Workshop for Google Smartphone Decimeter Challenge (SDC) 2023-2024

Michael Fu, Mohammed Khider, Frank van Diggelen, Dave Orendorff

September 12, 2023

ION GNSS+ 2023
Frank van Diggelen
Welcome!
Agenda

01  Welcome Note and SDC Introduction (Frank)
02  Data Collection Process (Mohammed)
03  Break
04  Data Format & Lessons Learned & Tips (Dave)
05  Getting Started with Python Notebook (Michael)
06  Other Lessons Learned & Tips (Mohammed)
07  Q&A
Google SDC 2023-2024 will launch at 3:30 PM MDT, September 12
**Google Smartphone Decimeter Challenge 2023**

Improve high precision GNSS positioning and navigation accuracy on smartphones.

**Time:** Sep 12, 2023 - May 23, 2024

**Platform:** [https://www.kaggle.com/competitions/smartphone-decimeter-2023/](https://www.kaggle.com/competitions/smartphone-decimeter-2023/)

**Sponsors:** Google, and Institute of Navigation.

**Data:**

- ADR* enabled phones:
  - Pixel, Samsung, Xiaomi.
- 196 different drive tests:
  - Tens of different routes
- All labeled with ground-truth positions and velocities collected with NovAtel SPAN ISA-100C, with precise lever-arm compensation.
- In-phone Raw GNSS measurements and Inertial Measurement Unit (IMU) readings

**Prizes:** $15,000 cash prizes, $1,700x3 travel incentive for presenting at ION GNSS+ 2024, hotel/free registration for ION GNSS+ 2024.

*ADR: Accumulated Delta Range = Carrier Phase*
Datasets, Score, and Leaderboards

⭐ 196 drive tests in 3 sets: training, public test, and private test.

⭐ Score = mean(50%, 95%)

Training dataset
- 156 traces
- Ground truth (GT) accessible.
- Multiple phones on each drive

Public test dataset
- 20 traces
- GT NOT accessible.
- One phone per drive
- Score displayed on public leaderboard.

Private test dataset
- 20 traces
- GT NOT accessible.
- One phone per drive
- Score recorded on private leaderboard
  NOT visible until end of competition.
The score: \text{mean}(50\%, 95\%)

A single number for quantitative evaluation of your results

**Definition**

mean of 50\% and 95\% distribution of horizontal (2d) errors

**Rationale: a single number**

A completely objective metric: Winners determined by this number only.

50\% and 95\% are both canonical metrics, and each are important.

**Interpreting a score**

The unit of the score is meters.

If \text{score} = x \text{(m)}, then median accuracy is roughly 0.6 \times x \text{(m)}.

---

1st place $7,000
2nd place $5,000
3rd place $3,000
## Differences from last year’s SDC

<table>
<thead>
<tr>
<th><strong>Larger Prizes</strong></th>
<th><strong>More Phones</strong></th>
<th><strong>More Varieties of Routes</strong></th>
<th><strong>Vetted Ground Truth</strong></th>
<th><strong>IMU Synchronization</strong></th>
</tr>
</thead>
</table>
| $15,000 cash + $1,700x3 travel incentive to present at ION GNSS+ 2024, hotel, registration to ION 2024 at Baltimore | **Previous:** Pixel, Samsung S, Xiaomi Mi8  
  ● High end phones  
  ● L1, L5  
**This year:** + Samsung A series  
  ● L1-only, noisier, more challenging  
  ● More affordable  
  ● Market majority (>50%) of ADR-enabled phones | Tens of different routes in SF Bay area and Los Angeles | Enhanced quality check against NovAtel SPAN solutions (detailed in later slides) | For some phones, chipset event-timestamps of IMU and raw GNSS measurement |
AI researchers use open datasets for validation.

GNSS community should adopt similar practices. Today typical papers have:
   ● very few real datasets to show results
   ● inconsistent metrics (50% here, rms there, etc)

The SDC data brings GNSS to the R&D standards of AI.

Use the training and test datasets in any paper you present on your position algorithms
Questions?
Data Collection Process

Google SDC 2023
Mohammed Khider
Collection Platform

Custom cars with:
- Phones mounted near the windshield
- Ground truth system mounted on the trunk
- Accurate lever arms measurements

Displacement of the example phone:
\[ x = 0, \ y = 0.80\text{m}, \ z = 0.15\text{m}. \]

Lever arm of the antenna:
\[ x = 0, \ y = 0, \ z = 1.3\text{m}. \]
How did we setup a Precisely Coordinated Test Vehicle?

- Built a stable mount for the reference receiver
- 3D printed phone mounts
- mm-level lever arm compensation using robotic arm

NovAtel SPAN

Alignment pins
GnssLogger v3.0.6.3
- Updated .txt log to include Location.isMock
- Fixed logging Automatic Gain Control to file and UI on Android 13 and up
- “Force Full Tracking” option
- “Keep Screen On” option
- Updated ground truth options for residual plot
- Many bug fixes / performance improvements

GNSS Analysis app v4.6.0.1
- Added a table of C/N0 comparisons, by constellation, and comparing L1 to L5.
- Added pseudorange rate residual plot.
- Adjustment related to changes of NASA CDDIS and BKG servers

GNSS tools available at g.co/gnsstools
**Vetting Phone Logs**

**Any phone log must have...**
- No cross-track bias vs WLS position
- SPAN RTK integer fixes > 90%
  - GNSS raw valid > 60%
  - GNSS valid ADR > 5%
- Discontinuous clock < 10%

**Any test set log must have...**
- Phone model appears in Train set at least once
- Not in train set of previous SDC
- Public/private test 50:50 random split
Ground Truth Vetting
Is there a “ground truth” for the ground truth?

- Verified with laser ranger.
- Prism installed on top of car
- Compared computed distances
- Correct then and there, not necessarily correct everywhere.
Horizontal Error Distributions of Weighted Least Squares (WLS) solutions

GT good

GT inaccurate

GT inaccurate
Cross-Track Error Distributions of WLS solutions

GT good

GT inaccurate

GT inaccurate
Summary of Improved Ground Truth Quality

*Ground truth is the foundation of Google SDCs. This year, we did these steps to improve its quality.*

- SPAN RTK integer fixes > 90%
- Unbiased-Zero-Mean of cross-track error from WLS positions
- Unbiased-Zero-Mean of horizontal error from WLS position
- Re-process traces with suspicious ground truth problems
Questions?
Break time
Dave Orendorff
Data directory

./train/2020-06-25-00-34-us-ca-mtv-sb-101/pixel4/

Collection name ("date-time-route") Phone

- device_gnss.csv → Raw and derived GNSS values (interpolated)
- device_imu.csv → Raw IMU values
- ground_truth.csv → Interpolated ground truth values at GNSS time
- supplemental/
  - gnss_log.txt → Same information, in traditional formats
  - span_log.nmea
### device_gnss.csv structure

58 columns...
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td>Prefix of sentence</td>
</tr>
<tr>
<td>utcTimeMillis</td>
<td>Milliseconds since UTC epoch (1970/1/1), converted from GnssClock</td>
</tr>
<tr>
<td>TimeNanos</td>
<td>GnssClock#getTimeNanos()</td>
</tr>
<tr>
<td>LeapSecond</td>
<td>GnssClock#getLeapSecond()</td>
</tr>
<tr>
<td>TimeUncertaintyNanos</td>
<td>GnssClock#getTimeUncertaintyNanos()</td>
</tr>
<tr>
<td>FullBiasNanos</td>
<td>GnssClock#getFullBiasNanos()</td>
</tr>
<tr>
<td>BiasNanos</td>
<td>GnssClock#getBiasNanos()</td>
</tr>
<tr>
<td>BiasUncertaintyNanos</td>
<td>GnssClock#getBiasUncertaintyNanos()</td>
</tr>
<tr>
<td>DriftNanosPerSecond</td>
<td>GnssClock#getDriftNanosPerSecond()</td>
</tr>
<tr>
<td>DriftUncertaintyNanosPerSecond</td>
<td>GnssClock#getDriftUncertaintyNanosPerSecond()</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>HardwareClockDiscontinuityCount</td>
<td>\texttt{GnssClock#getHardwareClockDiscontinuityCount()}</td>
</tr>
<tr>
<td>Svid</td>
<td>\texttt{GnssMeasurement#getSvid()}</td>
</tr>
<tr>
<td>TimeOffsetNanos</td>
<td>\texttt{GnssMeasurement#getTimeOffsetNanos()}</td>
</tr>
<tr>
<td>State</td>
<td>\texttt{GnssMeasurement#getState()}</td>
</tr>
<tr>
<td>ReceivedSvTimeNanos</td>
<td>\texttt{GnssMeasurement#getReceivedSvTimeNanos()}</td>
</tr>
<tr>
<td>ReceivedSvTimeUncertaintyNanos</td>
<td>\texttt{GnssMeasurement#getReceivedSvTimeUncertaintyNanos()}</td>
</tr>
<tr>
<td>CnoDbHz</td>
<td>\texttt{GnssMeasurement#getCnoDbHz()}</td>
</tr>
<tr>
<td>PseudorangeRateMetersPerSecond</td>
<td>\texttt{GnssMeasurement#getPseudorangeRateMetersPerSecond()}</td>
</tr>
<tr>
<td>PseudorangeRateUncertaintyMetersPerSecond</td>
<td>\texttt{GnssMeasurement#getPseudorangeRateUncertaintyMetersPerSecond()}</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>AccumulatedDeltaRangeState</td>
<td><code>GnssMeasurement#getAccumulatedDeltaRangeState()</code></td>
</tr>
<tr>
<td>AccumulatedDeltaRangeMeters</td>
<td><code>GnssMeasurement#getAccumulatedDeltaRangeMeters()</code></td>
</tr>
<tr>
<td>AccumulatedDeltaRangeUncertaintyMeters</td>
<td><code>GnssMeasurement#getAccumulatedDeltaRangeUncertaintyMeters()</code></td>
</tr>
<tr>
<td>CarrierFrequencyHz</td>
<td><code>GnssMeasurement#getCarrierFrequencyHz()</code></td>
</tr>
<tr>
<td>CarrierCycles</td>
<td><code>GnssMeasurement#getCarrierPhase()</code></td>
</tr>
<tr>
<td></td>
<td>Deprecated in API level 28 (Android P in 2018)</td>
</tr>
<tr>
<td>CarrierPhase</td>
<td><code>GnssMeasurement#getCarrierPhase()</code></td>
</tr>
<tr>
<td></td>
<td>Deprecated in API level 28 (Android P in 2018)</td>
</tr>
<tr>
<td>CarrierPhaseUncertainty</td>
<td><code>GnssMeasurement#getCarrierPhaseUncertainty()</code></td>
</tr>
<tr>
<td></td>
<td>Deprecated in API level 28 (Android P in 2018)</td>
</tr>
<tr>
<td>MultipathIndicator</td>
<td><code>GnssMeasurement#getMultipathIndicator()</code></td>
</tr>
<tr>
<td>SnrInDb</td>
<td><code>GnssMeasurement#getSnrInDb()</code></td>
</tr>
</tbody>
</table>
### CSV Format - Extended RAW sentence - Raw Values

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ConstellationType</strong></td>
<td><code>GnssMeasurement#getConstellationType()</code></td>
</tr>
<tr>
<td><strong>AgcDb</strong></td>
<td><code>GnssMeasurement#getAutomaticGainControlLevelDb()</code></td>
</tr>
<tr>
<td><strong>BasebandCnoDbHz</strong></td>
<td><code>GnssMeasurement#getBasebandCnoDbHz()</code></td>
</tr>
<tr>
<td><strong>FullInterSignalBiasNanos</strong></td>
<td><code>GnssMeasurement#getFullInterSignalBiasNanos()</code></td>
</tr>
<tr>
<td></td>
<td>Added in API level 30 (Android 11 in 2020)</td>
</tr>
<tr>
<td><strong>FullInterSignalBiasUncertaintyNanos</strong></td>
<td><code>GnssMeasurement#getFullInterSignalBiasUncertaintyNanos()</code></td>
</tr>
<tr>
<td></td>
<td>Added in API level 30 (Android 11 in 2020)</td>
</tr>
<tr>
<td><strong>SatelliteInterSignalBiasNanos</strong></td>
<td><code>GnssMeasurement#getSatelliteInterSignalBiasNanos()</code></td>
</tr>
<tr>
<td></td>
<td>Added in API level 30 (Android 11 in 2020)</td>
</tr>
<tr>
<td><strong>SatelliteInterSignalBiasUncertaintyNanos</strong></td>
<td><code>GnssMeasurement#getSatelliteInterSignalBiasUncertaintyNanos()</code></td>
</tr>
<tr>
<td></td>
<td>Added in API level 30 (Android 11 in 2020)</td>
</tr>
<tr>
<td><strong>CodeType</strong></td>
<td><code>GnssMeasurement#getCodeType()</code></td>
</tr>
<tr>
<td></td>
<td>Added in API level 29 (Android 10 in 2019)</td>
</tr>
<tr>
<td><strong>ChipsetElapsedRealtimeNanos</strong></td>
<td><code>GnssClock#getElapsedRealtimeNanos()</code></td>
</tr>
<tr>
<td></td>
<td>Added in API level 29 (Android 10 in 2019)</td>
</tr>
<tr>
<td>Field name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>ArrivalTimeNanosSinceGpsEpoch</td>
<td>from Arrival time</td>
</tr>
<tr>
<td>RawPseudorangeMeters</td>
<td>(arrival time - transmit time) x c</td>
</tr>
<tr>
<td>RawPseudorangeUncertaintyMeters</td>
<td>from receivedSvTimeUncertaintyNanos</td>
</tr>
<tr>
<td>SignalType</td>
<td>GPS_L1, GPS_L5, GLO_G1, etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IsrbMeters</td>
<td>Inter-Signal Range Bias wrt. GPS_L1</td>
</tr>
<tr>
<td>IonosphericDelayMeters</td>
<td>from Klouchar Model</td>
</tr>
<tr>
<td>TroposphericDelayMeters</td>
<td>from Hopfield model</td>
</tr>
<tr>
<td>WlsPositionXEcefMeters</td>
<td>Weighted Least Squares solution with</td>
</tr>
<tr>
<td></td>
<td>minimal heuristic filters or smoothers</td>
</tr>
<tr>
<td>WlsPositionYEcMeters</td>
<td></td>
</tr>
<tr>
<td>WlsPositionZEcefMeters</td>
<td></td>
</tr>
</tbody>
</table>
### Derived Values

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReceivedSvTimeNanosSinceGpsEpoch</td>
<td>Signal transmission time</td>
</tr>
<tr>
<td>SvPositionXEcefMeters</td>
<td></td>
</tr>
<tr>
<td>SvPositionYEcefMeters</td>
<td></td>
</tr>
<tr>
<td>SvPositionZEcefMeters</td>
<td></td>
</tr>
<tr>
<td>SvElevationDegrees</td>
<td>Satellite PVT from RINEX ephemeris.</td>
</tr>
<tr>
<td>SvAzimuthDegrees</td>
<td></td>
</tr>
<tr>
<td>SvVelocityXEcefMetersPerSecond</td>
<td></td>
</tr>
<tr>
<td>SvVelocityYEcefMetersPerSecond</td>
<td></td>
</tr>
<tr>
<td>SvVelocityZEcefMetersPerSecond</td>
<td></td>
</tr>
<tr>
<td>SvClockBiasMeters</td>
<td></td>
</tr>
<tr>
<td>SvClockDriftMetersPerSecond</td>
<td></td>
</tr>
</tbody>
</table>
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). 
Synchronize GNSS and IMU Measurements

<table>
<thead>
<tr>
<th></th>
<th>ChipsetElapsedRealtimeNanos in GNSS CSV</th>
<th>ElapsedRealtimeNanos in IMU CSV</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Case 1</strong></td>
<td>✅ Available</td>
<td>✅ Available</td>
</tr>
<tr>
<td><strong>Case 2</strong></td>
<td>❌ Not available</td>
<td>✅ Available</td>
</tr>
<tr>
<td><strong>Case 3</strong></td>
<td>✅ Available</td>
<td>❌ Not available</td>
</tr>
<tr>
<td><strong>Case 4</strong></td>
<td>❌ Not available</td>
<td>❌ Not available</td>
</tr>
</tbody>
</table>

**Best case (Case 1):**
- Sub-millisecond time sync precision. Recommend to use IMU.

**Other cases (Cases 2, 3, 4):**
- Not guaranteed for precise sync. Use IMU at your own risk.
CSV Format - Ground Truth *(ground_truth.csv)*  
*consistent with gnss_logger syntax*

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageType</td>
<td>Fix</td>
</tr>
<tr>
<td>Provider</td>
<td>GT</td>
</tr>
<tr>
<td>LatitudeDegrees</td>
<td>latitude</td>
</tr>
<tr>
<td>LongitudeDegrees</td>
<td>longitude</td>
</tr>
<tr>
<td>AccuracyMeters</td>
<td>horizontal accuracy (m)</td>
</tr>
<tr>
<td>BearingDegrees</td>
<td>bearing</td>
</tr>
<tr>
<td>UnixTimeMillis</td>
<td>Timestamp of measurement*</td>
</tr>
</tbody>
</table>

* interpolated to match raw measurement
NMEA File Format

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
...

GGA – Global Positioning System Fix Data
Time, position and fix related data for a GPS receiver.

- Differential reference station ID, 0000-1023
- Age of Differential GPS data
- Geoidal separation, meters
- Horizontal dilution of precision

$--GGA,hhmmsss,llllll.l,yyyy.yyy,a,x,x,x,x,xxxx,M,xx,M,xx,xxxx*hh<CR><LF>

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]
Questions?
Michael Fu
Get Started with Python Notebooks
Google SDC 2023
Collegial and Wide-Ranging Discussions

Popular notebooks:

- GSDC2 - baseline submission
  Updated 4mo ago
  Score: 4.87 - 12 comments - Google Smartphone Decimeter Challenge 2022

- Carrier Smoothing + Robust WLS + Kalman Smoother
  Notebook copied with edits from a private notebook - Updated 2mo ago
  Score: 3.027 - 11 comments - Google Smartphone Decimeter Challenge 2022

- Smartphone Competition 2022 [Twitch Stream]
  Updated 4mo ago
  Score: 4.823 - 5 comments - Google Smartphone Decimeter Challenge 2022

- Getting started with RTKLIB
  Updated 3mo ago
  Score: 4.75 - Google Smartphone Decimeter Challenge 2022

- GSDC22 - Coordinate with NearestNeighbors
  Notebook copied with edits from a private notebook - Updated 2mo ago
  Score: 3.889 - 3 comments - Google Smartphone Decimeter Challenge 2022

- GSDC2 - Savag Filter + Outlier Removal with BO
  Updated 4mo ago
  Score: 4.376 - 0 comments - Google Smartphone Decimeter Challenge 2022

- GSDC22 - Coordinate with NearestNeighbors [V2]
  Notebook copied with edits from a private notebook
  Score: 3.889 - 3 comments - Google Smartphone Decimeter Challenge 2022

- Deriving baseline WLS positions (in progress)
  Updated 4mo ago
  Score: 4 comments - Google Smartphone Decimeter Challenge 2022

- Carrier Smoothing + Robust WLS + Kalman Smoother v2
  Notebook copied from Taro - Updated 2mo ago
  Score: 3.994 - 3 comments - Google Smartphone Decimeter Challenge 2022

- GSDC22 - Coordinate with NearestNeighbors [V3]
  Notebook copied with edits from a private notebook - Updated 2mo ago
  Score: 3.996 - 4 comments - Google Smartphone Decimeter Challenge 2022

Example discussions:

- WLS
- Smoothing
- Visualization
- RTKLIB
- Ensemble
- Outlier removal

RTKLIB queries
Posted in smartphone-decimeter-2022 4 months ago

[Long post. Only relevant to anybody else who’s working with RTKlib.]

With many thanks to @Timeyett for his great work with RTKlib, I've been trying to get to grips with it for the first time and have a few questions. Sorry if these are very basic/stupid questions, I’m completely new to this.

[Edit: Questions struck through once I’ve fixed the problem] I am happy that I understand what’s going on.

RTKLIBexplorer • 6th in this Competition) • 4 months ago • Options • Report • Reply

Good questions! I'll attempt to answer them below:

1. To include a GNSS constellation in the solution, RTKLIB requires rover constellation. The base observations are in the identical format of the signal strength measurements by satellite and epoch.

2. Looking at the plotted residuals after plotting the solution in RTKLIB is

Question for top-teams: are your solutions real-time?

Posted in smartphone-decimeter-2022 2 months ago

May I ask if your solutions are real-time localization or need post-processing? One of my recent projects requires real-time precise localization, but I followed some public open-source solutions and found most solutions require whole route information in this competition (i.e., access to all data after route completion) and then do filtering. I appreciate it if you can help.

SolvewWorld • 127th in this Competition) • a month ago • Options • Report • Reply

Many people, myself included, are doing Kalman Smoothing, which effectively runs a Kalman filter forward, and then backward over the results, combining the results appropriately. This means that future data is used for positioning, and thus is not a real-time method.
Insights:
1. Breakthrough of 5m → 2m within 3 weeks. 2m → 1.2m takes the rest 10 weeks.
2. Breakthroughs were made by the leaders (e.g. RTKLIB introduced to the forum; Taro's two-step FGO). Then others followed closely.
3. Small improvements were made collectively, through forum discussions and solution sharings.

**Summary:** Common goal, platform, data, metrics, and award incentives enabled successful and rapid technical advancement!
Trend of **Best Score** and **Leading Solutions**
(Best score = 80% * best public score + 20% * best private score)

Details about leading solutions:

- **FGO (Factor Graph Optimization)**
  - Taro Suzuki: [paper](#), [post](#).

- **PPK float (Post-processed kinematic)**:
  - Tim Everett: papers (1, 2), post (1, 2), [code](#), and [RTKLIBExplorer](#).

- **FGO+TensorFlow**:
  - Akio Saito: [post](#).
Use Kaggle Notebooks

Michael Fu

Google
Submit your solutions in SDC 2023

**Step 1:**
Register as a Kaggle user, and log into SDC 2023: [https://www.kaggle.com/competitions/smartphone-decimeter-2023/](https://www.kaggle.com/competitions/smartphone-decimeter-2023/)

**Step 2:** Download data

```
> kaggle competitions download -c smartphone-decimeter-2022
```

**Step 3:** Submit and evaluate

OR

Put your solutions in one csv file
Questions?
Other Lessons Learned & Tips
Google SDC 2023
Example Filters for Noisy Measurements

Measurements from GNSS chipsets of mobile phones are often noisier and more erroneous.  

**Example of filters you 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
**Phase Center Offset (PCO) and Variations (PCV)**

**Phase Center Variations** is the additional delay added in each signal's propagation.

- **Phase center is:**
  - the point where GNSS signals are received

- **Antenna reference point is:**
  - the point a positioning engine should point to by definition,
  - and the center of primary screen,
  - and also the origin of Android sensor coordinate system.

Not provided, but would be helpful.
## Chipset-Specific Measurement Quality

Performance tuning: categorize by GNSS chipsets first, and then by phone models.

<table>
<thead>
<tr>
<th>Qualcomm</th>
<th>Broadcom</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Pixel 4 (XL), 5 (a)</td>
<td>● Pixel 6 (Pro), 7 (Pro), Xiaomi Mi8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Samsung LSI (Exynos)</th>
<th>MediaTek</th>
</tr>
</thead>
<tbody>
<tr>
<td>● S8+, S20/S21/S22/S23 Ultra (5G), A6, A20, A21s, A32 (5G), A50</td>
<td>● A22 5G</td>
</tr>
<tr>
<td>● GLONASS sometimes has quality issues*</td>
<td>● ADR state reset bit not being set*</td>
</tr>
<tr>
<td>● Had a reverted-ADR-sign issue before*</td>
<td></td>
</tr>
</tbody>
</table>

*They are known issues, have been fixed by the chipset manufacturers, and *might* have rolled out to the phones in test.
Datum and Tectonic Movement at California

Base station coordinate settings at NovAtel Inertial Explorer GUI:

- **Up to 50cm error** in station coordinate, when selecting 2010 instead of 2020.

Tectonic movement in Western US: 1-5cm per year

Source: UNAVCO
The SDC collection of >150 labeled datasets (>200,000 epochs) with verified GT can bring GNSS to the R&D standards of AI.

Use the training datasets to evaluate your position algorithms, and to present your results.
Thank you and good luck!

Questions?

Acknowledgments:

Thanks to the Institute of Navigation (ION) for co-sponsoring the Google Smartphone Decimeter Challenges.

Thanks to Tolu Ojelade, James Mitchell, Imad Fattouch, Vivek Gulati, and other Google and Kaggle colleagues for their contributions to data preparation and organization of the competition.