Site-specific ray generation for accurate estimation of signal power

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

Site-specific ray generation for accurate estimation of signal power

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 modelling approach that accurately estimates the signal power received by end users while taking into account the details of the environment in their vicinity. This accuracy is at the cost of high computational load and high memory consumption due to the heavy computation performed by processes such as 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 and the coverage area, our technique gives the minimum possible number of rays required in order to 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.

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 coverage map. However, empirical models are usually not very accurate because they finally depend on the environment where they were originally devised [8].

Deterministic models from their side use thorough details of the environment of propagation for path loss estimation. They take into account the complete 3D map and the characteristics of the environment. According to these models, waves' interaction with their environment is taken into account through reflection, refraction, scattering and diffraction, making them to be the most accurate ones among all the propagation models. However, this high precision is at the cost of a high memory consumption and computational load. Although very accurate, these models have the problem of not being practical for generating coverage maps in complex environments and are useless in real time scenarios [1][6][12]. Despite their slowness, deterministic models are still required when there is a need to accurately estimate the signal power received by end users [8]. 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. Our main objective is to reduce their complexity without compromising their accuracy.

Ray Tracing (RT) is based on the light/wave duality [12], 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 the 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, generated rays are modeled as tubes or cones centered by a line (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 sphere at the cross-section of the cones must be

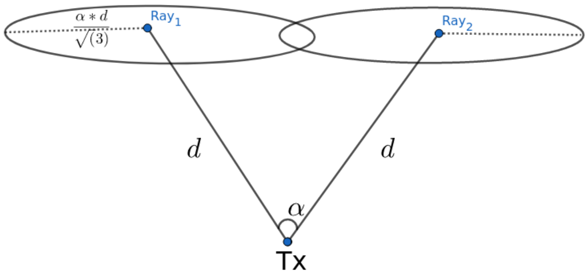

Figure 1: Ray cones overlap to avoid blank area in 2D

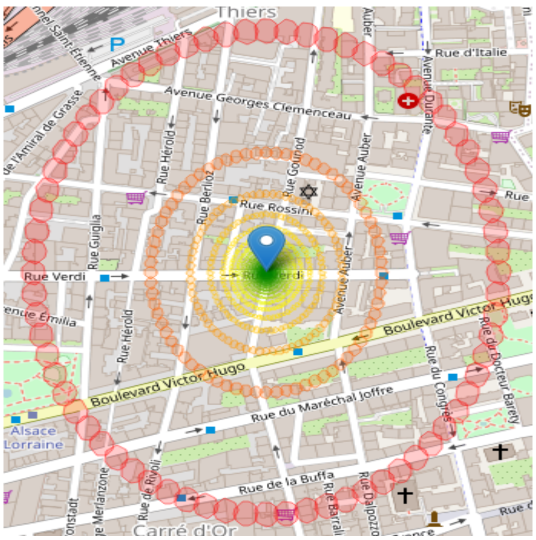

Figure 2: Example of ray cones launched with a constant angular separation α **=5°**

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 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 travelling the distance d_i is therefore given by (1) [2], with α_i being the maximum angular separation of the i^{th} ray with its neighbors.

$$R_i = \frac{\alpha_i d_i}{\sqrt{3}}. \tag{1}$$

Choosing the same constant value for the separation angle α works perfectly in 2D scenarios. However in 3D, this leads to inaccuracies and gaps in the propagation area [2] as shown in Figure 2. This problem is solved in the literature by using the icosahedron technique, which consists in subdividing the faces of the icosahedron into multiple uniform equilateral triangles. However, this technique is computationally slow and launches many rays that are wasted since with this technique, rays are launched in all possible directions regardless the position of the receiving antenna. The smaller the area of interest, the higher the number of wasted rays are, 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 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 rays that will never reach the receiver. Moreover, our technique overcomes the complexity and the computational slowness of the icosahedron technique. On a laptop of 16GB memory with 7 processors at 1.8GHz, our technique was able to launch up to one million rays within 5 seconds, hence reducing the overhead due to the ray generation process. Simulations were then performed with this new technique in different scenarios and a validation with respect to the state of the art model implementing the original icosahedron technique was carried out. Through those simulations, we could see that our solution for ray generation is flexible, robust and computationally fast at almost no cost.

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 the technique as well as its performance evaluation are presented in Sections 4 and 5. Finally, the conclusion of our work is presented in Section 6 with perspectives on our future research.

2 RELATED WORK

Due to the complexity of 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 [12][11]. This is because a naive Ray Tracer performs an intersection test between all the rays and all the buildings. The previous techniques help run the intersection test only on buildings that can be potentially hit by the rays, hence reducing RT complexity. BVH has been further improved by the use of spatial acceleration structures in order to ease the ray object intersection test [6]. Another technique used is based on the use of efficient Graphical Processing Unit (GPU). RT execution time is then drastically reduced by the hardware acceleration.

Some other techniques are used to by-pass the overhead incurred by the icosahedron technique. First, Matlab in their Ray Tracing implementation computes the direction and the maximum angular separation of each of the vertices of the geodesic structure obtained from the icosahedron offline. Three fixed tessellation frequencies are chosen, the computation of the vertices' coordinates and their maximum angular separation are computed offline and those values are then simply loaded in the memory [5]. Nevertheless, this approach lacks of flexibility because the number of rays launched is limited to those three choices making it impossible for Matlab users to adapt the number of rays to be launched depending on the scenario they have. Moreover, as it is using the icosahedron technique, many rays can be useless since rays are launched in all possible directions.

Further, the authors in [7] propose a technique based on a "golden spiral". With such an approach, it is possible to evenly generate points on a sphere in order to launch rays passing by them. This technique 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 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 only generates 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 generates more rays for an area of interest of 5000 meters radius and less for an area of interest of 500 meters. For a given scenario, our method can thus produce a range of number of rays to be launched. It is proved in this paper that within that range, one can choose any number of rays at almost no 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.

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 icosahedron into many other equilateral triangles. The subdivision of the faces is performed as follows [9]:

- 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

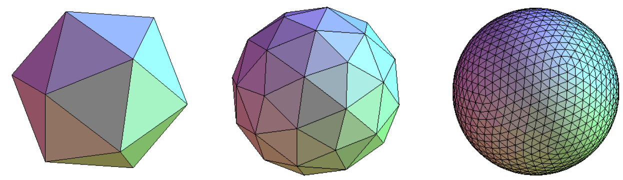

Figure 3: The first one is the original icosahedron (20 faces). The second and the third ones are the geodesic structures with tessellation frequencies $n = 2$ and $n = 10$ respectively.

- 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.

Figure 3 shows the original icosahedron and the geodesic structures obtained from it. All the $10n(n+1)$ [2][9] rays are then finally launched on the vertices of the obtained sphere. This technique has the advantage to fully cover 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 [7].

On the other hand, our ray generation technique is based on an iterative and adaptive approach. 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.

Proceedings 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).

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

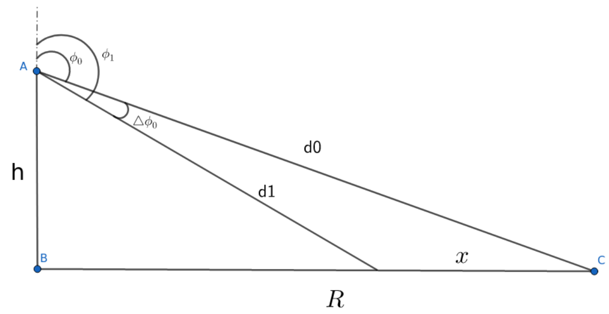

Figure 4: Illustration of our ray generation procedure

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

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

In order to avoid any gap between rays, we show by binary search that the maximum value of k which minimizes the overlap between rays while avoiding any gap is $k = 2\sqrt{3}$. Also, the value of α_i depends on the angular distance in elevation or in azimuth, i.e., $\alpha_i = max(\triangle\phi_{i-1}, \triangle\theta_i)$. 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.

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

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

Our method is site specific, i.e., we take the local information in order to generate the optimal number of rays in that specific scenario. Said differently, 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. 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. Since the azimuthal angular separation β is known in advance, the algorithm returns the elevations of the rays. Further, rays are launched using their Cartesian coordinates β and ϕ.

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.

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

$k \leftarrow 2\sqrt{3}$
$\triangle\phi \leftarrow 0$
$\phi \leftarrow \pi - \arctan(R/h)$
$Elevation \leftarrow []$
while $\phi \le \pi$ **do**
  $Elevation.add(\phi)$
  $\triangle\theta \leftarrow \cos^{-1}\left[(cos\beta - 1)\sin(\phi)^2 + 1\right]$
  $\alpha \leftarrow max(\triangle\theta, \triangle\phi)$
  $\triangle\phi \leftarrow \frac{2*\alpha}{(\alpha-k)*\tan\phi}$
  $\phi \leftarrow \phi + \triangle\phi$
end while

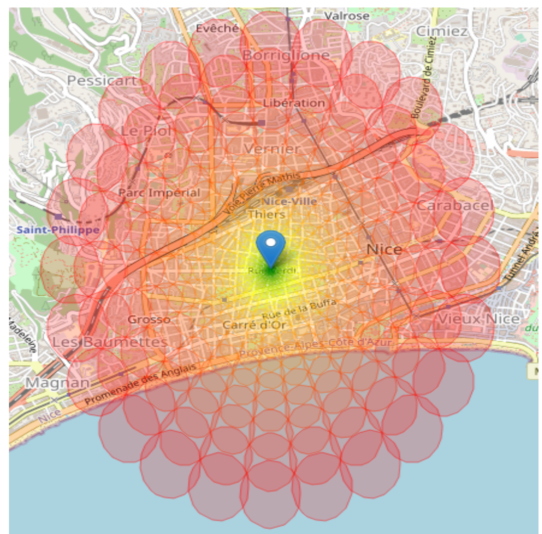

Figure 5: Our ray generation technique without any gaps and with minimum overlap between rays

Once rays are launched, subsequent RT processes are performed. An intersection test is performed between the rays and the buildings in order to determine if a ray hits a building and whether a ray 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 [10]. The signal power carried by received rays is computed using (6) [3][4].

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

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 distance crossed by the m^{th} ray until reaching 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.

The process explained above for one antenna can be generalized to multiple antennas. In this case, our technique is used to generate rays for each individual antenna and the power corresponding to each antenna is computed using (6). The SINR (Signal to Interference plus Noise Ratio) can then computed using the received powers from the different antennas within the same frequency band. The antenna with the highest signal power is considered as the source of signal and the others of same frequency are considered as sources of interference. Finally the Shannon capacity formula can be used to compute the download bitrate of the receiver.

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 in 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 starting with a $\phi_0 \geq \pi/2$ (below the horizon), 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. Hence, the number of rays launched in our case is set to $\approx 0.47*$Matlab. We fixed the radius of the coverage area to 5000m and the values for β (that determine the number of rays launched at each iteration) to 2.35°, 1.19° and 0.75° corresponding to the equivalent high, medium and low angular separation of Matlab.

We start by assessing the sensitivity of our technique regarding the number of rays launched by Matlab. We compare each of our cases to those 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.

Figure 6, 7 and 8 show a comparison between our high angular separation scenario and all the 3 other scenarios available in Matlab for different maximum number of reflections. These figures show that our signal power estimation has the same distribution as Matlab. 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 are able to accurately estimate the signal power received by end users. From the 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. Moreover, the mean error made by our technique regarding the one of Matlab increases slightly and is less than 3 Mbps in all cases. This remains

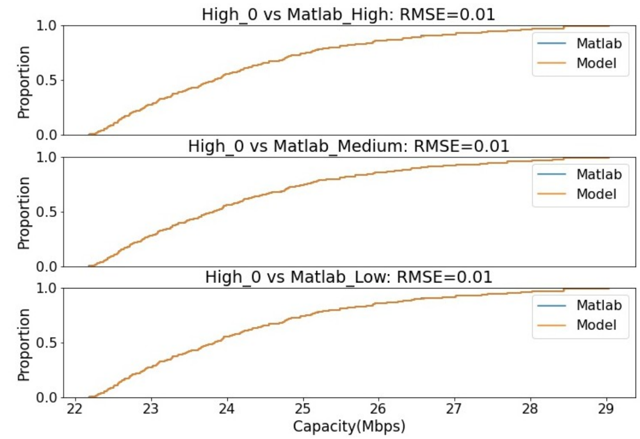

Figure 6: High angular separation in LOS

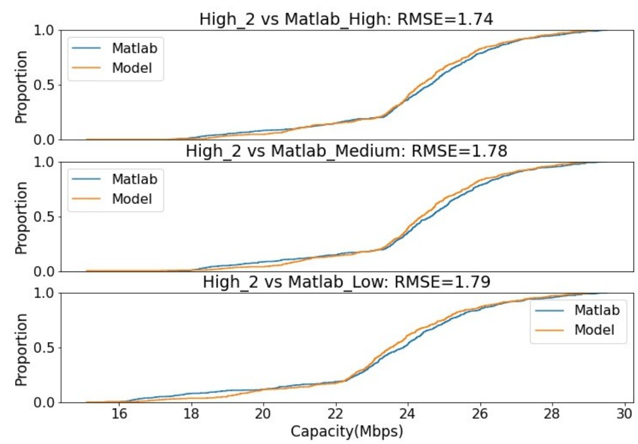

Figure 7: High angular separation with 2 maximum number of reflections

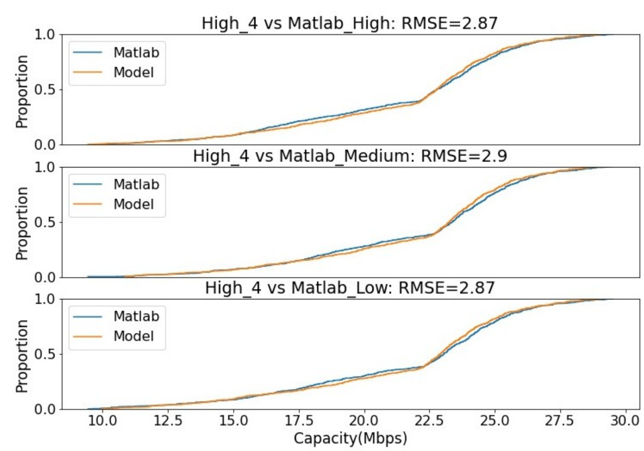

Figure 8: High angular separation with 4 maximum number of reflections

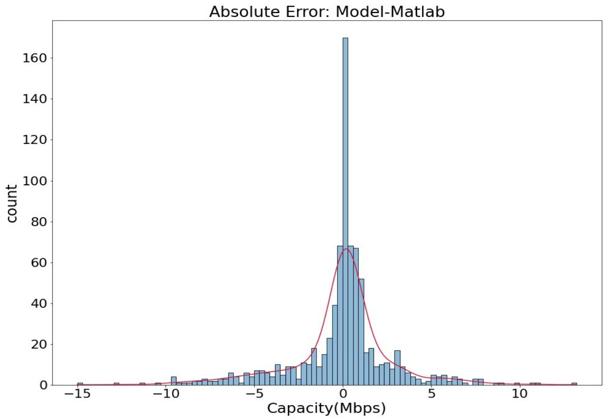

Figure 9: Mean Absolute Error distribution

Table 1: Results from 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

a good trade-off regarding the advantages offered by our technique as explained earlier. Note that all bitrate values were computed using the Shannon capacity formula for a bandwidth of 1MHz and a noise power of -107 dBm.

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 is a Gaussian 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 similar, only the plots for one terrain were shown in Figure 6, 7 and 8. However, we summarized 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, 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 Matlab ones. 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. Our method can be further validated by taking into account interference from other antennas by following the process explained in Section 3.

Table 2: Results from the 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: Results from 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 RESULTS

After explaining the correctness, robustness and the cost of our site-specific ray generation technique, in this section we dig deeper into the gain it offers. As a site-specific method, we aim at optimizing the number of rays launched in order to reduce the number of rays wasted and hence 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. 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. The y-axis determines the number of rays that are launched at each elevation represented in the figure by the radius of the coverage area d. The figure gives for a given scenario (100 meters radius for instance) 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. Second, the figure provides the minimum number of rays necessary to fully cover the propagation area without any gap. 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 in particular 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,

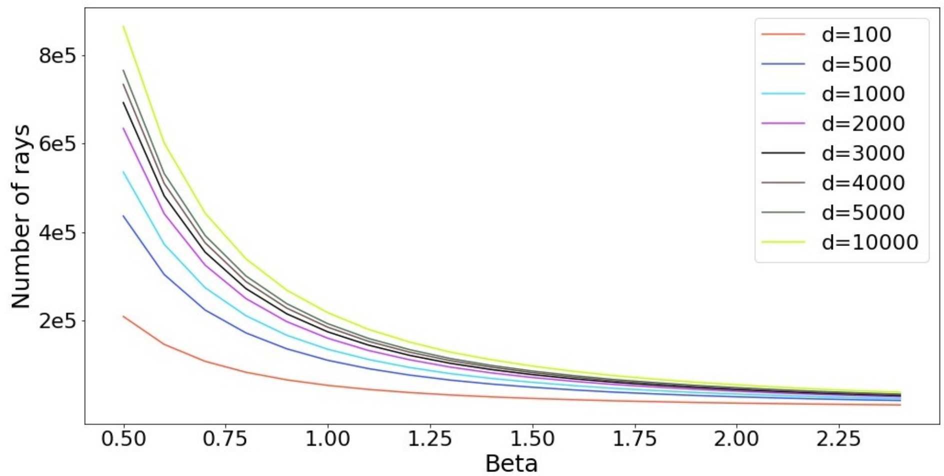

Figure 10: Number of rays launched in different scenarios

Table 4: RMSE 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

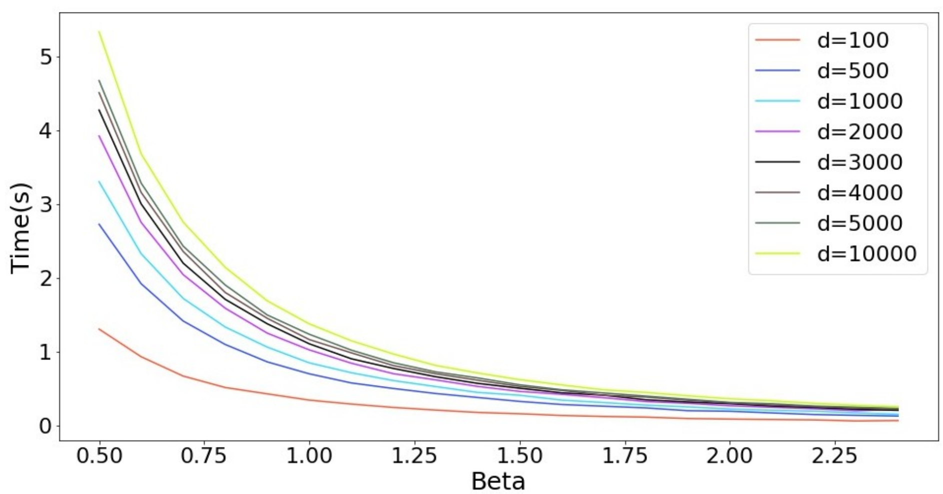

Figure 11: Time to generate rays in different scenarios

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 are able to 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 more or less 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 at 5000 meters and 30 meters respectively. Each of our simulation results was compared to Matlab. Since Matlab has only 3 possible angular separations, we compared our results to its *low* angular separation, because of its high accuracy compared to the 2 others. A range of

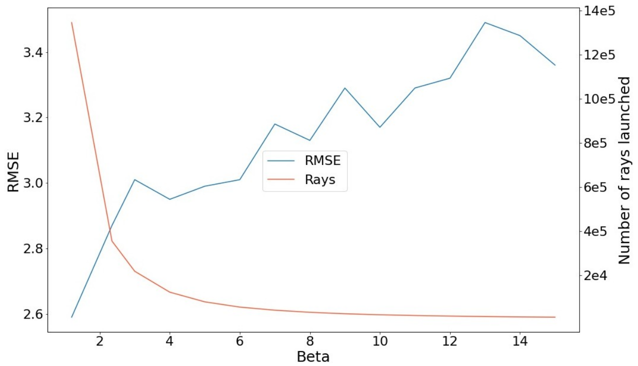

Figure 12: Sensitivity study on Terrain 3

β values were taken with 15° being the maximum, due to the small angle approximation performed on β.

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, our method can be parallelized and have almost the same execution time as shown in Figure 11 for multiple antennas than a single one.

6 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. In the future we plan to pursue the development and validation of our technique towards its integration in a operational tool for the cartography and management of signal power and bitrate in the cellular networks of mobile operators across a large area.

REFERENCES

[1] 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

[2] 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

[3] 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.

[4] 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.

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

[6] 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

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

[8] 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).

[9] 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.

[10] 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

[11] 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.

[12] 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