A novel approach for ray tracing optimization in wireless communication

Bernard Tamba Sandouno, Yamen Alsaba, Chadi Barakat, Walid Dabbous, Thierry Turletti

A novel approach for ray tracing optimization in wireless communication

Bernard Tamba Sandouno
Sogudo, Inria, Université Côte d'Azur
Sophia Antipolis, France
bernard-tamba.sandouno@inria.fr

Yamen Alsaba
Sogudo
Paris, France
yamen.alsaba@sogudo.com

Chadi Barakat
Inria, Université Côte d'Azur
Sophia Antipolis, France
chadi.barakat@inria.fr

Walid Dabbous
Inria, Université Côte d'Azur
Sophia Antipolis, France
walid.dabbous@inria.fr

Thierry Turletti
Inria, Université Côte d'Azur
Sophia Antipolis, France
thierry.turletti@inria.fr

ABSTRACT

Ray Tracing is a propagation modeling approach that accurately estimates the signal power received by end users while considering the details of the environment in their vicinity. The accuracy of this estimation is at the cost of high computational load and high memory consumption due to the heavy computation performed by processes such as the Ray Generation. In this paper, we introduce a site-specific ray generation technique able to generate up to 1 million rays within 5 seconds and a root mean square error for bandwidth estimation within 2 Mbps. Depending on the location of the antenna, the coverage area, the type of the terrain and the computational resources available, our technique gives the minimum possible number of rays required to accurately estimate end-users' signal power received and their download bitrate.

KEYWORDS

Ray tracing, ray launching, accurate signal power estimation

1 INTRODUCTION

Accurate electromagnetic signal power estimation has always been of great interest in both academia and industry. During the last decades, different sets of equations and algorithms called propagation models were developed in order to estimate the signal power received by end-users in a given location. These models differ from each other in the level of accuracy they offer and the time it takes them to perform the estimation. Some of the propagation models, namely the stochastic and empirical models, give poor level of accuracy while estimating the signal power; they still have the advantage of being fast. On the other hand, deterministic models give an accurate estimation of signal power but have the disadvantage of being computationally slow.

Stochastic models are computationally fast because they consider the environment of propagation as a set of random variables. Those variables make it possible to develop a model for the propagation channel using probability density functions, which allows to estimate the path loss with less input data. They are not accurate since exact details of the environment are not accounted for.

This work is supported by French ANRT agency under contract CIFRE 2020/1392.
An earlier version of this work has appeared in the proceedings of the ACM International Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (ACM MSWiM'22), October 24–28, 2022, Montreal, QC, Canada.

Empirical models on the other hand are built around a set of parametric equations for the characterization of radio wave propagation as a function of frequency, distance, and other conditions. These models are calibrated by measurements collected in a precise environment. Due to their low complexity, these models have low execution time. They are also very easy to implement either to estimate the path loss in a given location or to generate a whole Quality of service map. However, empirical models are usually not very accurate because they finally depend on the environment where they were originally devised [13].

Deterministic models from their side use thorough details of the environment of propagation for path loss estimation. They consider the complete 3D map and the characteristics of the environment. According to these models, waves' interaction with their environment is considered through reflection, refraction, scattering and diffraction, making them the most accurate ones among all the propagation models. However, this high accuracy is at the cost of a high memory consumption and computational load. Indeed, although very accurate, these models have the problem of not being practical for generating coverage maps in complex environments and are even more difficult to use in real time scenarios [2][7][17]. Despite their slowness, deterministic models are still required when there is a need to accurately estimate the signal power received by end users [13]. This is why our focus in this paper is on these models and specifically on *Ray Tracing* which is the most used deterministic model nowadays [2]. Our main objective is to reduce the complexity of Ray Tracing without compromising its accuracy.

Ray Tracing (RT) is based on the light/wave duality [17], i.e., waves can be treated as light rays. At high frequency, all the physical properties applied to light (rays) can also be applied to waves. The latter leads to replacing waves by rays in RT, i.e., they are the ones emitted and received by antennas. The slowness of RT mainly comes from the high computational load and high memory consumption of the processes involved in its workflow. A process of particular interest in its workflow and of particular need for estimating the so-called path loss between a transmitting and a receiving antenna is the ray generation one, also known as ray launching. This process is the main focus of our work.

Ray launching consists of launching rays in all directions in order to fully cover an antenna 3D radiation pattern. Ideally, an infinite number of rays is to be generated, which is not practically feasible. To approximate reality with a finite number of rays, rays

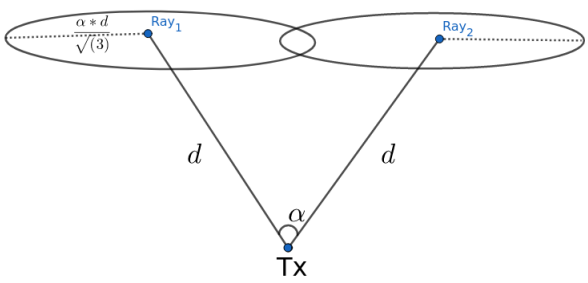

Figure 1: Ray cones overlap to avoid blank area in 2D

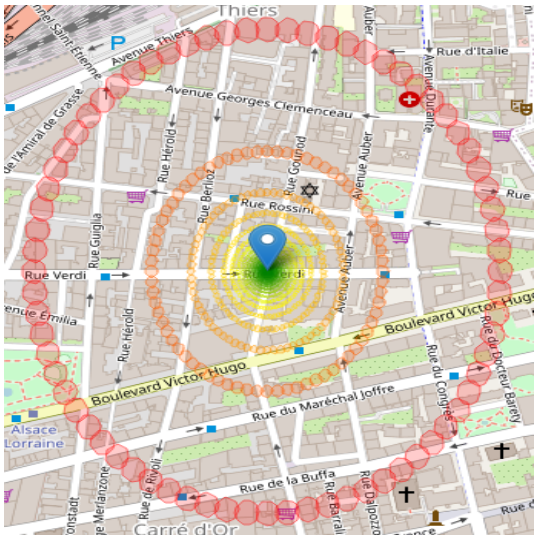

Figure 2: Example of ray cones launched with a constant angular separation α =5°

are generated in the form of tubes or cones centered at a line (called the ray), with the most used ones being ray cones. Ray cones are launched in such a way to fully cover all the propagation area around the antenna and to avoid blank zones. In order to meet this requirement, the radius of the spheres at the cross-section of the cones must be well chosen. If the ray cones only touch each other, this will lead to small areas between them that are not covered by the rays. To avoid this issue, the ray cones must overlap slightly as in Figure 1, which passes by increasing their radius by a constant multiplicative factor. The minimum radius increase factor that helps to fully cover the area between tangent spheres is $2/\sqrt{3}$. This can be easily derived from the property of the equilateral triangle composed by the centers of three adjacent spheres. The radius R_i of the i^{th} ray's sphere after traveling the distance d_i is therefore given by (1) [3], with α_i being the maximum angular separation between the i^{th} ray and its neighboring rays.

$$R_i = \frac{\alpha_i d_i}{\sqrt{3}}. \quad (1)$$

Choosing the same constant value for the separation angle α works perfectly in 2D scenarios. However in 3D scenarios, this still leads to inaccuracies and gaps in the propagation area [3] as shown in Figure 2. This problem is solved in the literature by using the icosahedron technique, which consists in subdividing the faces of an icosahedron into multiple uniform equilateral triangles. However, this technique is computationally slow and launches many rays that are wasted as the rays are launched in all possible directions regardless of the position of the receiving antenna. In real-life scenarios, the smaller the area of interest around the antenna, the larger the number of wasted rays is, hence increasing the computational load and the memory consumption of RT.

In this paper, we introduce a new ray generation technique that quickly and iteratively finds the optimal number of required rays to cover the receiving area. Our technique is site specific, i.e., for each scenario it gives the optimal number of rays that need to be launched in order to fully cover the area without any blank zone. By minimizing the number of rays that need to be launched, our method reduces the overhead incurred by those rays that never reach the receiver. Moreover, by generating rays in a flexible and iterative way, we were able to overcome the complexity and the computational slowness of the icosahedron technique. By utilizing our technique, we achieved the capability to launch up to one million rays within a span of 5 seconds on a laptop equipped with 16GB of memory and 7 processors running at 1.8GHz. As a result, we effectively minimized the overhead associated with the ray generation process. Simulations were then performed with this new technique in different real-life scenarios and a validation with respect to the state-of-the-art model implementing the original icosahedron technique was carried out. After validating our model, we made an extensive simulation study on the accuracy/complexity trade-off of Ray Tracing depending on whether we are in an urban, suburban, or rural area. This is meant to help one choose the suitable number of rays to launch given a specific scenario while having sufficient information about the complexity required to reach a sought level of accuracy. Through all these simulations, we were able to observe that our solution for ray generation was flexible, robust, and computationally fast at almost no cost compared to standard ray tracing.

The rest of the paper is organized as follows. In Section 2, we explain the techniques used in the literature to accelerate Ray Tracing in general and more specifically the ones to overcome the complexity of the icosahedron technique. Section 3 contains the technical details of the icosahedron technique and of our ray generation technique. The validation of our proposed technique and its performance evaluation are presented in Sections 4 and 5. This work has been first published in [11], to extend it further for the purpose of the current journal publication, we add here a site-specific accuracy/complexity study that we detail in Section 6. Finally, the conclusion of our work is presented in Section 7 with perspectives on our future research work.

2 RELATED WORK

Due to the complexity of Ray Tracing (RT), different solutions have been proposed in the literature in order to accelerate it. Space division techniques such as Uniform Division and Bounding Volume Hierarchy (BVH) are meant to reduce the number of Ray Object

Intersection tests [17][16]. This is because a naive Ray Tracer performs an intersection test between all the rays and all the buildings. The previous techniques help to run the intersection test only on buildings that can be potentially hit by the rays, hence reducing RT complexity. BVH has been further improved using spatial acceleration structures in order to ease the ray object intersection test [7]. Another technique used is based on the use of efficient Graphical Processing Unit (GPU). RT execution time is then drastically reduced thanks to hardware acceleration.

Some other techniques are developed to by-pass the overhead incurred by the icosahedron technique used to generate rays in a way to fairly cover the space around the emitting antenna. First, Matlab in their Ray Tracing implementation computes offline the direction and the maximum angular separation of each of the vertices of the geodesic structure obtained from the icosahedron. Three fixed tessellation frequencies are chosen, the calculation of the vertices' coordinates and their maximum angular separation are computed offline, and those values are then simply loaded in the memory [6]. Nevertheless, this approach lacks flexibility because the number of rays launched is limited to those three choices making it impossible for Matlab users to adapt the number of launched rays depending on the scenario they have under study. Moreover, the icosahedron technique leads to many useless rays since it assumes that rays are launched in all possible directions, which can go far beyond the simulated area.

Further, the authors in [12] propose a technique based on a "golden spiral". With such a technique, it is possible to evenly generate points on a sphere in order to launch rays passing by them. This approach has the advantage of being flexible, i.e., one can launch any number of rays. However, the technique gives no clue about the suitable number of rays to launch in a given environment. Hence one must try different number of rays depending on each scenario before finding the best setting. This can easily become cumbersome since the number of possible rays to launch can be huge. On the other hand, this technique is also as brute force as the icosahedron technique, i.e., rays are launched in all possible directions to be sure not to miss the receiver. With this, many rays are launched and most of them can be useless if the receiver is located just near the antenna for instance. Those useless rays will go anyway through all the subsequent RT process, like the intersection test with the buildings, hence adding more overhead to RT.

In this paper, we focus our work on solving the complexity related to the icosahedron technique used to launch rays in the literature. Our method outperforms existing solutions to the icosahedron technique issue, because on the first hand it solves the complexity of the icosahedron technique by its ability to generate a large number of rays in a reasonable time using an iterative and adaptive process. On the other hand, our method is site specific, i.e., depending on the coordinates of the antenna and the radius of the area of interest, our method can only generate the necessary rays that can be potentially received by receivers. For instance, our method will launch less rays for computing the signal power for a receiver located just near the antenna and will launch more for another one that is far away. In the same manner, it can adapt the number of rays to the simulated area, leading for example to more rays for an area of interest of 5000 meters radius as compared to an area of interest of 500 meters radius. For a given scenario, our method can thus produce an appropriate range for the number of rays to be launched. We will show in this paper that within this range, one can safely choose any number of rays at almost no additional cost. Hence, one can choose to launch the minimum possible number of rays while maintaining the accuracy of the signal power estimation. The latter has the advantage that, with less number of rays, less intersection tests with the buildings are needed as well as a lower number of reception tests, hence reducing the high computational load of RT.

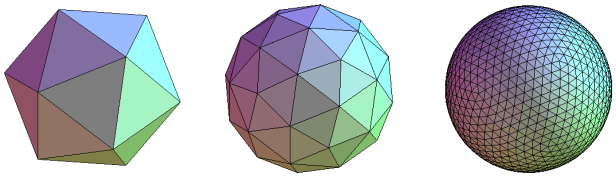

Figure 3: The first one is the original icosahedron (20 faces). The second and the third ones are the geodesic structures with tessellation frequencies $n = 2$ and $n = 10$ respectively.

This paper is an extended version of our previous conference paper [11]. We further incorporate here a sensitivity study whose purpose is to aid in determining the appropriate number of rays to be launched, considering factors such as zone type (urban, suburban, rural), available resources, and desired level of accuracy. This study addresses the tradeoff between accuracy and complexity in Ray Tracing, aiming to reduce the complexity of Ray Tracing while preserving its accuracy based on the number of rays launched. By the end, we provide recommendations regarding the optimal number of rays to be launched depending on the specific scenarios being considered.

3 SYSTEM MODEL

In the literature the ray generation technique used is the icosahedron technique despite its computational slowness. It consists in subdividing all the faces of the original icosahedron (20 faces) into many other equilateral triangles to be able to adapt the number of rays. The subdivision of the faces and the launching of rays is performed as follows [14]:

- Choose one edge of the face (that is also an equilateral triangle). Let AB be the chosen edge and C be the vertex facing the edge.
- Subdivide AB into n segments of equal length, n is the tessellation frequency. Let $P_0, P_1, ..., P_n$ be the chosen points.
- Trace the segment $CP_{n/2}$ and trace all the segments parallel to $CP_{n/2}$ and passing by the chosen points.
- Repeat this operation for all the edges of all the faces of the icosahedron. This will result in small equilateral triangles on each face of the original icosahedron.
- Project all the vertices of the small equilateral triangles on the circumscribed sphere to the icosahedron.
- All the $10n(n + 1)$ [3][14] rays are then finally launched on the vertices of the obtained sphere, with n being the tessellation frequency.

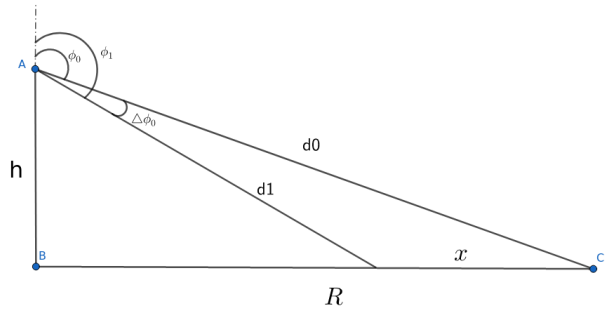

Figure 4: Illustration of our ray generation procedure

Figure 3 shows the original icosahedron and the geodesic structures obtained from it for two tessellation frequencies, $n = 2$ and $n = 10$. This technique has the advantage of fully covering the whole propagation area without any gap. Nevertheless, the geometrical algorithm explained above is computationally heavy and its complexity grows faster with n. Hence, generating rays using this technique is computationally slow [12].

On the other hand, our ray generation technique is based on an iterative and adaptive approach[11]. Given the height of the antenna and the radius of the area of interest, it generates the optimal number of rays required in order to fully cover the potential area where the receiver is located while minimizing the overlap between the adjacent ray cones. To meet this challenge, we set the elevation step $\triangle\phi$ to be iterative, i.e., its value to depend on the previous ray. Afterwards, we set the radius of all the rays to respect (1). Finally, given an elevation angle ϕ_i, the goal is to find a step $\triangle\phi_i$ so that rays on the next elevation ϕ_{i+1} overlap with their neighbors having ϕ_i as elevation angle in such a way to avoid any gap. Once $\phi_{i+1} = \phi_i + \triangle\phi_i$ is found, we continue the process to find ϕ_{i+2}, so on and so forth until $\phi \geq \pi$.

To illustrate further our idea, we consider the example in Figure 4. Given the elevation angle ϕ_0, we look for the $\triangle\phi_0$ so that the cross-sections of Ray 0 and Ray 1 located on the same azimuth overlap with each other in such a way to avoid any gap. The value of ϕ_1 found will help to find ϕ_2 and so on.

Proceeding this way has the advantage of being adaptive, i.e., the full area is covered with less possible number of rays launched. Nevertheless, it is not trivial to find how much the rays must overlap in order to remove any gap. Hence, we changed our constraint to be that the rays must only touch each other (without overlapping) and afterwards the gap removal was done with binary search.

From Figure 4, the cross-sections of Ray 0 and Ray 1 touch each other, if condition (2) is met, with d_0 being the distance travelled by Ray 0 and d_1 the distance Ray 1 should travel in order to meet (2).

$$x = Radius_0 + Radius_1 \Leftrightarrow x = \frac{\alpha_0 d_0 + \alpha_1 d_1}{\sqrt{3}} \tag{2}$$

Using the law of cosine, the Pythagorean theorem and the small angle approximation and the binary search (for overlapping), we derived (3), with $k = 2\sqrt{3}$.

$$\triangle\phi_i = \frac{2 * \alpha_i}{(\alpha_i - k) * \tan\phi_i}, \quad \pi/2 < \phi_i \leq \pi \tag{3}$$

Note that, the value of α_i depends on the angular distance in elevation and azimuth, $\alpha_i = max(\triangle\phi_{i-1}, \triangle\theta_i)$. The value of $\triangle\theta_i$ is found at each iteration using the angular distance formula in (4) with the constant value β being the angle between rays in azimuth. This constant value β has a big impact on the number of rays launched since it is the one that decides how many rays are launched for each given elevation $\triangle\phi_i$.

$$\triangle\theta_i = \cos^{-1}\left[(cos\beta - 1)\sin(\phi_i)^2 + 1\right]. \tag{4}$$

On the other hand, in (3), we have the condition $\pi/2 < \phi_i \leq \pi$; this is because when $\phi_i \leq \pi/2$, the signal is not received by any receiving antenna due to the fact that most of the building's facets are vertical and the transmitting antennas are usually higher than the receiving ones. However, following the same approach, interested readers can derive the formula for the case $\phi_i \leq \pi/2$.

Our method is said to be site specific, because we consider local information about the environment of interest in order to generate the optimal number of rays in that specific scenario. Practically, our method adapts the angle from where the first ray must be launched in order to fully cover the potential area where the receiver is. To do this, an initial value of elevation ϕ_0 is chosen as a starting point. This value is chosen with regards to the radius of the area and the height of the antenna used. From Figure 4, this radius is R and the value of ϕ_0 is derived in (5) with h being the height of the antenna.

$$\phi_0 = \pi - \arctan(R/h) \tag{5}$$

Our algorithm to iteratively identify the elevation angles of rays is summarized in Algorithm 1 shown below. The constant azimuthal angular separation β is known in advance and the algorithm returns the elevations of the rays. Further, rays are launched using their Cartesian coordinates $[0, 2\pi]$ (with a step of β) and ϕ.

Algorithm 1 Site-specific ray generation algorithm

Require: R, h, β ▷ R: Radius of the area ▷ h: Height of the antenna ▷ β: Azimuthal angular separation

```
k ← 2√3
△φ ← 0
φ ← π − arctan(R/h)
Elevation ← []
while φ ≤ π do
    Elevation.add(φ)
    △θ ← cos⁻¹ [(cosβ − 1) sin(φ)² + 1]
    α ← max(△θ, △φ)
    △φ ← 2∗α / ((α−k)∗tan φ)
    φ ← φ + △φ
end while
```

Figure 5 shows an example of rays launched using our site-specific and iterative ray launching technique. One can see that all the propagation area is covered without any gap and with minimum number of rays launched.

After using our technique to generate and launch rays, subsequent RT processes are further performed. The intersection test is performed between rays and the buildings in order to determine if

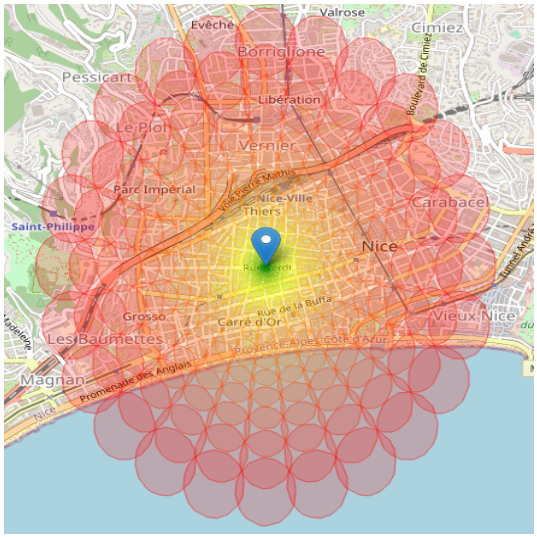

Figure 5: Our ray generation technique without any gaps and with minimum overlap between rays

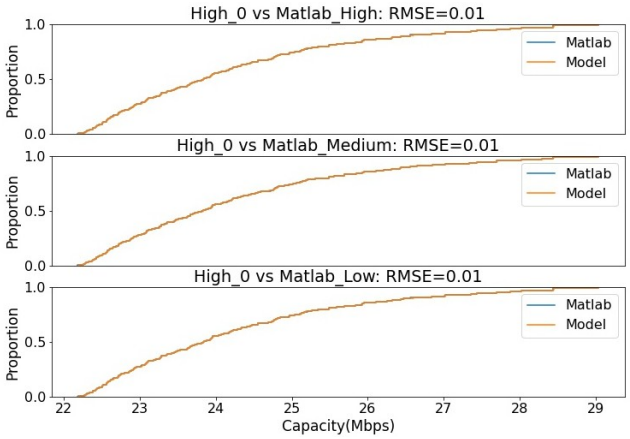

Figure 6: High angular separation in LOS

a ray hits a building and whether it is being reflected or diffracted. The complexity of this intersection test is tightly linked to the complexity of the environment and the number of rays launched. The more rays there are, the more intersection tests will be performed, hence the higher will be the computational load and the memory consumption. However, our technique helps to reduce the computational load of this process. The rest of the processes until rays reception by the receiver is detailed in [15]. The signal power carried by received rays is computed using (6) [4][5].

$$P_{rx} = P_{tx} G_{rx} G_{tx} \left(\frac{\lambda}{4\pi d_{los}}\right)^2 \left| \sum_{m=1}^{M} \prod_{n=1}^{N} R_{mn} \frac{d_{los}}{d_m} e^{j\frac{2\pi}{\lambda}(d_m - d_{los})} \right|^2 \tag{6}$$

In this equation, d_{los} denotes the line-of-sight (LOS) distance between the transmitter and the receiver. M is the number of rays received. N and d_m are respectively the number of reflections and the unfolded distance of the m^{th} ray from the transmitting antenna to the receiver. R_{mn} is the Fresnel coefficient at the n^{th} reflections of the m^{th} ray; it can be one of the parallel or perpendicular components of the Fresnel equation depending on the polarization of the antenna [4].

In case of multiple antennas involved in the environment considered, our technique will generate rays for each individual antenna and the power corresponding to each antenna is then computed using (6). When a receiver receives signals from different antennas, the SINR (Signal to Interference plus Noise Ratio) is further computed to consider interference from other antennas at the same carrier frequency. To compute the SINR, the antenna with the highest signal power is considered as the source of signal and the others at the same frequency as the signal source are considered as sources of interference. Finally, the Shannon capacity formula is used to compute the download bitrate of the receiver (7), where B is the channel frequency bandwidth.

$$Bitrate = B * log_2(1 + SINR) \tag{7}$$

4 NUMERICAL SIMULATIONS

We implemented our site-specific approach and validated it against the approach in the literature that uses the icosahedron technique for ray generation. As explained in Section 2, Matlab in their RT implementation uses the icosahedron technique. Hence, we checked the correctness of our technique by comparing it with the Matlab implementation of RT.

The main challenge with the validation of our technique against the icosahedron technique is the choice of the number of rays to launch. For a first validation of the accuracy of the received signal power, and for the purpose of fairness, we set the number of rays to be comparable in both cases. Indeed, in the icosahedron technique, rays are launched in all directions, while in ours, only the optimal number of rays useful in each scenario is launched. Since our technique launches rays only below the horizon (starting with a $\phi_0 \geq \pi/2$), we adapt for each of the 3 sets of number of rays available in Matlab, the number of rays that need to be launched in our case. Indeed, in the 3 sets of number of rays available in Matlab, 40962, 163842 and 655362 rays are launched respectively for the high, medium, and low angular separation [6]. For each of the latter case, we set the values of β (the constant azimuthal distance that determines the number of rays launched in azimuth at each elevation) to 2.35°, 1.19° and 0.75°, which corresponds respectively to 19252, 77005 and 308020 rays launched in our model. Simply said, where Matlab launches 40962 rays, 19252 rays are the equivalent for our model and so on. For the simulation setup, we set the radius of the coverage area to 5000m.

We start by assessing the sensitivity of our technique regarding the number of rays launched by Matlab. We compare each of our cases to Matlab's 3 resolutions. We repeat this process for different maximum numbers of reflections allowed: 0, 2 and 4. Moreover, we test the accuracy of our technique on 3 different urban environments in the city center of Nice, France. For the bitrate computation, we used the Shannon capacity formula (7) with a bandwidth of 1MHz and a receiver noise power of -107 dBm.

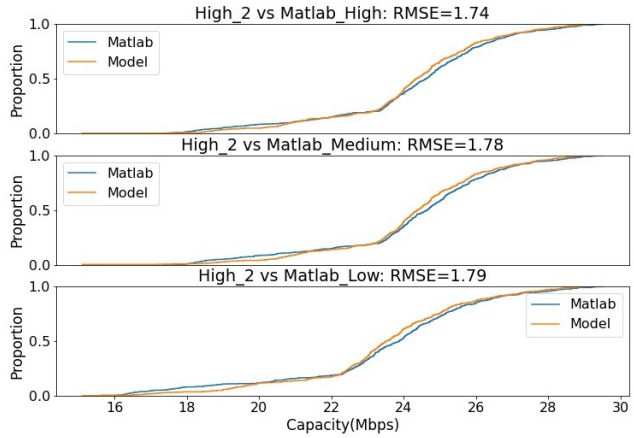

Figure 7: High angular separation with 2 maximum number of reflections

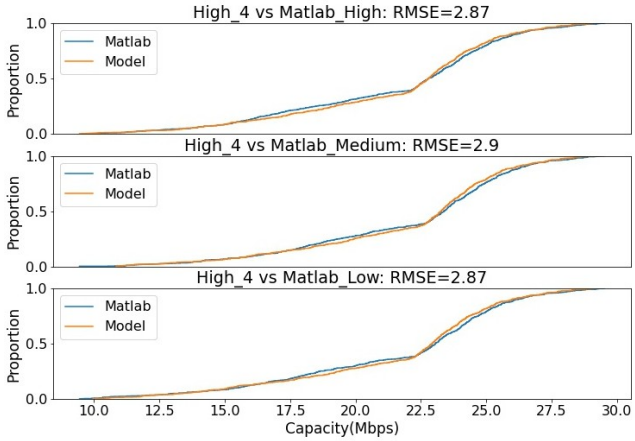

Figure 8: High angular separation with 4 maximum number of reflections

Figures 6, 7 and 8 show a comparison between our solution in the case of a high angular separation and all the 3 other scenarios available in Matlab for different maximum number of reflections. The figures show a direct comparison between our solution and Matlab using the Cumulative Distribution Function (CDF) of the bitrate estimation. We can observe from the figures that our bitrate or capacity estimation has the same distribution as Matlab, for the three resolution scenarios of Matlab (High, Medium, and Low) as compared to the *high* resolution scenario in our model. This gives an idea on the fact that by using our approach that is adaptive and gives the less possible number of rays in a reasonable time, we can accurately estimate the signal power and consequently the download bitrate of end users. From the Cumulative Distribution Function (CDF) we can also see that our method is not that sensitive to the number of rays launched in Matlab. We will explain this sensitivity more deeply in the next section. We can further observe that the mean error made by our technique regarding the one of Matlab increases slightly and is less than 3 Mbps in all cases. This

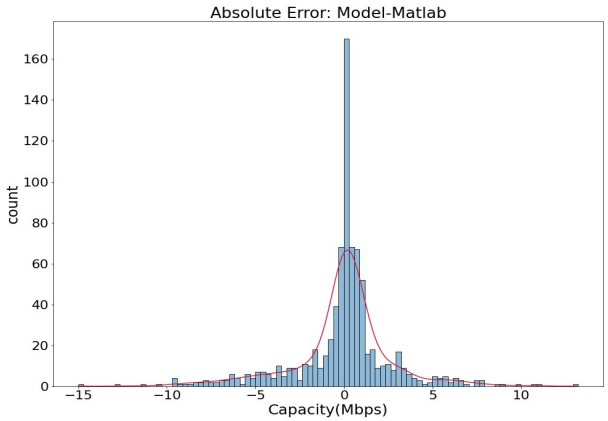

Figure 9: Mean Absolute Error distribution

Table 1: RMSE (in Mbps) for the 1st terrain

	High_0	**High_2**	**High_4**	**Med_2**	**Med_4**
Matlab_High	0.0	1.51	2.48	1.45	2.41
Matlab_Medium	0.0	1.58	2.45	1.47	2.41
Matlab_Low	0.0	1.55	2.54	1.4	2.49

remains a good trade-off regarding the advantages offered by our technique as explained earlier.

By zooming on the case of high angular separation with 4 reflections, Figure 9 shows the absolute error distribution for bitrate estimation between our solution and Matlab. We can see from this figure that the error follows a Gaussian distribution centered around 0, i.e., most of the errors made by our model are around 0. This highlights the ability of our model to accurately estimate the signal power and the bitrate as compared to the widely used icosahedron technique.

Our simulations were performed on 3 urban terrains in the city of Nice in France. Since the plots for the 3 terrains were bringing similar results, only the plots for one terrain were shown in Figures 6, 7 and 8. We summarize here the results obtained from the simulations performed on the 3 terrains in Tables 1, 2 and 3. For each terrain, the simulation was done with different angular separations and different number of reflections. Each column is a comparison between the estimation performed by our technique and the one of Matlab. For example, the column **Med_4** means that we are comparing our *medium* angular separation with 4 reflections to the 3 angular separations available in Matlab. Each cell represents the root mean square error (RMSE) in Mbps between our results and the ones of Matlab. The tables show slight variations of the RMSE from one terrain to another due to the differences in the terrains themselves. These small variations highlight the robustness and scalability of our technique, and its ability to be accurate regardless of the terrain used. With all these simulations, one can see that our method is correct, robust to terrain change and capable of maintaining the accuracy of RT while launching less rays.

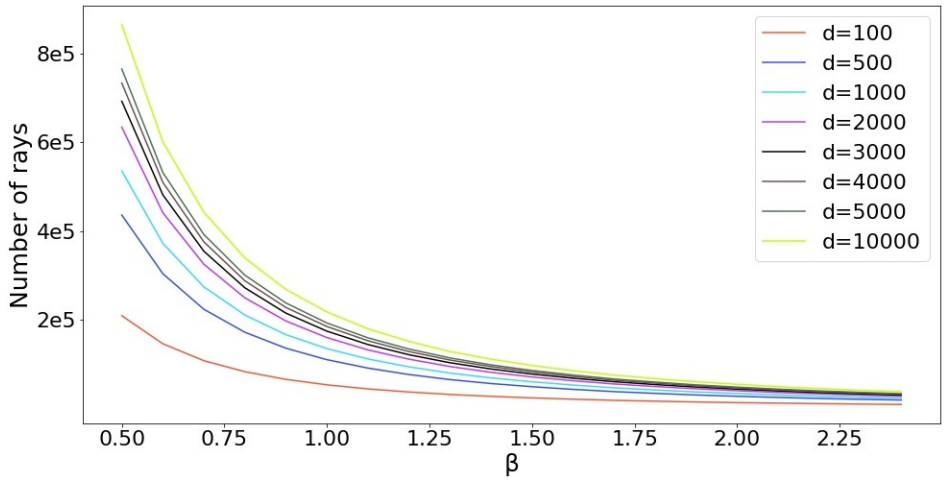

Figure 10: Number of rays launched in different scenarios

Table 2: RMSE (in Mbps) for 2nd terrain

	High_0	High_2	High_4	Med_2	Med_4
Matlab_High	0.0	1.39	2.14	1.29	1.94
Matlab_Medium	0.0	1.47	2.26	1.39	2.12
Matlab_Low	0.0	1.38	2.28	1.27	2.18

Table 3: RMSE (in Mbps) for the 3rd terrain

	High_0	High_2	High_4	Med_2	Med_4
Matlab_High	0.01	1.74	2.87	1.69	2.73
Matlab_Medium	0.01	1.78	2.9	1.74	2.69
Matlab_Low	0.01	1.79	2.87	1.76	2.59

5 SENSITIVITY ANALYSIS

After explaining the correctness, robustness, and the cost of our site-specific ray generation technique, we dig deeper in this section into the gain it offers through a sensitivity analysis of the results versus the number of rays. As a site-specific method, we aim at optimizing the number of rays launched by reducing the number of rays wasted which allows us to reduce the computational load of RT. In traditional RT, for covering an area of 100 meters radius, one need to launch as much rays as in the case of 5000 meters radius (i.e., with no consideration of the receiving area). However as shown in Figure 10, our technique optimizes the number of rays launched by taking into account the radius of the area of interest and the height of the antenna. The x-axis of this figure represents β, the constant azimuthal angular separation and the y-axis determines the number of rays launched. The figure gives for different coverage area radius value d, the number of rays that are launched as a function of β. The first intuitive observation is that the higher the azimuthal angular separation is, the less rays are launched, which is intuitive as the angular step between adjacent rays increases. Second, the figure provides the minimum number of rays necessary to fully cover the propagation area without any gap for each value of β and d. The curves in the figure thus help to have a sense of how many rays are effectively launched by our method, and consequently how many are saved compared to the icosahedron technique. We can observe that the gain obtained depends on the coverage area d: small coverage areas need less rays than bigger ones. We can noticeably see that at $\beta = 0.5°$, almost 1 million rays are required for 10000 meters radius while only 200000 rays are enough for the 100 meters case. Our method can then automatically save almost 800000 rays to be launched when switching between these two environments, which later helps to reduce the computational load and the high memory consumption of RT.

As our method reduces the complexity of the icosahedron technique by generating less rays in an adaptive and flexible way, we make different simulations to assess the time taken by our model to generate rays. Rays' generation time includes the time to find the azimuth and elevation of each ray at departure and the time to launch the rays given those angles. The mean time taken to do this is shown in Figure 11. The figure plots the generation time as a function of the azimuthal angular separation β for different areas of radius d. This helps to get an idea of what is the time required to launch a certain number of rays. Naturally we see that the smaller the coverage area's radius, the less time is required. For instance, at $\beta = 0.5°$, it takes 1.5 seconds to generate rays at $d = 100$ meters, while it takes 5.5 seconds for $d = 10000$ meters. This comes from the ability of our method to minimize the number of rays necessary in each specific scenario. Moreover, we see that almost 1 million rays can be launched by our technique in almost 5 seconds. This shows that our technique is clearly less complex than the icosahedron technique.

We move our performance evaluation further by assessing the sensitivity of our approach regarding the number of rays launched. Said differently, we seek to evaluate the change of accuracy of our signal power estimation compared to the icosahedron technique when rays are launched. We changed the number of rays by varying the value of the azimuthal angular separation β. We performed new simulations for different values of β by setting the values of the radius d and the height of the antenna to 5000 meters and 30

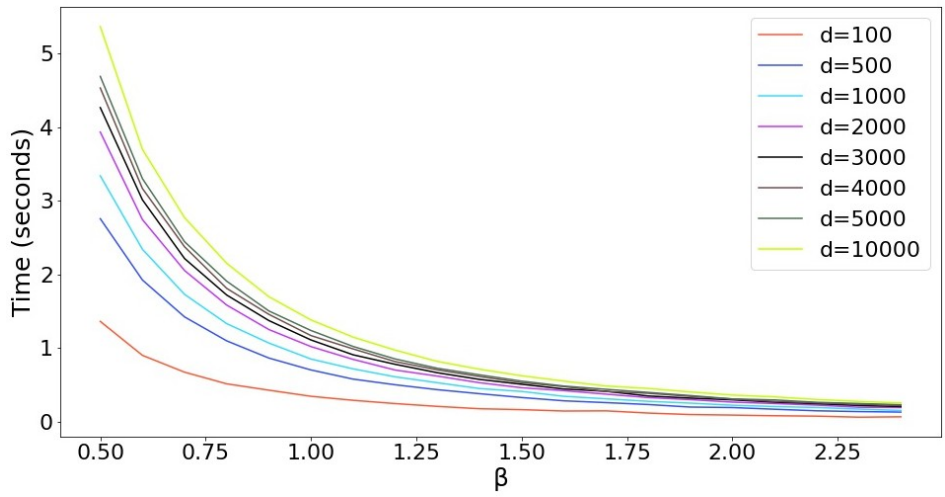

Figure 11: Time to generate rays in different scenarios

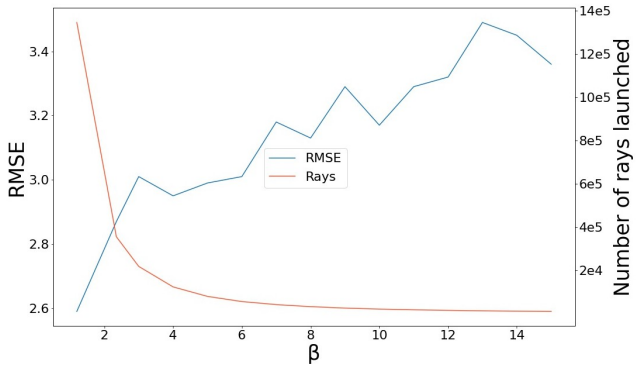

Figure 12: Sensitivity study on Terrain 3

meters respectively. Each of our simulation results was compared to Matlab. We compared our results to Matlab's *low* angular separation, because of its high accuracy compared to the high and medium ones. A range of β values were taken with 15° being the maximum, which is needed by the small angle approximation in our technique.

For each value of β in the above range, we compute the RMSE of the bitrate estimate with respect to the low resolution of Matlab and show it in Figure 12. The value of the number of rays launched is also given as a function of β to help having a better sense of the efficiency of our technique. From this figure we can see that our technique is not very sensitive to the value of β, i.e., the loss in terms of accuracy is very small, for instance for rays launched at 2° and the other ones at 15°. This property is very important since one can safely choose an azimuthal angular separation of 15°, hence launching less rays and ending up reaching almost the same level of accuracy as for other rays using a smaller value of β. Therefore, the minimum possible number of rays can be launched at a low cost. Since less rays are launched, the computational load and the memory consumption of RT can be reduced.

Figure 12 shows the studies for only one terrain. We performed the simulations on 2 different other terrains to check the robustness of our results. Table 4 shows the summary of the RMSE values of the bitrate estimation obtained in each case. Despite the slight variations from one terrain to another, we can observe from the table that the difference in terms of accuracy is still very small. This confirms the robustness of our technique and its ability to launch the minimum possible number of rays while keeping the overall accuracy within acceptable range. Furthermore, our method can be easily applied in a realistic scenario where there exist multiple antennas. Since antennas are independent, they are treated independently from each other.

The next section will move the study further by evaluating the impact of changing the number of rays on the accuracy of our estimation and the complexity of the computation while varying the number of rays in more complex and realistic scenarios including multiple antennas.

6 ACCURACY/COMPLEXITY TRADE-OFF

In the previous section we show the effect of a chosen value of β on the accuracy in order to help users choose the right number of rays to launch depending on their scenario. In this section, we dig in depth in this study by considering different types of terrains in large-scale scenarios involving multiple antennas. This is to fully help users get the most out of our site-specific technique. Our extensive study shows the pros and cons of a given number of rays in terms of accuracy and complexity. As the number of rays to be launched depends mostly on the value of β chosen, our study highlights on one hand the effect of β on the Ray Tracing's bitrate estimate and on the other hand, the computational complexity as a function of β and the type of the terrain. In a nutshell, we aim to emphasize on the accuracy/complexity trade-off of Ray Tracing based on the number of rays launched and the type of terrain. The latter is meant to help users have full control of the technique and to tune it depending on the level of accuracy sought and the available resources.

Table 4: RMSE (Mbps) vs beta for all the 3 terrains

β	**1.19**	**2.35**	**3.0**	**4.0**	**5.0**	**6.0**	**7.0**	**8.0**	**9.0**	**10.0**	**11.0**	**12.0**	**13.0**	**14.0**	**15.0**
Terrain 1	2.49	2.54	2.57	2.74	2.78	2.88	3.0	3.04	2.98	3.1	2.99	3.19	3.21	3.18	3.18
Terrain 2	2.18	2.28	2.39	2.51	2.6	2.73	2.7	2.86	2.88	2.84	2.82	2.95	2.97	2.93	2.99
Terrain 3	2.59	2.87	3.01	2.95	2.99	3.01	3.18	3.13	3.29	3.17	3.29	3.32	3.49	3.45	3.36

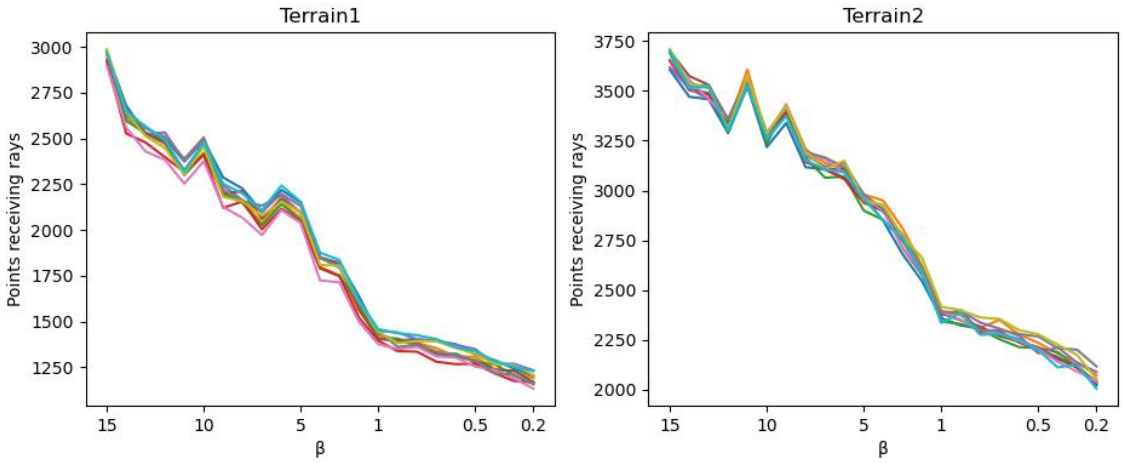

Figure 13: Urban areas

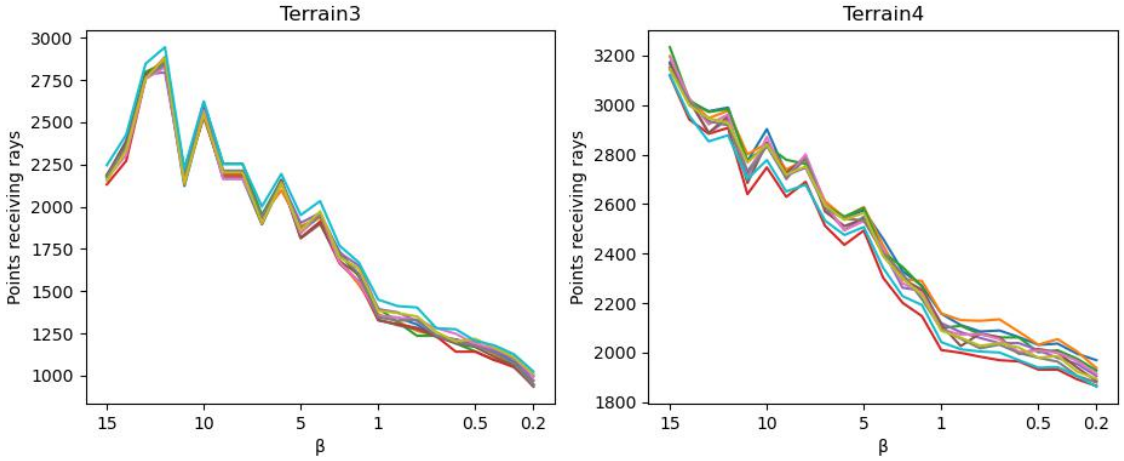

Figure 14: Suburban areas

Table 5: Terrains description

Terrain	Terrain 1	Terrain 2	Terrain 3	Terrain 4	Terrain 5	Terrain 6
Zone	Urban	Urban	Sub-urban	Sub-urban	Rural	Rural
Buildings	5141	5406	2861	1031	755	431
Antennas	8	6	2	5	1	1

To stress out the effect of the number of rays on the computational complexity, we simulated our model on a more powerful server featuring 32 processors and 94GB memory. This allows us to perform our complexity/accuracy trade-off based on real world data sets in order to show the effectiveness of our site-specific ray generation model in real life scenarios involving multiple antennas. We obtained the buildings' information from the French National Institute of Geographic and Forest Information (IGN) [9]. The information on the position and characteristics of transmitting antennas are obtained from the French telecommunications regulatory agency (ARCEP) [1]. Finally, we obtained the zone type from the French national statistics institute (INSEE) [8].

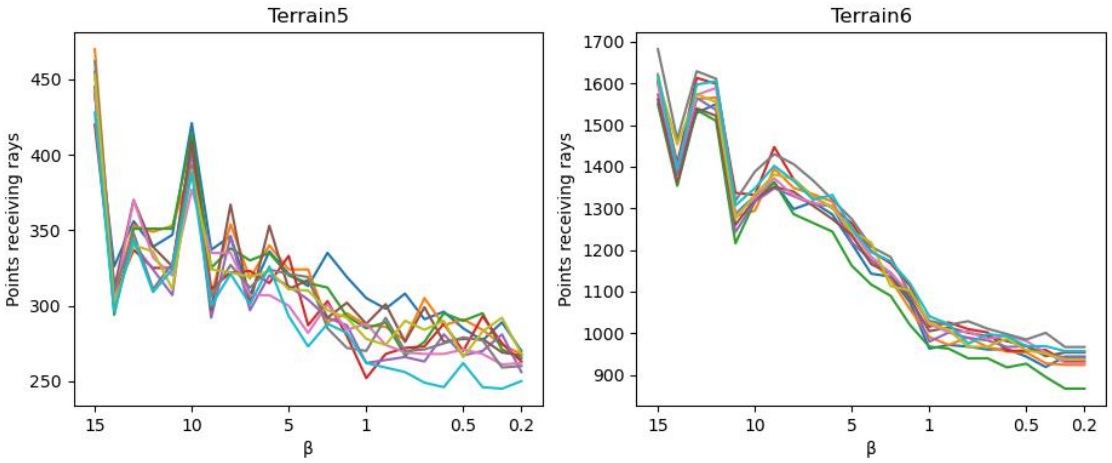

Figure 15: Rural areas

For each of these publicly available and up-to-date datasets, we built an Application Programming Interface (API) that, given a location and a radius of the considered area, gives the type of the terrain, all the buildings and all the transmitting antennas in that area. We developed this API based on these publicly available datasets that we stored in a database. This API has 3 endpoints that help to obtain all the information needed by our Ray Tracing model.

- The first endpoint accepts latitude and longitude as input, representing a geographical address. It then provides output indicating whether this address corresponds to an urban, suburban, or rural area.
- The second endpoint requires an address and a radius as input. It returns all the buildings within the specified radius. Each building is described by its shape, location, and height.
- The third endpoint also takes an address and a radius as input. It returns all the antennas within the given radius. This endpoint offers additional filtering parameters, such as the technology (4G, 5G) and the name of the mobile network operator. For each antenna, the output includes information about its geographical location, height, frequency, bandwidth, technology, and the distance between the antenna and the provided address.

For our study, we considered 3 different types of terrains: Urban, Suburban and Rural. For the sake of highlighting the robustness of our technique, we took 2 different cities for each terrain type. For all the 6 cities considered, the studied area is 1km radius around a given location in the city. Within this radius, 5000 random receiving points are taken, whose heights were set to 1.5 meters. The latter corresponds to the average height of a mobile phone hold by a person. To this height, we added the elevation from the sea level of the receiving point. The latter is obtained from the IGN elevation API [10]. Given an address, this IGN's API gives the altitude of the address regarding the sea-level.
To ensure the stability of the model, we perform our experiments 10 times on different 5000 random receiving points chosen by changing the seed used to generate them. Simply said, for each value of β we have 10 different experiments run on different sets of receiving points. Also, for each experiment, the value of β is the same for all the multiple antennas in the areas considered and the interference coming from surrounding antennas is accounted for when computing the receiver's download bitrate. Table 5 gives the description of the terrains considered in our study. We naturally see that the number of buildings and antennas decrease when moving from urban to rural areas.
For each terrain shown in Table 5, we performed the accuracy assessment as a function of β and also the computational complexity assessment in terms of execution time and memory consumption.

6.1 Accuracy

In our model, the higher is the value of β, the higher will be the angular separation of a ray cone with its neighbors and according to equation (1), the higher will be its radius. In an obstacle free environment, the space is covered by big overlapping circles when large values of β are used. With small values of β instead, the space is covered by small overlapping circles. However, in complex environments, the big radius of rays tends to overlap, or even fully cover, some obstacles, therefore showing places as covered by rays while in reality they are not. This typically happens in areas behind obstacles. We will be referring to this phenomenon as *ray overestimation*. In the case where small radius rays are launched, rays will be reflected instead by the obstacles and no overestimation should occur. In this section we assess how often this overestimation occurs and link it to the number of obstacles and the value of β used to generate rays. We further study how this phenomenon affects the accuracy of our model and give insights about the suitable values of β to use depending on the use case in order to ensure the best accuracy of the model.

First, to quantify the ray overestimation, we evaluate how many of the 5000 receiving points receive rays for each value of β. Knowing that small values of β are more precise than the large ones in terms of ray overestimation, we assess how many points receive rays for each value of β and how far are these values between large

and small values of β depending on the type of the terrain. Figures 13, 14 and 15 show how the number of receiving points varies as a function of β. The 10 line plots in the figures correspond to the same experiment repeated over 10 different sets of 5000 points (randomly selected). From these figures and as expected, we see that the number of points receiving rays decreases as the value of β decreases. However, as the value of β gets close to 0, the level of overestimation starts to be negligible, and the curves tend to be less and less inclined. Nevertheless, the extent of this overestimation is more or less severe from one terrain to another. In rural areas for instance as shown in Figure 15, the difference of points receiving rays between large and small values of β is relatively small. The decrease is in the order of 43%. However, in suburban and urban areas, the relative decrease is more than 50% for both cases. This high overestimation in urban and suburban areas typically comes from the complexity of the environment, i.e, the high number of obstacles that are present in those areas. This is to say that in complex environments, large values of β are subject to ray overestimation phenomenon. Nevertheless, in less complex environments, the overestimation is relatively small, i.e., one can launch a small or a high number of rays with only small losses in terms of precision.

We move further our study to assess the impact of the azimuthal angular separation β on the bitrate estimation accuracy. This is meant to give a clear idea about the loss and gain in terms of bitrate depending on the type of terrain. To do this, we run the same experiment over the same sets of 5000 receiving points and compute the download bitrate for each of those points that receive the signal power. The download bitrate considers interference coming from surrounding antennas. For each of the 10 experiments and for each value of β, we take the average bitrate over all the points receiving rays. As result, we obtain a matrix containing the average bitrate for each value of β and for all the 10 experiments. Figures 16, 17 and 18 are box plots showing these results for Terrain 1, 4 and 6. We can observe in these figures how the average bitrate varies as a function of β. In urban areas as shown in Figure 16, we see high fluctuations for a wide range of β values. It starts to converge around $\beta = 1°$. The latter means that if one wants to maintain a high accurate model in urban areas, it is better to choose small values of β, mainly where the model starts to converge. The complexity related to this choice will be highlighted in Section 6.2. According to Figure 17, in suburban areas, we see that the convergence is faster than in urban areas. Indeed, the model starts to converge around $\beta = 6°$. This means that medium values of β can be used to launch rays in suburban areas while still maintaining the model's high level of accuracy. Finally, from Figure 18, we see that the variation of the average value of the bitrate in rural areas is quite negligible, since the model converges around $\beta = 11°$. Moreover, the difference of bitrate between the highest value of β and where it starts to converge is around 2 Mbps. This is to say that the model remains accurate despite the value of β chosen in the range we consider. We can thus conclude that in environments with small number of obstacles, β has almost no effect on the model's estimation accuracy.

In summary, urban areas are more sensitive to the overestimation incurred by large values of β due to the large number of obstacles present in these environments. Moreover, the model's accuracy also suffers from this phenomenon. To have an accurate Ray Tracing download bitrate estimation in urban areas, it is better to choose

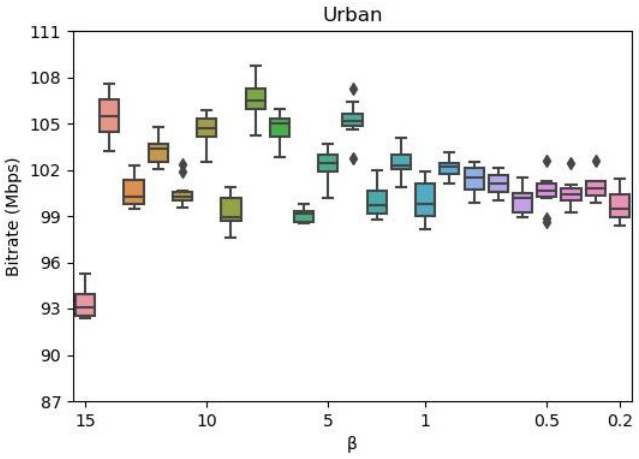

Figure 16: Average bitrate values in Terrain 1

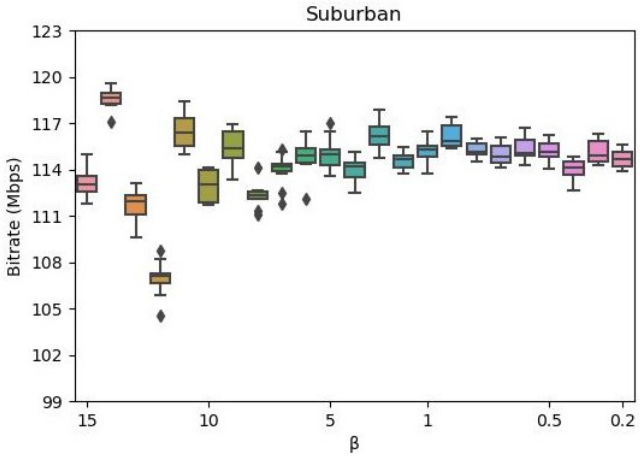

Figure 17: Average bitrate values in Terrain 4

small values of β. However, suburban, and mostly rural areas are less sensitive to this phenomenon. Thus, higher values of β can be chosen, i.e., a smaller number of rays can be launched in those environments with just a little effect on the model's overall accuracy.

6.2 Complexity

We study in this section how the memory consumption and the execution time varies depending on the value of β and the type of terrain. We performed these simulations on a Linux server having an Intel(R) Xeon(R) CPU @ 2.6 GHz with 32 processors and 94GB memory.

Figures 19, 20 and 21 show the memory consumption and the execution time needed to compute the bitrate for all the receiving points as a function of β in urban, suburban, and rural areas respectively. From these figures we can see that the memory consumption and the execution time have almost the same distribution and they decrease as the value of β increases. This behavior is normal since with large values of β, the number of launched rays is small leading to less computational complexity, which is the opposite to what

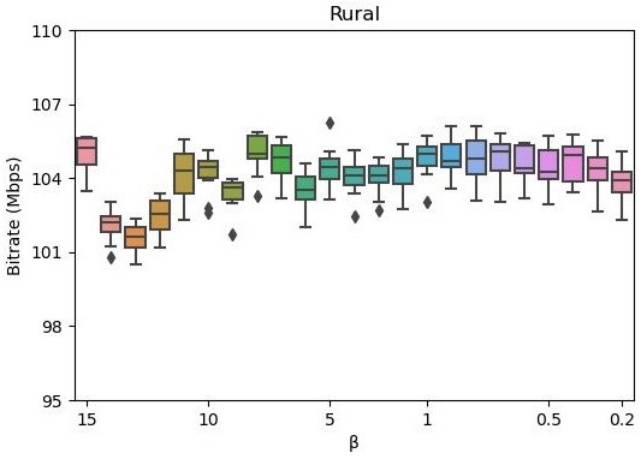

Figure 18: Average bitrate values in Terrain 6

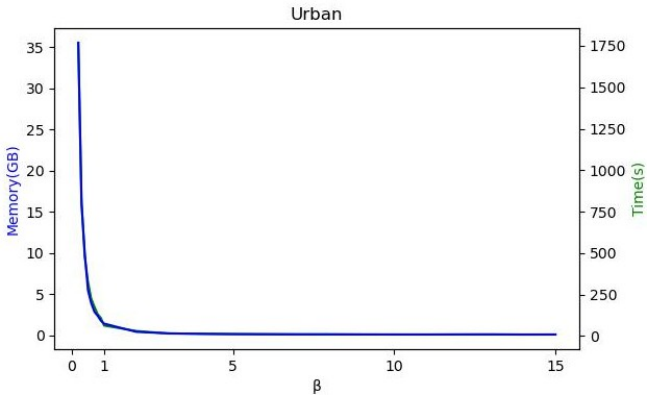

Figure 19: Complexity in Terrain 1

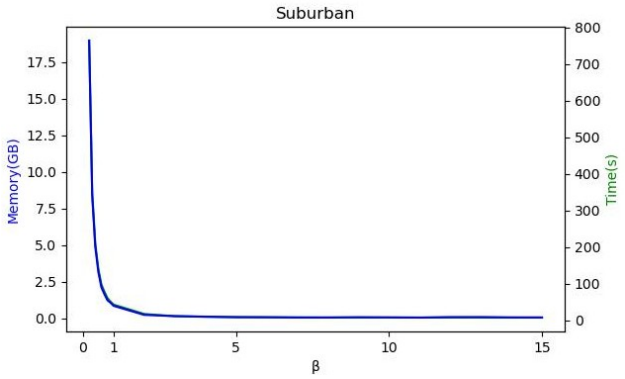

Figure 20: Complexity in Terrain 4

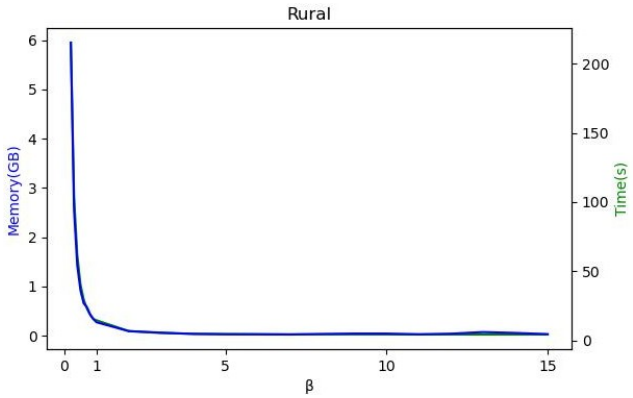

Figure 21: Complexity in Terrain 6

happens with smaller values of β where a large number of rays is generated.

Urban areas contain a lot of antennas, this means that the number of rays generated will be high as all the antennas are launching rays at the same time. This large number of rays leads to high computational complexity as opposed to sub-urban and rural areas and as can be seen in Figure 19. From this figure, we see that more than 35GB of memory is consumed with an execution time around 30 minutes for the smallest value of β considered. While in suburban areas, less antennas and buildings are present leading to a lower complexity. From Figure 20, the smallest value of β consumes more than 18GB of memory with an execution time around 13 minutes. However, in rural areas, resource consumption is way less as opposed to the previous ones due to the small number of antennas present in the considered areas. According to Figure 21, the memory consumption with the lowest value of β is less than 6GB with an execution time around 5 minutes. From all these figures, we can see that for $\beta > 1$, the memory consumption and the execution time drop drastically as compared to the case of $\beta < 1$.

6.3 Accuracy vs. Complexity Trade-off

Our study on both complexity and accuracy of our RT solution helps to choose the right number of rays to generate depending on the level of accuracy sought and the number of resources available. This allows to tackle the accuracy/complexity trade-off that is crucial in Ray Tracing. In practice, a user with unlimited resources willing to run Ray Tracing with high accuracy can go with small values of β regardless the terrain. Another one with limited resources willing to have an acceptable accuracy can go with any values of β in rural areas, and medium values of β in suburban and urban areas. For sub-urban areas this does not impact accuracy, however for urban areas, one must accept to lose a bit in accuracy to adapt to the constraint on computing resources.

7 CONCLUSION

In this paper we presented a new ray generation technique. Being site specific, our technique generates the minimum number of rays necessary to fully cover the area of interest without any gaps. Although less rays are launched compared to the state-of-the-art approach, we prove that our technique still maintains the accuracy of Ray Tracing. Since we launch less rays, we reduce the high computational load and the high memory consumption of Ray Tracing. Moreover, we solve for the computational slowness of

the state-of-the-art technique by the ability of our new technique to generate thousands of rays in few seconds. We also introduce a practical trade-off between accuracy and complexity, allowing users to select the appropriate number of rays based on the terrain type and available resources. In the future we plan to pursue the development and validation of our technique towards its integration in an operational tool for the cartography and management of signal power and bitrate in the cellular networks of mobile network operators across a large area.

REFERENCES

[1] Autorité de régulation des communications électroniques des postes et de la distribution de la presse. 2022. Le marché du haut et très haut débit fixe (déploiements). https://www.data.gouv.fr/fr/datasets/le-marche-du-haut-et-tres-haut-debit-fixe-deploiements/ Last accessed 02 February 2022.

[2] Olaonipekun Oluwafemi Erunkulu, Adamu Murtala Zungeru, Caspar K. Lebekwe, and Joseph M. Chuma. 2020. Cellular Communications Coverage Prediction Techniques: A Survey and Comparison. *IEEE Access* 8 (2020), 113052–113077. https://doi.org/10.1109/ACCESS.2020.3003247

[3] Stephen Kasdorf, Blake Troksa, Cam Key, Jake Harmon, and Branislav M. Notaroš. 2021. Advancing Accuracy of Shooting and Bouncing Rays Method for Ray-Tracing Propagation Modeling Based on Novel Approaches to Ray Cone Angle Calculation. *IEEE Transactions on Antennas and Propagation* 69, 8 (2021), 4808–4815. https://doi.org/10.1109/TAP.2021.3060051

[4] C. G. Liu, E. T. Zhang, Z. P. Wu, and B. Zhang. 2013. Modelling radio wave propagation in tunnels with ray-tracing method. In *2013 7th European Conference on Antennas and Propagation (EuCAP)*. 2317–2321.

[5] A. Maltsev, R. Maslennikov, A. Sevastyanov, A. Lomayev, and A. Khoryaev. 2010. Statistical channel model for 60 GHz WLAN systems in conference room environment. In *Proceedings of the Fourth European Conference on Antennas and Propagation*. 1–5.

[6] Mathworks. 2022. Matlab Ray Tracing. https://fr.mathworks.com/help/comm/ref/rfprop.raytracing.html Last accessed 13 June 2022.

[7] Jan Reitz Moritz Alfrink and Jürgen Roßmann. 2021. Improving Ray Tracing Based Radio Propagation Model Performance Using Spatial Acceleration Structures. In *Proceedings of the 17th ACM International Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet '21)*. Alicante, Spain. ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3479242.3487318

[8] Institut national de la statistique et des études économiques. 2022. La grille communale de densité à 4 niveaux. https://www.insee.fr/fr/information/2114627 Last accessed 02 February 2022.

[9] Institut national de l'information géographique et forestière. 2021. Base de données Topologique. https://geoservices.ign.fr/documentation/donnees/vecteur/bdtopo Last accessed 01 February 2022.

[10] Institut national de l'information géographique et forestière. 2021. Calcul altimétrique (REST). https://geoservices.ign.fr/documentation/services/api-et-services-ogc/calcul-altimetrique-rest Last accessed 02 February 2022.

[11] Bernard Tamba Sandouno, Yamen Alsaba, Chadi Barakat, Walid Dabbous, and Thierry Turletti. 2022. Site-Specific Ray Generation for Accurate Estimation of Signal Power. In *Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems* (Montreal, Quebec, Canada) *(MSWiM '22)*. Association for Computing Machinery, New York, NY, USA, 39–46. https://doi.org/10.1145/3551659.3559058

[12] Sayer, Lawrence G. 2020. Enhanced Radio Propagation Modelling for Future Wireless Networks. (2020), 65–69. http://research-information.bristol.ac.uk

[13] Purnima Sharma and R K Singh. 2010. Comparative Analysis of Propagation Path loss Models with Field Measured Data. *International Journal of Engineering Science and Technology* 2 (06 2010).

[14] Wikipédia. 2021. Géode (géométrie) — Wikipédia, l'encyclopédie libre. http://fr.wikipedia.org/w/index.php?title=G%C3%A9ode_(g%C3%A9om%C3%A9trie)&oldid=188739072 Last accessed 22 February 2022.

[15] Zhengqing Yun, M.F. Iskander, and Zhijun Zhang. 2001. Development of a new shooting-and-bouncing ray (SBR) tracing method that avoids ray double counting. In *IEEE Antennas and Propagation Society International Symposium. 2001 Digest. Held in conjunction with: USNC/URSI National Radio Science Meeting (Cat. No.01CH37229)*, Vol. 1. 464–467 vol.1. https://doi.org/10.1109/APS.2001.958892

[16] Zhengqing Yun, M.F. Iskander, and Zhijun Zhang. May 2000. Fast ray tracing procedure using space division with uniform rectangular grid. *Electron. Lett* 36, 10 (May 2000), 895–897.

[17] Zhengqing Yun and Magdy F. Iskander. 2015. Ray Tracing for Radio Propagation Modeling: Principles and Applications. *IEEE Access* 3 (2015), 1089–1100. https://doi.org/10.1109/ACCESS.2015.2453991