Android Raw GNSS Measurement Datasets for Precise Positioning

Guoyu (Michael) Fu, *Mohammed Khider, Frank van Diggelen,*
Google, Inc.

Session C3, Sep 24, 2020
ION GNSS+ Virtual 2020

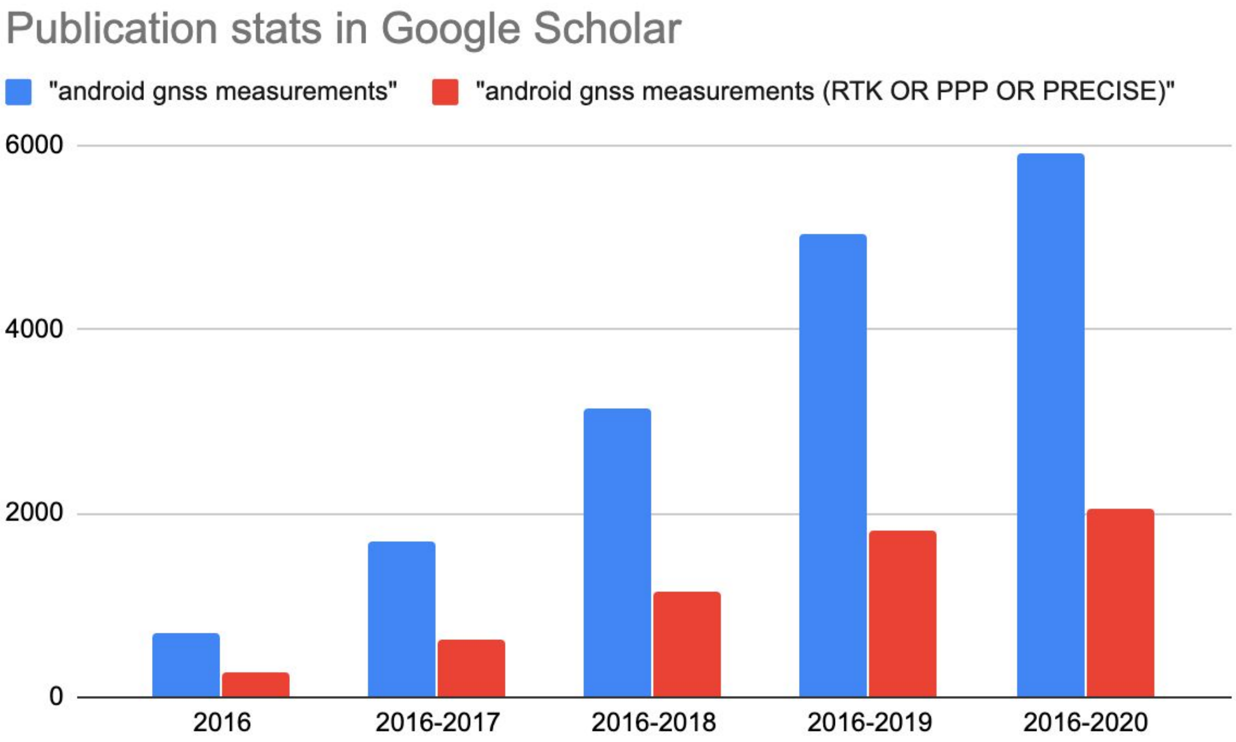
Authors

Dr. Guoyu (Michael) Fu
guoyuf@google.com

Dr. Mohammed Khider
gomo@google.com

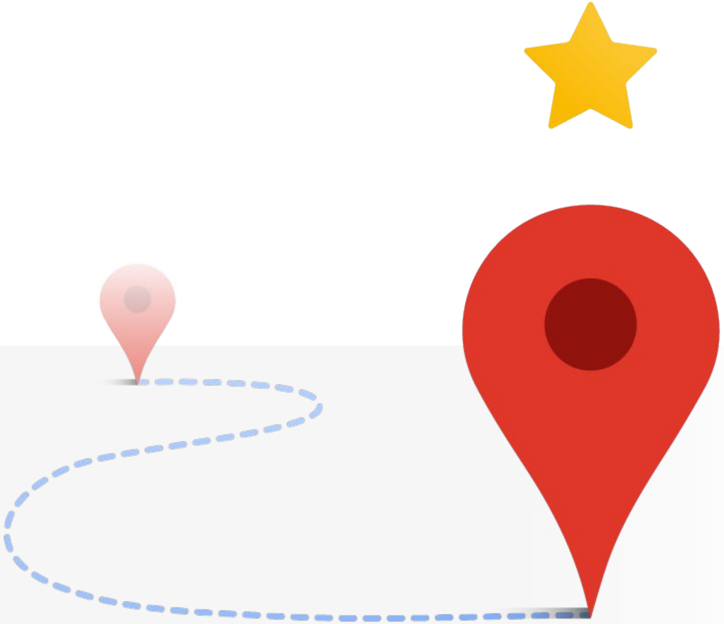
Dr. Frank van Diggelen
fvandiggelen@google.com

Over **1/3** of publications focus on high precision positioning



References: [2][3]

Motivations



High precision smartphone positioning is helpful for:

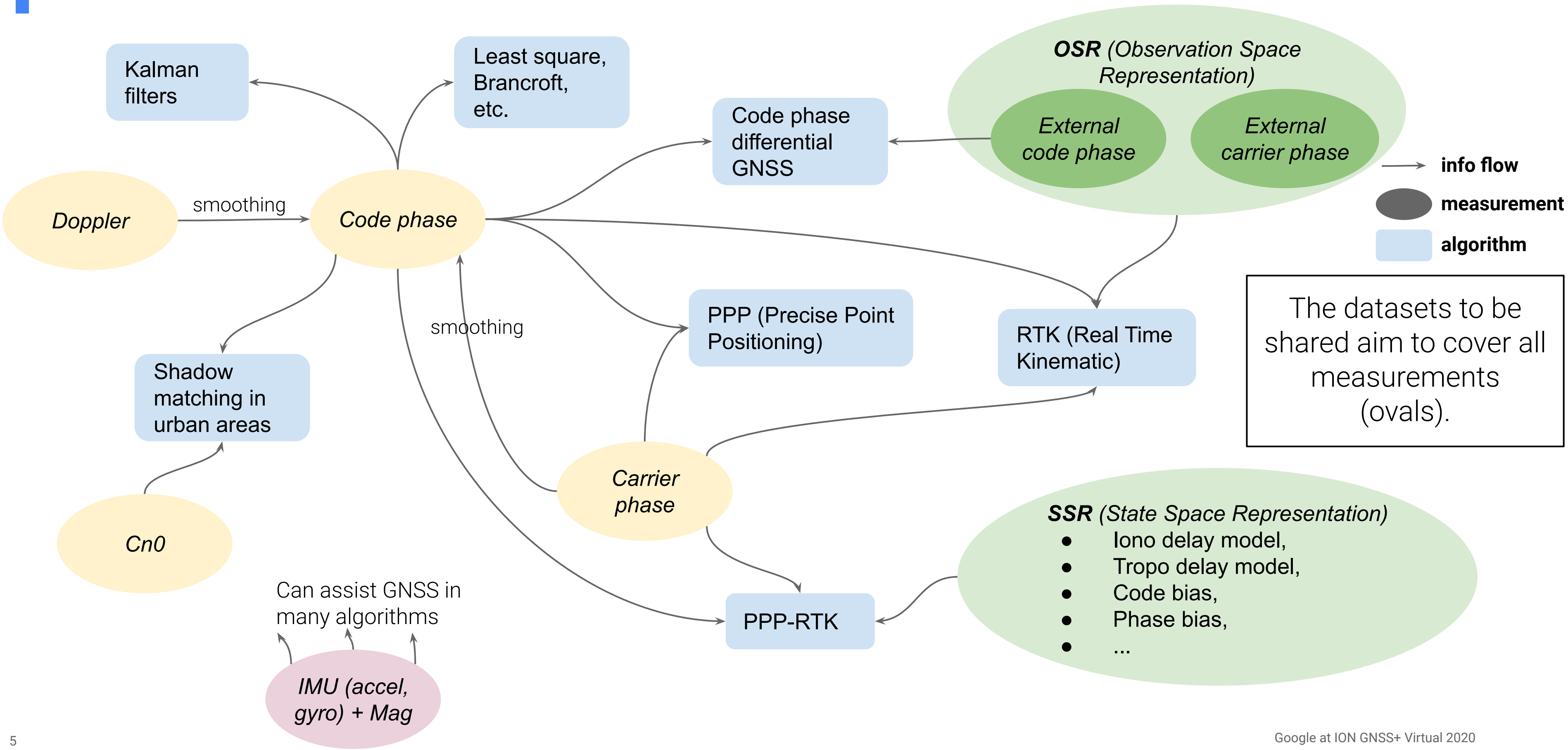
- Lane level mapping, traffic monitoring, navigation
- Ubiquitous geo-surveying
- Entertaining precision agriculture
- Precise augmented reality (AR) applications
- Better indoor mapping and positioning
- Other novel applications ...

Google will share smartphone test datasets to:

- provide a set of benchmark tests with ground truth, and
- a common evaluation tool with standard positioning metrics.

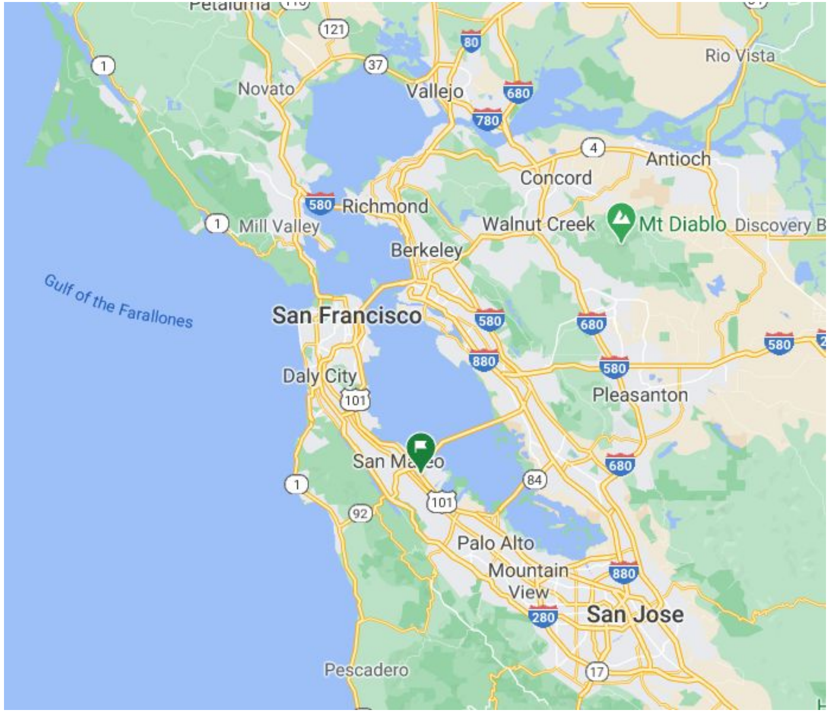
Measurements & Common Positioning Algorithms

Google



The datasets to be shared aim to cover all measurements (ovals).

We release …



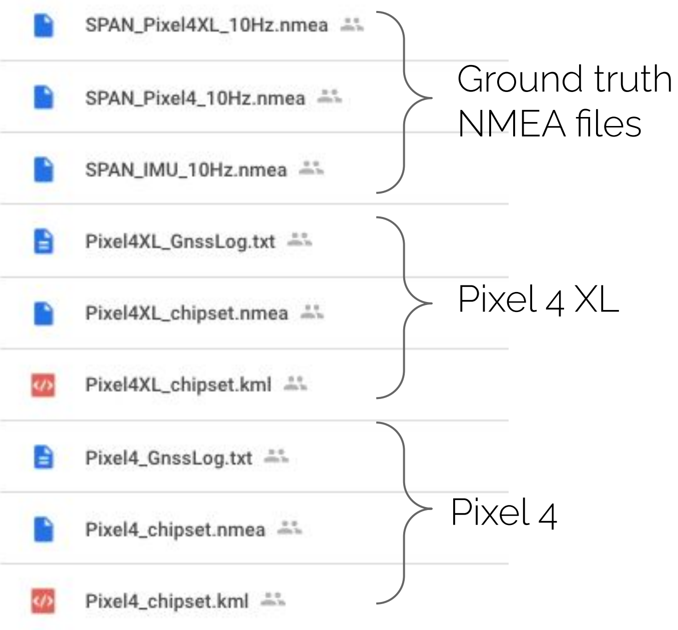
Google will release in [GitHub]() in October 2020:

- ~60 traces collected between May and Aug, 2020
  - GnssLog with raw GNSS and IMU readings (mostly from Pixel phones)
  - Converted RINEX observation files
  - Chipset positions in NMEA format
  - Ground truth NMEA files
  - OSR (Observation Space Representation) RINEX files from nearby stations owned by Verizon, Inc.
  - SSR (State Space Representation) text files from SwiftNav, Inc.
- A set of Matlab scripts for evaluating the result NMEA files

Open Data Directories

- **Datasets/YYYY-MM-DD-{Country code in 2 letters}-{Region}-{Collection index}** (e.g. 2020-05-14-US-MTV-1)
  - **{Phone name}_GnssLog.txt**: GnssLogger log file
  - **SPAN_{Phone name}_{Report rate}.nmea**: Ground truth positions at the phone's position
  - **{Phone name}_chipset.nmea**: Chipset positions in NMEA format
  - **{Phone name}_chipset.kml**: Chipset positions in KML format

- **Corrections/YYYY-MM-DD-{Country code in 2 letters}-{Region}-{Collection index}/**
  - **OSR/**: RINEX files of nearby base stations by Verizon
  - **SSR/:** SwiftNav SSR corrections in JSON



Outline

SPAN IMU-to-DUT lever arm

***NmeaUtils*: DUT NMEA vs GT NMEA**

Content of GnssLog

NMEA format

- SPAN_Pixel4XL_10Hz.nmea
- SPAN_Pixel4_10Hz.nmea
- SPAN_IMU_10Hz.nmea

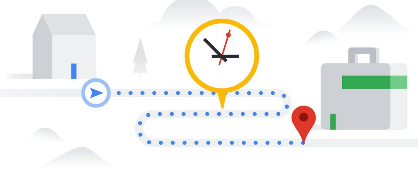
Ground truth NMEA files

- Pixel4XL_GnssLog.txt
- Pixel4XL_chipset.nmea
- Pixel4XL_chipset.kml

Pixel 4 XL

- Pixel4_GnssLog.txt
- Pixel4_chipset.nmea
- Pixel4_chipset.kml

Pixel 4



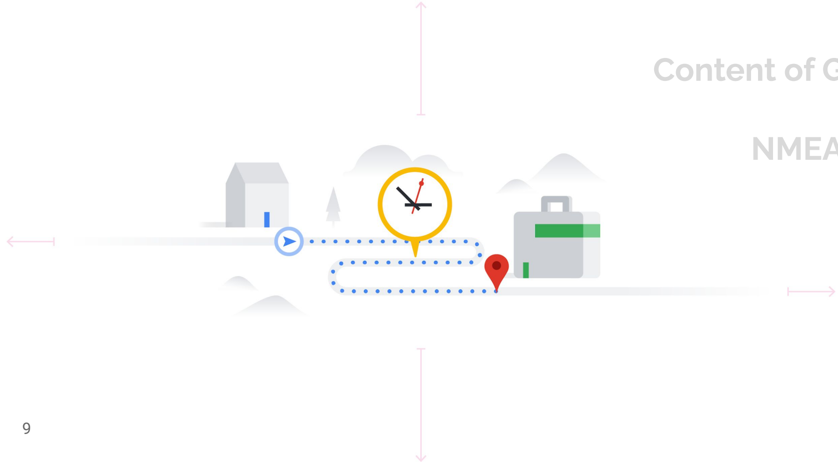
Outline

SPAN IMU-to-DUT lever arm

NmeaUtils: DUT NMEA vs GT NMEA

Content of GnssLog

NMEA format



- SPAN_Pixel4XL_10Hz.nmea
- SPAN_Pixel4_10Hz.nmea
- SPAN_IMU_10Hz.nmea

Ground truth NMEA files

- Pixel4XL_GnssLog.txt
- Pixel4XL_chipset.nmea
- Pixel4XL_chipset.kml

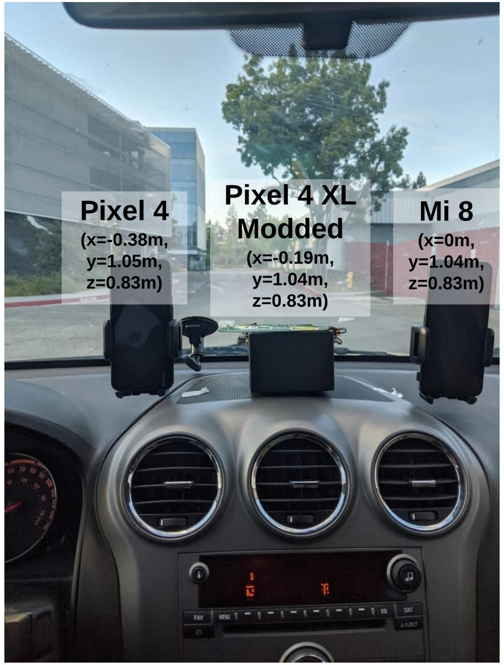
Pixel 4 XL

- Pixel4_GnssLog.txt
- Pixel4_chipset.nmea
- Pixel4_chipset.kml

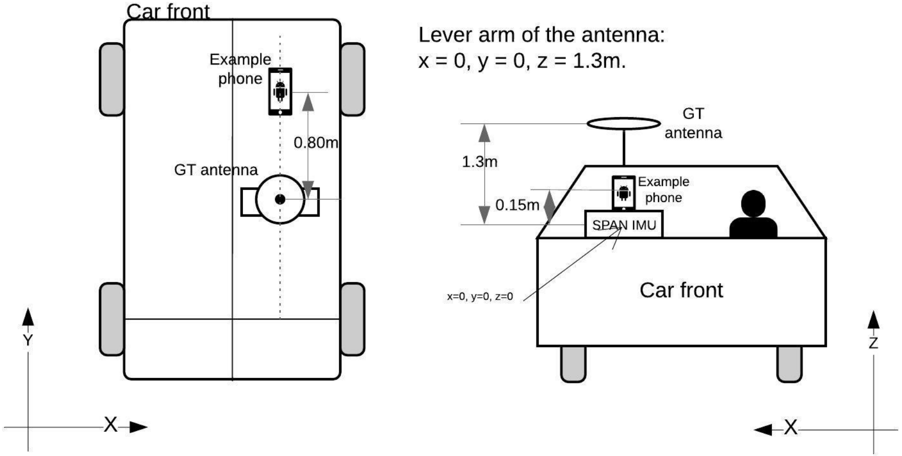
Pixel 4

SPAN IMU-to-DUT Lever Arm

- The vector from the SPAN IMU to the DUT.
- Lever arm of each DUT **has been compensated** in its GT NMEA file.



Example illustration of SPAN vehicle frame



Displacement of the **example** phone:
x = 0, y = 0.80m, z = 0.15m.

Lever arm of the antenna:
x = 0, y = 0, z = 1.3m.

Outline

SPAN IMU-to-DUT lever arm

NmeaUtils: DUT NMEA vs GT NMEA

Content of GnssLog

NMEA format

- SPAN_Pixel4XL_10Hz.nmea
- SPAN_Pixel4_10Hz.nmea
- SPAN_IMU_10Hz.nmea

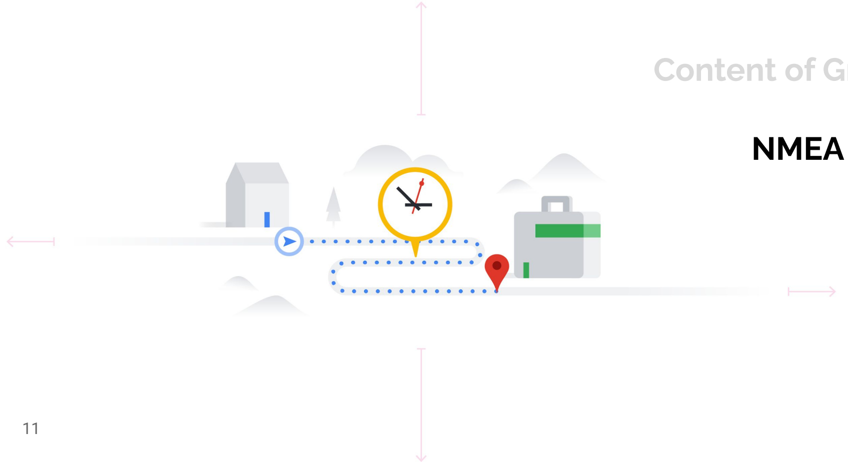
Ground truth NMEA files

- Pixel4XL_GnssLog.txt
- Pixel4XL_chipset.nmea
- Pixel4XL_chipset.kml

Pixel 4 XL

- Pixel4_GnssLog.txt
- Pixel4_chipset.nmea
- Pixel4_chipset.kml

Pixel 4



NMEA File Format

Google

Example NMEA snippet (for one position):

...

$GPGGA,234339.00,3718.8750459,N,12157.0677483,W,1,15,0.7,44.77,M,-32.69,M,,*64

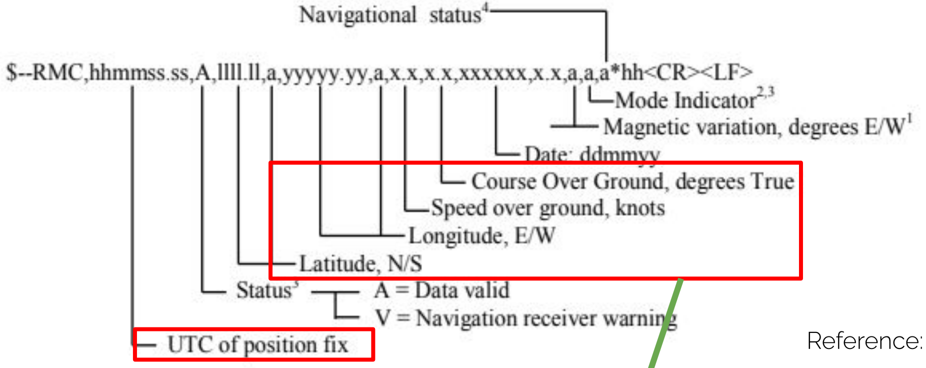
$GPRMC,234339.00,A,3718.875,N,12157.068,W,0.0,266.5,270720,,,A*21

...

RMC – Recommended Minimum Specific GNSS Data

Time, date, position, course and speed data provided by a GNSS navigation receiver. This sentence is transmitted at intervals not exceeding 2-seconds and is always accompanied by RMB when a destination waypoint is active.

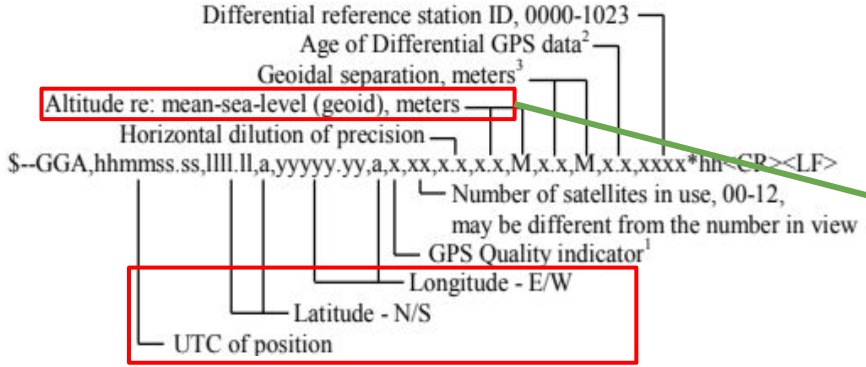
RMC and RMB are the recommended minimum data to be provided by a GNSS receiver. All data fields must be provided, null fields used only when data is temporarily unavailable.



Reference: [1]

GGA – Global Positioning System Fix Data

Time, position and fix related data for a GPS receiver.



- Timestamp
- Latitude
- Longitude
- Altitude (**GGA**)
- Speed over ground (**RMC**)
- Course over ground (**RMC**)

Outline

SPAN IMU-to-DUT lever arm

NmeaUtils: **DUT NMEA vs GT NMEA**

Content of GnssLog

NMEA format

- SPAN_Pixel4XL_10Hz.nmea
- SPAN_Pixel4_10Hz.nmea
- SPAN_IMU_10Hz.nmea

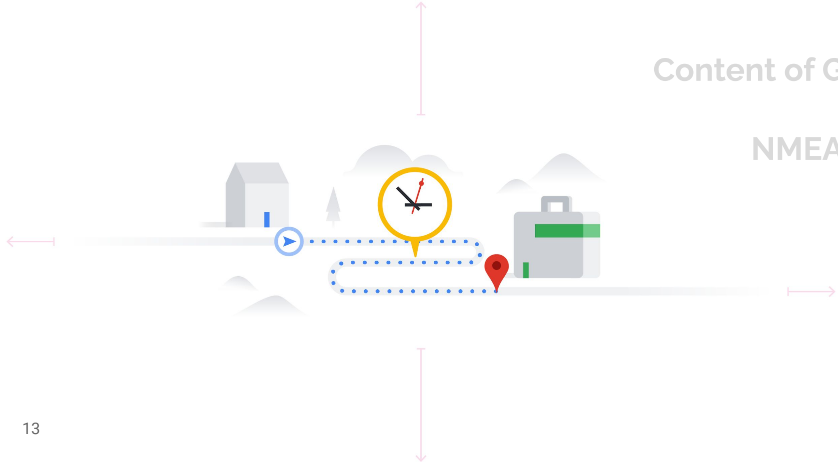
Ground truth NMEA files

- Pixel4XL_GnssLog.txt
- Pixel4XL_chipset.nmea
- Pixel4XL_chipset.kml

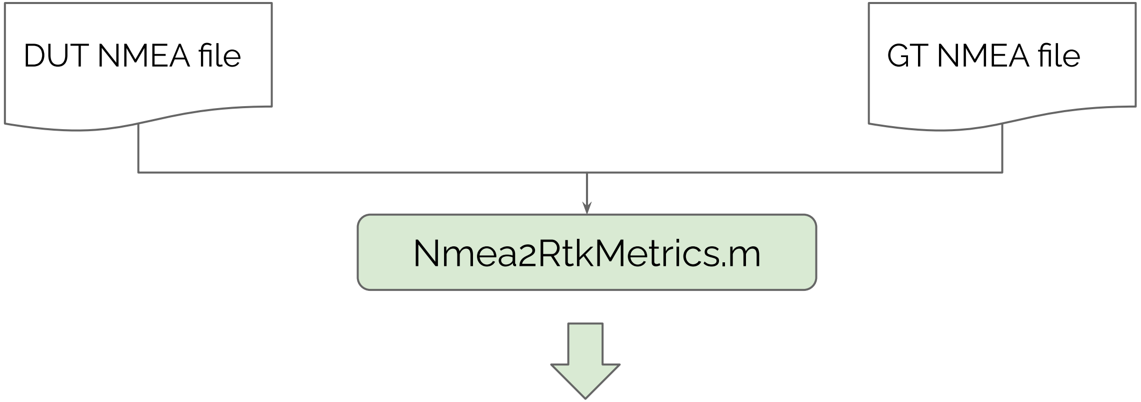
Pixel 4 XL

- Pixel4_GnssLog.txt
- Pixel4_chipset.nmea
- Pixel4_chipset.kml

Pixel 4



NmeaUtils: Nmea2RtkMetrics.m



NumValid Pts	50% (m)	80% (m)	95% (m)	XTrack 50% (m)	XTrack 80% (m)	XTrack 95% (m)	AlongTrack 50% (m)	AlongTrack 80% (m)	AlongTrack 95% (m)
1630	2.64	3.26	4	0.79	1.14	1.75	2.49	3.11	3.77

High Precision Positioning Metrics

NumValidPts: Number of valid solution points.

50% (m): 50-percentile of horizontal error (meters)
80% (m): 80-percentile of horizontal error (meters)
95% (m): 95-percentile of horizontal error (meters)

Horizontal positioning errors

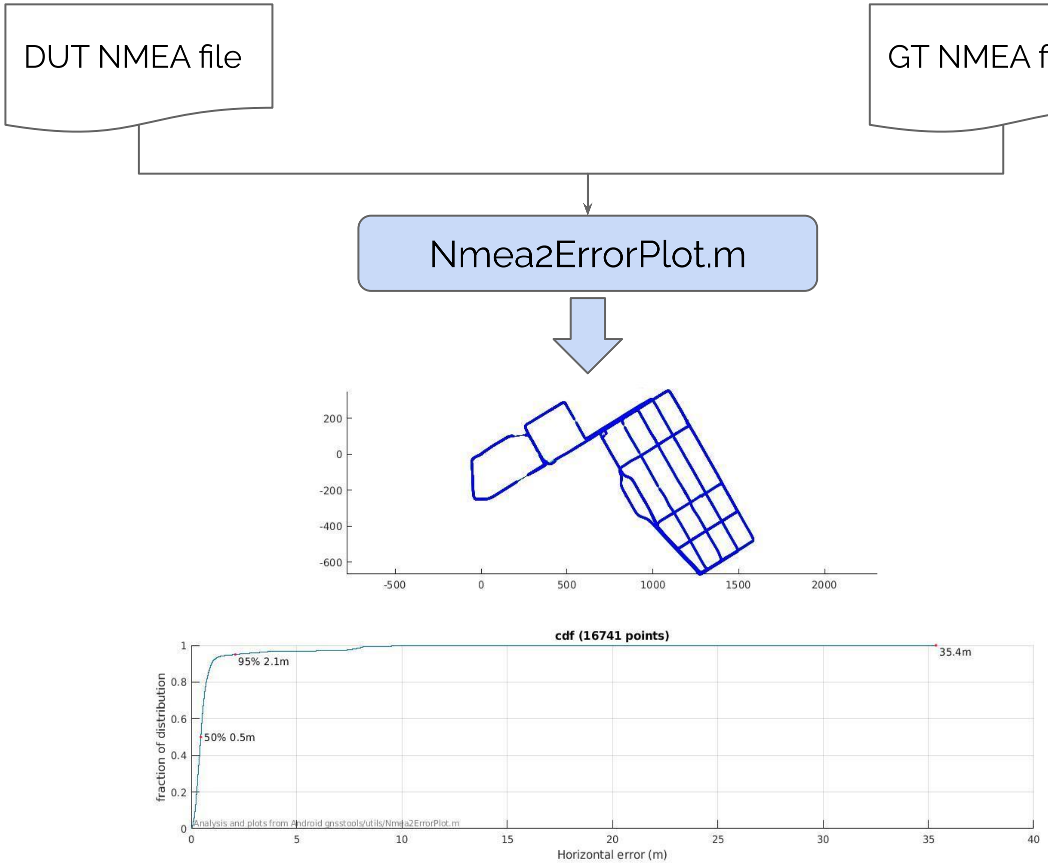
TT1M (s): Time (seconds) to 1-meter horizontal error
TA1M (s): Average consecutive time (seconds) of 1 meter or less horizontal error
AA5S (m): Horizontal accuracy (meters) at the 5th second
AA10S (m): Horizontal accuracy (meters) at the 10th second

Convergence metrics

X-track 50% (m): 50-percentile of cross-track horizontal error (meters)
X-track 80% (m): 80-percentile of cross-track horizontal error (meters)
X-track 95% (m): 95-percentile of cross-track horizontal error (meters)
Along-track 50% (m): 50-percentile of along-track horizontal error (meters)
Along-track 80% (m): 80-percentile of along-track horizontal error (meters)
Along-track 95% (m): 95-percentile of along-track horizontal error (meters)

Cross-track & along-track errors

NmeaUtils: Nmea2ErrorPlot.m



Outline

SPAN IMU-to-DUT lever arm

NmeaUtils: DUT NMEA vs GT NMEA

Content of GnssLog

NMEA format

- SPAN_Pixel4XL_10Hz.nmea
- SPAN_Pixel4_10Hz.nmea
- SPAN_IMU_10Hz.nmea

Ground truth NMEA files

- Pixel4XL_GnssLog.txt
- Pixel4XL_chipset.nmea
- Pixel4XL_chipset.kml

Pixel 4 XL

- Pixel4_GnssLog.txt
- Pixel4_chipset.nmea
- Pixel4_chipset.kml

Pixel 4

GnssLog Format - RAW sentence

Example RAW GNSS measurement sentence:

Raw,1595893567433,13326953000000,18,,-1279915458480313484,-0.7581386566162109,36.676308809546754,5.250991947053456,10.31209756655846 4,2788,8,0.0,16399,171985355993445,14,33.7,-440.52789306640625,0.20000000298023224,21,-11431.158460699902,0.002397700231676362,1.57542003 E9,,,,0,,1,1.85,,,,,,C,9534437134304

Field name	Description
Raw	Prefix of sentence
utcTimeMillis	Milliseconds since UTC epoch (1970/1/1), converted from GnssClock
TimeNanos	GnssClock#getTimeNanos()
LeapSecond	GnssClock#getLeapSecond()
TimeUncertaintyNanos	GnssClock#getTimeUncertaintyNanos()
FullBiasNanos	GnssClock#getFullBiasNanos()
BiasNanos	GnssClock#getBiasNanos()
BiasUncertaintyNanos	GnssClock#getBiasUncertaintyNanos()
DriftNanosPerSecond	GnssClock#getDriftNanosPerSecond()
DriftUncertaintyNanosPerSecond	GnssClock#getDriftUncertaintyNanosPerSecond()

Signal arrival time

Signal transmit time

Doppler

Field name	Description
HardwareClockDiscontinuityCount	GnssClock#getHardwareClockDiscontinuityCount()
Svid	GnssMeasurement#getSvid()
TimeOffsetNanos	GnssMeasurement#getTimeOffsetNanos()
State	GnssMeasurement#getState()
ReceivedSvTimeNanos	GnssMeasurement#getReceivedSvTimeNanos()
ReceivedSvTimeUncertaintyNanos	GnssMeasurement#getReceivedSvTimeUncertaintyNanos()
CnoDbHz	GnssMeasurement#getCn0DbHz()
PseudorangeRateMetersPerSecond	GnssMeasurement#getPseudorangeRateMetersPerSecond()
PseudorangeRateUncertaintyMetersPerSecond	GnssMeasurement#getPseudorangeRateUncertaintyMetersPerSecond()

More details: [5][6]

GnssLog Format - RAW sentence (Cont')

Field name	Description
AccumulatedDeltaRangeState	GnssMeasurement#getAccumulatedDeltaRangeState()
AccumulatedDeltaRangeMeters	GnssMeasurement#getAccumulatedDeltaRangeMeters()
AccumulatedDeltaRangeUncertaintyMeters	GnssMeasurement#getAccumulatedDeltaRangeUncertaintyMeters()
CarrierFrequencyHz	GnssMeasurement#getCarrierFrequencyHz()
CarrierCycles	GnssMeasurement#getCarrierPhase() Deprecated in API level 28 (Android P in 2018)
CarrierPhase	GnssMeasurement#getCarrierPhase() Deprecated in API level 28 (Android P in 2018)
CarrierPhaseUncertainty	GnssMeasurement#getCarrierPhaseUncertainty() Deprecated in API level 28 (Android P in 2018)
MultipathIndicator	GnssMeasurement#getMultipathIndicator()
SnrInDb	GnssMeasurement#getSnrInDb()

Carrier phase

Inter-Signal Biases

Chipset timestamp in Android time scale (used to sync IMU measurement)

Field name	Description
ConstellationType	GnssMeasurement#getConstellationType()
AgcDb	GnssMeasurement#getAutomaticGainControlLevelDb()
BasebandCn0DbHz	GnssMeasurement#getBasebandCn0DbHz() Added in API level 30 (Android 11 in 2020)
FullInterSignalBiasNanos	GnssMeasurement#getFullInterSignalBiasNanos() Added in API level 30 (Android 11 in 2020)
FullInterSignalBiasUncertaintyNanos	GnssMeasurement#getFullInterSignalBiasUncertaintyNanos() Added in API level 30 (Android 11 in 2020)
SatelliteInterSignalBiasNanos	GnssMeasurement#getSatelliteInterSignalBiasNanos() Added in API level 30 (Android 11 in 2020)
SatelliteInterSignalBiasUncertaintyNanos	GnssMeasurement#getSatelliteInterSignalBiasUncertaintyNanos() Added in API level 30 (Android 11 in 2020)
CodeType	GnssMeasurement#getCodeType() Added in API level 29 (Android 10 in 2019)
ChipsetElapsedRealtimeNanos	GnssClock#getElapsedRealtimeNanos() Added in API level 29 (Android 10 in 2019)

More details: [5][6]

Example Filters for Noisy Measurements

Measurements from GNSS chipsets of mobile phones are often noisier and more erroneous. Example of filters your can apply (to exclude) are:

1. Full bias nanoseconds is zero or invalid
2. BiasUncertaintyNanos too large
3. Arrival time is negative or unrealistically large
4. Unknown constellation
5. TimeNanos is empty
6. State is not STATE_TOW_DECODED
7. ReceivedSvTimeUncertaintyNanos is high
8. AdrState violating this condition: ADR_STATE_VALID == 1 & ADR_STATE_RESET == 0 & ADR_STATE_CYCLE_SLIP == 0
9. AdrUncertaintyMeters is high
10. … … more filters to be applied … …

GnssLog Format - Assisted sensors sentences

Uncalibrated Accelerometer

Header:
UncalAccel,**utcTimeMillis**,**elapsedRealtimeNanos**,UncalAccelXMps2,UncalAccelYMps2,UncalAccelZMps2,BiasXMps2,BiasYMps2,BiasZMps2
Example content:
UncalAccel,1594250738570,20337419021212,-1.1348436,9.876386,1.1284244,0.0,0.0,0.0

Uncalibrated Gyroscope

Header:
UncalGyro,**utcTimeMillis**,**elapsedRealtimeNanos**,UncalGyroXRadPerSec,UncalGyroYRadPerSec,UncalGyroZRadPerSec,DriftXRadPerSec,DriftYRadPerSec,DriftZRadPerSec
Example content:
UncalGyro,1594250738568,20337417901212,0.12336553,0.02968888,-0.014162418,4.348146E-4,3.291696E-4,-0.0012910228

Uncalibrated Magnetic Field

Header:
UncalMag,**utcTimeMillis**,**elapsedRealtimeNanos**,UncalMagXMicroT,UncalMagYMicroT,UncalMagZMicroT,BiasXMicroT,BiasYMicroT,BiasZMicroT
Example content:
UncalMag,1594250738582,20337431900795,30.27242,-59.439495,-27.946125,21.504183,-10.548593,-11.250341

SensorEvent#timestamp

- Chipset timestamp in Android time scale (elapsed time since **device boot**)
- Used to sync GNSS measurement for API 29 and above

utcTimeMillis

- Milliseconds since UTC epoch (1970/1/1)
- = **elapsedRealtimeNanos** + estimated UTC time at **device boot** after a network sync (NTP).

RINEX observation files converted from GnssLogs

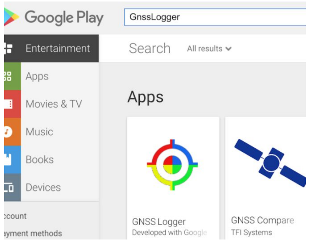
RINEX Data [4]	Android GnssMeasurementEvent APIs
Epoch time	Process GnssClock to compute the signal arrival time ("arrivalTime"). Floor it to 100 nanosecond level (F11.7). Adjust measurements using Doppler.
Code pseudorange (C)	1. Process GnssClock to compute arrivalTime. 2. Use GnssMeasurement.getReceivedSvTime*() to compute satellite transmit time. 3. Compute raw pseudorange by subtracting the two times above.
Carrier phase (L) cycles (Required in cycles. Please refer Page 11 Section 3.3 [4])	As per ADR(m) = wavelength * ADR(cycles) GnssMeasurement.getAccumulatedDeltaRangeMeters() / wavelength
Doppler (D) in Hz (Table A2 [4])	-1 * GnssMeasurement.getPseudorangeRateMetersPerSecond() / wavelength
Signal Strength (S) in dB-Hz	GnssMeasurement.getCn0DbHz()
Loss of Lock Indicator bit	ADR state is retrieved using GnssMeasurement.getAccumulatedDeltaRangeState() ● Blank = Unknown loss of lock status / ADR state is invalid or reset. ○ ADR state invalid or ADR state Reset ● 0 = OK ○ Default ● bit 1 (least significant bit) = loss of lock/cycle slip (observation should be discarded) ○ ADR state cycle slip ● bit 2 = half-cycle slip (observation should be discarded) ○ ADR state half cycle reported
Signal Strength Indicator bit	This is straightforward from section 5.7 of [4]

Summary

- Google will release:
  - ~60 smartphone traces in October, which include:
    - GnssLog files,
    - RINEX observation files,
    - chipset NMEA files,
    - ground truth NMEA files,
    - OSR and SSR corrections
  - a set of Matlab scripts to evaluate positioning results.

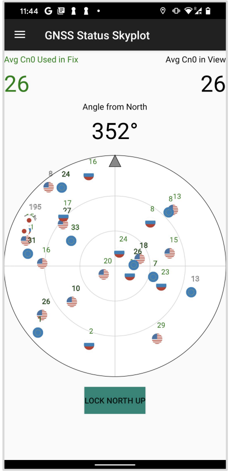
New updates in ***GnssLogger*** (App) and ***GnssAnalysisTools*** (Matlab program)

Download ***GnssLogger*** v3
in [Google PlayStore](https://play.google.com)
- Launch in Sep 2020

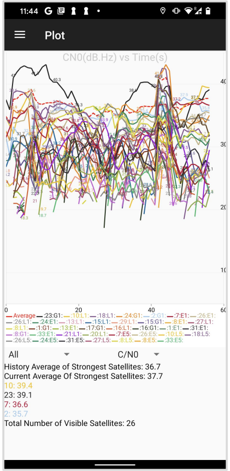


Download ***GnssAnalysisTools***
in g.co/gnsstools
- Launch in Oct 2020

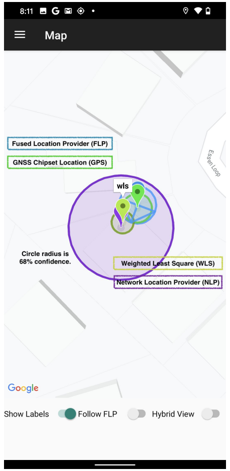
GnssStatus Table View



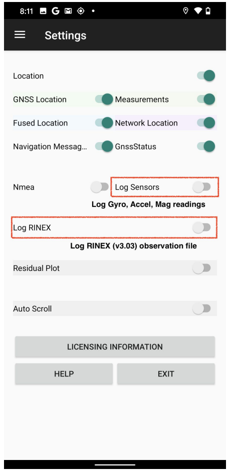
GnssStatus Skyplot View



CN0 & Residuals Plots



Locations Map View



Sensor & RINEX Logs

```
     3.03           OBSERVATION DATA    M                   RINEX VERSION / TYPE
GnssLogger          Google 10           20200813 215658 UTC PGM / RUN BY / DATE
GnssLogger                                                  MARKER NAME
        0.0000        0.0000        0.0000                  APPROX POSITION XYZ
G    8 C1C L1C D1C S1C C5Q L5Q D5Q S5Q                      SYS / # / OBS TYPES
R    4 C1C L1C D1C S1C                                      SYS / # / OBS TYPES
E    8 C1C L1C D1C S1C C5Q L5Q D5Q S5Q                      SYS / # / OBS TYPES
  2020    08    13    21    56   58.0000000     GPS         TIME OF FIRST OBS
 7 R04  6 R05  1 R06 -4 R07  5 R19  3 R20  2 R21  4         GLONASS SLOT / FRQ #
                                                            END OF HEADER
> 2020 08 13 21 56 58.4356890  0 21
G08  24465871.69525  -1255001.20025      3201.00425       33.20025  24463514.12724
0.00024      2391.10024        23.70024
G10  20951028.77025  -2837668.85425       910.11825       33.50025  20948693.98605
-1264681.89405       679.66605        30.40005
G15  23068868.22023        -0.00023     -2914.00023       21.60023
```

Android Raw GNSS Measurements in RINEX v3.03 Observation file

References

1. *National Marine Electronics Association. NMEA 0183--Standard for interfacing marine electronic devices. NMEA, 2002.*
2. *Google Scholar, searches with "android gnss measurements", result URL:* https://scholar.google.com/scholar?q=android+gnss+measurements+&hl=en&as_sdt=0%2C5&as_ylo=2016&as_yhi=2019
3. *Google Scholar, searches with "android gnss measurements (RTK OR PPP OR PRECISE)", result URL:* https://scholar.google.com/scholar?as_ylo=2016&q=android+gnss+measurements+(RTK+OR+PPP+OR+PRECISE)&hl=en&as_sdt=0,5
4. *RINEX the receiver independent exchange format v3.03 (2015).*
5. *The GSA GNSS Raw Measurements Task Force. [White Paper] Using GNSS Raw Measurements on Android devices, 2017. URL:*

   https://www.gsa.europa.eu/newsroom/news/available-now-white-paper-using-gnss-raw-measurements-android-devices
6. *Frank van Diggelen, Mohammed Khider. GNSS Analysis Tools from Google. InsideGNSS Magazine, Mar 2018. URL:* https://drive.google.com/open?id=1Y9yhJQXim2EsdH-3MoG0Z-jf98vBLH71

Thank you!

Questions? Please send to
android-gps-datasets@google.com.