + GLONASS L1/L2 professional grade GNSS receivers using method of relative static and 20 measurement sessions lasting minimally 20 min. The reference coordinates of control point were obtained through several measurement campaigns which have lasted for more than 24 h in total. The estimated accuracy of the coordinates of control points is expected to be at few mm level which is well suitable for comparison with smartphone measurements.

The specific goal of this test was not only to verify improvement in accuracy of CroCoord application compared to other Android-based smartphone GNSS processing software, but also to investigate achievable accuracy of smartphone GNSS surveying for three positioning methods: SPP, PPP, and DGNSS. For SPP and PPP testing, only smartphone collected measurements were used, whereas the performance of DGNSS was tested using the additional measurements from permanent station ZAGREB which is a part of CROPOS GNSS network. In DGNSS method, ZAGR station was fixed with coordinates $X = 4,282,037.523$ m, $Y = 1,224,886.395$ m, $Z = 4,550,534.641$ m. Smartphone measurements on control point were collected with dual-frequency Xiaomi Mi Mix-3.

Measurements were processed using several commercial and free software packages including Trimble Business Center 5.00, Leica GeoOffice 8.4, RTKLIB, Google GNSS

analysis software, as well as with online post-processing services such as Rokubun Jason, Trimble VRX, and NRCan CSRS-PPP.

Comparison of definite plane coordinates (Easting, Northing) for each solution was performed in conformal Gauss-Krüger projection using the following parameters: $\varphi_0 = 0^\circ$, $\lambda_0 = 16.5^\circ$, linear scale 0.9999, shift in Easting 500,000 m. The accuracy of each solution was analysed using positional (plane) residuals between map projection coordinates (Easting, Northing) of two quantities: reference coordinates E , N of control point and E , N of each solution (coordinates obtained by processing of measured data):

$$\sigma = \sqrt{\Delta E^2 + \Delta N^2} = \sqrt{(E_{control} - E_{solution})^2 + (N_{control} - N_{solution})^2} \quad (1)$$

In Table 1 and on Fig. 5 we present and describe only the best solutions for each GNSS positioning method, so the results obtained with other software and measurements from other constellations are not shown. The first two solutions (S1 and S2) are obtained by using static data from the smartphone and permanent station in network-DGNSS method. The solution S1 represents the optimal solution in which coordinate of the point was obtained by using network-DGNSS method in Leica GeoOffice where both GPS and Galileo dual-frequency were exploited. The position residual between reference coordinate and this solution is 9 cm. Solution S2 is obtained in the same way as S1, but

Table 1. The results of GNSS data processing for measurement session on December 31st, 2019, from 09:00 to 11:00 UTC

Solution #	Software	Description	SPP/PPP/DGNSS	ΔE [m]	ΔN [m]	Error [m]
Reference solution (true)	TBC		Relative static			
S1	Leica Geooffice	GPS + Galileo, L1/E1/L5/E5	Network-DGNSS	-0.08	-0.03	0.09
S2	RTKLIB	GPS + Galileo, L1/E1/L5/E5	Network-DGNSS	-0.22	0.27	0.35
S3	CSRS-PPP	GPS L1/E1/L5	PPP	-0.31	-0.33	0.45
S4	RTKLIB	GPS + Galileo, L1/E1/L5/E5	PPP	-0.01	0.73	0.73
S5	Google GNSS analysis tools, Android-default	GPS L1	SPP	-0.15	2.12	2.12

Static measurements were performed using smartphone on control point. **S4** is the solution obtained by CroCoord application, whereas **S5** is the solution obtained in smartphone by its default routines from Google GNSS analysis tools

RTKLIB was used. The positional residual is 35 cm, which is almost 4 times worse than solution S1. Leica GeoOffice as commercial software seems to be more accurate than RTKLIB, even if RTKLIB has more customizability than Leica Geofice.

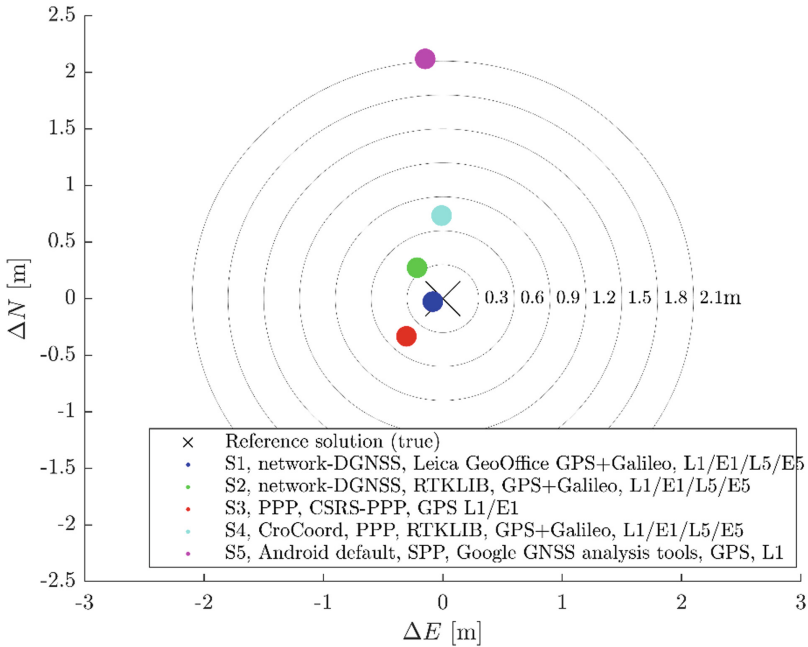


Fig. 5. The results of GNSS data processing for measurement session on December 31st, 2019, from 09:00 to 11:00 UTC

The last three solutions are the analysis of SPP or PPP solutions using three software. S3 solution, obtained from Canadian CSRS-PPP online post-processing service, is the most optimal of these three compared. The solution used only GPS measurements on two frequencies. The positional residual is 45 cm, which is nearly the same as S2 solution in network-DGNSS method. PPP using RTKLIB resulted in positional residual of 73 cm. This result shows that we can expect the positional accuracy of CroCoord application to be around ± 1 m for shorter and medium observation times (<1 h), and sub-meter accuracy for sessions longer than 1 h. Solution S5 is the SPP solution which can be expected from default smartphone GNSS positioning applications which do not take care of any corrections and usually use only GPS on one-frequency. The positional residual is 2.1 m for our two-hour long session. The accuracy of such positioning might be in the range between 2 and 5 m. The comparison between S4 and S5 solutions is important as it shows that the positional accuracy of CroCoord is supposedly to be higher for around 65% compared to other android GNSS applications.

Definite coordinates of all solutions are given in Table 2.

Table 2. Definite coordinates of all solutions

Solution #	Geodetic on GRS80			Cartesian geocentric on GRS80			Plane	
	φ [°]	λ [°]	h [m]	X [m]	Y [m]	Z [m]	E [m]	N [m]
Reference solution (true)								
S1	45.8082391	15.9658304	164.835	4,281,943.22	1,225,064.34	4,550,534.07	497,343.982	5,074,772.381
S2	45.8082389	15.9658293	165.038	4,281,943.40	1,225,064.30	4,550,534.20	497,343.899	5,074,772.355
S3	45.8082416	15.9658276	163.927	4,281,942.49	1,225,063.90	4,550,533.61	497,343.766	5,074,772.655
S4	45.8082361	15.9658265	164.794	4,281,943.51	1,225,064.10	4,550,533.81	497,343.677	5,074,772.048
S5	45.8082457	15.9658303	163.500	4,281,941.83	1,225,063.93	4,550,533.63	497,343.972	5,074,773.115
	45.8082582	15.9658285	175.920	4,281,949.23	1,225,065.90	4,550,543.50	497,343.834	5,074,774.500

5 Conclusion

In order to explore the potential of smartphones which may collect raw multi-frequency multi-GNSS measurements, we developed CroCoord v1.0 application which provides the position of surveyed points in WGS84, and official geodetic reference coordinate systems of the Republic of Croatia. We debugged and verified application for completeness and consistency. The application can be used to collect static multi-frequency GNSS measurements from all global satellite navigation systems and for any session duration. The speed of the application is nearly real time (user gets results in the smartphone almost immediately after the end of measurements) and is limited solely by the internet connection needed for the transfer of the data to and from the server and time needed for the processing of GNSS measurements. It is possible to replace RTKLIB as GNSS post-processing tool with any other software having the option for simultaneous batch processing.

We verified that CroCoord application has greater accuracy, using 2-h long static session and SPP/PPP processing method, CroCoord yielded 65% more accurate results compared to other android based GNSS processing algorithms and software (0.73 m with CroCoord compared to default smartphone solution with 2.12 m accuracy), as it exploits benefits of more sophisticated software RTKLIB running on an external server. The analysis of our comparison non-related to CroCoord showed that on-line GNSS post-processing service such as CSRS-PPP also has potential for processing of mobile-phone data in PPP mode as it outperformed the accuracy of RTKLIB processing for 38% (from 0.45 m with CSRS-PPP to 0.73 m with RTKLIB). Finally, our test showed high potential for the usage of corrections and GNSS data from surrounding permanent GNSS stations in DGNSS, where even sub-dm positional accuracy for longer observation periods can be expected.

The executables of application can be obtained upon the request to the corresponding author, after which instructions for usage will be given. Authors are still working on developing and further improving the reliability of the post-processing, and to extend the capabilities of the application regarding the choice of parameters and positioning methods.

Note

Measurements data can be downloaded from following link <https://doi.org/10.5281/zenodo.3633374>. Data include measurements from smartphone (in android-based format, Rinex 2.11 and 3.03), permanent station ZAGR (in Rinex 2.11 and 3.03), along with all supplementary data authors used in processing such as broadcast and precise ephemerides (in Rinex 2 and MGEX).

References

1. Zrinjski, M., Barković, Đ., Matika, K. (2019). Razvoj i modernizacija GNSS-a. *Geodetski list*, 73(1), 45–65.
2. Pokupić, M. (2019). CroCoord - Android application for multi-GNSS surveying (Master's thesis). (<https://www.bib.irb.hr/1037146>).

3. Dabove, P., Di Pietra, V. (2019). Towards high accuracy GNSS real-time positioning with smartphones. *Advances in Space Research*, 63(1), 94–102.
4. Humphreys, T. E., Murrian, M., Van Diggelen, F., Podshivalov, S., Pesyna, K. M. (2016). On the feasibility of cm-accurate positioning via a smartphone's antenna and GNSS chip. In 2016 IEEE/ION Position, Location and Navigation Symposium (PLANS) (pp. 232–242). IEEE.
5. Pesyna Jr, K. M., Heath Jr, R. W., Humphreys, T. E. (2014). Centimeter positioning with a smartphone-quality GNSS antenna. In *Radionavigation Laboratory Conference Proceedings*.
6. Masiero, A., Guarnieri, A., Pirotti, F., Vettore, A. (2014). A particle filter for smartphone-based indoor pedestrian navigation. *Micromachines*, 5(4), 1012–1033.
7. Wang, L., Li, Z., Zhao, J., Zhou, K., Wang, Z., Yuan, H. (2016). Smart device-supported BDS/GNSS real-time kinematic positioning for sub-meter-level accuracy in urban location-based services. *Sensors*, 16(12), 2201.
8. Wang, L., Li, Z., Yuan, H., Zhao, J., Zhou, K., Yuan, C. (2016). Influence of the time-delay of correction for BDS and GPS combined real-time differential positioning. *Electronics Letters*, 52(12), 1063–1065.
9. Adjrad, M., Groves, P. D. (2018). Intelligent urban positioning: Integration of shadow matching with 3D-mapping-aided GNSS ranging. *The Journal of Navigation*, 71(1), 1–20.
10. Al-Azizi, J. I., Shafri, H. Z. M. (2017). Performance evaluation of pedestrian locations based on contemporary smartphones. *International Journal of Navigation and Observation*, 2017.
11. Fissore, F., Masiero, A., Piragnolo, M., Pirotti, F., Guarnieri, A., Vettore, A. (2018). Towards Surveying with a Smartphone. In *New Advanced GNSS and 3D Spatial Techniques* (pp. 167–176). Springer, Cham.
12. European GSA, dual-frequency. Available online: <https://www.gsa.europa.eu/newsroom/news/market-understands-value-dual-frequency> (accessed on 30 January 2020).
13. Quectel LC79D chip. Available online: <https://www.quectel.com/infocenter/news/563.htm> (accessed on 30 January 2020).
14. OnNmeaMessageListener Java class. Available online: <https://developer.android.com/reference/android/location/OnNmeaMessageListener.html> (accessed on 27 January 2020).
15. Android location API. Available online: <https://developer.android.com/reference/android/location/package-summary> (accessed on 27 January 2020).
16. Van Diggelen, F. S. T. (2009). *A-GPS: Assisted GPS, GNSS, and SBAS*. Artech House.
17. European GNSS Agency (GSA). Using GNSS raw measurement on android devices. https://www.gsa.europa.eu/system/files/reports/gnss_raw_measurement_web_0.pdf.
18. Takasu, T. (2013). RTKLIB ver. 2.4. 2 Manual. RTKLIB: An Open Source Program Package for GNSS Positioning, 29–49.
19. RTKLIB software. Available online: <http://www.rtklib.com/> (accessed on 28 January 2020).
20. Rokubun Jason software. Available online: <https://paas.rokubun.cat/> (accessed on 28 January 2020).