

Q What are the challenges of ray-tracing for GNSS applications?

GNSS Solutions is a regular column featuring questions and answers about technical aspects of GNSS. Readers are invited to send their questions to the columnist, **Dr. Mark Petovello**, Department of Geomatics Engineering, University of Calgary, who will find experts to answer them. His e-mail address can be found with his biography below.



MARK PETOVELLO is a Professor in the Department of Geomatics Engineering at the University of Calgary. He has been actively involved in many aspects of positioning

and navigation since 1997 including GNSS algorithm development, inertial navigation, sensor integration, and software development. Email: mark.petovello@ucalgary.ca

Simulating the propagation and reception of GNSS signals in complex environments is a challenging task. Indeed, the user always has to trade off between the computation time and the reliability of the output. Moreover, the motion of GNSS satellites, atmospheric effects, and building geometry are always difficult to model.

This article describes how ray-tracing methods can produce consistent and reliable output. It also provides an example of validation of a software suite using a graphics processing unit (GPU)-based ray-tracing algorithm to model obscuration and multipath.

Methods for Modeling GNSS Signal Propagation

Simulations can help us assess the impact of the environment on the propagation of GNSS signals in constrained surroundings such as urban settings. Indeed, simulations are relevant alternatives to test the performance of GNSS systems and services, compared to measurement campaigns in a live environment. Simulators enable users to control completely the operational parameters and then to test systems or services in different conditions. They also provide users with data to forecast the reception in a given area (for instance, for mission planning) or to analyze what happened in the field (for instance, for mission debriefing).

Several methods exist to compute the propagation of GNSS signals in constrained environments. These are summarized in **Figure 1** and can be sorted into four categories according to the methods used to model the environment and the propagation.

Pure statistical methods (such as Rayleigh, Rice, Nakagami, and so forth) provide results quickly, but their accuracy and relevance can be low. Moreover, these methods rely on empirical parameters that can be difficult to tune to fit to real environments. Hybrid methods may partially overcome this drawback by using a deterministic approach to define either the environment or the propagation model.

Deterministic methods return accurate data but have two major disadvantages. Firstly, their reliability relies largely on the quality of models (or *synthetic environments*) used to define the environment (satellites locations, clocks, atmospheric conditions, buildings, vehicles, receiver properties, etc.). Indeed, the user can achieve a Laplace demon for a while only if the environment modelling (3D virtual

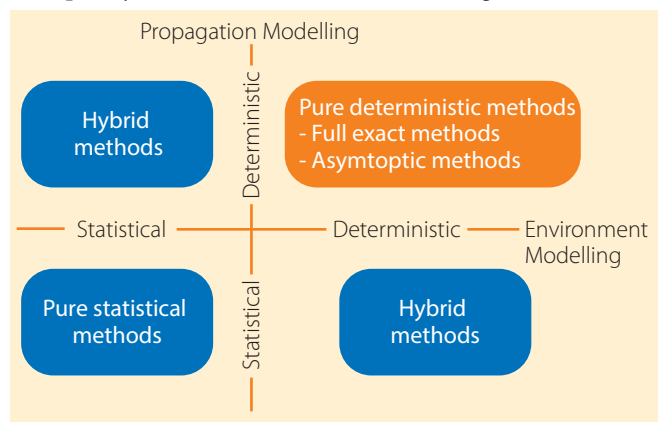


FIGURE 1 Methods of computing GNSS signal propagation in constrained environments, sorted according to how the environment and signal propagation are modeled



GPS NETWORKING



TEST & MEASUREMENT



DAS/WIRELESS



SURVEY & MAPPING



FIRST RESPONDERS



MILITARY/ GOVERNMENT



SIGNAL DISTRIBUTION



SIGNAL ACQUISITION



TIMING AND SYNCHRONIZATION



CELLULAR INFRASTRUCTURE

The World's Leading Provider of GPS/GNSS Signal Distribution Components

Proven Customer Service and Quality Products You Can Rely On!

GPS Networking, Inc. is the first company to specialize in providing GPS (Global Positioning System) products and solutions to enable you to effectively distribute the GPS/GNSS (Global Navigation Satellite System) signal throughout your facility. We are continually expanding & improving the line of GPS/GNSS products which allow you to design the GPS/GNSS network that meets your application needs, whether it is in Military, Telecom, Utilities, Public Safety, Manufacturing or Surveying.

With over 20 years of experience, GPS Networking, Inc. has been the world leader and your GPS source for providing GPS/GNSS Antenna Splitters, Amplifiers, Rack Mount Splitters, Reradiating Kits and Fiber Optic Antenna Link systems. If you need information on how to accomplish your particular GPS/GNSS Network project, please call us at 800-463-3063 or email us at salestech@gpsnetworking.com and let us put our skills and experience to work for you.

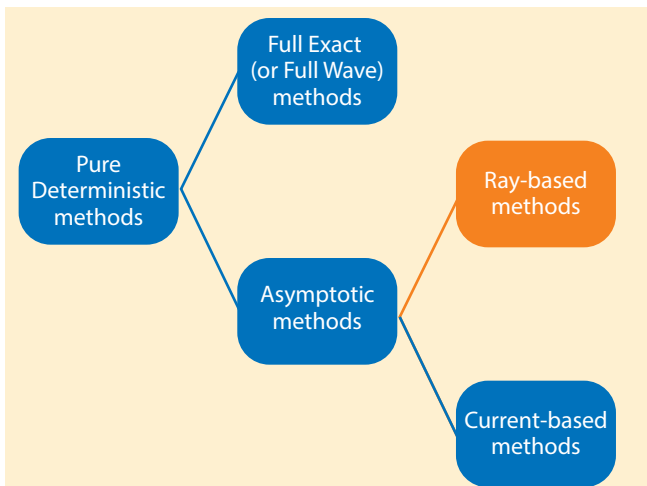


FIGURE 2 Breakdown of pure deterministic methods

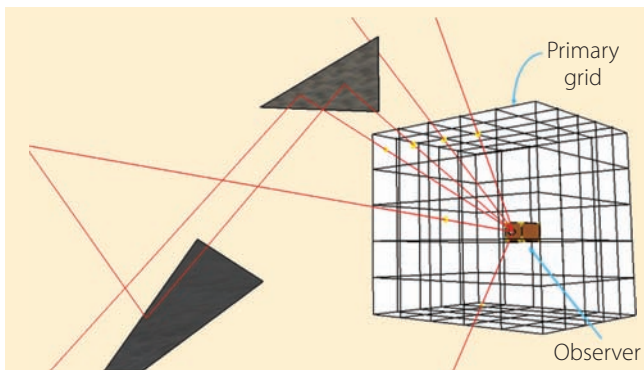


FIGURE 3 Primary grid definition. A primary ray is cast towards each center of the primary cells. The resolution of the primary grid is one of the simulation parameters

scene and simulation parameters) perfectly matches reality. Unfortunately, it is not possible to create such a model with limited time and effort. Moreover, these methods are often time-consuming in complex environments such as urban areas.

Pure deterministic methods can be sorted in two categories as shown in **Figure 2**: *full exact* (such as *Finite Element* methods) and *asymptotic* methods. Unlike full exact methods, asymptotic approaches use approximations to solve the Maxwell equations but with greater efficiency. Specifically, they assume the dimensions of objects are greater than the wavelength of the emitted signal.

Asymptotic methods are divided in two main categories: ray-based and current-based algorithms. The first category uses *geometrical optics* (GO) and its extension for diffractions (*uniform theory of diffraction* or UTD) to compute the propagation of the signal whereas the second category relies on *physical optics* (PO) laws.

The objective of this article is to describe how ray-based asymptotic methods can be chosen to assess the propagation of GNSS signals in constrained environments and explain their added values and limitations.

What is Ray-Tracing?

A ray can be defined by three parameters: O , location of the emission source; dir , the direction vector; t , distance crossed by the ray.

The equation of the ray is: $ray = O + \vec{dir} * t$

The objective of ray-tracing is to find all the possible paths from the receiver (also called the *observer*) to the emitter (also called the *source*), considering a limited number of interactions per emitted rays. A ray-tracer (or ray-tracing algorithm) uses a primary grid to cast primary rays, as shown in **Figure 3**. Then, it iteratively computes the possible interactions between these rays and the virtual scene (often defined using triangles). If those interactions exist (i.e., if they are compliant with the laws of physics) and if the number of interactions to reach the emitter is below the maximum number of interactions set by the user, a ray (or multipath) is created.

Ray-tracing techniques have two major issues: they require time-consuming algorithms and their accuracy depends on the resolution of the primary grid. Indeed, depending on the complexity of the scene (basically the number of triangles), a combinatorial problem to find the possible multipaths reaching the receiver makes the ray-tracer very slow. That is the reason why the most difficult task to achieve during the coding of a ray-tracing algorithm is to develop acceleration techniques to quicken the computation process.

Several solutions exist to either improve the intersection determination (for instance, based on spatial hierarchies such as *bounding volume hierarchies* or BVH techniques), or to decrease the number of cast rays (often based on adaptive sampling techniques), or even to replace rays with beams or cones. Moreover, today we can use the resources of a computer's graphic board to accelerate the computation. Indeed, as ray-tracing can be coded by a large number of primary functions that can be treated simultaneously, it can be easily ported into a GPU.

As for the dependency of a ray-tracing technique's accuracy on the resolution of the primary grid, details and therefore rays may be missed if the three-dimensional scene is made up of small details. This issue is called *aliasing*. Aliasing artefacts appear, for instance, in parts of the scene with abrupt changes (such as edges) or in complex areas with lots of constituent objects. Solutions (or anti-aliasing techniques), such as adaptive or stochastic samplings, exist that can deal with this phenomenon.

Ray Tracing for GNSS

Ray-tracing techniques are often used to model the propagation of GNSS signals in constrained environments. Indeed, by means of a ray-tracing algorithm, software can simulate two phenomena of propagation (see **Figure 4**):

- obscurations, line-of-sight (LOS) signals that cannot reach the receiver due to the presence of buildings between the emitters and the receivers, which may lead to availability problems
- multipaths, reflected, diffracted, transmitted, scattered signals that generate positioning errors and fading effects.



FIGURE 4 Left: obscuration representation. The receiver is located at the center of the red virtual sphere. Right: example of GPS reception close to the City Hall of Toulouse, France. In white, incoming rays from the satellites, in blue, diffracted rays, in red reflected rays (simulation)

As ray-tracing is a deterministic method, specular multipaths can be displayed in 3D and their attributes (receiver power, phase, polarization, Doppler, geometry of the ray, and so forth) are known. It is therefore easy to see the contribution of some part of the environment on the propagation of the signal.

The Synthetic Environment

The main input of a ray-tracing algorithm is the synthetic environment, a virtual modeling of a real or realistic

environment in terms of both geometry and physics. The geometries of the buildings or the objects define the shape of the rays. The definition of the intrinsic parameters of the materials of the objects (permittivity, conductivity, and thicknesses of the walls, and so on) enables computation of the interaction between the signal and the objects.

These synthetic environments may be sorted into three categories:

- **Fictitious mock-ups.** These 3D scenes do not model *any real or realistic* environment. The mock-

ups are mainly used to assess a GNSS system or service, in particular, geometrical and/or physical configurations, or to study the impact on the results when varying some parameters, such as the influence of a modification of roof shapes (buildings with realistic roofs, with flat roofs, without any roof, and so forth) on the visibility of the satellites.

- **Geotypical mock-ups.** These mock-ups model a *realistic* environment, i.e., an environment that does not exist but that looks familiar (e.g., dense European city, a mountainous area). They are based on real parameters such as real footprints and/or heights of buildings but some parts (for instance, the shapes and the locations of the doors or the windows or the roofs) are randomly defined. Templates of buildings are used to characterize the global shapes of the buildings to be created. Geotypical environments are

Inertial Navigation System



0.1° Roll & Pitch
0.2° Heading
2 cm RTK



Ellipse-D Dual GNSS/INS

- » Immune to magnetic disturbances
- » L1/L2 GNSS receiver
- » Accurate heading even under low dynamics
- » Post-processing

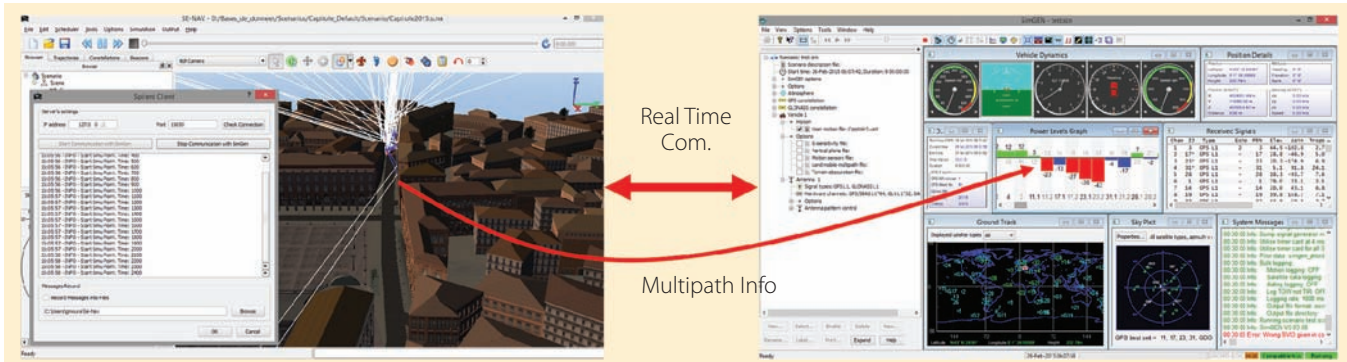


FIGURE 5 Left: ray-tracing software. Right: GNSS signal simulator software. Both programs communicate in real time; the simulator sends computation requests and the ray-tracer returns the obscuration and multipath modeling

mainly used to assess the availability and performance of GNSS systems in a typical environment, for instance for statistical studies.

- **Geospecific mock-ups.** These mock-ups model a *real* environment, and are accurate descriptions of a real scene in terms of geometry and physics. These 3D scenes are used to either reproduce actual signal reception (e.g., for mission debriefing), or to forecast a reception (e.g., for mission planning) or even to populate a measurement campaign with simulated data. In such cases, the quality of the 3D mock-ups has a strong effect on the consistency and the reliability of the results. Indeed, the geometries of the buildings or obstacles (such as street lights, road signs, vehicles, etc.) must mimic the local environment as closely as possible in order to be compared to measurements.

Validation – Example of Simulated Data

A measurement campaign was recently conducted to validate the performance of a simulator based on ray-tracing algorithms. Two parts comprised the

simulation system (screenshots shown in **Figure 5**): a typical GNSS simulator capable of modeling the constellation and relevant receiver characteristics and a GPU-based ray-tracing algorithm to deterministically compute obscuration and multipath parameters.

Both software applications communicate in real-time through a TCP/IP protocol. The computation frequency has been set to 10 hertz.

We selected a U.S. street for the simulation and for the measurement campaign with the local environment created from raw data. These raw data, which model the geometry of the large structures (buildings, large objects), have been created using photogrammetry techniques from satellite stereoscopic imagery. The resolution of the input images was 50 centimeters.

Then, we post-processed these data to add geometrical details (buildings windows, doors, street objects, and so on) and to define the building materials. **Figure 6** shows how the model evolves from low to high resolution.

Figure 7 compares measured and simulated data when only obscuration has been considered (no multipath).

The results of the measurement campaign indicates that general trends are globally reproduced: the simulated levels of the received power are fair for GPS-05 and GLO-09. Regarding GPS-02, the drop of power is caused by the obscuration of the LOS signal as explained by the simulated data. However, the simulated power does not correctly mimic the small variations of power, mainly due to multipath interactions and interferences.

Figure 8 shows the evolution of the simulated carrier-to-noise-density ratio (C/N_0) when multipath is considered: a maximum of four reflections and one diffraction per multipath have been modeled and 10 multipaths per channel have been taken into account.

The graphs in **Figure 8** show a better consistency between measured and simulated data. The variation of power is more erratic and the visibility of the satellites is similar to actual measurements of signals.

Nevertheless, some discrepancies can be observed in the power level. Several causes can explain such discrepancies, such as quality of the scene (especially for low elevation satellites with only non-LOS reception) or the surrounding traffic during the test (vehicle or pedestrian) and the vegetation (for instance, isolated trees), which have not been modeled in the simulation.

Deterministic simulations never produce data that fits perfectly with real measurements. Indeed, they highly depend on the realism of the synthetic environment, and it is not



FIGURE 6 The synthetic environment models a narrow street. White buildings are defined in low resolution. Buildings close to the receiver are detailed in order to model the influence of small structures on the reception.

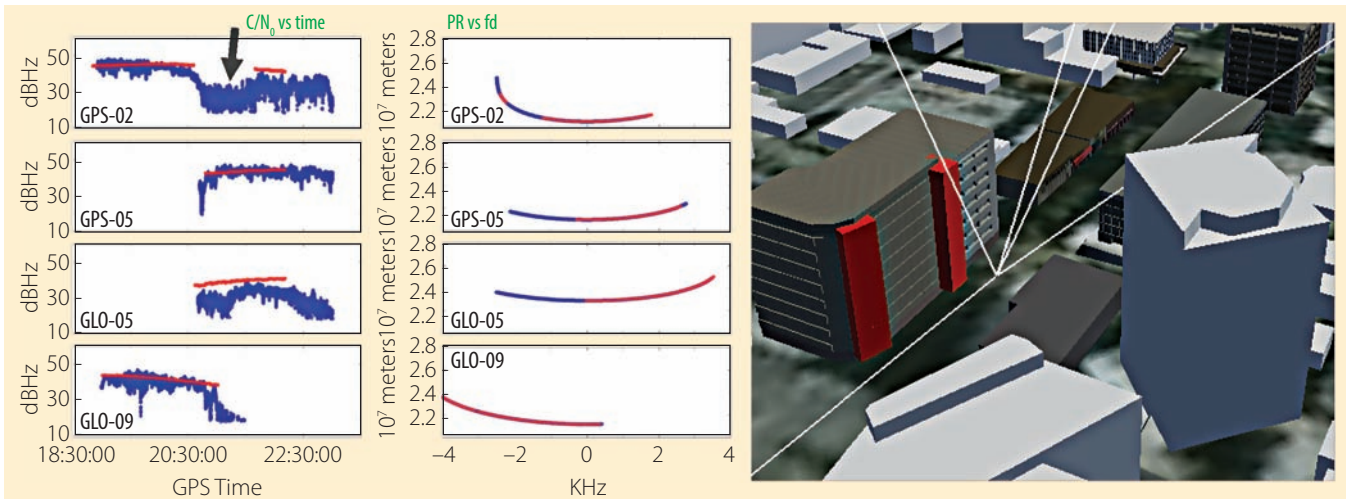


FIGURE 7 Evolution of carrier-to-noise-density ratio for various satellites considering only obscuration of signals. In blue: live measurement of signals; in red: simulation (only LOS signals considered). Left graphs: C/N_0 evolution for different satellites. Middle graphs: pseudorange versus doppler frequency. Right image: simulation with direct path from the satellites displayed in white

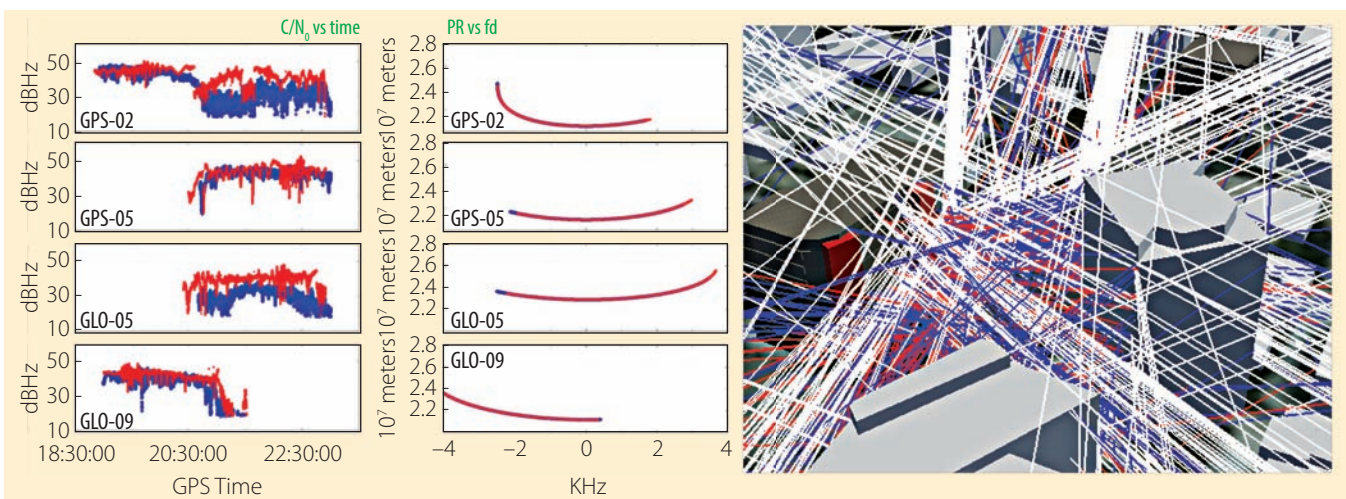


FIGURE 8 Evolution of live and simulated carrier-to-noise-density ratio for various satellites incorporating multipath effects. Left graphs: in blue: live measurement; in red: simulation (LOS + multipath signals considered) Middle graphs: pseudorange versus Doppler frequency. Right image: Simulation with reflections in red, diffractions in blue.

possible (with limited time and effort) to perfectly mimic a real configuration. However, the added value of using such algorithms is certain when we compare measurements with statistical and deterministic simulations.

Ray-tracing algorithms will not perfectly reproduce what happened or what could/will happen, but they can supply data that match real measurements as closely as possible, according to the quality of the definition of the synthetic environment.

Manufacturers

OKTAL Synthetic Environment (OKTAL-SE), Vigoulet-Auzil, France,

and the positioning technology business unit of **Spirent Communication plc** based Paignton, United Kingdom, have recently carried out measurement campaigns to validate the performance of a simulator based on ray-tracing method. SimGEN: the simulator software suite, developed by Spirent, produced the simulation scenario and modeling of the constellation and the receiver. SE-NAV, the propagation kernel, developed by OKTAL-SE, providing SimGEN with all the data needed to model deterministically multipath. SE-NAV uses a proprietary GPU ray-tracing algorithm to compute the obscuration as well as the mul-

tipath. The raw data used to create the Synthetic Environment was created by **Siradel**, Saint-Grégoire., France.

Author



GRÉGORY MOURA graduated from the top French Institute of Aeronautics and Space (ISAE-SUPAERO). Since 2005, he has been working as a project manager and expert in the field of

GNSS propagation simulation. He is currently managing the GNSS activities of the French company OKTAL Synthetic Environment. Moura is in charge of the management of the simulator SE-NAV that computes the propagation and the reception of GNSS signals in constrained environments. 