On the feasibility of resolving Android GNSS carrier-phase ambiguities

Jianghui Geng¹, Guangcai Li¹

1. GNSS Research Center, Wuhan University, Wuhan, China

E-mail: jgeng@whu.edu.cn; guangcai.li@whu.edu.cn

Abstract: High-precision navigation using low-cost handsets has profound potential for mass-market applications, which has been being boosted by the release of raw GNSS data from Google Android smart devices. However, integer ambiguity fixing for centimeter-level GNSS positioning is prevented by the unaligned chipset initial phase biases (IPBs) found within Android carrier-phase data. In this study, we thus investigate the temporal behaviors of those chipset IPBs using zero baselines where smart devices are linked to external survey-grade antennas, and find that the IPBs are generally stable over time as the mean standard deviation of single-epoch IPB estimates derived from continuous carrier-phase data is as low as 0.04 cycles for all satellites. Unfortunately, these chipset IPBs differ randomly among satellites and change unpredictably if carrier-phase signals are re-tracked, discouragingly suggesting that the chipset IPBs cannot be pre-calibrated or even calibrated on the fly. We therefore have to presumably correct for them in a post-processing manner with the goal of inspecting the potential of Android GNSS ambiguity resolution if hopefully the IPBs can be gone. For a vehicle-borne Nexus 9 tablet with respect to a survey-grade receiver located 100-2000 m away, we achieve the first ambiguity-fixed solution within 321 s and finally 51.6% of all epochs are resolved; the ambiguity-fixed epochs can achieve a positioning accuracy of 1.4, 2.2 and 3.6 cm for the east, north and up components, respectively, showing an improvement of 30%-80% compared to the ambiguity-float solutions. While all smart devices above are connected to external survey-grade antennas, we find that a Xiaomi 8 smartphone can be coupled effectively with a miniaturized portable patch antenna, and then achieve commensurate carrier-phase tracking and ambiguity-fixing performance to those of a commercial μ -blox receiver with its dedicated patch antenna. This is encouraging since a compact and inexpensive patch antenna paired with smart devices can promote the democratization of high-precision GNSS.

Keywords: Android smart devices; Ambiguity resolution; Unaligned chipset initial phase bias; External GNSS antennas

Introduction

The release of raw GNSS data from Google Android smart devices in 2016 has been catalyzing the innovation of high-precision positioning using the democratized low-cost handset for mass-market applications (Banville and Diggelen, 2016; Malkos 2016). In particular, mobile- and safety-related applications, e.g., multimodal logistics and mobile health diagnosis, can potentially benefit from sub-meter or better positioning precisions, while a sub-decimeter or centimeter precision available at smartphones may facilitate semi-professional tasks such as personal or crowd-source mapping (European GNSS Agency 2017; Geng, et al. 2019). Banville and Diggelen (2016) demonstrated that carrier-phase data from smart devices had the potential to provide decimeter-level or better positioning accuracy. However, they pointed out that the embedded antennas were the main obstacle to realizing this potential, and an external antenna might be a solution.

Smartphone antennas are passive ones with linear polarization rather than the right-hand polarized patterns employed by survey-grade antennas, and thus suffer from the loss of 3 dB signal power caused by polarization mismatch (Zhang et al. 2013). In this case, smart device GNSS signals are especially susceptible to multipath effects, and its carrier-to-noise ratio (C/N_0) is about 10 dB-Hz lower than that of geodetic receivers (Geng et al. 2018; Zhang et al. 2018). Even worse, the GNSS antennas of smart devices are omnidirectional rather than hemispherical, which further deteriorates the multipath contamination. Riley et al. (2017) tried a float real-time kinematic (RTK) experiment using a Nexus 9 tablet and only 1.5 m horizontal positioning accuracy could be achieved at most due to severe multipath and cycle slip effects. Even in an open roof or a rural environment, Pirazzi et al. (2017) and Realini et al. (2017) had to struggle to obtain a 0.5 m and 0.1 m horizontal positioning

accuracy for Huawei P10 and Nexus 9, respectively, not to mention ambiguity resolution which was hardly achievable using smart device antennas (cf. Pesyna et al., 2014). To overcome this problem, Geng et al. (2018) replaced the embedded GNSS antenna of a Nexus 9 tablet with an external survey-grade GNSS antenna. They successfully obtained a high signal quality comparable to that of geodetic receivers and achieved centimeter-level relative positioning accuracy with Nexus 9 carrier-phase data. However, carrier-phase ambiguities were kept float due to unmitigated chipset biases.

To be specific, Geng et al. (2018) found that a peculiar initial phase bias (IPB) was present within Nexus 9 GNSS chipsets whenever they locked carrier-phase signals of a satellite, and out of expectation, this bias differed from satellite to satellite and changed randomly if the carrier-phase signal of a particular satellite was re-tracked due to loss of phase lock. Similarly, despite the overwhelming multipath interference on embedded GNSS antennas, Riley et al. (2017) still discerned random offsets of a few tenth cycles among satellites in their ultra-short-baseline (a few decimeters) carrier-phase residuals of Nexus 9, and explained that the Nexus 9 did not deliver the true full carrier-phase, but instead an accumulated delta range with arbitrary IPBs. In the following, we name these biases as the chipset IPBs which are unaligned across tracked satellites. Actually, such biases might have also been found within other smartphone GNSS chipsets, even before Google officially released Android raw GNSS data. For example, placing a specially designed Samsung Galaxy S5 within an anechoic chamber, Humphreys et al. (2016) found drifts of its carrier-phase measurements which grew approximately linearly over time, but the drift rates differed from satellite to satellite. Therefore, even though inspecting the carrier-phase observations collected by embedded GNSS antennas only, Realini et al. (2017) and Håkansson (2018) definitively concluded that the double-difference carrier-phase ambiguities of Nexus 9 (and other similar smart devices) were not of integer nature, and only ambiguity-float solutions could be achievable ultimately.

Integer ambiguity resolution is crucial to centimeter-level GNSS positioning. The objective of this study is to investigate the characteristics of unaligned IPBs within smart device chipsets, and try to resolve single-frequency (L1) ambiguities by presumably calibrating such phase biases. We select a Nexus 9 tablet as a representative of all similar smart devices since its carrier-phase measurements are continuous without any duty-cycle issues, and notably we also replace its embedded GNSS antenna with an external survey-grade antenna to reduce multipath effects for efficient integer-cycle resolution.

Theories

Unaligned chipset initial phase biases (IPBs) across satellites

In order to generate continuous carrier-phase within GNSS receiver chipsets, both integer and fractional parts of the local replica signals from the numerically controlled oscillator (NCO) must be stored simultaneously, and the integer component of the NCO phase should be counted during the continuous carrier-phase tracking (Hauschild 2017). Phase locked loop (PLL) drives the difference between the incoming intermediate frequency (IF) phase and the total NCO phase to zero in order to achieve a phase-lock status (Misra and Enge, 2006). Suppose that the phase tracking for satellite s commences at time t^s ; once a phase-lock status is achieved and provided that the intermediate frequency $f_{IF} \neq 0$, the carrier-phase measurement can be obtained from the total NCO phase as (O'Driscoll and Petovelle, 2010)

$$\varphi_r^s(t_r) = \frac{\rho(t_r)}{\lambda} + \varphi_{r,Mix}(0) + t^s f_{IF} - \varphi_{r,NCO,IF}^s(t^s) - \varphi^s(0) + N_r^s(t_r) \quad (1)$$

where $\varphi_r^s(t_r)$ is the phase measurement in cycles by receiver r at epoch t_r ; $\rho(t_r)$ is the geometric distance between satellite s and receiver r ; λ is the signal wavelength; $\varphi_{r,Mix}(0)$ is the initial mixer phase, $\varphi_{r,NCO,IF}^s(t^s)$ is the initial value of the IF component of the NCO phase; $\varphi^s(0)$ is the initial phase at the transmitter of

satellite s ; $N_r^s(t_r)$ is the integer ambiguity.

Note that both $t^s f_{IF}$ and $\varphi_{r,NCO,IF}^s(t^s)$ depend on t^s when phase tracking begins. Therefore, they differ from satellite to satellite and will change if carrier-phase signals of a particular satellite are re-tracked. To eliminate this discrepancy, GNSS chipsets normally provide an initial value of the IF component of the NCO phase as

$$\varphi_{r,NCO,IF}^s(t^s) = t^s f_{IF} - t_0 f_{IF} \quad (2)$$

where t_0 is the epoch when the receiver is turned on. Substituting Eq. 2 into Eq. 1,

$t^s f_{IF} - \varphi_{r,NCO,IF}^s(t^s)$ becomes $t_0 f_{IF}$, which is favorably the same for all satellites and can thus be absorbed by $\varphi_{r,Mix}(0)$. This implementation requires the chipsets to store a globally accessible IF phase accumulator, and its phases of each tracked satellite are updated continuously during receiver operation, which will definitely complicate the algorithm of chipset's phase lock loop.

On the contrary, if the adverse impacts of $t^s f_{IF}$ and $\varphi_{r,NCO,IF}^s(t^s)$ are not eliminated, we have to combine them with $\varphi_{r,Mix}(0)$ to form the chipset IPB

$$\varphi_{r,t^s,0} = \varphi_{r,Mix}(0) + t^s f_{IF} - \varphi_{r,NCO,IF}^s(t^s) \quad (3)$$

which is also a function of the epoch when phase tracking begins. It is different among satellites and will change randomly if the carrier-phase of a particular satellite is re-tracked. However, it keeps constant over a continuous tracking of a particular satellite. Therefore, we refer to $\varphi_{r,t^s,0}$ as the chipset IPB.

Geng et al. (2018) found that Nexus 9 carrier-phase data exhibit such IPBs, which are different among satellites and randomly change if satellite signals are re-tracked. We speculate that such smart devices may not provide the initial value of the IF component of the NCO phase as done by geodetic receivers, or those provided does not satisfy Eq. 2. Therefore, we call such IPB the unaligned chipset IPB of smart

devices throughout.

Double-difference observations including unaligned chipset IPBs

For relative positioning on a short-baseline between receivers r and b , the satellite orbit errors, clock offsets, hardware delays and the atmospheric effects can be eliminated in the double-difference carrier-phase and pseudorange observations. For GPS, such double-difference measurements in units of meters can be

$$\begin{aligned}\Phi_{rb,i}^{jk} &= \rho_{rb}^{jk} + \lambda_i B_{rb,i}^{jk} + M_{rb,\Phi_i}^{jk} + \varepsilon_{\Phi_{rb,i}^{jk}} \\ P_{rb,i}^{jk} &= \rho_{rb}^{jk} + M_{rb,P_i}^{jk} + \varepsilon_{P_{rb,i}^{jk}}\end{aligned}\quad (4)$$

where $\Phi_{rb,i}^{jk}$, $P_{rb,i}^{jk}$ and ρ_{rb}^{jk} refer to the double-difference carrier-phase, pseudorange and geometric range between satellites j, k and receivers r, b , respectively. λ_i is the signal wavelength at frequency i , and $B_{rb,i}^{jk}$ is the double-difference carrier-phase bias including the integer ambiguity $N_{rb,i}^{jk}$. M_{rb,Φ_i}^{jk} and M_{rb,P_i}^{jk} represent the multipath effects. $\varepsilon_{\Phi_{rb,i}^{jk}}$ and $\varepsilon_{P_{rb,i}^{jk}}$ represent the observation noise.

Whenever a smart device with unaligned IPBs is involved in Eq. 4, they cannot be eliminated by double differencing. For Eq. 4, we presume that a reference geodetic receiver is used without any unaligned IPBs, while a smart device is employed as a rover. We use $\varphi_{r,t^j,0,i}$ and $\varphi_{r,t^k,0,i}$ to represent the unaligned IPBs at frequency i of the smart device r with respect to satellites j and k , respectively. The resulting double-difference IPB is then formulated as

$$\begin{aligned}\varphi_{rb,0,i}^{jk} &= \left((\varphi_{r,t^j,0,i} - \varphi_{0,i}^j) - (\varphi_{r,t^k,0,i} - \varphi_{0,i}^k) \right) - \left((\varphi_{b,0,i} - \varphi_{0,i}^j) - (\varphi_{b,0,i} - \varphi_{0,i}^k) \right) \\ &= \varphi_{r,t^j,0,i} - \varphi_{r,t^k,0,i}\end{aligned}\quad (5)$$

where $\varphi_{b,0,i}$ is the IPB of the geodetic receiver b which is the same for all satellites and therefore can be eliminated by differencing between satellites; $\varphi_{0,i}^j$ and $\varphi_{0,i}^k$ are

the IPBs within satellites j and k , respectively, which can be eliminated by differencing between stations; however, since both $\varphi_{r,t^j,0,i}$ and $\varphi_{r,t^k,0,i}$ are unaligned random values, their difference is unlikely to be an integer or zero constant, and as a result, the double-difference IPB $\phi_{rb,0,i}^{jk}$ will be absorbed by $B_{rb,i}^{jk}$, that is

$$B_{rb,i}^{jk} = \phi_{rb,0,i}^{jk} + N_{rb,i}^{jk} \quad (6)$$

which explains why in Eq. 4 we call $B_{rb,i}^{jk}$ the carrier-phase bias instead of integer ambiguity. On account of Eq. 6, we are no longer able to fix $B_{rb,i}^{jk}$ to integers straightforwardly. The IPB $\phi_{rb,0,i}^{jk}$ has to be corrected before resolving $B_{rb,i}^{jk}$.

Identification of the fractional parts of unaligned chipset IPBs

It is difficult to estimate undifferenced IPBs due to rank deficiency. Fortunately, we only need to separate double-difference IPB $\phi_{rb,0,i}^{jk}$ from the double-difference carrier-phase ambiguity $N_{rb,i}^{jk}$. Since $N_{rb,i}^{jk}$ are lumped together with the integer part of $\phi_{rb,0,i}^{jk}$, we estimate only the fractional part of $\phi_{rb,0,i}^{jk}$ while combining its integer part with $N_{rb,i}^{jk}$. The fractional part of $\phi_{rb,0,i}^{jk}$ is defined as

$$\bar{\varphi}_{rb,0,i}^{jk} = \phi_{rb,0,i}^{jk} - [\phi_{rb,0,i}^{jk}] \quad (7)$$

where $[\cdot]$ is an integer rounding operation. Based on Eq. 6, Eq. 7 now becomes

$$\begin{aligned} \bar{\varphi}_{rb,0,i}^{jk} &= (B_{rb,i}^{jk} - N_{rb,i}^{jk}) - [B_{rb,i}^{jk} - N_{rb,i}^{jk}] \\ &= B_{rb,i}^{jk} - [B_{rb,i}^{jk}] \end{aligned} \quad (8)$$

which shows how we can derive $\bar{\varphi}_{rb,0,i}^{jk}$ from the original double-difference phase bias estimate $B_{rb,i}^{jk}$. Note that we still call $\bar{\varphi}_{rb,0,i}^{jk}$ the IPB rather than the fractional IPB for brevity. In addition, to obtain satellite-specific IPBs, we assume that the IPB for a

particular satellite j is zero. Then we can compute the IPB for the remaining satellites using

$$\bar{\varphi}_{r,t^k,0,i} = \bar{\varphi}_{r,t^j,0,i} - \bar{\varphi}_{rb,0,i}^{jk} \quad (9)$$

where $\bar{\varphi}_{r,t^j,0,i}$ and $\bar{\varphi}_{r,t^k,0,i}$ are IPBs for reference satellites j and another satellite k , respectively.

Therefore, integer ambiguity resolution on the carrier-phase data from smart devices will require the calibration of IPBs, that is

$$\begin{aligned} \bar{\Phi}_{rb,i}^{jk} &= \Phi_{rb,i}^{jk} - \lambda_i (\bar{\varphi}_{r,t^j,0,i} - \bar{\varphi}_{r,t^k,0,i}) = \rho_{rb}^{jk} + \lambda_i N_{rb,i}^{jk} + M_{rb,\Phi_i}^{jk} + \varepsilon_{\Phi_{rb,i}^{jk}} \\ P_{rb,i}^{jk} &= \rho_{rb}^{jk} + M_{rb,P_i}^{jk} + \varepsilon_{P_{rb,i}^{jk}} \end{aligned} \quad (10)$$

where $\bar{\Phi}_{rb,i}^{jk}$ is the IPB-corrected double-difference carrier-phase observation in the units of meters.

Data collection and processing

Since we focus on studying the chipset IPBs and the possibility of resolving smart device carrier-phase ambiguities, we replaced the embedded GNSS antenna with an external survey-grade GNSS antenna to avoid the seriously multipath effects throughout. All smart devices and geodetic GNSS receivers were connected with a TRM57971.00 antenna using a signal splitter, as shown in Fig. 1. To be specific, the geodetic receiver Trimble NetR9 (TRI9) supplied power to TRM57971.00 through the splitter while receiving the radio frequency (RF) signals. Another geodetic receiver Trimble NetR9 (TRIA) and a Nexus 9 tablet were connected to the second and third outlets of the splitter. The RF signals were transmitted to the embedded antenna feeding point within Nexus 9. Note that although the Nexus 9 tablet was connected to an external antenna, its embedded GNSS antenna was not cut off and consequently an RF shielding box was used to prevent Nexus 9 from receiving GNSS signals using its self-contained antenna (Fig. 1). Note that all carrier-phase and pseudorange data

discussed in this study were on GPS L1 signals. We tried different versions of Geo++ RINEX (Receiver INdependent EXchange format) Logger and GnsLogger applications to obtain Android GNSS data from smart devices, as shown in Table 1.

Fig. 1 Fundamental configuration of a smart device connected to an external geodetic GNSS antenna, exemplified by a Nexus 9 tablet and a TRM57971.00 antenna. This smart device can also be a Samsung Galaxy S8 or a Huawei Honor 8 handset. A geodetic receiver Trimble NetR9 (TRIA, TRIB or TRI9) will be always connected to this TRM57971.00 antenna through a signal splitter to provide benchmark positioning solutions. During the data collection, the radio frequency (RF) shielding box was used to prevent Nexus 9’s embedded antenna from receiving GNSS signals.

Table 1 Zero-baseline, short-baseline and vehicle-borne smart device experiments collecting 1-Hz GPS raw single-frequency data. “NEX9”, “SAMS”, “HONO”, “TRI9”, “TRIA” and “TRIB” abbreviate Nexus 9, Samsung Galaxy S8, Huawei Honor 8, Trimble NetR9 (5.30), Trimble NetR9 (5.03) and Trimble NetR9 (4.85), respectively. The GNSS chipsets used by NEX9, SAMS, HONO are Broadcom

BCM4752, BCM4774 and BCM47531, respectively. TRIB, TRIA, TRI9 and WUHN can play the roles of reference stations. The true coordinate of the rover was determined using data from TRI9 with respect to WUHN, or TRIA in the vehicle-borne experiment. Note that the embedded GNSS antennas of NEX9, SAMS and HONO have been replaced by the TRM57971.00 antenna as shown in Fig. 1.

Types	Baseline (Length)	Duration (UTC)	Data logger (Version)
Zero baselines (static)	NEX9-TRI9 (0 m)	3:30-6:00, March 4,	Geo++ RINEX Logger (1.2.1)
	TRIA-TRI9 (0 m)	2018	
	SAMS-TRI9 (0 m)	8:20-8:30, June 6,	Geo++ RINEX Logger (1.2.4)
	HONO-TRI9 (0 m)	2018	
	SAMS-TRIB (0 m)	15:45-15:55, August 27, 2019	GnssLogger (2.0.0.1)
	HONO-TRIB (0 m)		
	NEX9-TRIB (0 m)		
Short baseline (static)	NEX9-WUHN (480 m)	3:30-6:00, March 4, 2018	Geo++ RINEX Logger (1.2.1)
Vehicle-borne (kinematic)	NEX9-TRI9 (0 m) NEX9-TRIA (100- 2000m)	2:55-3:35, January 15, 2019	Geo++ RINEX Logger (2.0.1)

We used zero baseline experiments to study the temporal properties of the GPS L1 chipset IPBs (Table 1). The positions of both receivers on zero baselines were fixed to the known ground truths and no atmosphere delays were estimated. We presumed that geometric delays and multipath effects totally canceled in this process, and only ambiguities alongside random observation noise were reserved ultimately. In this case, we could easily extract the fractional parts of those ambiguities as the

estimates of chipset IPBs, albeit noisy. To be specific, carrier-phase bias parameters were re-estimated at each epoch to avoid multi-epoch filtering on IPB estimates, and the fractional parts of those epoch-wise ambiguities were separated by a rounding operation as the double-difference IPBs. Since the reference station is a geodetic receiver without IPB, assuming that the IPB for a specific satellite is zero, we can obtain the satellite-specific IPBs by Eq. 9.

In contrast, the short-baseline data on NEX9-WUHN were experimented on to demonstrate high-precision positioning using Nexus 9 single-frequency GPS data by presuming that unaligned chipset IPBs could be pre-calibrated anyway (Table 1) (see Eq. 10). On average 8 satellites were tracked at each epoch. Note that in this study we are not designing a practical algorithm to calibrate chipset IPBs for smart devices, but investigating the possible achievement of Android GNSS ambiguity resolution if hopefully the IPBs can be gone. The IPB corrections used in this study were IPB estimates from the zero baseline experiments. For the short-baseline processing, we employ extended Kalman filter (EKF) to obtain the final solutions. An elevation-dependent weighting function is specified for determining the variance of observations (Euler and Goad 1991), which is used to construct the covariance matrix of double-differenced measurement errors by means of the law of error propagation (Takasu and Yasuda 2010). The position was estimated as a white-noise like parameter with a variance $\sigma_p^2 = 30^2 m^2$ while no atmosphere delays were computed. This time the ambiguities were estimated as constants over continuous arcs. Carrier-phase biases with a filtering duration of less than 10 epochs are excluded for ambiguity resolution. We re-attempted integer ambiguity resolution at each epoch to avoid any catastrophic consequences of incorrect fixing. We quantified the success rate of ambiguity resolution using the ratio between the number of ambiguity-fixed epochs and the number of all epochs. Note that an ambiguity-fixed epoch was achieved only when at least four double-difference ambiguities were fixed to integers successfully with the ratio test passing the threshold of 3.0 (Euler and Schaffrin 1990).

Fig. 2 A vehicle-borne experiment in a Wuhan suburban area. The vehicle drove along a near square route of about 7 km denoted in red escorted by trees and buildings. A reference station TRIA equipped with a Trimble NetR9 receiver and a TRM115000.00 antenna was established approximately in the middle of the area. A Nexus 9 tablet (NEX9) and a Trimble NetR9 receiver (TRI9) were mounted in the vehicle sharing a TRM57971.00 antenna as shown in the inset pictures. In addition, the number of satellites (abbreviated as “NS”) and the PDOP (Position Dilution of Precision) values for TRI9 and NEX9 are also presented in the inset where the vertical black line indicates the time when the vehicle began to drive (i.e., 3:10:40 UTC).

To augment the short-baseline experiment above, we also carried out a vehicle-borne experiment on January 15, 2019 in a Wuhan suburban area using a Nexus 9

tablet (NEX9) and a Trimble NetR9 receiver (TRI9) co-connected to a survey-grade antenna TRM57971.00 on top of the vehicle (Table 1). Fig. 2 describes the vehicle trajectory along an approximate square route spanning a 1.5×2 km area escorted often by trees and buildings. A reference station equipped with another Trimble R9 receiver (TRIA) was placed approximately in the middle of the area to provide benchmark positioning solutions using dual-frequency GPS data; in particular, the ambiguity-fixed epochs account for 99.2% of all epochs for the baseline solution between TRIA and TRI9. The vehicle kept static at first for about 16 minutes until 3:10:40 (UTC) and then drove for about 24 minutes at a speed of 25-35 km/h. We can see that 6-7 satellites could be observed during the static period which however dropped often to 5 after vehicle driving, while in contrast the collocated TRI9 could track more than 7 satellites most of the time albeit frequent loss of signals from a few satellites.

Results

Temporal properties of unaligned chipset IPBs

Fig. 3 shows the fractional parts of the double-difference carrier-phase bias estimates of all three smart devices (Nexus 9, Samsung Galaxy S8 and Huawei Honor 8) with respect to a geodetic receiver TRI9 or TRIB collocated on zero baselines. We can find that all fractional parts related to smart devices are non-zero and differ clearly from satellites to satellites, which can deviate up to several tenth cycles and hence destroy the integer nature of ambiguities definitely. Even worse, the fractional parts for G12 observed by NEX9-TRI9 change abruptly by about 0.1 cycles at 03:34:24 (UTC) due to a signal re-locking event after a cycle slip. In contrast, for the zero baseline TRIA-TRI9 which consists of two geodetic receivers, their fractional parts are close to zero with a much smaller scatter and stay the same over the duration of observations for all visible satellites, which hence ensures the manifestation of integer double-difference carrier-phase ambiguities. Moreover, this fact actually confirms the usual recognition that modern geodetic receivers keep identical their IPBs for different satellites

belonging the same GNSS. Unfortunately, smart devices represented by Nexus 9 do not obey this rule, and this problem seems shared by other smart devices employing similar GNSS chipsets (i.e., Samsung Galaxy S8 and Huawei Honor 8), even though they used different data logging software and versions. We thus demonstrate that the carrier-phase bias estimates from smart device GNSS chipsets are composed of not only integer ambiguities, but also the non-integer IPBs. These IPBs are not kept aligned or uniform across all satellites tracked by a smart device chipset, and therefore they cannot cancel through differencing between satellites, resulting in the non-integer carrier-phase bias estimates evidenced in Fig. 3.

Fig. 3 Ten minutes of fractional parts of the double-difference carrier-phase bias estimates color coded for 18 satellites referring to G28, G32 or G14 over seven zero baselines. An external survey-grade antenna TRM57971.00 was used for all smart devices. The four vertical lines indicate when the duty cycle is on. Note that the three panels from left to right correspond to three reference satellites (G28, G32 and G14), because the three zero-baseline experiments were carried out on different days. The Android data logger software used in these three zero-baseline experiments were Geo++ RINEX Logger (Ver. 1.2.1), Geo++ RINEX Logger (Ver. 1.2.4) and GnssLogger (Ver. 2.0.1), as shown in Table 1.

In addition, unlike NEX9-TRI9 with almost constant IPBs over the 10 minutes, the fractional parts of the double-difference carrier-phase bias estimates from HONO-TRI9 and SAMS-TRI9 only keep constant for the first five minutes, but suddenly diverge drastically once their duty-cycle settings were turned on automatically. This is because the duty-cycle settings will render the carrier-phase tracking of Samsung Galaxy S8 and Huawei Honor 8 discontinuous, or in other words, the carrier-phase data will have cycle slips and IPB reset at each epoch. Paziewski et al. (2019) discerned long-term drift in the double-difference carrier-phase data of their ultra-short-baseline (2.5 m) in case of the activated duty-cycling mode of Huawei P20. Since the long-term drift phase bias is much larger than one full cycle of carrier phase, fractional IPBs cannot be separated without ambiguities. Even worse, Paziewski et al. (2019) and Li and Geng (2019) confirmed that the carrier-phase noise far exceeded 0.1 cycles after duty-cycle on, which indicates that such discontinuous carrier-phase can hardly be used for ambiguity resolution. Therefore, high-precision positioning with such devices is quite challenging, and the phase-reconstruction technique may be a potential solution (Pesyna, et al. 2015).

In contrast, the Nexus 9 tablet disabled the duty-cycle setting, thus being able to provide continuous carrier-phase observations and constant IPBs alongside. To study longer-term temporal properties of unaligned chipset IPBs, we therefore only used the GPS single-frequency data from Nexus 9 throughout.

Fig. 4 Double-differenced ambiguity fractional parts (cycle) for several representative color coded satellite pairs computed at each epoch for the NEX9-TRI9 zero baselines using GPS L1 data in a static and a kinematic experiment lasting for about 150 and 40 minutes, respectively (cf. Table 1). An external survey-grade antenna TRM57971.00 was used for NEX9. An averaging over every 50 epochs is applied to filter out noise, producing the black curves ultimately. Any discontinuity of black curves implies a cycle slip. Note that the vertical black line in the bottom panel denotes the epoch when the vehicle started to move.

Accordingly, Fig. 4 shows the chipset IPB estimates of the Nexus 9 tablet collocated with TRI9 over several tens of minutes on a zero baseline in a static and a kinematic experiment (cf. Table 1). We only exhibit several representative satellites to avoid overlap of symbols. Since the raw single-epoch IPB estimates are quite noisy, we average them every 50 epochs and plot the results in black curves. In the top panel

of Fig. 4, disregarding the sudden jumps, the IPB estimates change over time with peak-to-peak variations of about 0.2 cycles, but without any drifting tendency overall, similar to those found by Geng et al. (2018). The IPB jumps or discontinuities are caused by accidental loss of track of satellite signals, which can be especially illustrated by G22 revealing two abrupt jumps before 4:06 UTC. Despite such jumps, we still find continuous IPB estimates for G22 in the top panel lasting for almost two hours until 6:00 UTC whose mean is 0.32 cycles and the standard deviation is 0.04 cycles (1σ), demonstrating the overall temporal stability of IPBs. Interestingly, the continuous IPB estimates seem to be modulated with a number of “boxcar” signatures of various widths but nearly identical heights, which uniquely characterize most temporal instabilities of the IPB estimates within the top panel of Fig. 4. We guess that this phenomenon might be caused by some unknown self-adjustment of the phase lock loop (PLL) regulated by Nexus 9 GNSS chipsets, rather than by the variation of IPBs themselves. Moreover, in the bottom panel of Fig. 4 for the kinematic experiment, the IPB estimates are also quite stable within 0.1 cycles except for a peculiar uniform sudden decline of about 0.1 cycles at around 3:04 UTC. In contrast, after the vehicle drove, loss of track of satellite signals took place frequently with IPBs reset in the mean time and thus jumping randomly as well as chaotically. Nevertheless, irrespective of such jumps, the IPB estimates are generally stable over time, notwithstanding a kinematic scenario.

According to Figs. 3 and 4, we can find that the chipset IPBs differ among satellites and changes randomly if the carrier-phase of a particular satellite is re-tracked, but fortunately they are generally stable over time since the mean standard deviation of all raw single-epoch IPB estimates derived from continuous carrier-phase (without cycle slips) is as low as 0.04 cycles (1σ) for all satellites in the two NEX9-TRI9 zero-baseline experiments (cf. Table 1). Since a chipset IPB for a satellite changes whenever the signals of this satellite are re-tracked, we unfortunately cannot provide a priori satellite-specific IPB corrections to enable integer ambiguity resolution for real-time applications. One remedy is that the GNSS chipset

manufacturers for smart devices align the IPBs for all tracked satellites, as always done by geodetic receivers. However, for post-processing, we could try to remove a mean IPB from each continuous carrier-phase arc to recover the integer property of double-difference ambiguities. We should bear in mind that zero baseline solutions are preferred in this process since float double-difference ambiguities can converge efficiently and sufficiently to facilitate precise IPB computation.

Ambiguity resolution by calibrating IPBs for short baselines

As suggested above, we computed a mean IPB correction for each satellite over its each continuous carrier-phase arc using the single-epoch IPB estimates from the two NEX9-TRI9 zero-baseline experiments (cf. Table 1). Since the boxcar signatures with an amplitude of up to 0.1 cycles presented in Fig. 4 was not taken as part of the IPBs, they might degrade the success rate of ambiguity resolution and also bias, if discernibly, position estimates once ambiguities were fixed to integers. We then applied these corrections back to the two baselines and attempted integer ambiguity resolution. In particular, the LAMBDA (Least-squares AMBiguity Decorrelation Adjustment) method was employed to identify the best integer candidates for the double-difference carrier-phase ambiguities (Teunissen 1995). A fixed solution could be achieved once at least four ambiguities were fixed to integers successfully.

Fig. 5 GPS single-frequency carrier-phase positioning errors (cm) on short baselines using a Nexus 9 tablet (NEX9) with respect to geodetic receivers (WUHN and TRIA). The NEX9-WUHN is a static baseline while NEX9-TRIA is a kinematic one (cf. Table 1). Both ambiguity-float (pink) and ambiguity-fixed (dark green) solutions are shown for the RMS of positioning errors in the east, north and up components; the benchmark solutions were computed using a Trimble NetR9 receiver sharing the same antenna TRM57971.00 with NEX9; note that all RMS were computed only for the ambiguity-fixed epochs. Ambiguity resolution was enabled by calibrating Nexus 9 chipset IPBs, and the ratio values for ambiguity validation are plotted in the fourth row of panels with the red horizontal lines denoting a threshold of 3.0 and the percentages (86.0% and 51.6%) of the fixed epochs plotted therein. Note that the ratio values larger than 20 are simply plotted as 20 for brevity. The time-to-first-fix were 03:50:37 and 03:00:21 UTC for the two experiments. The bottom panels show the number of visible satellites (blue) and the PDOP values (orange); the black vertical line denotes the time when the vehicle began to drive.

Fig. 5 shows the positioning differences of both ambiguity-float and ambiguity-fixed solutions from the benchmark positions. For the static baseline NEX9-WUHN spanning about 480 m, the time-to-first-fix is about 20 minutes with about 86.0% of all epochs resolved successfully; moreover, the RMS of the positioning differences from benchmark solutions after ambiguity fixing are 0.8, 0.7 and 1.8 cm for the east, north and up components, respectively, in sharp contrast to 15.4, 3.1 and 12.5 cm for the ambiguity-float solution. This achievement implies that the boxcar-like noise reported in Fig. 4 does minimum harm to the ambiguity-fixed positioning accuracy. As a comparison, the baseline solution for TRI9-WUHN collocating with NEX9-WUHN is also computed using the same satellites where the time-to-first-fix is 304 s and 96.3% of all epochs are resolved.

Fig. 6 Double-difference pseudorange and carrier-phase residuals for the ambiguity-fixed baseline solutions NEX9-WUHN and TRI9-WUHN. Note that the baseline solution NEX9-WUHN has been shown in Fig. 5. The standard deviations (STD) of

the residuals are presented at the top right corner of each panel. Different colors represent different satellite pairs which are not explicitly shown here.

To examine why geodetic receivers perform so much better than Nexus 9, we plot Fig. 6 to show the double-difference pseudorange and carrier-phase residuals after the baseline solutions above. It can be seen that the pseudorange residuals of NEX9-WUHN are almost 7 times larger than those of TRI9-WUHN in terms of standard deviations. This demonstrates that the pseudorange of Nexus 9 cannot ensure as fast convergences of carrier-phase ambiguities as those boosted by Trimble NetR9 pseudorange. In the mean time, compared to the carrier-phase residuals of TRI9-WUHN, those of NEX9-WUHN exhibit an anomalous near-periodic signal, which we believe is directly associated with the boxcar signatures found in the ambiguity fractional parts show in Fig. 4. Since the amplitude of this near-periodic signal or boxcar signature can be up to 0.1 cycles, they will definitely compromise the accurate identification of integer ambiguities in the process of baseline solutions and ultimately prolong the time-to-first-fix for Nexus 9. To quantify the impact of the boxcar signatures on NEX9 ambiguity resolution, we estimated one IPB for each boxcar section (i.e., boxcar-specific IPBs) identified in the top panel of Fig. 4, instead of one IPB for each continuous carrier-phase arc (i.e., arc-specific IPBs), aiming at mitigating unaligned IPBs as well as those boxcar noise. We then experimented on the short baseline NEX9-WUHN by correcting for the arc-specific or the boxcar-specific IPBs, and re-attempted ambiguity resolution by starting the processing from the epoch $3:30+5n$ UTC where $n=0, 1, 2 \dots 18$ until one hour after. In total, we collected the time-to-first-fix and the percentages of ambiguity-fixed epochs for 19 times of ambiguity resolution trials. On average, 590 s of data were required to achieve ambiguity-fixed solutions with about 84% of epochs resolved in case of arc-specific IPB corrections, whereas 441 s of data were needed with 88% of epochs fixed in case

of boxcar-specific IPBs. This shows that the boxcar signatures delay the time-to-first-fix by about 34% regarding baseline NEX9-WUHN.

For the vehicle-borne short baseline solution, we achieved the first fixed solution within 321 s and finally 51.6% of all epochs were resolved. After the vehicle began to drive at 3:10:40 UTC, the ambiguity-fixed solution persisted for another 456 s until 3:18:16 UTC, though occasionally some epochs afterward could still be fixed successfully. We attribute the loss of ambiguity-fixed solutions to the frequent loss of track for satellites in the vehicle driving stage, as shown by the bottom panel of Fig. 5. In this case, new ambiguities can hardly converge to high enough precisions for a successful integer-cycle resolution. In contrast, the collocated baseline TRI9-TRIA with two geodetic receivers could achieve single-frequency ambiguity resolution at more than 98% of all epochs, even though the vehicle was driving. TRI9 received satellite signals from more satellites (cf. Fig. 2), which in part explains its higher fixing rate, while however we argue that the definitely better carrier-phase and pseudorange tracking quality of geodetic receivers also contribute significantly. We envision that a dedicated GPS cycle-slip repair algorithm for Nexus 9 would improve the mobile high-precision positioning quality (cf. Geng et al. 2010). Despite the poor fixing achievement during the vehicle driving period, ambiguity-fixed epochs can still achieve a positioning accuracy of 1.4, 2.2 and 3.6 cm for the east, north and up components, respectively, showing an improvement of 30%-80% compared to the ambiguity-float solutions.

In addition, it took about 20 minutes for baseline NEX9-WUHN to achieve the first ambiguity-fixed solution, while only 321 s for baseline NEX9-TRIA in the static stage. We believe that this is because NEX9-WUHN does not have as good a satellite geometry as NEX9-TRIA has in the early stage of the experiments (see the bottom panels of Fig. 5). Finally, the results above confirm that the correction of Nexus 9 chipset IPBs is critical to the recovery of the integer nature of its carrier-phase ambiguity, and successful ambiguity resolution has to potential to improve Nexus 9

GPS single-frequency positioning accuracy to the centimeter level.

Discussion

Chipset phase biases beyond IPBs

On May 31, 2018, the world's first dual-frequency (L1/E1 + L5/E5) GNSS smartphone Xiaomi 8 was released, equipped with a Broadcom BCM47755 chipset. Later, another similar handset Huawei Mate 20 was launched on October 26, 2018 embedding a HiSilicon Kirin 980 Hi1103 GNSS chipset. We used the same zero-baseline processing strategy as that for Table 1 to compute the fractional parts of their double-difference carrier-phase bias estimates epoch by epoch referring to the collocated TRI9.

Fig. 7 Fractional parts (cycle) of double-difference GPS L1/L5 and Galileo E1/E5a carrier-phase bias estimates color coded for all observed satellite pairs by a Xiaomi 8 (XMI8) smartphone on a zero baseline with respect to TRI9. Note that the observation periods for GPS and Galileo satellites are different.

Accordingly, Fig. 7 shows these fractional parts for Xiaomi 8. We can see that both L1 and L5 fractional parts are close to zero this time, showing that the unaligned IPBs across GPS satellites present in Nexus 9, Samsung Galaxy S8 and Huawei Honor 8 above have been gone favorably. In this case, we can resolve GPS ambiguities without being obsessed or trapped by unaligned IPBs. Conversely, Xiaomi 8 carrier-phase ambiguities can be fixed to integers analogous to geodetic receivers. Xiaomi 8 Galileo E1 fractional parts show similar characteristics to those on its GPS L1/L5 signals. At this point, we may be optimistic that Broadcom has fixed the IPB problem in their latest smart device GNSS chipsets, albeit the increased algorithm complexity. Unfortunately, E5a signals in Fig. 7 can still manifest IPBs among the Galileo satellites, which is quite abnormal and hardly understandable at the moment on account of the preferable IPB situation for GPS L1/L5 and Galileo E1. Moreover, in spite of the absence of IPBs on GPS signals, the boxcar signatures of comparable amplitude to that for Nexus 9 remain in the fractional parts of Xiaomi 8 GPS/Galileo double-difference ambiguities, which might still impair the ambiguity fixing efficiency in a significant manner as demonstrated for baseline NEX9-WUHN above.

Fig. 8 GPS L1/L5 and Galileo E1/E5a carrier-phase biases (cycle) after removal of integer ambiguities color coded for all observed satellite pairs by a Huawei Mate 20 (HWM2) smartphone on a zero baseline with respect to TRI9. Do not ignore the amplification factors of 10^6 and 10^5 for the vertical annotations for Galileo carrier-phase biases. Note that we used Android P duty-cycle control to force duty-cycle off within Huawei Mate 20.

Unlike the Nexus 9 tablet and the Xiaomi 8 handset, however, the Huawei Mate 20 (HWM2) smartphone exhibits divergent carrier-phase biases among all GPS and Galileo satellites. Fig. 8 shows the carrier-phase bias estimates after removal of their integer parts for all continuous arcs observed by a zero baseline HWM2-TRI9. We can see that the GPS L1/L5 carrier-phase biases do not converge, but diverge

drastically to about 10 cycles over a few tens of minutes for all satellites, even in a zero-baseline processing with respect to a geodetic receiver. This phenomenon is even more serious in case of Galileo E1/E5a carrier-phase biases since they diverge to even millions of cycles within 20 minutes. Such catastrophic divergences will totally invalidate the usability of carrier-phase data from Huawei Mate 20 in any phase-relevant positioning practices, not to mention ambiguity resolution. We guess that this problem might be caused by the asynchronous lock of carrier-phase signals from different satellites, though a common lock time is reported in the RINEX file, which however needs more investigation by the smartphone manufacturer.

Portable patch antennas for smart devices

Heretofore, we have been constantly using an external survey-grade antenna to enable Android GNSS ambiguity resolution over short baselines. However, survey-grade antennas are normally of large size and not easily portable. It is thus unrealistic or even unimaginable to ask the mass market to favor the idea of combining smart devices with survey-grade antennas for semi-professional assignments, or persuade common users to carry a survey-grade antenna always in case of any high-precision crowd-sourcing tasks. Since patch antennas are usually of small size and naturally portable, it will be preferable to use them instead of survey-grade antennas to pair with smart devices for democratized high-precision GNSS. We hence experimented on a representative μ -blox ANN-MS patch antenna in connection with a Xiaomi 8 smartphone to study if ambiguity resolution is still achievable. Fig. 9 depicts the configuration of this experiment lasting for two hours on November 21, 2018. At first, an ANN-MS antenna was placed on an experiment board while connected to a Xiaomi 8 smartphone which is placed within an RF shielding box (cf. Fig. 1); another Xiaomi 8 was simply anchored on the board to collect GNSS data using its embedded antenna; in the meantime, a second ANN-MS antenna was also fixed on the board, but connected to a μ -blox C94-M8P-1 receiver; a fourth solution was carried out by

linking an RF-shielded Xiaomi 8 smartphone to a survey-grade antenna (TRM57971.00) close to this board (cf. Fig. 1). Note that no antenna splitter was used here because the ANN-MS antenna was not compatible with our splitter. However, since the u-blox ANN-MS patch antenna was an active antenna, we also provided an additional direct-current (DC) power supply module. The benchmark position of the two ANN-MS antennas and the Xiaomi 8 smartphone on the board had been predetermined with respect to station WUHN which was about 480 m away (Table 1). Since all antennas were close to each other, we presumed that they were observing identical GPS satellites and suffering from the same multipath effects and atmosphere refractions. We processed only GPS L1 data and the baseline processing strategies were the same as those for the experiments in section “Results”, except that no IPB corrections need to be taken into account for Xiaomi 8.

Fig. 9 An experiment investigating the impacts of patch antennas on Xiaomi 8 GNSS data quality. The ANN-MS antenna connected to a Xiaomi 8 smartphone which is held in an RF shielding box, the Xiaomi 8 smartphone using its embedded antenna and the ANN-MS antenna connected to a μ -blox C94-M8P-1 receiver are fixed on the experimental board from left to right. The RF-shielded Xiaomi 8 smartphone shown in the bottom right corner of the figure connects to TRM57971.00 antenna through a signal splitter (cf. Fig. 1). The distance between the board and the TRM570971.00 antenna is about 1.6m.

Fig. 10 GPS satellites tracked by Xiaomi 8 smartphones and a μ -blox C94-M8P-1 receiver from 8:00 to 10:00 UTC on November 21, 2018. The vertical red line segments mark all cycle slip events whose occurrence rates are plotted within each left panel. ANN-MS is a μ -blox patch antenna. Correspondingly, the right panels

show the carrier-phase residuals (cycle, blue dots) and carrier-to-noise ratios (C/N_0 , orange dots) for satellite pair G10-G14 with regard to each handset-antenna pair. Note that all carrier-phase residuals were computed in short baseline solutions with respect to station WUHN, and their RMS are plotted in the bottom right corner of each right panels.

Fig. 10 shows the GPS satellites tracked by the four handset-antenna pairs. We can see that Xiaomi 8's embedded antenna introduced a large number of cycle slips which occurred at about 8% of all epochs, even in case of a static baseline configuration and an open skyview environment. Moreover, in the right panels of Fig. 10, the carrier-phase residuals of satellite pair G10-G14 for Xiaomi 8's embedded antenna are exemplified which manifest pronounced and chaotic fluctuations of up to 0.3 cycles with an RMS of 0.113 cycles over the two hours. Correspondingly, the carrier-to-noise ratios (C/N_0) of the two satellites also reveal the problematic observational quality of the embedded antenna, which is an omnidirectional passive linearly polarized antenna (Li and Geng, 2019). On the contrary, other handsets connected to the survey-grade and patch antennas suffered much fewer cycle slips, which occurred at less than 1.5% of all epochs.

Since the other handset-antenna combinations deliver carrier-phase residuals of less than 0.15 cycles with RMS of less than 0.06 cycles, we believe that the bad carrier-phase residuals in the top right panel of Fig. 10 should be attributed to the embedded linearly polarized antenna, rather than the spatially-correlated errors (e.g., atmosphere refractions, orbit and clock errors) or the potential chipset defects of smart devices. Finally, Fig. 10 demonstrates that the Xiaomi 8 smartphone can be connected effectively to a portable patch antenna and achieve commensurate carrier-phase tracking performance (i.e., low cycle-slip rate and good C/N_0) to that of a μ -blox receiver with its dedicated patch antenna, more encouragingly approaching the performance of Xiaomi 8 with a survey-grade antenna.

Fig. 11 Short-baseline positioning accuracy (cm) for the east, north and up components over a period of two hours for a Xiaomi 8 smartphone and a μ -blox receiver (C94-M8P-1) connected to different antennas with respect to station WUHN on November 21, 2018. ANN-MS is a patch antenna manufactured by μ -blox. The RMS statistics are calculated using the positions spanning from 8:15 to 10:00 UTC. The vertical dashed lines indicate the time to first fix, and the horizontal red lines mark the threshold of 3.0 for the ratio test. Note that the ratio values of larger than 20 are plotted as 20. The bottom panels show the number of visible satellites (sky blue) and the PDOP values (orange).

Fig. 11 shows the positioning accuracy of the Xiaomi 8 smartphone connected to different antennas. We can see that Xiaomi 8 smartphone cannot fix the ambiguity correctly when using its embedded antenna, and only obtained the ambiguity-float solutions with the RMS of the positioning errors are 24.26, 9.92 and 10.47 cm for the

east, north and up components, respectively. On the contrary, other handsets connected to the survey-grade and patch antennas obtained ambiguity-fixed solutions. To be specific, for the Xiaomi 8 smartphone connected to a survey-grade antenna (TRM57971.00), its time-to-first-fix is 422 s with about 94.1% of all epochs resolved successfully; moreover, the RMS of the positioning errors after ambiguity fixing are 0.58 cm, 0.65 cm and 2.33 cm for the east, north and up components, respectively. Meanwhile, for the Xiaomi 8 smartphone connected to a patch antenna (ANN-MS), its time-to-first-fix was increased to 670 s with about 90.2% of all epochs resolved successfully; the RMS of the positioning errors after ambiguity fixing were increased to 1.10 cm, 1.11 cm and 3.36 cm for the east, north and up components, respectively. Their positioning performance is approaching to that of a μ -blox receiver with its dedicated patch antenna. Its time-to-first-fix is 595 s with about 91.4% of all epochs resolved successfully, and its RMS of the positioning errors after ambiguity fixing are 0.75 cm, 0.37 cm and 1.97 cm for the east, north and up components, respectively. The results above confirm that the Xiaomi 8 smartphone can be connected effectively to a portable patch antenna and achieve commensurate positioning performance to that of a μ -blox receiver with its dedicated patch antenna. Therefore, for those users who use smart devices for high-precision semi-professional assignments, smaller and inexpensive patch antenna pairing smart devices could be a viable solution.

Conclusions

We investigated the feasibility of resolving Android GNSS carrier-phase ambiguities. Based on the raw GNSS data from some smart devices, we found unaligned chipset initial phase biases (IPBs). Subsequently, we conducted zero-baseline, short-baseline and vehicle-borne smart device experiments to investigate the temporal properties of the unaligned chipset IPBs and the ambiguity resolution performance by calibrating IPBs for smart devices. Experimental results show that the smart devices including

Nexus 9, Samsung Galaxy S8 and Huawei Honor 8 all have the unaligned chipset IPB issues. We select a Nexus 9 tablet as representative of smart devices since its carrier-phase measurements are continuous. The unaligned chipset IPBs differ among satellites and changes randomly if the carrier-phase of a particular satellite is re-tracked, but fortunately they are generally stable over time since the mean standard deviation of all raw single-epoch IPB estimates derived from continuous carrier-phase is as low as 0.04 cycles for all satellites.

We computed a mean IPB correction for each satellite over its each continuous carrier-phase arc using the single-epoch IPB estimates from the zero-baseline experiments. By calibrating IPBs, we implemented ambiguity resolution on the Nexus 9 tablet connected with an external survey-grade antenna. For a ~480 m static short-baseline, the time-to-first-fix is about 20 minutes with about 86.0% of all epochs resolved successfully; moreover, the RMS of the positioning differences from benchmark solutions after ambiguity fixing are 0.8, 0.7 and 1.8 cm for the east, north and up components, respectively, in contrast to 15.4, 3.1 and 12.5 cm for the ambiguity-float solution. For the vehicle-borne short baseline solution, we achieved the first fixed solution within 321 s and finally 51.6% of all epochs were resolved. Despite the poor fixing achievement during the vehicle driving period due to the frequent loss of track for satellites, ambiguity-fixed epochs can still achieve a positioning accuracy of 1.4, 2.2 and 3.6 cm for the east, north and up components, respectively, showing an improvement of 30%-80% compared to the ambiguity-float solutions. The results above confirm that the correction of Nexus 9 chipset IPBs is critical to the recovery of the integer nature of its carrier-phase ambiguity, and successful ambiguity resolution has to potential to improve Nexus 9 GPS single-frequency positioning accuracy to the centimeter level.

Finally, we discuss that the chipset phase biases beyond IPBs for dual-frequency GNSS smartphone Xiaomi 8 and Huawei Mate 20, and further discuss the portable patch antennas for smart devices. For Xiaomi 8 smartphone, in spite of the absence of

IPBs on GPS L1/L5 and Galileo E1 signals, E5a signals can still manifest IPBs among the Galileo satellites. Moreover, the boxcar signatures of comparable amplitude to that for Nexus 9 remain in the fractional parts of Xiaomi 8 GPS/Galileo double-difference ambiguities, which might still impair the ambiguity fixing efficiency. Huawei Mate 20 smartphone exhibits divergent carrier-phase biases among all GPS and Galileo satellites. The GPS L1/L5 carrier-phase biases diverge drastically to about 10 cycles over a few tens of minutes for all satellites, while Galileo E1/E5a carrier-phase biases since they diverge to even millions of cycles within 20 minutes, which makes ambiguity resolution impossible. We experimented on a representative μ -blox ANN-MS patch antenna in connection with a Xiaomi 8 smartphone to study if ambiguity resolution is still achievable. Encouragingly, Xiaomi 8 smartphone can be connected effectively to a portable patch antenna and achieve commensurate carrier-phase tracking performance and positioning performance to that of a μ -blox receiver with its dedicated patch antenna. When Xiaomi 8 smartphone connected to an ANN-MS patch antenna in this study, its time-to-first-fix is 670 s with about 90.2% of all epochs resolved successfully; the RMS of the positioning errors after ambiguity fixing are 1.10 cm, 1.11 cm and 3.36 cm for the east, north and up components, respectively. For those users who use smart devices for high-precision semi-professional assignments, smaller and inexpensive patch antenna pairing smart devices could be a viable solution.

Author Contribution Statement: J. G. conceived the project and the main conceptual ideas. J. G. and G. L. worked out almost all of the technical details, and performed the numerical calculations for the suggested experiments; G. L. analyzed the data; J. G. and G. L. wrote the paper. All authors provided critical feedback and helped to shape the research, analysis and manuscript.

Data Availability Statement: All data collected for the study are available from

the authors by request.

Acknowledgment

This work is funded by National Key Research and Development Program of China (No. 2018YFC1504002) and National Science Foundation of China (No. 41674033).

We used Google GnssLogger apps and Geo++ RINEX Logger apps to obtain raw GNSS data from smart devices. We thank two anonymous reviewers for their valuable comments.

References:

- Banville S, Diggelen FV (2016) Precise GNSS for Everyone: Precise Positioning Using Raw GPS Measurements from Android Smartphones. *GPS World* 27(11):43-48
- Euler HJ, Goad CC (1991) On optimal filtering of GPS dual frequency observations without using orbit information. *Bulletin Géodésique* 65(2):130-143.
- Euler HJ, Schaffrin B (1990) On a measure of the discernibility between different ambiguity solutions in the static-kinematic GPS mode. In: Schwarz KP, Lachapelle G (eds) *Kinematic systems in geodesy. Surveying and remote sensing*. Springer, New York, pp 285-295
- European GNSS Agency (2017) *Using GNSS raw measurements on Android devices*. Publications Office of the European Union, Luxembourg
- Geng J, Li G, Zeng R, Wen Q, Jiang E (2018) A Comprehensive Assessment of Raw Multi-GNSS Measurements from Mainstream Portable Smart Devices. *Proc. ION GNSS+ 2018*, Institute of Navigation, Miami, Florida, USA, September 24 - 28, 392-412
- Geng J, Jiang E, Li G, Xin S, Wei N (2019) An Improved Hatch Filter Algorithm

towards Sub-Meter Positioning Using only Android Raw GNSS Measurements without External Augmentation Corrections. Remote Sens-Basel 11(14). doi:10.3390/rs11141679

Geng J, Meng X, Dodson AH, Ge M, Teferle FN (2010) Rapid re-convergences to ambiguity-fixed solutions in precise point positioning. J Geodesy 84(12):705-714. doi:10.1007/s00190-010-0404-4

Håkansson M (2018) Characterization of GNSS observations from a Nexus 9 Android tablet. Gps Solut 23(1):21. doi:10.1007/s10291-018-0818-7

Hauschild A (2017) Basic Observation Equations. In: Teunissen PJG, Montenbruck O (eds) Springer Handbook of Global Navigation Satellite Systems. Springer International Publishing, Cham, pp 561-582

Humphreys TE, Murrian M, Diggelen FV, Podshivalov S, Pesyna KM (2016) On the feasibility of cm-accurate positioning via a smartphone's antenna and GNSS chip. Proc. IEEE/ION PLANS 2016, Savannah, GA, April 11-14, 232-242

Li G, Geng J (2019) Characteristics of raw multi-GNSS measurement error from Google Android smart devices. GPS Solut 23(3):90. doi:10.1007/s10291-019-0885-4

Malkos S (2016) User location takes center stage in new Android OS: Google to provide raw GNSS measurements. GPS World 27(7):36.

Misra P, Enge P (2006) Global Positioning System: signals, measurements and performance, 2nd edn. Ganga-Jamuna Press, MA

O'Driscoll C, Petovelle M (2010) GNSS Solutions: Carrier-phase and its measurement for GNSS. Inside GNSS 5(4):18-22.

Paziewski J, Sieradzki R, Baryla R (2019) Signal characterization and assessment of code GNSS positioning with low-power consumption smartphones. GPS Solut 23(4):98. doi:10.1007/s10291-019-0892-5

- Pesyna KM, Heath RW, Humphreys TE (2014) Centimeter Positioning with a Smartphone-Quality GNSS Antenna. Proc. ION GNSS 2014, Tampa, Florida, USA, September 8 - 12, 1568-1577
- Pesyna KM, Kassas ZM, Heath RW, Humphreys TE (2015) A Phase-Reconstruction Technique for Low-Power Centimeter-Accurate Mobile Positioning. IEEE T Signal Proces 62(10):2595-2610.
- Pirazzi G, Mazzoni A, Biagi L, Crespi M (2017) Preliminary performance analysis with a GPS+Galileo enabled chipset embedded in a smartphone. Proc. ION GNSS 2017, Institute of Navigation, Portland, Oregon, USA, September 25-29, 101-115
- Realini E, Caldera S, Pertusini L, Sampietro D (2017) Precise GNSS Positioning Using Smart Devices. Sensors-Basel 17(10):2434.
- Riley S, Lentz W, Clare A (2017) On the Path to Precision - Observations with Android GNSS Observables. Proc. ION GNSS 2017, Institute of Navigation, Portland, Oregon, USA, September 25-29, 116-129
- Takasu T, Yasuda A (2010) Kalman-Filter-Based Integer Ambiguity Resolution Strategy for Long-Baseline RTK with Ionosphere and Troposphere Estimation. Proceedings of International Technical Meeting of the Satellite Division of the Institute of Navigation 7672(6):161-171.
- Teunissen PJG (1995) The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. J Geodesy 70(1):65-82.
doi:10.1007/BF00863419
- Zhang X, Tao X, Zhu F, Shi X, Wang F (2018) Quality assessment of GNSS observations from an Android N smartphone and positioning performance analysis using time-differenced filtering approach. GPS Solut 22(3):70.
doi:10.1007/s10291-018-0736-8
- Zhang Y, Yao Y, Yu J, Chen X, Zeng Y, He N (2013) Design of a novel quad-band circularly polarized handset antenna. Proceedings of the International Symposium

on Antennas & Propagation, Nanjing, China, October 23-25, 146-148