April 2020

Sequential Likelihood Updates for User Position Estimation with GNSS Signal Strength Matching

Kevin Watts

Sequential Likelihood Updates for User Position Estimation with GNSS Signal Strength Matching

Abstract:

This publication describes apparatuses, methods, and techniques for performing Global Navigation Satellite System (GNSS) shadow matching that can increase user location accuracy in an urban environment by appropriately calculating and reporting sequential likelihood (probability) updates for user location estimation. To do so, an electronic device (*e.g.*, a smartphone) utilizes an Urban Canyon Positioning Algorithm, which uses particle filters to solve filtering problems arising in Bayesian statistical inference. The Urban Canyon Positioning Algorithm performs sequential likelihood updates for user position estimation by weighting various particles of the particle filters to account for numerous factors, such as the number of GNSS signals, the observed shape of a "constellation" of satellites of the GNSS, unmeasured physical features (*e.g.*, buildings), local physical features, and so forth.

Keywords:

Global Navigation Satellite System, GNSS, Global Positioning Satellite, GPS, shadow matching, GNSS shadow, GPS shadow, urban canyon, satellite signal, line-of-sight signal, LOS, non-line-of-sight signal, NLOS, particle filter, Bayesian statistical inference, geolocation.

Background:

User equipment (UE), such as smartphones, often utilize accelerometers, gyroscopes, magnetometers, barometers, Global Navigation Satellite System (GNSS) technology (*e.g.*, Global Positioning System (GPS), Galileo, BeiDou, GLONASS, Indian Regional Navigation Satellite

System (IRNSS), Quasi-Zenith Satellite System (QZSS)), proximity sensors, ambient light sensors (ALS), touchscreen sensors, radar technology, cameras, microphones, and various other sensors that are embedded in or on the smartphone, which enhance the user experience and can play a role in the functionality of many application software (applications).

Many applications installed on a UE, such as navigation applications, social media applications, business locator applications, and so forth, depend on accurate user location. To this end, UE manufacturers and operating system (OS) developers can use algorithms that leverage different technologies to increase the user location accuracy, as is illustrated in Figure 1.

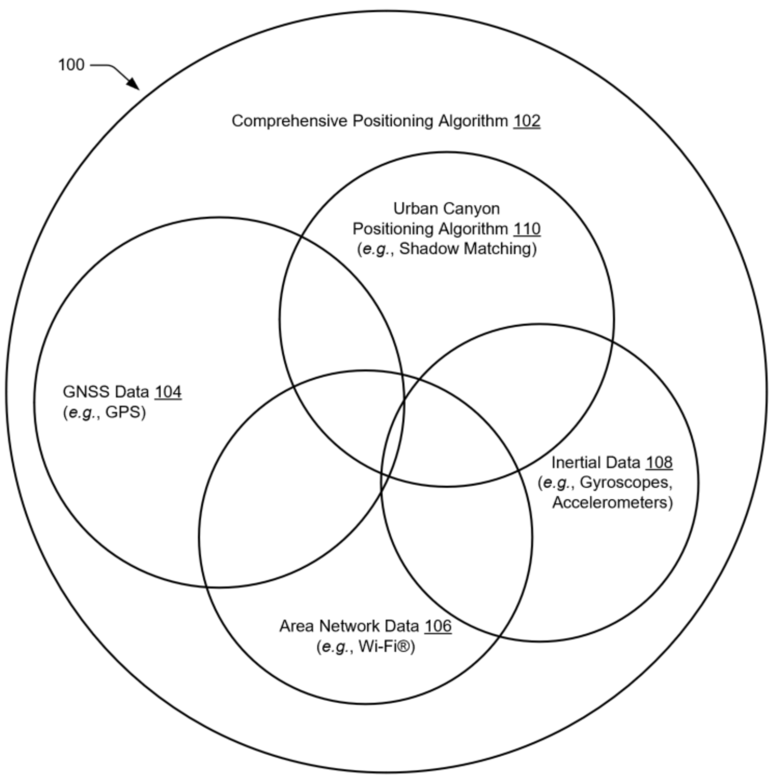

Figure 1

Figure 1 illustrates a Venn diagram 100 illustrating a comprehensive positioning algorithm 102. The comprehensive positioning algorithm 102 can utilize GNSS data 104 (*e.g.*, GPS), area network data 106 (*e.g.*, Wi-Fi® hot spots), inertial data 108 (*e.g.*, gyroscopes, accelerometers), and an Urban Canyon Positioning Algorithm 110 to determine user location, when a user is standing still, walking, running, or driving. The Urban Canyon Positioning Algorithm 110 augments the GNSS data 104 that the smartphone receives from signals sent from a "constellation" of satellites of the GNSS. Further, the Urban Canyon Positioning Algorithm 110 is useful when the user is in an urban environment.

When the user is in an urban environment, the user's smartphone can receive line-of-sight (LOS) signals and/or non-line-of-sight (NLOS) signals of satellites of the GNSS. NLOS signals have various excess path lengths and can distort the estimated user location, as is illustrated in Figure 2.

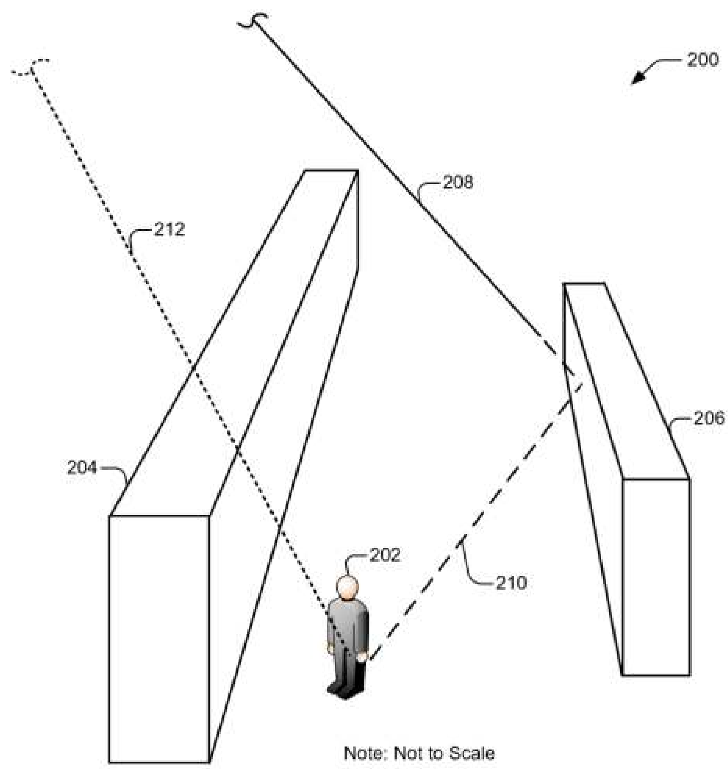

Figure 2

Figure 2 is an example urban environment 200, in which a user 202 is located between building 204 and building 206. The user 202 is carrying their smartphone, but the smartphone is not in a line of sight of a satellite (the satellite is not illustrated in Figure 2) of the GNSS. Instead, the smartphone receives a reflected signal (an NLOS signal) of the satellite. In Figure 2, the solid line represents the direction of a signal 208 of the satellite of the GNSS, the dashed line represents an excess path length 210, and the dotted line represents the distance 212 of the user to the satellite of the GNSS. If the smartphone cannot differentiate an LOS signal of a satellite from an NLOS signal of the satellite, the smartphone may falsely estimate the user as being farther from their location by a distance equal to the illustrated excess path length 210.

The NLOS received signals reflected off physical features (*e.g.*, buildings) in a dense urban environment are often weaker than LOS signals. As such, in many cases, the smartphone can differentiate LOS signals from NLOS signals by measuring various signal-strength parameters, such as:

- C/N_0—a carrier-to-noise density (signal strength) of the signals, which is measured in decibels per Hertz (dB/Hz);
- μ—the median of the C/N_0; and
- σ—the standard deviation of the C/N_0.

The Urban Canyon Positioning Algorithm 110 can use the various signal-strength parameters to estimate a user's current position, in time intervals, such as every second. Assume the Urban Canyon Positioning Algorithm 110 estimates a user being in a street in San Francisco, California, in the United States of America (USA), during a first-time interval. Then, the Urban Canyon Positioning Algorithm 110 estimates the user being in the street in San Francisco, during a second time interval. Then, the Urban Canyon Positioning Algorithm 110 estimates the user being in Reno, Nevada, USA, during a third time interval. The probability of all three estimated user locations being accurate is virtually zero. Therefore, it is desirable for the Urban Canyon Positioning Algorithm 110 to correctly calculate and estimate sequential likelihood (probability) updates for user position estimation with GNSS signal strength matching.

Description:

This publication describes techniques that can increase user location accuracy in an urban environment by appropriately calculating and reporting sequential likelihood updates for user location estimation by using GNSS signal strength matching. In one aspect, the techniques involve performing GNSS shadow matching by applying appropriate weights of multiple variables and parameters that are used to estimate user location, in time intervals.

GNSS Shadows

Physical features (*e.g.*, buildings, bridges, tunnels) in the urban environment can block LOS signals of one or more satellites in the constellation of satellites of the GNSS, as is illustrated in Figure 3.

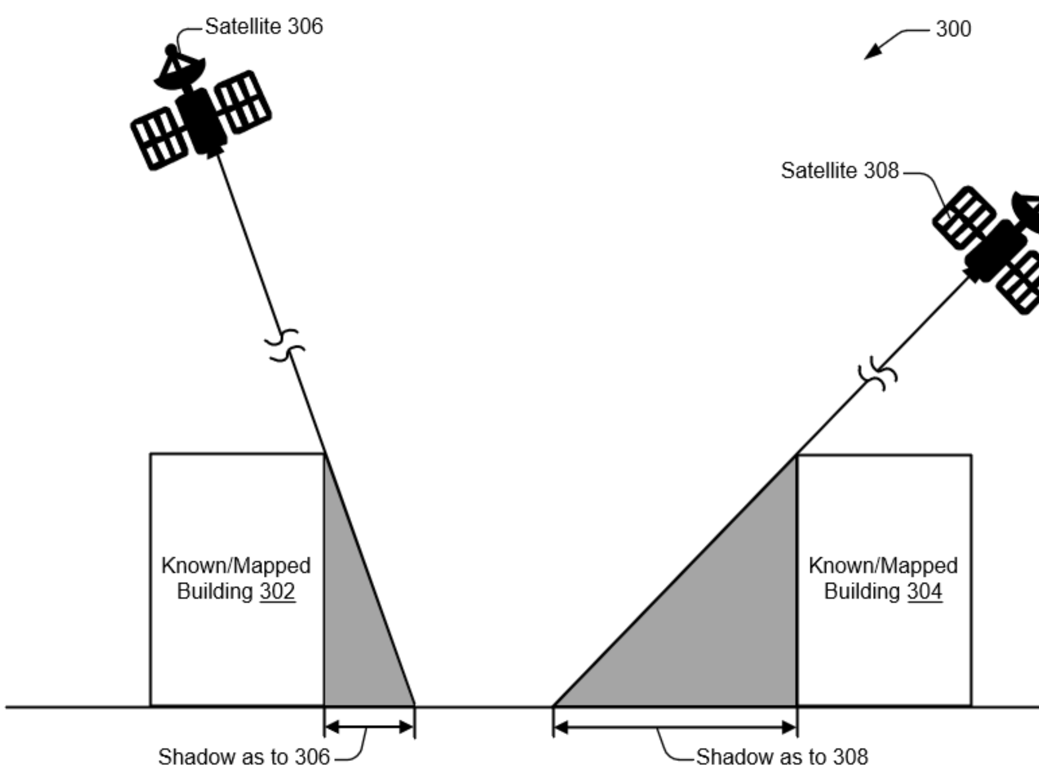

Figure 3

Figure 3 illustrates a portion of an urban environment 300, which includes building 302 and building 304. The smartphone of the user receives LOS and/or NLOS signals of satellites 306 and 308, which are part of the constellation of satellites of the GNSS. Locations that do not receive LOS signals may be considered to be shadow locations. Similar to light from a light source (*e.g.*, the Sun, a streetlight) leaving shadows when the light shines on a physical feature, physical features can block LOS signals of one or more satellites of the GNSS, leaving invisible "shadows." Consequently, when the smartphone does not have a line of sight to a satellite, the smartphone is in a shadow. Figure 3 illustrates a shadow near the building 302, where the building 302 blocks LOS signals of the satellite 306 (illustrated as "shadow as to 306"). Also, Figure 3 illustrates a shadow near the building 304, where the building 304 blocks LOS signals of the satellite 308 (illustrated as "shadow as to 308"). In the shadow as to 306, the smartphone often detects a weaker signal (lower C/N_0) of the satellite 306. Similarly, in the shadow as to 308, the smartphone often detects a weaker signal of the satellite 308.

As is illustrated in Figure 3, in the shadow as to 306, the smartphone can receive NLOS signals of satellite 306. Still, in the shadow as to 306, the smartphone can receive LOS signals from the satellite 308. Similarly, in the shadow as to 308, the smartphone can receive NLOS signals of the satellite 308. Still, in the shadow as to 308, the smartphone can receive an LOS signal of the satellite 306. When the user is located between the buildings 302 and 304 and between the illustrated shadows, the smartphone can receive LOS and NLOS signals of both satellites (306 and 308). Alternatively, although not illustrated as such in Figure 3, the user may only receive NLOS signals when the user is located in an urban environment, because the user may not be in a line of sight of any satellite of the GNSS (*e.g.*, the user may be under a bridge).

The Urban Canyon Positioning Algorithm 110 matches the GNSS signal strength of the LOS and NLOS signals received by the smartphone of the user with a map of the urban environment. This is referred to as "GNSS shadow matching." Note that an increased number of satellites in the constellation of satellites of the GNSS can increase user location accuracy.

Sequential Estimation Algorithms

In one aspect, the Urban Canyon Positioning Algorithm 110 uses particle filters or Sequential Monte Carlo (SMC) methods to solve filtering problems arising in Bayesian statistical inference. The Bayesian statistical inference is a method of statistical inference in which Bayes's theorem is used to update the probability for a hypothesis as more evidence or information becomes available. The Bayesian statistical inference is useful in dynamic analysis of a sequence of data, as is shown in Equation 1.

$$P(x|\{y\}) = \left(\prod_j \frac{P(y_j|x,\{y_{j+1}\cdots\})}{P(y_j|\{y_{j+1}\cdots\})}\right) \cdot P(x) \qquad \textbf{Equation 1}$$

In Equation 1, the various parameters are defined as follows:

- $P(x)$ — a prior probability of a user position *x*;
- $\prod_j \frac{P(y_j|x,\{y_{j+1}\cdots\})}{P(y_j|\{y_{j+1}\cdots\})}$ — a current probability of GNSS shadows, which may also be referred to as a "likelihood or measurement update"; and
- $P(x|\{y\})$ — a posterior probability of the user location *x*, given the measurement *y*.

In a general aspect, updates to the Urban Canyon Positioning Algorithm 110 may also be shown using Equation 2.

$$P(x|t+1) = \big(P(x|GNSS\ shadows)\big) \cdot P(x|t) \qquad \textbf{Equation 2}$$

In Equation 2, the various parameters are defined as follows:

- $P(x|t)$ — a prior probability of the user position *x,* given time *t*;
- $P(x|GNSS\ shadows)$ — a current probability of GNSS shadows at the user position *x*, which may also be referred to as a "likelihood surface"; and
- $P(x|t+1)$ — a posterior probability of the user position *x*, given time *t+1*.

The Urban Canyon Positioning Algorithm 110 relies on proper updates by weighting various particles of the particle filters. Some factors in weighting the various particles include:

- GNSS signal strength measurements not being independent;
- Relative likelihood surfaces for nearby positions;
- Sequential updates; and
- Unmodeled physical features (*e.g.*, unmodeled or unmapped buildings) in an urban environment.

Non-Independent GNSS Signal Strength Measurements

As is shown in Equation 2, the posterior probability, $P(x|t+1)$, of the user location *x*, given time *t+1*, depends on the likelihood surface, $P(x|GNSS\ shadows)$. The likelihood surface includes multiple input observations that are not independent. In this context, not independent does not mean dependent because the multiple input observations do not necessarily depend on one another. The multiple input observations, however, may depend on similar factors that affect the GNSS signal strength, such as an unmapped building in the vicinity of the user, the user's smartphone can receive signals from a few (*e.g.*, two, three) satellites, the user may be close to a parked over-sized truck, and so forth. In these cases, the Urban Canyon Positioning Algorithm 110, which utilizes the particle filters, can de-weight each probability curve of each input observation to create a more-trusting probability curve of the posterior probability, $P(x|t+1)$, of the user location *x*, given time *t+1*. Equation 3 details some techniques that the particle filters can use to de-weight the likelihood updates.

$$P(x|t+1) = f(x|The\ Urban\ Canyon\ Positioning\ Algorithm\ Updates) \cdot P(x|t)$$

Equation 3

In Equation 3, the various parameters are defined as follows:

- $P(x|t)$ — a prior probability of the user position *x,* given time *t*;
- $P(x|t+1)$ — a posterior probability of the user position *x*, given time *t+1*; and
- $f(x|The\ Urban\ Canyon\ Positioning\ Algorithm\ Updates)$ — a function of the user position *x*, given updates of the Urban Canyon Positioning Algorithm 110, where the function of the user position *x* depends on numerous factors including:
    - The number (count) of GNSS signals;
    - Observed constellation of satellites;
    - Unmeasured physical features; and
    - Local physical features.

The Number of GNSS Signals

When the user's smartphone receives signals from a higher number of satellites (*e.g.*, fifteen, twenty) versus a lower number of satellites (*e.g.*, two, three), the Urban Canyon Positioning Algorithm 110 can increase the user location accuracy. In one aspect, the function of the user position, $f(x|The\ Urban\ Canyon\ Positioning\ Algorithm\ Updates)$, includes multiplying each GNSS signal to determine the posterior probability, $P(x|t+1)$, as is detailed in Equation 4:

$$P(x|GNSS\ signals) = \prod_i P(signal_i|x) \cdot P(x) \qquad \textbf{Equation 4}$$

where i =1, 2, 3, …, (N-1), N. Note that in mathematics, Greek capital Pi ($\prod$) represents the product of the subsequent terms. Hence, a greater i increases the probability of the user position x, given the GNSS signals.

Nevertheless, to apply a proper weight according to the number of the GNSS signals, the function of the user position, $f(x|The\ Urban\ Canyon\ Positioning\ Algorithm\ Updates)$, incorporates scaling the output probability by the number of the GNSS signals to determine the posterior probability, $P(x|t+1)$, as is detailed in Equation 5:

$$P(x|GNSS\ signals) = \prod_i P(signal_i|x) \cdot \frac{P(x)}{scale\_factor} \qquad \textbf{Equation 5}$$

where *scale_factor* depends on the number of the GNSS signals.

Observed Constellation of Satellites

In addition to the number of GNSS signals that are received by the smartphone of the user, the Urban Canyon Positioning Algorithm 110 can also consider a shape of the constellation of the satellites. Assume the smartphone of the user estimates that there is a building north to the user. Also, assume the smartphone receives signals of only satellites north to the user, instead of from south, east, and/or west. In that case, the Urban Canyon Positioning Algorithm 110, which uses the particle filters, can de-weight signals coming from one direction instead of multiple directions.

Unmeasured Physical Features

In one aspect, each GNSS signal probability may be modeled as $P(GNSS\ signals|x)$. $P(GNSS\ signals|x)$ is the probability of the GNSS signals, given the user location x. The user location accuracy can be increased by de-weighting received GNSS signals to account for unmeasured, incorrectly measured, unmapped, and/or incorrectly mapped physical features (*e.g.*, buildings) in the urban environment. Thus, the probability of each GNSS signal may be modeled as $P(GNSS\ signals|x, building\ models)$. $P(GNSS\ signals|x, building\ models)$ is the probability of the GNSS signals, given the user location x and the building models. Note that the probability of the building models, $P(building\ models)$, is not independent. Hence, de-weighting the probability of the building models makes the particle filter updates for the Urban Canyon Positioning Algorithm 110 more robust.

Local Physical Features

The probability of the GNSS signals may be modeled as $P(GNSS\ signals|x, building\ models)$. One way to de-weight the probability of the building models, $P(building\ models)$, is according to local physical features. For example, assume a user walks in a dense urban environment, such as Manhattan, NY, USA. The probability of the building models needs to reflect the dense urban environment of Manhattan, instead of open spaces, wide boulevards, grassy fields, and so forth. If suddenly, the user's smartphone, which utilizes the Urban Canyon Positioning Algorithm 110, receives updates that the probability of the building models, $P(building\ models)$, represents open spaces, the Urban Canyon Positioning Algorithm 110 can de-weight such updates. One reason to de-weight such updates, instead of ignoring them altogether, may be to account for any recent construction activities, such as demolition of an older building.

Further to the descriptions above, a user may be provided with controls allowing the user to make an election as to both if and when systems, programs, or features described herein may enable collection of user information (*e.g.*, information about a user's social network, social actions, social activities, profession, a user's preferences, or a user's current location), and if the user is sent content or communications from a server. In addition, certain data may be treated in one or more ways before it is stored or used, so that personally identifiable information is removed. For example, a user's identity may be treated so that no personally identifiable information can be determined for the user. Thus, the user may have control over what information is collected about the user, how that information is used, and what information is provided to the user. The user can also disable the Urban Canyon Positioning Algorithm 110 and may choose to rely only on GNSS data, area network data, and/or inertial data when utilizing applications that depend on user location (*e.g.*, navigation applications).

Conclusion

The Urban Canyon Positioning Algorithm 110 performs sequential likelihood updates for user position estimation by weighting various particles of the particle filters to account for numerous factors, such as the number of GNSS signals, the observed constellation of satellites, unmeasured physical features, local physical features, and so forth.

References:

[1] Patent Publication: US20170131409A1. System and method for localization and tracking using GNSS location estimates, satellite SNR data and 3d maps. Priority Date: July 17, 2015.

[2] Patent Publication: US20190094379A1. Three-dimensional city models and shadow mapping to improve altitude fixes in urban environments. Priority Date: September 28, 2017.