

iag-comm4-2022-4, updated on 27 Nov 2022
<https://doi.org/10.5194/iag-comm4-2022-4>
2nd Symposium of IAG Commission 4 "Positioning and Applications"
© Author(s) 2022. This work is distributed under
the Creative Commons Attribution 4.0 License.



Computing the GPS Sky-View Factor in Urban Landscapes for Autonomous Driving Simulation

Ganesh Kumar¹, Sharnam Shah¹, Yongbo Qian¹, Nahid Pervez¹, and Tyler Reid²

¹Ford Greenfield Labs, CA, United States of America

²Xona Space Systems, British Columbia, Canada

Computing the GNSS Sky-View Factor in Urban Landscapes for Autonomous Driving Simulation

GANESH P KUMAR, SHARNAM SHAH, YONGBO QIAN, NAHID PERVEZ

Ford Greenfield Labs • Palo Alto • California • 94043 USA

Email: (gkumar29, sshah89, npervez2, yqian17)@ford.com

TYLER G R REID

Xona Space Systems • Vancouver • British Columbia • V6R 2G6 Canada

Email: tyler@xonaspace.com

Keywords: GNSS, Sky-View Factor (SVF), Line of Sight (LOS), Multipath, Autonomous Vehicle (AV)

Motivation

It is often cost- and risk-effective to test the response of an Autonomous Vehicle (AV)'s planning and control module to simulated perception output from its sensors, an exercise called *perception simulation* [1]. The AV's GNSS Sensor (called *the ego GNSS*) computes its position in a world coordinate frame (e.g., WGS-84) and feeds directly into the AV's localization module, influencing downstream operations such as map-relative localization and sensor fusion. Consequently, predicting or simulating GNSS output is extremely useful to determine roadways wherein AV localization may experience degraded performance.

However, GNSS output is generally challenging to simulate [4] due to the multiple time- and location-varying sources of error that impact its operation. City landscapes pose modelling challenges in that their buildings and urban canopies (referred to as *topographic elements* from now) limit the ego GNSS' view of satellites in the sky, causing *sky-impairment* [14] that impacts GNSS *availability*. Further, these topographic elements also precipitate multipath and non-line of

sight (NLOS) effects that impact GNSS *accuracy*.

Sky-View Factor (SVF): Since modeling (or simulating) these effects in their entirety for a given urban landscape is computationally nontrivial - primarily due to the need to capture the interference of radio-frequency waves interacting with each topographic element - we will focus on the more tractable problem of computing the *Sky-View Factor (SVF)* of the landscape. We define the SVF of the (ego) GNSS with respect to a landscape to be fraction of the sky visible to it, unobscured by topographic elements [13]; the SVF is thus a dimensionless quantity between zero (representing a completely obscured sky) and unity (representing a fully unobscured sky), representing the *complement* of sky impairment. When its SVF is unity, the GNSS' sky visibility is blocked only by the earth's curvature, and the GNSS receiver can view the maximum possible number of satellites in its line of sight (LOS).

The motivation behind the choice of the SVF as our metric of interest are: its computation is a tractable geometric problem determined only by the shapes of the topographic elements and the GNSS receiver location; the number of satellites visible to the user in its LOS may be determined from it; the data structures used in its computation may be used as a precursor to more complex models of accuracy and availability; it distills a landscape into a single scalar metric that measures how close the AV is located to a city center (or a location rich in topographic elements) - and it may thus be used to characterize cities; and prior work does not compute it directly except for specific dispositions of topographic elements [13].

Thus, the *novelty* of this abstract lies in identifying and solving the problem of computing the SVF, suggesting approaches to speed up the computation (at the expense of accuracy) for real time applications and outlining further applications of SVF-related data structures.

Terminology

We paraphrase the following definitions from [14]. A GNSS *constellation* consists of a satellite set that provides *position, navigation, and timing (PNT)* information to a GNSS Receiver that is usually located on the earth's surface. Traditional GNSS constellations reside in Medium Earth Orbit, for example, GPS at an altitude of approximately 20,200 km. Historical GNSS constellation orbital data is available online for example, at [10], although this framework also allows us to examine future satellites including commercial Low Earth Orbit (LEO) Position, Navigation, and Time (PNT) satellites via simulation. We denote the altitude of satellites in a constellation of interest by R_{sat} . We will also use *GPS* to mean *GNSS receiver* throughout. The *pseudo-range equation* is used to compute the ego position on the earth's surface using the satellites visible to the GPS. *Multipath* refers to the reflection of GPS signals off multiple surfaces (e.g., those of buildings) before reaching the GPS receiver, leading to degraded accuracy.

Prior Work

The pseudo-range GPS equation, sources of GPS error, and satellite navigation performance metrics including availability and accuracy are detailed in [14]. The significance and challenges of GPS modeling for perception simulation are noted in [4, 6], while [11, 20] specify approaches to computing multipath effects in urban environments. Multipath and NLOS effects are computed using simulators in [15, 21]. Our prior work [17] measures the difference between automotive and RTK GPS receiver accuracy over North American Highways. The Sky-View Factor, a term largely used in building and environmental research, is defined and computed in [13].

Problem Statement

Given the following data: 1. a discrete time interval $(t_0, t_0 + \delta t, t_0 + 2\delta t, \dots, t_f)$ sampled every δt seconds (the sampling frequency or GPS Epoch), 2. the map of an urban landscape defined by topographic elements $T = \{T_i: 1 \leq i \leq n\}$, specified in WGS-84 coordinates (taken, e.g., from OpenStreetMap or Google Street View, comprising latitude, longitude and altitude), with each roadway taken to be a polygon and each topographic element taken to be a polyhedron specified by its vertices or faces; 3. the position of an AV specified in WGS-84 coordinates (on e.g., a roadway) in the map; 4. satellite position data for the given time interval (e.g., from RINEX files), for satellite set $S = \{S_i: 1 \leq i \leq l\}$, each orbiting the earth at altitude R_{sat} above the earth's surface.

Problem: Determine the SVF of the ego GPS over the given time interval. Further, determine those satellites that are in the GPS LOS.

Computing SVF for a single time point: Take a single epoch t in the given time interval. At t let $G = (lat, lon, alt)$ be the vehicle position on the earth. Ignoring topography, the area of the sky that is visible to the GPS receiver in the absence of topographic elements is given by the base of a cone with apex G , slant distance R_{sat} , and a planar apex angle θ_1 . This visible area A_{vis} lies between 0 and $4\pi R_{sat}^2$, where the factor of 4π corresponds to the solid angle subtended by a sphere at its center; the resulting SVF is unity (unless $A_{vis} = 0$, in which case the SVF is undefined).

We will now determine the impact of a single (polyhedral) element T_1 on the SVF. First, we compute the polygonal face of T_1 visible G to by constructing straight line segments from G to each vertex of T_1 and discarding those vertices that are occluded from by another point in T_1 . We call this polygon $Vis(T_1)$, and suppose that its vertices are $\{V_i\}$. Subsequently, we construct a *blocked spherical pyramid* Pyr_1 with apex G and edges of the form GV_i extended to distance R_{sat} ; none of the points in the interior of Pyr_1 is visible to G . The base of this pyramid rests on a subset of the unobscured visible area A_{vis} computed earlier. We may compute the area of this pyramidal base to be A_1 . Since this area is obscured, the resulting visible area shrinks to $A_{vis} - A_1$, and the SVF becomes $1 - A_1/A_{vis}$.

To extend this computation to an urban landscape comprising multiple polyhedral elements, we first compute their union using (e.g.) [2] and apply the algorithm above to the resulting solid to determine the region of blockage. Note that this blocked region is in general a concave union of convex polyhedra, resting on a spherical base.

Extension of SVF to a time interval: Repeating this algorithm at every time point in the given interval is inefficient. To extend the computation from time instant to the next epoch $t + \delta t$, we drop the contributions from topographic elements visible at time t , and correspondingly add the contributions of those visible at $t + \delta t$. Elements common to consecutive time instants may be identified by pre-processing the urban landscape for the entirety of the time interval, and thus used to speed up the computation over the interval.

Speeding up SVF Computation: When each topographic element is a cuboid oriented vertically to the earth's surface (e.g., each is a building), this computation can be simplified using a plane sweep algorithm [3]. To get a quick (if inaccurate) estimate of the SVF in real time, we may replace every polyhedron with its (cuboidal) bounding box and repeat the sweep algorithm.

Determining satellites in the LOS: At any time instant t we may determine if satellite S_i is in the LOS by determining if the line segment GS_i lies fully outside the blocked region of the GPS receiver. This may be done (e.g.) by testing whether the segment intersects any of the faces of the blocked region, using standard approaches from computational geometry [22]. We repeat this computation for every satellite to determine the subset of the satellite set that lies in the receiver's LOS.

Current Plan and Future Work

We are currently examining various cityscapes from OpenStreetMap building data [16] to give as input to our algorithm, and thereby model GNSS geometry and visibility (to a first approximation) in perception simulation. We are also currently working with external partners who specialize in creating virtual 3-D models of the earth for higher fidelity modelling. In future work, we will use the techniques in [14] to get a first approximation of the multipath and NLOS component of the measurement noise. This is expected to be a significant extension to our current algorithm.

References

1. Applied Intuition. "From Research to Production: Strategies for Perception Simulation (Part 1 of 2)". Available: <https://blog.applied.co/blog-post/research-to-production> (2021).
2. Aronov, Boris, Micha Sharir, and Boaz Tagansky. "The union of convex polyhedra in three dimensions." *SIAM Journal on Computing* 26, no. 6 (1997).
3. De Berg, Mark, et al. "Computational geometry: introduction." *Computational geometry: algorithms and applications* (2008).
4. Durst, Phillip Jefferson, and Christopher Goodin. "High fidelity modelling and simulation of inertial sensors commonly used by autonomous mobile robots." *World J. Model. Simul* 8.3 (2012): 172-184.
5. Elango, G. Arul, and G. F. Sudha. "Design of complete software GPS signal simulator with low complexity and precise multipath channel model." *Journal of electrical systems and information technology* 3.2 (2016): 161-180.
6. Elmquist, Asher, and Dan Negrut. "Methods and models for simulating autonomous vehicle sensors." *IEEE Transactions on Intelligent Vehicles* 5.4 (2020): 684-692.
7. Gowdayyanadoddi, Naveen S., et al. "A ray-tracing technique to characterize GPS multipath in the frequency domain." (2015).
8. von Hünerbein, Karen, and Grégory Moura. "A new tool for simulation of geospecific multipath and obscuration of GPS/GNSS signals with relation to realistic 3-D city models."
9. Hussain, Arif, et al. "Adaptive GNSS Receiver Design for Highly Dynamic Multipath Environments." *IEEE Access* 8 (2020): 172481-172497. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.927.3102&rep=rep1&type=pdf> (2022).
- International GNSS Service (IGS), "Downloadable Constellation Data". Available: <https://www.igs.org/network#downloadable> (2022).
10. Lau, Lawrence, and Paul Cross. "Development and testing of a new ray-tracing approach to GNSS carrier-phase multipath modelling." *Journal of Geodesy* 81.11 (2007): 713-732.

11. Li, ChengJun, et al. "The design and implement of GPS software simulation platform." 2008 Asia Simulation Conference-7th International Conference on System Simulation and Scientific Computing. IEEE, 2008.

12. Miao, Chunping, et al. "Review of methods used to estimate the sky view factor in urban street canyons." *Building and Environment* 168 (2020): 106497.

13. Morton, Y. Jade, et al., eds. *Position, navigation, and timing technologies in the 21st century: Integrated satellite navigation, sensor systems, and civil applications*, volume 1 and 2. John Wiley & Sons, 2021.

14. Nievinski, Felipe G., and Kristine M. Larson. "An open source GPS multipath simulator in Matlab/Octave." *Gps Solutions* 18.3 (2014): 473-481.

15. OpenStreet MapBuilding Data. Available: <https://taginfo.openstreetmap.org/keys/building> (2022).

16. Reid, Tyler GR, et al. "Standalone and RTK GNSS on 30,000 km of North American Highways." *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*. 2019.

17. Spirent. "GNSS, Multipath Simulation in 3D - Spirent." *Multipath and Obscuration Simulation*, www.spirent.com/products/multipath-and-obscuration-simulation-sim3d (2022).

18. StackExchange Inc. "How much of the sky is visible from a particular location?". Available: <https://physics.stackexchange.com/questions/129317/how-much-of-the-sky-is-visible-from-a-particular-location> (2022).

19. Suzuki, Taro, and Nobuaki Kubo. "Correcting GNSS multipath errors using a 3D surface model and particle filter." *Proceedings of the 26th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2013)*. 2013.

20. Suzuki, Taro, and Nobuaki Kubo. "Simulation of GNSS satellite availability in urban environments using Google Earth." *Proceedings of the ION 2015 Pacific PNT Meeting*. 2015.

21. Tananyagtár ,Gyires Béla. "Point Containment Algorithms in Computer Graphics". Available: <https://gyires.inf.unideb.hu/KMITT/a52/ch10s04.html#:~:text=Polyhedron%2Dpoint%20containment%20test.,number%20of%20intersections%20is%20counted> (2022).

22. US Department of Defense, "GPS Standard Positioning System Performance Standard". Available: <https://www.gps.gov/technical/ps/2020-SPS-performance-standard.pdf#page=67> (2020)