Mobile Devices as Digital Sextants for Zero-Permission Geolocation

Lorenz Schwittmann, Matthäus Wander and Torben Weis
Distributed Systems Group, University of Duisburg-Essen, 47048 Duisburg, Germany

Keywords: Geolocation, Tracking, Privacy, Ambient Light Sensor, Mobile Web Security.

Abstract: Sensors in mobile devices can be used to infer information about a user's context, most notably the location. Android apps and websites shown in Firefox for Android allow software to read the ambient light sensor, gyroscope and accelerometer without asking the user for permission. We show that these three sensors are sufficient to determine the rough geographical location of a user by turning the mobile device into a digital sextant. Despite low-quality sensor data, our approach is able to determine the position of the sun and thereby the geographical area where the user is located. Our approach works even if the user holding the device does not cooperate in being located or employs location-disguising techniques such as a VPN. We analyze in detail the different error sources and show in which settings and situations our approach works best. The location accuracy was at best 146 km with a medium accuracy better than 500 km. Truncating the positional sensor readings minimizes the privacy threat, while truncation of the ambient light sensor has almost no effect.

1 INTRODUCTION

Geolocation determines the location of a user's device in the world. Positioning or geolocating services are a standard feature offered by smartphones and other mobile devices. They achieve a high accuracy by combining satellite navigation (e.g., GPS, Galileo or GLONASS), cell tower and Wi-Fi triangulation. The location of a mobile device is synonymous to the location of its user and therefore subject to privacy concerns. Thus, operating systems like Android or iOS, or platforms like web browsers ask for user permission before giving location access to applications.

Unfortunately, adversaries can utilize side channels to determine the user's location without consent. A prominent example is IP address geolocation, which allows country-level or city-level geolocation (Triukose et al., 2012). Hiding the IP address protects from this type of geolocation, for example with a VPN, proxy server or Tor. Prior work has shown that freely accessible inertial sensors provide enough information to infer the trajectory of moving mobile devices such as acceleration of metro lines (Hua et al., 2017) or cars in city streets (Han et al., 2012). Even the phone's power meter can be used to infer the location based on cellular radio power consumption (Michalevsky et al., 2015). This enables the geolocation of the user even when hiding the IP address with a VPN. These approaches have in common that they require a systematic charting of a given area subject to specific features before geolocation becomes possible.

In this paper, we propose a zero-permission geolocation method based on mobile sensors, which does not require prior charting nor user permission. We determine the altitude of the unobstructed sun by creating a digital sextant based on the *ambient light sensor* (ALS) and *accelerometer*. Combined with a compass, this allows for geolocation without prior training or cartography anywhere on Earth where the sun is visible. However, *magnetometers* in smartphones used as compasses are too inaccurate for this purpose, especially because we cannot expect the user to calibrate them when they are unaware of being geolocated. Our method compensates for magnetometer deficiencies by inferring the sun's movement with time-delayed measurements, from which we can derive the user's location on Earth.

The sensors used by our method are accessible by Android apps or even websites on Firefox for Android without requesting user permission. The major challenge are inaccurate sensor readings by uncalibrated sensors on consumer devices. Yet, our web-based implementation achieves an accuracy of better than 500 km in 50% of our measurements. This shows that malicious websites can perform a country-level geolocation even when the user employs a VPN to hide their location.

Schwittmann, L., Wander, M. and Weis, T.
Mobile Devices as Digital Sextants for Zero-Permission Geolocation.
DOI: 10.5220/0007254000550066
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019)*, pages 55-66
ISBN: 978-989-758-359-9

The **contributions** of this paper include:

1. A novel approach to locate mobile devices without any infrastructure support.
2. A systematic evaluation of its accuracy, depending on latitude and sensor error.
3. An analysis of countermeasures based on reduced sensor resolution.

The paper is organized as follows: Section 2 motivates an attack use case. Section 3 provides astronomical background that the attack relies on. Section 4 describes assumptions about the threat model and Section 5 the geolocation method. Section 6 describes the implementation and how to handle practical interferences. Section 7 analyzes the practical applicability, while Section 8 evaluates the location accuracy and causes of measurement errors systematically. Section 9 discusses countermeasures and their effectiveness. Section 10 compares our work with related approaches.

2 USE CASE

Our digital sextant achieves an accuracy suitable for country-level geolocation. Although this is a coarse result, it can be used as part of a multi-level approach to bootstrap a more accurate geolocation method. There is a number of approaches for location tracking, which require either the starting point or at least the approximate area of where the user resides (Li et al., 2018; **?**; **?**; **?**; **?**; Han et al., 2012). Based on an approximate position, they allow to infer vehicular movement on a street map or similar approaches. Without any prior areal indication at all, there are too many potential matches and the resources required to process global map data renders the geolocation attempt infeasible. Our method thus yields the approximate area of the user, which can be then narrowed down with a computationally-expensive method to a specific location.

3 BACKGROUND

Our method is based on knowledge about planetary movements and celestial navigation combined with sensors available in smartphones.

3.1 Celestial Navigation

Celestial navigation relies on knowledge about perceived positions of celestial bodies depending on observation time and place on Earth. To describe celestial positions from a local observer's perspective, astronomers use the horizontal coordinate system (Figure 1). In it, every celestial object can be defined using two angles: *altitude* describes elevation from the observer's local horizon and *azimuth* the clockwise angle between north and that point on the horizon below the celestial body.

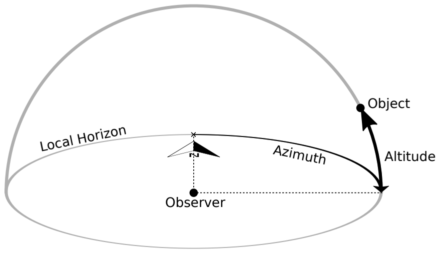

Figure 1: The horizontal coordinate system.

Combined with a precise location information of the observer (latitude/longitude) and time of the observation, this uniquely defines a celestial position and allows for identification of that celestial body. In the opposite case, if horizontal coordinates, time and celestial body are given, the observer's location is uniquely defined, which we utilize in our method.

3.2 Planetary Movements

While the general model of Earth orbiting the sun is well-known, various factors have to be taken into consideration to precisely predict its celestial position. Among these are ecliptic (tilt between rotational axis and orbital axis), exact orbital period (leap years/seconds), gravitational influences, day of year and time of day. Even then, there is no universal formula to describe celestial bodies precisely. Instead astronomers rely on fundamental ephemeris (i.e., tables of positions of celestial objects and their movements) which can be used to predict future positions. These calculations can be performed by broadly available astronomic programming libraries such as PyAstronomy (Czesla, 2013) with high accuracy.

3.3 Smartphone Sensors

Due to its brightness the sun is especially suited for detection using an ambient light sensor (ALS). Usually, ALS are mounted on the mobile device's front near the camera and prefaced by a lens to sample ambient light from a wide angle. Silicon photodiodes used in ALS are sensitive to a broad spectrum of light. To approximate human perception of illuminance and restrict that sensor sensitivity to visible light, filtering techniques are used to ultimately yield illuminance in

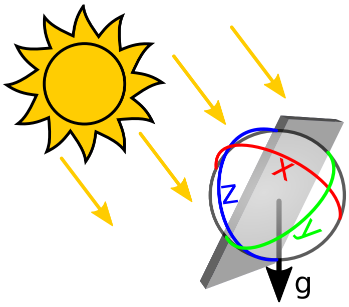

Figure 2: The mobile device records ambient light and positional sensors.

the photometry unit Lux. Nearly every smartphone and tablet is equipped with an ALS to adjust screen brightness, allowing for a sensor-based trade-off between screen readability and energy consumption. Besides its user interface usage, ALS readings are also exposed to applications on Android and in the Web API without any dedicated permission requirements. Both APIs yield illuminance as IEEE 754 floating points.

4 THREAT MODEL

As illustrated in Figure 2, a user holds their smartphone or another smart device while being exposed to direct sunlight. The user takes reasonable measures to hide their location such as using a VPN and thereby rendering IP address-based geolocation useless. Unconscious movements by the user cause the smart device to be facing in various directions.

An attacker aims to geolocate the user without their consent. The attacker can execute code without special permissions on the user's device, in particular without access to the geolocation API. The malicious code reads the ambient light sensor, accelerometer and magnetometer of the user's device. This assumption is met by installing a seemingly harmless app or simply visiting an HTML5 website, since the sensors of interest to our method are exposed via Web APIs (Tibbett et al., 2016; Kostiainen et al., 2017).

We furthermore assume the attacker runs another measurement on the user's device after one to six hours and observes a different position of the sun. This could be achieved by running the app in background or because the user visits the website a second time.

5 METHOD

Our method to locate a mobile device consists of 1. measuring directional ambient light at two points in time, 2. processing these measurements to find out the sun's altitude and azimuth, 3. calculating location candidates and 4. finally aggregating them to a position. We will discuss each step in detail in the following sections.

5.1 Measurements

We start by continuously reading accelerometer, magnetometer and ambient light sensors with corresponding time stamps. Since Android apps or websites do not require additional permissions for this, this data collection can be conducted without the user's permission or even awareness.

Since our approach does not control the user's movement, we merely assume the ambient light sensor eventually points towards the sun. We refer to this set of collected sensor readings as one *measurement*. To overcome inaccurate magnetometer values, we use at least two measurements from different points in time, which will be merged together in Section 5.3.

5.2 Preprocessing

Given a measurement, we want to locate the sun and calculate its altitude and azimuth.

For this, all sensor readings are transformed to a horizontal coordinate system. The accelerometer presents its values as a three dimensional vector (x, y, z) where z represents the front/back forces acting on the device. Assuming gravitational force is the main component, we can use the normal vector of the xy-plane to calculate the device's facing direction, which gives us the altitude in the horizontal coordinate system (cf. Figure 1). When interpreted as a compass, magnetometer readings can be used to determine the azimuth. As we cannot expect a calibrated magnetometer, the resulting azimuth is shifted with an unknown offset error. We combine these positional sensors with ALS readings to generate a directed lighting map. An example of such a transformed measurement is shown in Figure 3. The x-axis represents the (shifted) azimuth as derived from magnetometer readings while the y-axis shows altitude calculated from accelerometers. The color corresponds to recorded luminance. Figure 3 therefore shows the path where the mobile device faced the sky with corresponding color-encoded brightness.

While this gives us luminance values of the user's surroundings, the sun's position is not directly obvi-

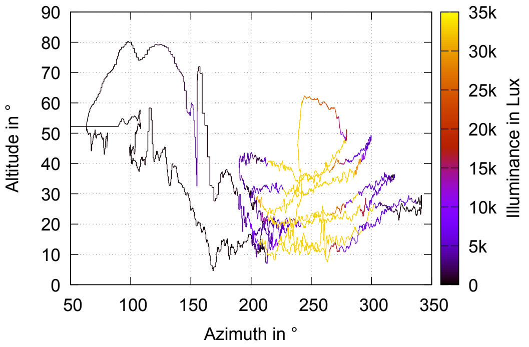

Figure 3: Sensor values transformed to horizontal coordinate system with color encoded illuminance.

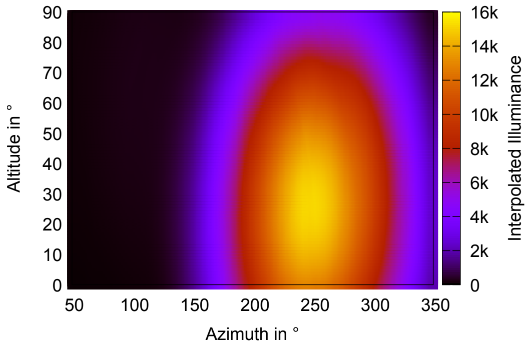

Figure 4: Interpolated luminance.

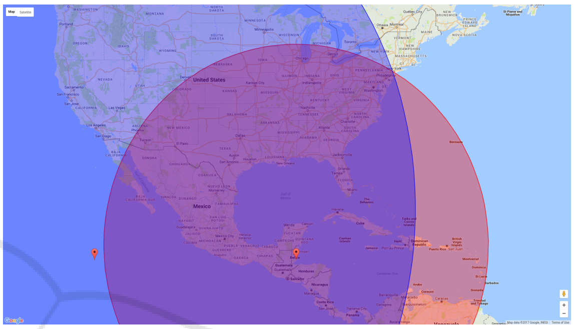

Figure 5: Circles defined by two altitude measurements in New York on 2017-08-06 at 14:00/16:00. Simulated.

ous. We need to fill the areas not passed by the path as the sun's center might be there. We therefore interpolate a global luminance model using double-powered inverse distance weighting by applying a convolution matrix $(k_{x,y})$ of order N with

$$k_{x,y} = \begin{cases} \left(1 - \left(\frac{\sqrt{x^2+y^2}}{\frac{N}{2}}\right)^2\right)^2 & \text{if } \sqrt{x^2+y^2} < \frac{N}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

We chose this matrix as it has the following properties: *a)* elements in the center of the matrix $\left(\frac{N}{2}, \frac{N}{2}\right)$ are weighted with 1, *b)* as the distance to the center increases, elements are weighted from 1 to 0 with an eased function and *c)* the corners of the matrix outside of a radius of $\frac{N}{2}$ from the center are weighted with 0.

We then apply this convolution matrix to our directed lighting map (Figure 3). For every altitude/azimuth pair, the matrix $(k_{x,y})$ is centered on that pair and the sum of its weighted neighborhood is the value of that coordinate pair in a new transformed lighting map. Figure 4 shows an example of this convolution applied to Figure 3. The color in that figure corresponds to interpolated luminance in that direction.

Finally, we estimate the sun's altitude and azimuth by finding the global maximum. We refer to a pair of estimated altitude and azimuth at point in time t_i as an *observation* denoted by $\text{Obs}(t_i)_{\text{alt}}$ and $\text{Obs}(t_i)_{\text{azi}}$.

5.3 Location Candidates

One observation defines the sun's position nonambiguously in the horizontal coordinate system and therefore the user's location. However, preliminary tests showed an error of up to ±40° in azimuth due to inaccurate magnetometer readings. This is in line with prior research which reported a compass error of $10-30°$ on mobile devices (Blum et al., 2012). We therefore keep the altitude that we derived from the accelerometer, and use observed azimuth only as a cue. Actual azimuth is determined with a second measurement. Considering only the altitude of the sun, one observation defines a set of locations where such an apparent altitude can be observed at a given time. Geometrically this set forms a circle on the Earth ellipsoid.

Determining perceived altitude and azimuth of a celestial object from a local observer at a certain point of time is a standard task of astronomy programming libraries but requires knowledge of the observer's position which is unknown in our approach. However, we do know the sun's horizontal position and can therefore numerically approximate possible observer locations. The center of that circle of possible observer locations is the *subsolar point*, where the sun is at 90° altitude.

Given two observations at different points in time, the observed altitude differs and two distinct circles are defined (Figure 5). Assuming the user is still near their location from the first measurement, we perform a circle-circle intersection to reduce the number of location candidates to two[1] If this assumption is not met the user's position difference adds a linear error to our approach. Compared with the empirical accuracy of

[1] Mathematically, also zero, one or an infinite number of intersection points are possible, which is easily detectable.

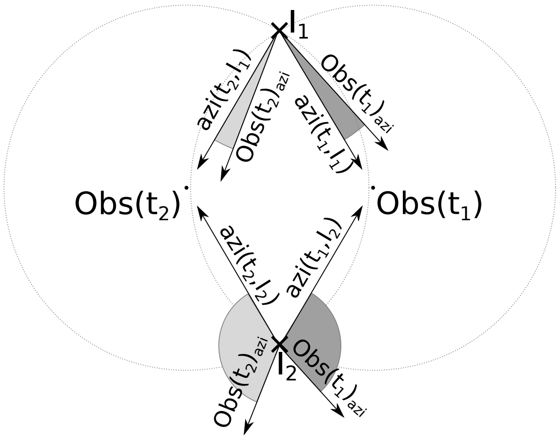

Figure 6: Intersection of location circles.

our approach (cf. Sec. 8) this error is negligible in most cases.

Figure 6 shows the approach for two observations Obs(t_1) and Obs(t_2). Intersecting them yields I_1 and I_2 as intersection points and therefore two location candidates. To select the correct location out of these two, we use the magnetometer azimuth readings Obs$(t_1)_{\text{azi}}$ and Obs$(t_2)_{\text{azi}}$. Although we argue that the magnetometer is too inaccurate to represent the azimuth, it still points in the general direction and suffices to make the correct selection out of opposing candidates. We therefore calculate the expected azimuth $azi(t_i, I_j)$ of both intersection points for both points in time using an astronomy library. The expected azimuth values are then compared to the ones yielded by the magnetometer readings and the intersection with least divergence is chosen. In Figure 6, the smaller azimuth deviation at I_1 indicates that this is the correct candidate, because the measured azimuth is closer to the expected azimuth than at I_2. We call the selected candidate the *intersection result*.

5.4 Location Aggregation

In the previous section we showed how we compensate azimuth inaccuracies and calculate device locations. If redundant measurements are available, we can utilize them to mitigate altitude measurement errors and thus improve the accuracy of geolocation. Redundant measurements occur when the user keeps using their mobile device even after we have computed one observation Obs(t_i).

Figure 7 shows how this redundancy integrates in the whole approach. Instead of two measurements, we perform k measurements for both points in time, which allows us to perform k^2 circle intersections and thus yields k^2 intersection results. We then select the median latitude and the median longitude of all intersections as the final result. We refer to the whole process as one *location determination*.

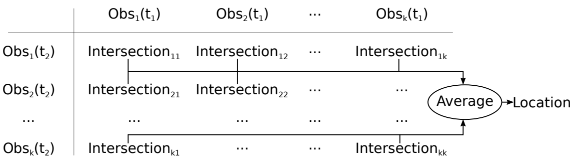

Figure 7: Increasing accuracy by using redundant observations.

6 IMPLEMENTATION

We use a web-based prototype that collects sensor data with the Ambient Light Sensor API (Kostiainen et al., 2017) and DeviceOrientation API (Tibbett et al., 2016). The implementation also records the device location as reported by the GPS-based Geolocation API (Popescu, 2016) for evaluation purposes to determine the accuracy of the digital sextant. The processing of the collected data is implemented in Python and uses PyAstronomy (Czesla, 2013) as astronomy library.

Dealing with Interferences. Reflections may interfere with the result subject to the reflecting surface. In one measurement series we found a light source at an altitude $< 0°$, which was due to a reflecting window board. While this is trivial to detect and filter, reflections from perpendicular surfaces (for example windows) are more challenging. In this case altitude is not altered but azimuth will be misestimated and can lead to an inaccurate location if the angle is large enough. As a plausibility check we have implemented a duplicate peak detection: if more than one significant light source above a threshold (the sun and its reflection) is found while searching the global maximum (cf. Section 5.2), the measurement will be marked as indecisive and rejected.

The sun threshold may vary between mobile devices due to different sensors used. If the device model is known (e.g., based on the user agent), the threshold can be preset subject to the sensor datasheet or to empirical data. If the model is unknown, the threshold can be set to 95% after collecting sensor data sufficiently long, assuming that the sun has been recorded eventually.

As our method requires exposure to direct sunlight, being indoor or under a clouded sky prevents geolocation. However, these cases can be identified trivially as the sun is orders of magnitude brighter on the Lux scale than other light sources. By prefiltering measurements for sunlight exposure we can avoid mislocating the user in theses cases.

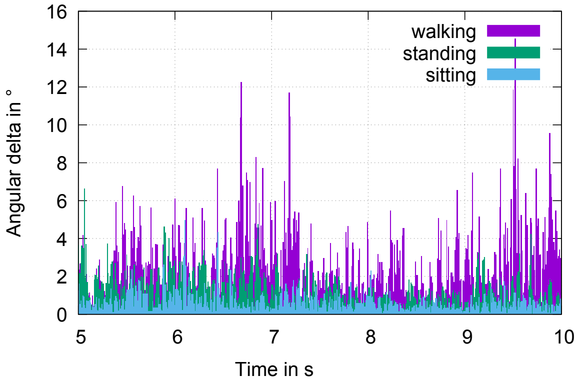

Figure 8: Impact of user behavior on accelerometer readings.

Table 1: Average angular velocity ω of various data sets.

	Table	Hand	Sit	Stand	Walk
ω in $\frac{\circ}{s}$	10.41	27.44	115.70	157.04	229.80

7 PRACTICAL APPLICABILITY

An essential requirement of our method is to face the sun, which leads to the question, whether this is realistic during everyday smartphone activities. To provide an indication for the applicability we performed the following tests with a Samsung Galaxy S7 running Firefox for Android 48:

1. Stand with the sun in the back (*standing*)
2. Walk 25 m towards the sun, turn around and walk 25 m away from the sun (*walking*)
3. Sit with the sun in the back (*sitting*).

In each setting the test person looked at the smartphone while touching, scrolling and reading the screen, which causes the tilting phone to eventually face the sun. Measurements were 40 seconds long and took place in Duisburg, Germany on an unclouded day at 11:00, 13:00 and 15:00. Each altitude/azimuth observation has been repeated $k = 5$ times, which amounts to a total of 45 measurements.

7.1 Movement Profiles

Figure 8 shows the angular difference between consecutive accelerometer vectors (data points) for an excerpt of each test. We see that during walking the smartphone moves more than during standing, and during standing more than during sitting, leading to different *movement profiles*. The sensor readings are not only influenced by user movement, but also by sensor jitter. This is demonstrated with two other tests: 1) the test person holds the smartphone in their hand without deliberate movement (*hand*), 2) the smartphone rests flat on a table (*table*). Table 1 shows the average angular velocities of each test: the phone resting on the table without any observable movement measures an angular velocity of $\omega = 10.41 \frac{\circ}{s}$, showing that sensor jitter has indeed an influence.

Table 2: Accuracy in each test.

	t_2	Error (km)	Spread (median/km)
Walk	13:00	1196.53	636.79
	15:00	1083.73	424.73
Stand	13:00	808.98	960.48
	15:00	777.08	231.13
Sit	13:00	380.24	300.79
	15:00	146.43	131.44

7.2 Location Determination

We apply our approach to the 11:00 observations of each test, paired with their corresponding observation at t_2 two or four hours later. Table 2 shows the location accuracy, i.e., the error of the determined location compared with the actual location. We can see a clear influence of movement profiles on accuracy: as (unconscious) motions of the user are reduced, the distance between determined and actual location reduces.

We conclude that while motion influences our approach, there are plausible scenarios that achieve an accuracy usable for country-level geolocation, in spite of sensor noise. A remaining open question is how often these scenarios occur with everyday smartphone usage of unaware users. This depends on the users' habits and we leave it for future work to collect sensor data of multiple persons during everyday activities to quantify the occasions for our geolocation method.

Based on the observation that too much macro human movement deteriorates the accuracy, we can restrict the method to run in situations with low-movement profiles only. This can be achieved by analyzing angular velocity or by using an existing method (Kwapisz et al., 2011) to determine the user's current activity.

8 SYSTEMATIC EVALUATION

Now that we have an indication for the practical applicability of our approach, we systematically analyze

Table 3: Accuracy comparison of Nexus 7 and Galaxy S7.

Nexus 7		Galaxy S7	
Error (km)	Spread (median/km)	Error (km)	Spread (median/km)
154.1	592.9	388.2	223.0
174.7	426.1	393.9	274.8
326.5	208.1	469.9	213.9
360.9	243.6	487.0	134.9
456.2	239.5	688.3	177.4
463.4	258.5	753.3	250.3
464.3	329.0	763.7	201.7
527.7	390.6	976.5	169.1
1 052.4	353.5	1 023.9	339.7
1 993.8	487.3	1 575.0	543.8

various factors that influence the geolocation accuracy. In the following measurements, we hold the mobile device in one hand and tilt it in two dimensions while pointing at the sun (cf. Figure 2).

We performed 10 location determinations in Duisburg, Germany with a Google Nexus 7 (2013) and a Samsung Galaxy S7 running Firefox for Android 48. First measurements were conducted at noon, the second ones two hours later at 14:00. Each altitude/azimuth observation has been repeated $k = 5$ times yielding a total of 100 measurements per device.

Our results are shown in Table 3. Column "Error" shows the error of 10 location determinations between our method and the true location as recorded by the Geolocation API. Accuracy ranges from 154.1 km to 1993.8 km with a median error of 459.8 km for the Nexus 7. For the S7, the accurancy ranges from 388.2 km to 1575.0 km. While the error range is smaller, the median error is 720.8 km and thus 56% higher than for the Nexus 7.

Because each location determination consists of $k^2 = 25$ redundant intersection results, we can examine the spread of these intermediate results as well. Column "Spread" shows the median error to the correct position of all 25 intersection results for each location determination. Interestingly, a larger spread within each location determination does not negatively impact the final accuracy. For example, the best result of the Nexus 7 in the first row has the highest spread of all location determinations for this device. Systematically, accuracy and spread correlate weakly with a Pearson correlation coefficient of $r = 0.2367$ for the Nexus 7. For the S7, this value is significantly larger with $r = 0.7553$. We will discuss differences of these devices and possible explanations in Section 8.2. Since there is no general high correlation between accuracy and spread, we conclude that our approach is robust and is not easily influenced by random measurement errors. In other words, each redundant measurement contributes to the accuracy and does not distort the final result.

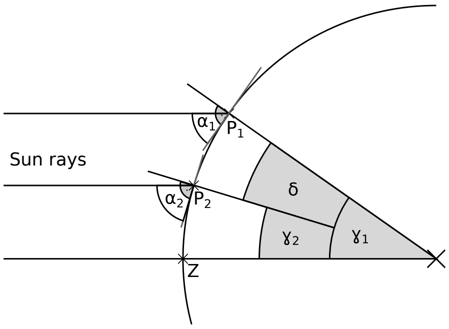

Figure 9: Location error depending on altitude.

8.1 Effect of Altitude Error

We now analyze the impact of an altitude estimation error in order to determine the expected results with more (or less) accurate sensors.

Analysis. Altitude alone—without considering non-linear influences of circle intersection—yields a location error of $\frac{\delta}{360°}E_{circ}$ for an assumed altitude error of δ. This can be derived from the Earth cross section diagram in Figure 9. Parallel sun rays reach Earth and are correctly observed at P_1 with an altitude of α_1. Angular distance to the subsolar point Z (i.e., angular radius of the circle) amounts to $\gamma_i = 90° - \alpha_i$ by corresponding angles. In case there is an altitude estimation error and $\alpha_2 = \alpha_1 + \delta$ is observed this will yield an erroneous location P_2. Solving these equations yields an angular distance between P_1 and P_2 of $\delta = \gamma_1 - \gamma_2$ which corresponds to a location offset by $\frac{\delta}{360°}E_{circ}$.

Simulation. To simulate the non-linear parts above this lower error bound we created two groups of artificial observations with a time difference of two hours. Each simulation initially assumes perfect altitude and azimuth observations as computed by PyAstronomy for this time and place. For each group we then generate simulated observations with altitude deviations of $\pm\delta$ and a probing width of 0.25°. Since we are considering the worst case impact, we perform our location determination approach on each pair out of both groups and then use the maximum location error as result.

The results are presented in Figure 10 and Figure 11. The worst case location error is significantly larger than the linear lower bound estimation. We can also see the worst case gradient grows as altitude deviation increases, which means the maximum error is non-linear. This is due to one circle growing so large that it almost covers the other shrunken circle com-

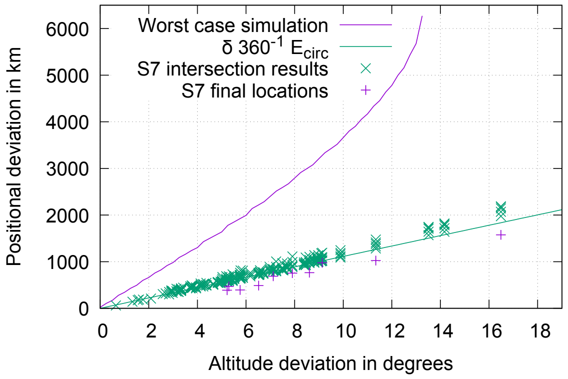

Figure 10: Simulated worst case observations compared to S7 based measurements.

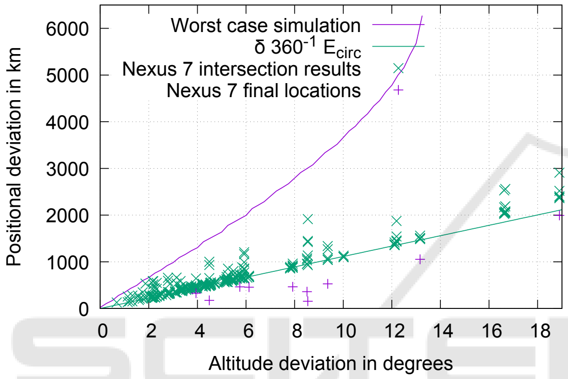

Figure 11: Simulated worst case observations compared to Nexus 7 based measurements.

pletely. The effect is illustrated in Figure 12. The small intersection angle cause any additional radius difference/altitude deviation to yield an even higher location error. Measurements with altitude deviations greater than 13.5° are not guaranteed to yield intersecting circles causing the result of the function to be undefined thereafter.

Practical Evaluation. While the error could be very high in theory, we now examine whether this has happened during our practical evaluation. Since we know the correct altitudes in our experimental setup, we can calculate altitude deviations for the intersection results (i.e., intersections before calculating a median and yielding a final position) and final positions in these figures. For each data point we aggregate the altitudes of all involved observations as a maximum to get a worst case view. In terms of location error the measurements are close to the linear bound. This indicates that the empirical average case is close to the linear bound and that the theoretical worst case does not occur in practice. Interestingly, the final positions are below the linear threshold, suggesting that measurement errors cancel each other out and thus redundant measurements produce a more accurate location result.

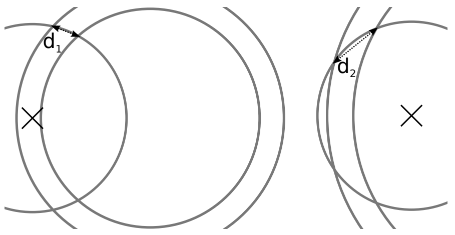

Figure 12: Effect of circles intersection angles on distance, $d_1 < d_2$.

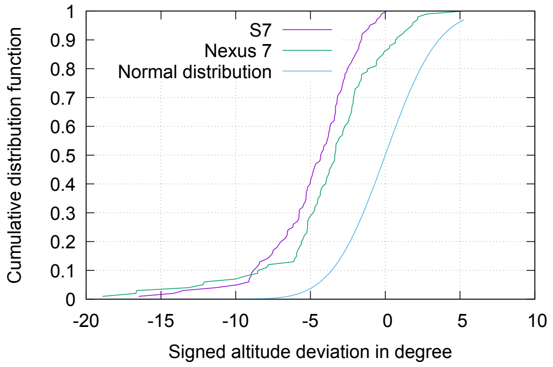

Figure 13: Distribution of signed altitude deviation.

8.2 Systematic or Random Error

We now investigate why the results of the two mobile devices scatter to a different extent and whether the sensor error is random or systematic.

A difference between Figure 10 and Figure 11 lies in the vertical spread of intersections. While they stick to the lower limit in the former they spread more towards the upper bound in the later. This means the same absolute altitude deviation has a different impact on both devices. To analyze this anomaly we took the sign of the altitude deviation into consideration.

Figure 13 shows a cumulative error distribution function of the signed altitude error for both devices. If deviations were merely due to random errors we would assume a uniform distribution around 0. Instead there is strong bias towards negative deviations, i.e., measuring the sun at a lower position as expected. The effect is even stronger on the Galaxy S7, where we observed almost no positive deviations, suggesting a device-specific systematic error. This distinction also provides an explanation for increased vertical spread: A positive deviation intersected with a negative deviation yields a higher location error than an intersection between two equally signed deviations. To verify this finding we removed intersections with a positive deviation on Nexus 7 and plotted the

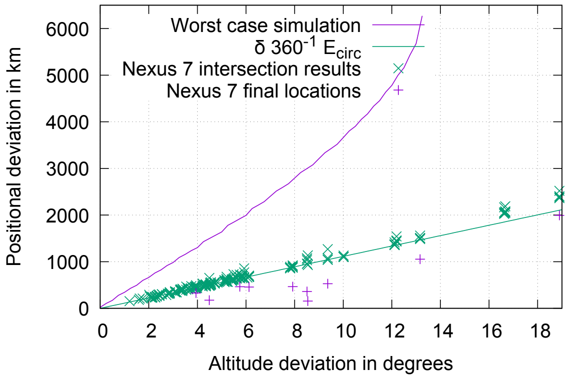

Figure 14: Simulated impact of altitude error on Nexus 7 without negative altitude deviated measurements.

result in Figure 14. The vertical variance has been drastically reduced, which confirms a device-specific systematic error.

This device-specific difference in altitude deviation also provides a viable explanation for the different correlation coefficient outcomes from Section 8. Since location determinations performed with the Nexus 7 contain more outliers in the intersection results (see Figure 11), the results will have a higher spread than with the Galaxy S7. We calculate a median out of these intersections to obtain the final position. Since medians are in general robust against influences by outliers, there is no strong correlation between the spread of a location determination and its positional deviation on the Nexus 7.

This has implications for our redundancy parameter k. If there is a systematic bias like for the Galaxy S7, then additional measurements will not increase the accuracy of our approach. On the other hand a random error will be compensated by an increase of k and successive averaging. Telling these errors apart outside of an experimental setup where no correct altitude is known is non-trivial. If a systematic error depended solely on the device model, an attacker could classify various models and act accordingly. However, analyzing this dependency is outside the scope of this paper.

8.3 Sensor Sampling Rate

Another difference between both mobile devices is how often they provide new sensor values. We now analyze whether this has an effect on our method.

Table 4 shows the sensor frequency per device across all measurements from Section 8. Concerning steadiness, the S7 performs better due to its significantly lower standard deviation for both the ambient light and acceleration sensor. This could be due to the faster processor and the general technological advancements during the 3 years between the release of both devices. Since our JavaScript implementation runs inside a web browser on top of a non-real-time operating system, there are several components involved being possible causes of this difference. The frequency of the Nexus 7 light values scatters widely with a standard deviation of 1.30 from its average of 3.67 Hz. However, this does not seem to affect our method as the Nexus 7 achieves a better accuracy (cf. Tab 3). The only sensor values more in favor of the Nexus 7 are acceleration events per second, which are nearly twice as frequent on the Nexus 7 than on the S7. Although this correlates with a higher location accuracy, it does not provide a plausible explanation. Our approach uses both accelerometer *and* ambient light readings to determine the altitude of the sun. Even if the accelerometer frequency is doubled and if the device orientation is more accurate, the luminance in that direction still lacks behind—especially when the ambient light sensor has a low sampling frequency.

Table 4: Frequency of sensor readings (Hz).

	Nexus 7		Galaxy S7	
	$\bar{x}$	σ	$\bar{x}$	σ
Light sensor	3.67	1.30	5.61	0.04
Accelerator	192.60	2.37	99.09	0.41

9 COUNTERMEASURES

As the geolocation method bears the risk of violating the user's desire for privacy, we now investigate potential countermeasures. An obvious remedy is to disable sensor access completely for apps and for websites. For example, the Orfox Browser[2], a mobile Tor Browser, restricts the use of any sensors and thus limits the possibilities to leak information to websites. While this prevents a whole class of sensor-based attacks, it also limits the potential of the Web as an application platform. Likewise, iOS does not expose an API to access ambient light sensor values preventing our attack on that platform.

Another remedy is to artificially reduce the sensor resolution to provide a compromise between privacy concerns and legitimate use cases. We analyze this possibility by truncating the sensor data of the Nexus 7 from Section 8, both regarding ALS and accelerometer. The truncation consists of rounding sensor readings to next multiples of a variable sensor truncation factor ε. This simulates a sensor with a reduced resolution.

[2] https://guardianproject.info/apps/orfox/

Table 5: Accuracy with ALS truncated to binary scale.

Nexus 7	
Distance (km)	Spread (median/km)
104.7	378.8
230.6	375.7
318.7	176.0
361.2	269.3
382.0	235.6
449.8	366.1
460.8	347.4
487.1	370.0
979.4	628.7
2 736.2	1 112.5

9.1 Ambient Light Sensor

Reducing the ALS resolution has been shown to prevent information leakage in other use cases (Schwittmann et al., 2016). Interestingly, our approach did not yield significantly worse results while iteratively increasing the truncation factor ε.

In extreme case, we round the ambient light sensor readings to binary values. Table 5 shows the results of such a binary truncation. Compared with regular results from Table 3 the results have shifted: while the average location error increased slightly by 9%, some results have become more accurate.

Although this result appears surprising, it is reasonable due to how our approach works. The sun is significantly brighter than everything else recorded like diffuse reflections or artificial lighting. This will cause all values to become 0 except those measurements directly pointed towards the sun. Our inverse distance weighting will then yield the center of these points as the altitude. As potential interferences do not pass the binarization filter, some results become more accurate while others become worse due to a loss of information.

From these results we conclude that reducing ALS resolution definitely does not provide an obstacle for our approach.

9.2 Accelerometer

Similarly, we simulate the impact of truncating the 3D accelerometer on location accuracy with truncation factors of $\varepsilon \in \{1,2,3,4,5,6,7,8,9\}\ \frac{m}{s^2}$. To put these values into perspective with Earth's gravitational acceleration of $9.81\frac{m}{s^2}$, the coarsest sensor resolution in our simulation should only be able to differentiate rotations multiple of 90°.

Figure 15 visualizes the simulation output. While up to $\varepsilon = 4$ the median location error does not become significantly worse, the number of successful location determinations drops as ε increases. This is due to observational circles not intersecting when the altitude error is too large or due to contradicting global maxima caused by the threshold function.

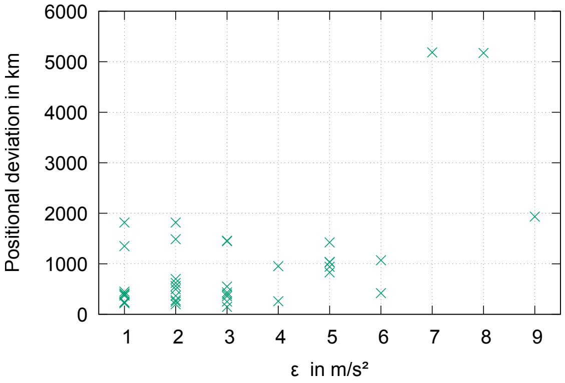

Figure 15: Impact of truncated accelerometer resolution.

This shows that a truncation of the accelerometer impairs the geolocation approach, but may still leak coarse information with a large error to an attacker.

10 RELATED WORK

The user's location is generally regarded as a privacy-sensitive information with a large body of research dedicated to ways of utilizing various available sources to infer it.

Locating a device by looking up its IP address in a database is a heuristic implemented by various commercial products. The observation behind this approach is that Internet service providers distribute IP addresses in geographical proximity. While this approach is straightforward once a database has been created, it fails when a privacy-aware user employs techniques to obfuscate their IP address (e.g., VPN, proxy server, Tor).

Powerspy (Michalevsky et al., 2015) derives the location from the phone power meter, which is available without special permissions in the Android API. Cellular radio power consumption depends on location due to obstructions and cellular tower placement allowing trajectory reconstructions using a dedicated coverage database. While providing more accurate results than our approach, it bears additional requirements: both an app has to be installed on the device and the user has to move in a coverage charted area.

Using camera images is another way of locating a user. It is feasible (Guan et al., 2013) to determine positions in a city by reading camera images and inertial sensors. Taking the sun into consideration (Ma et al., 2017) it is possible to locate a user on a map. Both approaches show remarkable accuracies but require camera permissions. A privacy-aware user is unlikely to grant this to a dodgy website or app.

In some scenarios inertial sensors alone provide enough information to locate users, if users move along given paths. Metro lines (Hua et al., 2017) have distinctive accelerations patterns allowing to track users. ACComplice (Han et al., 2012) uses solely the accelerometer to identify car trajectories in trained data. Another approach uses magnetometer and gyroscope readings (Narain et al., 2016) to locate a car using publicly available cartographic data without any training. While this increases practical applicability, feasibility on a global scale remains open. Compared to our approach this also requires cartographic material which might not be available for the subject's location.

Wi-Fi BSSID-based approaches are a standard way of locating smart devices. Even without access to the operating systems's BSSID queries, an android application without location permission can read the BSSID of the connected access point and perform its own BSSID lookup on application level (Zhou et al., 2013). Another side channel discovered by Zhou et al. consists of the speaker API. Every application can query whether any other application is currently playing sounds. Originally designed to provide apps with means of coordination, this allows to measure playback duration and deduce announcements made by a GPS navigation app.

Ambient light has been considered as well for location determination. SurroundSense (Azizyan et al., 2009) looks at a combination of ambient sound, light and color to acquire fine-grained location fingerprints to distinguish shops in a mall. Epsilon (Li et al., 2014) uses visible light beacons that are broadcasting their position. High frequency pulse width modulations on LED bulbs makes this flickering indistinguishable from dimming to the human eye. While both approaches promote non-malicious use cases, especially Epsilon is suited to violate the user's privacy since it could be implemented using a zero-permission app or website. However, it would require to deploy an infrastructure of beacons to track users.

11 CONCLUSIONS

In this paper we have presented a novel approach to locate mobile devices using sun-based measurements. The approach utilizes mobile device sensors that are accessible on platforms like Android without asking the user for permission. Unlike related work, a prior training or cartography of the user's area is not necessary, as we are relying on the well-known movement of celestial bodies.

In our experimental evaluation we achieved a median accuracy of better than 500 km, which is sufficient for country-level geolocation. The location accuracy will improve with more accurate sensors in mobile devices. Our analysis has shown that both random and systematic sensor errors influence the result, where the random error portion can be minimized by utilizing redundant measurements.

For future work we would like to improve our approach to cope with indirect sunlight. So far we rely on direct exposure to calculate altitude and azimuth. With an advanced sky model and sensor calibration it could be possible to estimate altitude based on a path not intersecting or tangent to the sun.

Privacy. In line with previous work in this field, our zero-permission geolocation approach once again shows that it is unforeseeable what high-level information might be concluded from seemingly harmless sensor values. One way to cope with this threat in general is to truncate sensor readings by default, which helps to preserve privacy in several cases while still providing a value to legitimate applications.

In our case, truncating the ambient light sensor has almost no effect on location accuracy and thus does not help. Truncation of the accelerometer worsens the accuracy and number of location determinations when rounded to multiples of 4 or more. Such an impairment likely affects legitimate use cases, thus questioning the adequacy of such a tradeoff.

A mitigation strategy might be to ask the user for permission before allowing sensor access at all—not for individual sensors as this affects the usability, but for a group of sensors that are less privacy-invading than camera or microphone access while still revealing some contextual information about the user, including accelerometer, barometer, magnetometer and ambient light sensor. An important element would be to disclose to the user what information is being collected and what it is used for. A technical enforcement combined with a documented privacy policy allows users to make an informed choice whether they approve the disclosure of contextual information.

REFERENCES

Azizyan, M., Constandache, I., and Roy Choudhury, R. (2009). Surroundsense: Mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, pages 261–272, New York, NY, USA. ACM.

Blum, J. R., Greencorn, D. G., and Cooperstock, J. R. (2012). Smartphone sensor reliability for augmented reality applications. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 127–138. Springer.

Czesla, S. (2013). PyAstronomy. https://github.com/sczesla/PyAstronomy. Accessed on 2018-05-16.

Guan, T., He, Y., Gao, J., Yang, J., and Yu, J. (2013). On-device mobile visual location recognition by integrating vision and inertial sensors. *IEEE Transactions on Multimedia*, 15(7):1688–1699.

Han, J., Owusu, E., Nguyen, L. T., Perrig, A., and Zhang, J. (2012). Accomplice: Location inference using accelerometers on smartphones. In Ramakrishnan, K. K., Shorey, R., and Towsley, D. F., editors, *Fourth International Conference on Communication Systems and Networks, COMSNETS 2012, Bangalore, India, January 3-7, 2012*, pages 1–9. IEEE.

Hua, J., Shen, Z., and Zhong, S. (2017). We can track you if you take the metro: Tracking metro riders using accelerometers on smartphones. *IEEE Transactions on Information Forensics and Security*, 12(2):286–297.

Kostiainen, A., Langel, T., and Turner, D. (2017). Ambient light sensor. W3C working draft, W3C. https://www.w3.org/TR/2017/WD-ambient-light-20170814/.

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82.

Li, L., Hu, P., Peng, C., Shen, G., and Zhao, F. (2014). Epsilon: A visible light based positioning system. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 331–343, Seattle, WA. USENIX Association.

Li, Z., Pei, Q., Markwood, I., Liu, Y., Pan, M., and Li, H. (2018). Location privacy violation via gps-agnostic smart phone car tracking. *IEEE Transactions on Vehicular Technology*, pages 1–1.

Ma, W. C., Wang, S., Brubaker, M. A., Fidler, S., and Urtasun, R. (2017). Find your way by observing the sun and other semantic cues. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299.

Michalevsky, Y., Schulman, A., Veerapandian, G. A., Boneh, D., and Nakibly, G. (2015). Powerspy: Location tracking using mobile device power analysis. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 785–800, Washington, D.C. USENIX Association.

Narain, S., Vo-Huu, T. D., Block, K., and Noubir, G. (2016). Inferring user routes and locations using zero-permission mobile sensors. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 397–413.

Popescu, A. (2016). Geolocation API specification 2nd edition. W3C recommendation, W3C. https://www.w3.org/TR/2016/REC-geolocation-API-20161108/.

Schwittmann, L., Matkovic, V., Wander, M., and Weis, T. (2016). Video recognition using ambient light sensors. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–9.

Tibbett, R., Volodine, T., Block, S., and Popescu, A. (2016). Deviceorientation event specification. W3C candidate recommendation, W3C. https://www.w3.org/TR/2016/CR-orientation-event-20160818/.

Triukose, S., Ardon, S., Mahanti, A., and Seth, A. (2012). *Geolocating IP Addresses in Cellular Data Networks*, pages 158–167. Springer Berlin Heidelberg, Berlin, Heidelberg.

Zhou, X., Demetriou, S., He, D., Naveed, M., Pan, X., Wang, X., Gunter, C. A., and Nahrstedt, K. (2013). Identity, location, disease and more: Inferring your secrets from android public resources. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS '13, pages 1017–1028, New York, NY, USA. ACM.