IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 3, 2020, accepted October 23, 2020, date of publication November 11, 2020,
date of current version November 25, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3037135

An Efficient Ray-Tracing Based Model Dedicated to Wireless Sensor Network Simulators for Smart Cities Environments

T. ALWAJEEH, P. COMBEAU, AND L. AVENEAU
XLIM Laboratory, UMR CNRS 7252, Poitiers University, 86073 Poitiers, France

Corresponding author: P. Combeau (pierre.combeau@univ-poitiers.fr)

This work was supported by the French National Research Agency, through the project titled 3D virtual platform for wireless sensor network simulation [PlateformE viRtuelle 3D pour la Simulation des rEseaux de caPTEURs (PERSEPTEUR)], under Grant ANR-14-CE24-0017-03.

ABSTRACT This article presents an efficient ray-tracing site-specific channel model compatible with wireless sensor network (WSN) simulator applications for smart cities. It relies on two 2D approaches dedicated to the main propagation modes; i.e. horizontal and vertical. The first is based on a pre-processing of the propagation environment, involving the calculation of an exact visibility graph. Computed from the discrete geometry concept called the super-cover model, it is improved to reduce its size and is used to very efficiently compute the most significant propagation paths in 3D. The specific WSN context, characterized by the sensor's radio range and mobility, is exploited to limit the size of the propagation environment and to pre-compute and store a set of visibility graphs, which are finally loaded and used on demand. The second approach is an over-rooftop model that re-uses the super-cover model to very efficiently extract the vertical profile containing the sensors, and proposes an original solution for the electrical field prediction where there are multiple diffractions in the transition zone. The results are validated against measurements, and show markedly better performance compared to others recent ray-tracing models. Finally, integration of the proposed overall solution for channel modeling in a WSN simulator is proposed, and interest in using such a model compared to conventional statistical models is demonstrated.

INDEX TERMS Radio propagation, ray-tracing, wireless sensor network.

I. INTRODUCTION

It was estimated in 2014 that 54% of the world's population lived in urban areas; roughly 3.3 billion people. This proportion is expected to increase to 66% by 2030, around 5 billion people [1]. This massive increase has encouraged new avenues of research into approaches to manage cities and offer urban services in a different way. The increase in interconnected elements in cities' infrastructure due to new technologies has also reshaped the vision for managing cities [2]. This, in turn, has led to the concept of the ''Smart City''. In general, a smart city refers to new technologies and innovative ideas to improve both the quality and efficiency of urban services, and to improve people's lives.

Technically, smart city communications rely on wireless sensor networks (WSNs). To develop tomorrow's smart city, researchers have to study, develop, test and evaluate new WSNs. These performance tests can be conducted using either experimental test beds or simulations. Although test beds are more realistic and more reliable, they are fairly complex, time-consuming, costly, and may be practically unfeasible for deploying a large number of sensors [3]. Simulation is an appropriate alternative for studying network parameters before deployment, especially for large-scale WSNs, as it provides a cost-effective, rapidly deployable, and fairly reliable solution. A comparison between simulation tools and test beds is given in [4] as a reference for choosing between them. In fact, simulation is used by the majority of the research community [5].

WSN simulators handle dynamic virtual networks, to observe and evaluate their operation. They rely on radio channel models along with their related protocols and algorithms. One challenge is the accuracy and the computation speed of the radio channel model. A wide range of available WSN simulators is presented in the literature [5]–[7]. Unfortunately, most of them use very simple and unrealistic

The associate editor coordinating the review of this manuscript and approving it for publication was Matti Hämäläinen.

propagation models [8]–[19], allowing fast simulations even for large networks but lacking in precision. Some simulators incorporate more realistic site-specific models [20]. The most widely used radio channel models are based on ray-tracing (RT) combined with geometrical optics (GO) and uniform theory of diffraction (UTD). Unfortunately, they often become computationally expensive when the environment contains a large number of obstacles, or when a high order of ray interactions is considered. This applies particularly in the WSN context, with a high number of mobile sensors necessitating continuous updating of the channels estimation.

The most computationally efficient models use 2D pre-processing of the propagation environment [21], [22] to compute an accelerating structure. A radio propagation model based on such a structure relies on two steps. Firstly and from a given transmitter, this structure is built to encode the wave propagation with some reflexions and diffraction. The second step uses this structure to calculate all valid paths for given receivers. In other words, the complexity is split in two parts; the calculation of the structure, and the calculation of the paths. In previous studies, the authors have proposed a 2D accelerating structure, the visibility graph (VG) [23]. This is an accurate solution, since it calculates all ray contributions in the plane like any other exact 2D structure. Its fast computation and use in ray-path determinations have already established its high level of efficiency [24], [25].

In this article we propose a global solution (with horizontal and vertical propagation) of a RT site-specific channel model compatible with urban WSN simulator applications. It is based on an improved VG structure and new contributions addressing the specific context of WSN:

- A new VG computation is proposed that reduces its size, and so reduces both the memory consumption and the computation time for the final ray-paths.
- A very efficient algorithm is proposed to limit the ray-path computations to the most significant ones. This reduces the ray-path computation time while limiting the increase in the power estimation error.
- The radio range of the sensors, depending on their technology, is exploited to limit the size of the simulated environment, and thus the computation time.
- A new channel pre-process is proposed, based on the relative spatial stationarity of the channel. It consists of pre-computing and storing a set of VGs associated with a grid of virtual transmitters. These VGs are then loaded from the hard drive and used on demand.
- Finally, a new and efficient method is proposed for over-rooftop transmission (ORT). It relies on fast vertical profile extraction, and an original solution to avoid electrical field divergence when multiple diffractions occur in the transition region, which is often the case in dense urban environments.

This paper is organized as follows. Section II presents the state of the art in radio channel modeling. Section III discusses the computation of exact VGs, and presents our contribution to minimizing their size and thus the corresponding ray-path computation times. Section IV details our new contributions, addressing the specific context of WSN. Section V shows how to handle ORT modeling, with low computation times and divergency-free results. Section VI presents the overall performance results of the proposed model. Section VII discusses the channel model integration into the WSN simulator, and illustrates how it produces added value. Section VIII concludes this paper.

II. STATE OF THE ART

Of the different kinds of propagation channel models, only site-specific models can provide accurate field predictions in a particular environment. Such models require detailed data about the propagation environment [26], usually from a Geographic Information System (GIS). Basic RT models involve impracticable complexity even for low geometry and interaction numbers. Where n denotes the number of faces, and i the number of interactions (reflection plus diffraction), the computation time is in $O(n^i)$. Many models try to reduce this.

A. 3D RT MODELS

Some of these models can be parametric, and although they are based on site-specific geometry they need some measurements for calibration purposes [27], [28]. Others are fully deterministic and rely on exact or approximate numerical solutions of Maxwell's equations. The full wave solutions are obviously not usable in WSN simulators due to the need to sample the propagation environment at a sub-wavelength order, leading to prohibitive computation costs. Therefore ray-tracing models based on the geometrical optics (GO) concept and its extensions appear to be better solutions [29], [30]. Many RT models proposed in the literature have been proved accurate. By considering all the main physical phenomena, namely line of sight, reflection, diffraction and their combinations, it is possible to simulate complex three dimensional (3D) rays. However, 3D RT can drastically slow down multi-path determination in a realistic city environment. Some optimization methods have been developed to keep computation times down to an acceptable level, such as AZ buffer [31], space-division methods [32]–[42], visibility pre-processing [43], [44], GPU implementation [45] and machine learning approach [22]. The calculation times for these methods are often not given. When they are available, they show either that these methods can only deal with very simple environments (an interior scene with 4 rooms [46] or an exterior scene made up of 64 buildings [47]), or that they are based on a non-exact and costly visibility pre-process [43], which would miss some significant contributions. In the context of WSN smart city simulators, the potentially high number of sensors and their dynamic behavior imply the need for continuous updating of the calculated channel estimate in very complex environments. Therefore, 3D RTs are not efficient enough.

Other approaches convert the three-dimensional problem into a two-dimensional (2D) hybrid one, depending on the distance between transmitting/receiving antennas, and their relative heights compared to the average height of buildings in the environment. Hence, 2D RT models fall into two categories depending on the main plane in which the rays are computed: *2D vertical propagation* and *2D horizontal propagation*.

B. 2D VERTICAL PROPAGATION MODEL

Reference [30] computes RT in a vertical plane by launching some rays in vertical half-planes to identify the reflection and diffraction points. When reflection occurs, a new vertical plane containing the reflected ray is used to find the next intersected building and so on until a stop criterion is met. Thus, 3D ray-paths can be computed from 2D vertical ray-launching. Good agreement with measurements is found in [48], but this algorithm needs many intersection tests to validate the ray-paths, and therefore still involves high computation times.

ORT models are efficient and accurate solutions when the transceivers are very far from each others. They only compute the 2D ray-paths propagating in the vertical plane containing the transceivers. They are based first on the extraction of geometrical obstacles involved in the vertical plane, and then on the estimation of the electrical field propagating in this plane by successive diffractions from the horizontal edges of the roofs of buildings. Obviously, this drastically reduces the model complexity compared to basic RT models, since only one propagation path is computed.

Several solutions have been proposed to deal with the multiple diffraction problem. The multiple knife-edge diffraction method is a recursive approach for estimating the overall diffraction loss due to multiple diffraction. Using this method, the obstacles are represented by simple geometry as infinitely thin edges (knife-edges). The most widely used multiple knife-edges methods are those of [49], [50]. They give relatively good results when there are large differences in building heights, but lead to large errors with grazing incidence, which is common case in dense urban environments. In [51], Vogler proposed a general multiple integral solution, valid even at grazing incidence, but the computation cost increases exponentially with the number of edges [52]. The well-known Uniform Theory of Diffraction [53], although computationally efficient and valid in the transition region, still gives inaccurate results in the case of multiple diffractions in the transition zone. Some improvements involving the addition of a higher order diffracted field have been proposed by Holm [54], but they need to consider a very high order for a realistic number of edges (100 order for 10 edges), and so still imply huge computation times. As an alternative, Andersen and Rizk [55], [56] enforced the diffracted field continuity at the shadow boundary by adding a second order diffracted field with a modified distance parameter L in the definition of the UTD coefficient. This solution gives good results but for separate wedges only, not for two joint wedges as found in a conventional building shape, and their iterative behavior leads to significant computation times as the number of wedges increases. Finally, Capolino and Albani [57], [58] proposed a closed-form solution for high-frequency diffraction problems from a perfectly-conducting thick screen. It is efficiently computable, strictly continuous in the transition region and still valid when the thickness becomes vanishingly small. In this article, we use this coefficient jointly with the conventional UTD coefficient for simple diffraction problems to predict the electrical field for any realistic building roof shape, which is an advantage with regard to the continuous improvement of GIS data quality.

C. 2D AND 2.5D HORIZONTAL PROPAGATION MODELS

2D horizontal models are used when the most significant ray-paths propagate horizontally in the streets by interacting with building walls and vertical edges. The simulated results usually compare well with measurements [59]. 2.5D models are built on 2D ones, but using the correct elevations of the interaction points for field calculations. Although 2D and 2.5D models are more computationally efficient than 3D RT thanks to simpler computations and less geometry, they often use some accelerating structures based on pre-processing of the propagation environment.

The first VG was proposed by [60]. From the image theory, it iteratively builds illumination zones, i.e. lit regions from a given source's image. These illumination zones are not exact visibility zones, since they contain some blockers (building faces). Then this VG is applied using a two-step RT approach: first, all zones containing a given receiver are identified; second, obstacles present in these illumination zones are tested to validate propagation paths in the RT process. Clearly, this reduces the computation time compared to a brute force RT, because fewer contributions are built and checked to obtain valid outcomes. Nevertheless, numerous intersection tests are still needed. As with all VG methods, the global complexity may be expressed from the complexity of two successive steps, building and RT:

$$\text{Build}(n, i) + \text{RT}(n, i). \tag{1}$$

In [61], the illumination zones are more accurately computed: an image is not valid for the whole wall, but only for the part of the wall illuminated by the parent image. However, illumination zones are still not exact visibility zones.

In [62], an exact VG is computed from the association of image theory and a polar sweep algorithm, which is next reused in [39]. Another optimization is to use the bounding box of buildings to limit the number of walls and edges on which intersection tests are needed to validate ray-path segments in the RT process. Unfortunately, the computation time is not given, and both its computational complexity and the fact that this VG is emitter dependent (as in [61]) suggest that it is not suitable for urban WSN simulators.

More recently, the illumination zone concept was reused by [63]. Each illumination zone, called a lit visibility polygon, is associated with a dark visibility polygon corresponding to

the part of the zone shadowed by one blocker (building). A ray segment is validated if the receiver is both inside the lit polygon and outside all the dark ones at a given depth of the graph. This operation is repeated for the lit polygon of the previous order image and so on to validate a propagation path. The dark polygons lead to numerous intersection tests for each ray-path, and thus high RT computation times. Moreover, the proposed pre-process is very time consuming, as shown in [66]. In [64], the solution was optimized for the treatment of several receivers, by mapping in pre-process the lit polygons to a coarse grid to limit the number of intersection tests needed in RT. In [65], the use of the polar sweep algorithm [62] is added to compute an intra visibility matrix in pre-process. It stores the visibility relationship between each pair of geometrical elements (faces or edges), as the extreme angles for which some parts of the two elements are visible. Another pair of angles is computed to establish the lit polygon of an image with regard to a wall. Thus in pre-process, validation of wall visibility is obtained by comparing its visibility angles with the lit polygon ones. A 50% reduction in pre-process computation time is obtained against [64], but because the intra visibility matrix is still non-exact, costly intersection tests are still needed to validate the potential ray-paths in RT, so the computation times remain too high for WSN simulators. In [66], the intra visibility matrix is reused jointly with a visibility table containing the list of faces/edges visible from an emitter moving along a linear route. This allows significant acceleration of the pre-processing time compared to [65] in the case of a mobile emitter, but it still takes a lot of time. In [67], a new algorithm is proposed to provide a dynamic visibility table taking into account the modifications of the visibility relations between a mobile emitter and mobile obstacles. A computation time reduction of only 20% is achieved with regard to a brute-force RT-process. Hence, it remains too large for WSN simulators. In the rest of this article, the efficiency of the proposed solution will be compared to those of [65] and [66].

As stated in Section 1, an exact 2D accelerating structure called a visibility graph (VG) is proposed in [23]. Its main advantage compared to other visibility structures is the exact nature of the visibility relation, which leads to a very simple and efficient usage for a given receiver. Indeed, since the visibility zones are mapped to a discrete grid, all those containing a given receiver are extracted linearly, in a very short time. Not only does this lead to far fewer zones; in addition it doesn't require checking the existence of ray-paths since each zone is exact. Furthermore, the building of this structure is optimized in many ways, such that even the pre-processing is relatively fast. In order to use this structure in WSN simulators for 2D and 2.5D horizontal propagation, some new specific optimizations are proposed in this article, allowing a reduction in the complexity of the two steps.

More precisely, the solution proposed here to establish a channel model suited to urban WSN applications is composed of two components:

- A 2.5D RT model associated with a 2D exact VG. Its efficient computation is presented in Section III, and the optimizations dedicated to the WSN context are detailed in Section IV.
- An ORT model dedicated to distant transceivers combining a very efficient vertical plane extraction and an accurate field prediction, which is detailed in Section V.

III. 2D EXACT VISIBILITY GRAPH

The visibility graph proposed here is the main acceleration structure used for the 2.5D model. A graph contains some elements called nodes, linked one by one by an arc providing a kind of layered structure.

A new VG implementation was written compared to [23], and a new optimization was added to minimize the number of nodes. Furthermore, some explanations are missing in [23]. This is why we detail the VG computation in this section before presenting its validation and performance evaluation.

A. VG COMPUTATION

In contrast to [60], [63], the VG used in this article is exact. There is no dark zone inside the illumination zones; i.e. the initial illumination zones are cut according to the blockers located inside. In this way, every point located in a zone is really visible from the transmitter (after reflections/diffractions). Thus, the VG exploitation is very efficient, because no intersection test is needed for ray-path validation, which drastically reduces the RT process. To ensure efficient zone use, a zone's outline is composed of only 3 points, i.e. a triangle for line-of-sight (LOS) and diffraction, or 4 points, i.e. a quadrangle for reflection. All zones are convex.

As noted in Section II, a VG has a layered structure. Its levels are built iteratively, according to the electromagnetic interactions that occur. Its root node corresponds to the transmitter; nodes at the first level of the VG represent the visible zones; and nodes at the next levels are zones lit after some reflections and/or diffractions from the geometrical elements (walls or vertical edges). The VG computation principle is presented by considering only reflection at first. The treatment specific to diffraction is presented below. Fig.1(b) illustrates the computation of an initial illumination zone by the image method for a first order reflection on the face F_1 of building B_1, from the image $T_{x'}$ of the emitter T_x. Buildings B_2 and B_3 are two blockers that must be taken into account to obtain the exact reflection zone in Fig.1(d). The remaining problem involves efficiently cutting any illumination zone according to the blockers' shapes. We call this process *zone reduction*.

The corresponding treatment is based on a discrete geometry concept, called the super-cover model, which discretizes a line segment as a set of pixels. Fig.1(a) illustrates this principle for the line D of equation $ax + by + c = 0$, where $(a, b, c) \in \mathbb{R}$ are the line coefficients. The super-cover of D

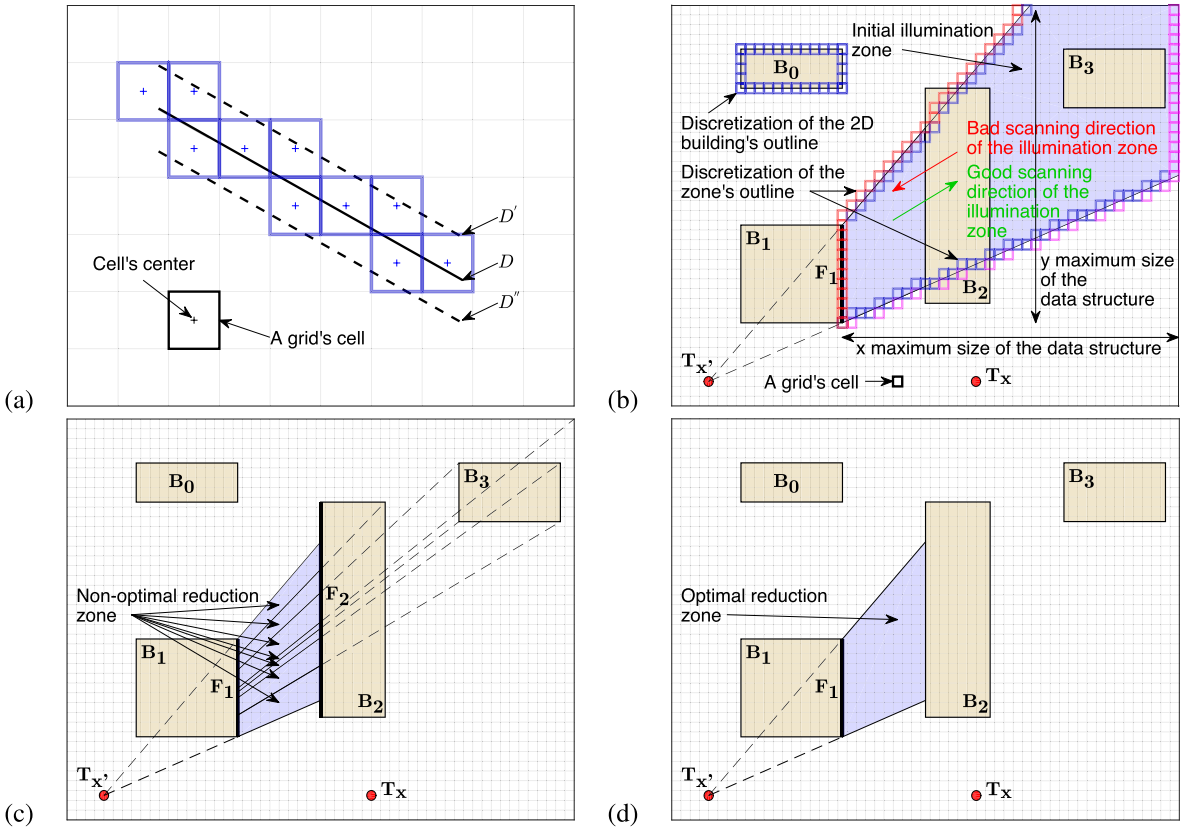

FIGURE 1. (a) Line super-cover, (b) buildings mapping and *zone reduction* principle, (c) non-optimal and (d) optimal *zone reduction*.

corresponds to the pixels whose centers are between the two lines D' and D'' (see Fig.1(a)) defined by:

$$D' = a(x + \frac{1}{2}) + b(y + \frac{1}{2}) + c = 0 \quad (2)$$

$$D'' = a(x - \frac{1}{2}) + b(y - \frac{1}{2}) + c = 0. \quad (3)$$

For example, the super-cover of D in Fig.1(a) is composed of all the blue pixels (x, y) given by the following double inequation:

$$-\frac{(\lfloor a \rfloor + \lfloor b \rfloor)}{2} \leq ax + by + c \leq \frac{(\lfloor a \rfloor + \lfloor b \rfloor)}{2}, \quad (4)$$

where $\lfloor \rfloor$ defines the integer part function.

In the first step, all the horizontal edges of the propagation environment are mapped, using the super-cover model, into a regular 2D grid, as illustrated in Fig.1(b) for building B_0. Hereafter, this first step is referred to as *scene2D mapping*. Thus, each grid's cell (i.e. pixel) stores the list of building edges located in this cell. This step is done only once, independently of any transceiver location. This mapping is very much more accurate than the mapping of the buildings' bounding box proposed in [62], which leads to map large empty areas.

When a new illumination zone is computed at a given VG level (i.e. for a given interaction order), it extends first from the source (previous image) of the zone to the edges of the scene's outline (cf. Fig.1(b)). This zone is thus too wide and contains some dark zones as in [60], [63]. To become exact, i.e. that all its points are visible from the source, it has to be cut according to the shape of the blockers. To do this, the proposed solution is based on determining all the edges in the environment that intersect the zone. As presented above, this is achieved by using a discrete geometrical approach. Here the problem is twofold: first, being able to provide an accurate zone's discretization in the most possible efficient way; and second, scanning the zone's discretization in the best direction, i.e. from the closest pixels to the most distant ones.

This last point is quite logical: by scanning the zone's discretization from the pixel farthest from the zone's source ($T_{x'}$ in the example given in Fig.1(b)), we will probably find a first edge that in fact is shadowed by many others. In this case the zone will be cut in a non-optimal manner, leading to many zones as illustrated in Fig.1(c). Even if the number of resulting zones is not optimal, they remain exact. In contrast, scanning the zone's discretization from the closest pixel to the source towards the most distant one (see Fig.1(b)) increases the probability of first finding a visible edge and so minimizing the number of resulting zones (see Fig.1(d)). This second approach firstly avoids loss of time in dealing with shadowed parts of the illumination zone, and secondly minimizes the memory consumption for VG storage, since it creates fewer zones at the current VG level, leading to fewer *zone reduction* steps in the next VG levels. As can be seen in

Section III-B, this is important for the ray-path computations. The second solution is used in this work.

In our implementation, these two problems are solved by the two next elements.

Firstly, four potential directions are considered for the zone scanning:

- from the minimum to the maximum x-axis value;
- from the maximum to the minimum x-axis value;
- from the minimum to the maximum y-axis value;
- from the maximum to the minimum y-axis value.

Secondly, the zone's discretization is stored in a specific data structure. Because the illumination zone is convex, its discretization can easily be obtained from its outline, computed from the super-cover model. Fig.1(b) illustrates the super-cover of the zone's outline as colored pixels. This data structure is an array in which each cell contains two integer values. The semantics of these two values depends on the direction chosen for the scanning of the zone's discretization, as indicated previously. So *zone reduction* comes down to the following process:

1) computation of the best scanning direction of the zone;
2) computation of the super-cover of the zone's outline;
3) scanning of the zone's discretization in the best direction.

The third step simply consists of two nested loops (x and y-axis). It is not time consuming because a very small number (often 0) of edges is contained in each pixel. Each edge stored in the corresponding pixel is tested as a valid blocker. If it is a valid blocker, the zone is cut according to this blocker and will be potentially cut again, either by another blocker in the same pixel, or by a blocker in another pixel in the case of a non-optimal scanning direction.

Applied to Fig.1(b), the best scanning direction is from the minimum to the maximum y-axis, and from the minimum to the maximum x-axis. The starting pixel is the closest red one to the source. The data structure's size, containing the discretization of the zone's outline, is given by the extreme pixels on the x and y-axis respectively (in red and purple in Fig.1(b)). The discretization of the zone's outline corresponds to the colored pixels. *Zone reduction* will start as soon as a pixel containing a building edge corresponds to a pixel in the zone's discretization, i.e. pixels lying between the red and purple ones in Fig.1(b), by scanning the zone in the previously defined direction. Finally, the main parameter of *zone reduction* is the resolution of its grid. In [23], we found that any sizes between 3 and 10 m gave almost the best performances. Therefore, for the rest of this paper we consider a constant cell size of 5 m x 5 m.

As mentioned above, diffracted zone computations need specific attention. They are independent of the parent node because they are linked only to buildings' vertical edges. As in [24], diffraction sub-graphs are thus computed once, before the *VG computation*, according to all the diffraction edges of the propagation environment. We call this step *DG computation*. Starting from a diffraction edge (the sub-graph root), the first DG level corresponds to diffraction zones. The next levels are reflection zones, iteratively computed until the maximum number of considered reflections is reached. During *VG computation*, each diffraction sub-graph is stored only once; i.e. the first time that its diffraction edge is lit by a VG zone. When a diffraction on the same edge occurs elsewhere in the VG, the corresponding sub-graph is re-used. From a programming perspective, this is performed by only copying pointers on the already computed sub-graph. This technique allows optimization of the VG's size and memory consumption.

According to the grid resolution, a pixel can store several edges; even a full building in the case of a coarse grid. Consequently, when a zone has to be cutted according to the edges located in a given pixel, it is possible to first cut it with a bad edge, i.e. one that is shadowed by another edge located in the same pixel. This situation can arise independently of the scanning direction. Thus, the result of the *zone reduction* will be a set with too many (but exact) visible zones, as in Fig.1(c). To solve this problem, a new step is proposed in this article. It consists of merging the zones resulting from a non-optimal reduction. These zones are identified if they satisfy the following two conditions: they have the same source and they share the same cutting edge (face F_2 in Fig.1(c)). If this is the case, the two zones are easily merged using knowledge of their respective outlines.

B. MULTI-PATHS COMPUTATION

The ray-path computation simply consists of browsing the nodes of all the VG's levels to determine whether they include the receiver. However, complex environments and large numbers of interactions lead to a high number of zones (up to several tens of thousands), each having a very low probability of containing the receiver. Hence, performing an exhaustive inclusion test for every zones would become very time-consuming. Consequently, we define a more efficient method called *VG mapping*. First, we use a second 2D discrete grid into which the VG's list of zones is mapped. Thus, each pixel of the grid contains the list of zones that intersect this pixel. Finally, the pixel containing a given receiver is identified, and the inclusion test is reduced to the list of zones stored in this pixel; i.e. the zones with a very high probability of containing the receiver. This step ensures the high efficiency of this method.

Zone mapping can be done in a very accurate way by reusing the same concept of 2D discretization used to compute the VG (cf. Section III-A). A zone is stored in a given pixel p if one pixel of its discretization corresponds to p. The mapping is also very efficient because of the use of the simple data structure presented in Section III-A to store the discretization of the zones' outlines. The concept is illustrated in Fig.2, corresponding to a simple scene with 3 rectangular buildings (B_1 to B_3). It shows the zones obtained by considering a maximal order of interactions equal to one. A total of 23 zones is found (10 visible zones: V_1 to V_{10}; 9 reflected zones: R_1 to R_9; and 4 diffracted zones: D_1 to D_4; for the sake of simplicity, diffracted zones are only shown for the edge P).

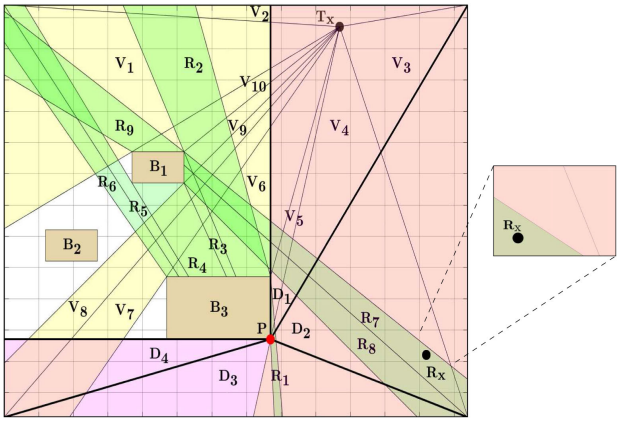

FIGURE 2. Extraction of the list of zones intersecting a pixel of the 2D grid. Here, for the given R_x location, only four zones will be tested: V_3, V_4, R_7, and D_2. Final inclusion test will remove V_3.

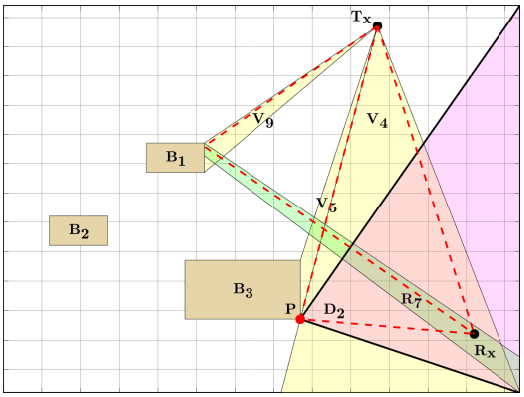

FIGURE 3. Resulting 2D paths.

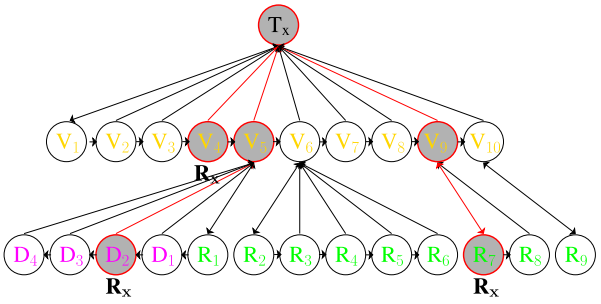

FIGURE 4. Extraction of the zones list intersecting a pixel of the 2D grid.

Let us consider the receiver R_x. Instead of conducting inclusion tests for 23 zones (for this simple configuration, with a limited number of interactions, and without considering all the diffracted zones), the test is done only for four zones, namely V_3, V_4, R_7, and D_2, to identify the ones that really contain the receiver, which are V_4, R_7, and D_2.

The mapping of the VG to a 2D grid is quite similar to the process used in [64]. However, the crucial difference is that in [64] the illumination zones are not cut according to the blockers. They are wider and contain dark zones, which are also mapped to the grid, increasing the number of inclusion tests needed. As a result, the grid mapping in [64] is much less efficient than ours: the increase in grid resolution leads to only a minor decrease of the computation cost due to the inclusion tests, because the dark zones are stored in many cells. In contrast, our VG contains exact visibility zones only. Since dense urban environments contain a very high number of blockers, the visibility zones are relatively small. Consequently, the use of a high grid resolution (small grid cells) is very efficient. Indeed, each cell stores a small number of zones, reducing the number of inclusion tests for a given receiver location. The increase in the grid's resolution obviously increases the computation cost due to the mapping process for the VG's zones, but it is kept very low thanks to the efficiency of the zone discretization algorithm. The grid's resolution is empirically fixed to 5 m x 5 m in the rest of this article.

Once this node identification process is completed, the nodes containing the receiver are known, as shown in Fig.3. Another process is required to go up through each identified branch from the node to the root (T_x) to compute the received paths (cf. Fig.4). As explained in Section III-A, each node stores all the necessary data to build the received paths (nature of node: reflection or diffraction, geometry: face or edge, zone boundary, zone source, etc.). This information, from the receiver to the transmitter and in the right order, is the key advantage of our VG structure. It allows an almost instantaneous ray-path computation. There is no need for intersection tests to validate the ray-path segments since the VG's construction ensures it contains only exact visible zones. Since in Fig.4 R_x is included in zones V3, R6, and D2, three paths (direct, reflected, and diffracted paths) are easily identified. These paths are built first in 2D using the source image technique, as shown by the red dotted lines in Fig.3. This is very fast because the order, the type of interactions and the objects that generated these interactions are all known. Then, from the knowledge of the transceivers and buildings heights, and from Fermat's principle of least time travel, the heights of the intermediate reflection and diffraction points along the path are adjusted to transform all the 2D paths into real 3D paths [25]. Finally, the proposed model is further improved by introducing the ground reflected ray because it has a great impact, especially in near-ground scenarios.

As a last step, the complex electrical field associated with each 3D path is computed as follows:

$$E_i = E_0 F_{T_x} F_{R_x} \prod_m R_m \prod_n A_n(s', s) D_n \frac{e^{-jkd}}{d}, \qquad (5)$$

where E_0 represents the reference field, F_{T_x} and F_{R_x} the transmitting and receiving antenna radiation patterns, R_m the reflection coefficient for the m^{th} reflector, D_n the diffraction coefficient for the n^{th} diffracting wedge, $A_n(s', s)$ the divergence factor of the diffracted rays, and e^{-jkd} the propagation phase factor due to the path length d ($k = 2\pi/\lambda$, with λ representing the wavelength).

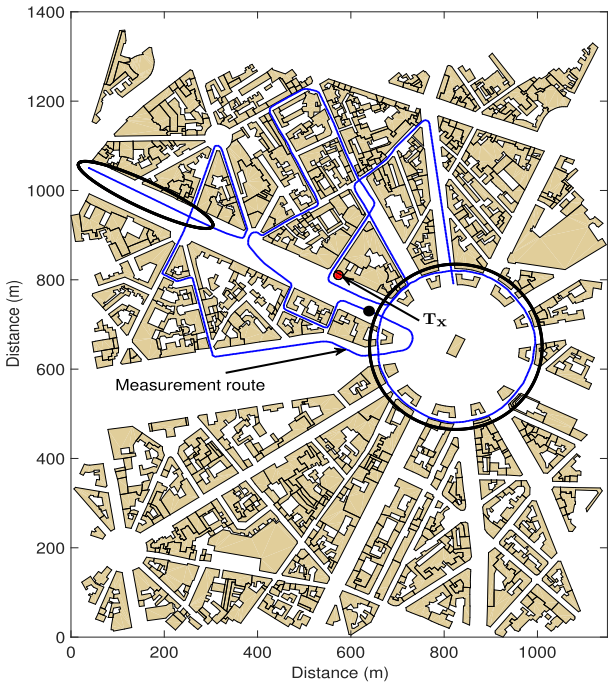

FIGURE 5. Measurement route in downtown Paris.

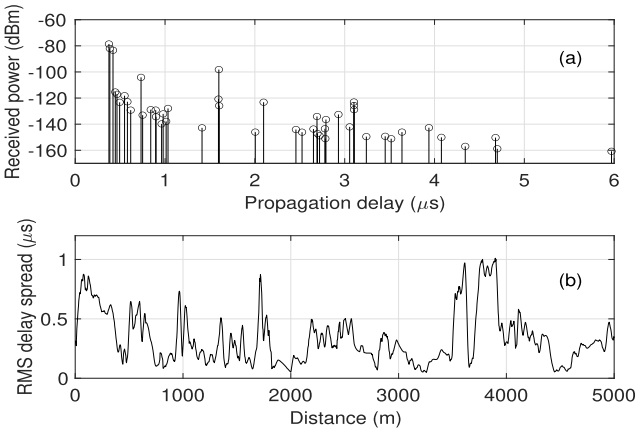

FIGURE 6. (a) Power delay profile example, (b) RMS delay spread evolution along the measurement route.

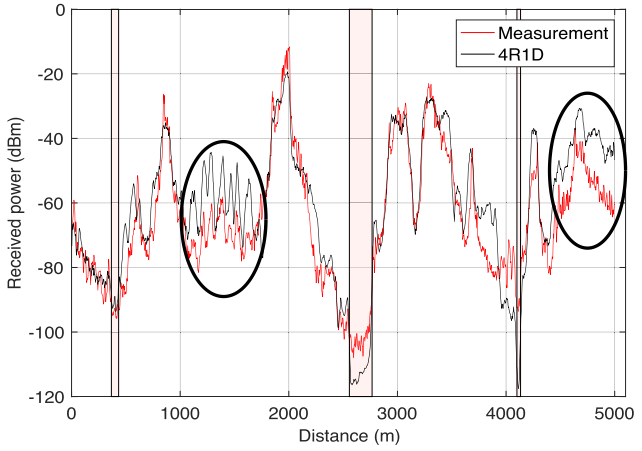

FIGURE 7. Simulation vs measurement.

C. PERFORMANCES

Model validation is based on field measurements that were conducted by France Telecom R&D in the *Charles de Gaulle - Étoile* neighborhood of downtown Paris (*cf.* Fig.5). The measurement route of 5 km (blue curve) corresponds to 1,650 equidistant reception points at a height of 1.5 m. The transmitter was a vertical dipole antenna located at a height of 7 m with a transmission power of 45 dBm at 1.8 GHz. This is a typical urban configuration with quite high buildings, where the dominant propagation paths are mainly in the horizontal plane. The geometry of the propagation scene, provided by the IGN (*Institut National Géographique*), includes 813 buildings (10,276 faces). The buildings are modeled as typical concrete blocks (relative permittivity $\epsilon_r = 9$, conductivity $\sigma = 0.01$ S/m), and the ground as a perfectly flat concrete surface.

As example, Fig.6(a) illustrates the power delay profile obtained at the receiver position depicted by a black point on Fig.5. This emphasizes the wideband character of our ray-tracing channel method. From this result, we can deduce all the wide-band parameters, like RMS delay spread whom the evolution along the measurement route is shown on Fig.6(b).

Fig.7 shows the received power obtained from the proposed 2.5D model (the black curve). It is parametrized for 4 reflections and 1 diffraction (4R1D), except for the pink parts of the route where a second diffraction is needed to reach the corresponding receiver locations. The 4R1D notation means that the VG is computed up to the fifth order of interactions, combining all the possible interaction arrangements from 0 (LOS) to 5 order, including a maximum of 4 reflections and 1 diffraction. In other words, the diffraction may occur in positions 1 to 5 or even be absent.

There is good agreement between the simulation and the measurements (the red curve), except for the pink parts where it is difficult to accurately predict the field strength, even with a second diffraction. The remaining errors, as circled in Fig.7, and corresponding to a stroll through the *Arc de Triomphe* and the portion of the route along the *Champs Élysée* (*cf.* black circled zones in Fig.5), can be justified by the vehicular traffic and the vegetation that were not considered in the GIS data. The mean error is 4.15 dB and the standard deviation of the error is 8.87 dB.

Table 1 presents the performances obtained on a PC running with a processor Intel CORE i7 3.1 GHz and 16 Go RAM. Notice that the results presented in this paper were all obtained on this computer. The computation time associated with the *scene2D mapping* plus the *DG computation* is 9 s, whereas the time for the *VG computation* plus the *VG mapping* (cf. Sections III-A and III-B) is 13 s. The VG computation time is almost the same as in [23]. The additional computation time due to the zone merging step is compensated for by the technological advances in processor performance. However, the number of resulting zones has decreased from

TABLE 1. Performances of the 2.5D model.

	Charles de Gaulle, 10,276 faces	*Munich part*, 1,734 faces
Scene2D mapping *+ DG computation*	9 s	1.7 s
VG computation + VG mapping	4R1D: 13 s, Nb zones: 98,888	5R0D: 0.12 s, Nb zones: 1,242
Path computation	1,650 R_{xs}: 7 s	1 R_x: 10 ms
Pre-process in [66]	-	5R0D: 3,458 s

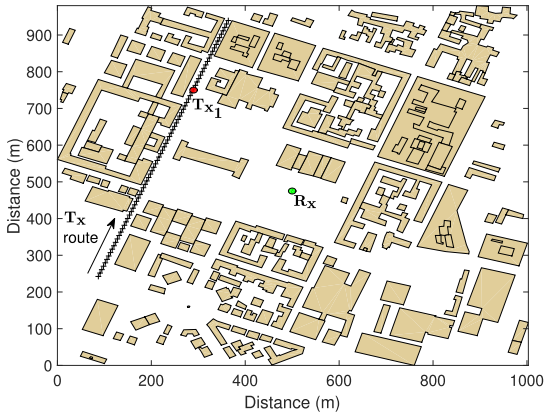

FIGURE 8. Environment for performances comparison with [66].

270,168 in [23] to 98,888 with the zone merging step. This reduction factor of about 2.7 leads to a significant time gain in the path computation step. Hence, the multi-path contributions between the emitter and all the 1,650 receivers of the route presented in Fig.5 are computed in 7 s using the VG described in Section III-B.

To further assess performance, we also present in Table 1 results obtained in a second environment, described in Fig.8, corresponding to a part of the city of Munich (Germany) used to evaluate performances of another recent published RT-method [66]. The environment's size is 1,000 m x 1,000 m and is composed of 1,734 vertical faces.

In [66], the polar sweep algorithm is used jointly with the intra visibility matrix to generate the visibility list for a fixed transmitter location, to accelerate the RT process. This pre-process is computed using a maximum reflection depth of 5, which corresponds in our formalism to 5R0D. The resulting computation time given in [66] is 262,800 s for 76 emitter locations, extrapolated to 3,458 s for a single emitter location. These results were obtained on a computer running with an Intel CORE i5 3.3 GHz. The difference between CPUs in our work and [66] should imply a constant but small scale factor in computation time. Since the visibility structure is not exact, the computation time of the RT-process must be added. Although it is not given in [66], it should be significant because of the need for numerous intersection tests to validate the potential ray-path segments.

As presented in Table 1, the computation times for the method proposed in this article is 1.7 s for the *Scene2D mapping* plus the *DG computation* and 0.12 s for the VG (*computation + mapping*), for a total of about 2 s. Finally, using the VG to compute ray-paths for one receiver location takes on average 10 ms per receiver.

Although these performances are relatively good, the next section introduces other processes dedicated to the specific context of WSN, which allow further improvements.

IV. OPTIMIZATIONS IN A WSN CONTEXT

This section proposes more new optimizations developed specifically to address the urban WSN context. Section IV-A presents an original VG application to very efficiently limit multi-path computations to the most significant ones. Section IV-B shows how we exploit the WSN sensors' characteristics to limit the size of the simulation environment in an efficient and accurate way. Finally, Section IV-C illustrates the proposed solution to take account of the WSN's sensors mobility.

A. LIMITATION TO THE MOST SIGNIFICANT PATHS

In many cases, a large number of propagation paths does not significantly contribute to the total field, but takes time to be computed. A simple optimization would involve only computing the most significant paths, without degrading the field prediction quality [68].

The VG nodes are roughly sorted from the most significant to the least. The leveled structure of the VG naturally leads to the nodes of level N being more significant than those of level N+1, since it has undergone one less interaction. Furthermore, since reflection is less penalizing than diffraction, the reflected zones are always placed before the diffracted ones for the same VG level, so the reflected paths are computed before the diffracted ones. Therefore, optimization simply consists of calculating the paths by increasing levels of interaction (the VG's depth), from the VG's root, until the desired number of paths is reached. Table 2 shows the results obtained for the same environment as Fig.5, always using the 1,650 receivers composing the measurement route. The absolute mean error is computed by taking the computation of all the paths as the reference. It shows that the path computation becomes very fast as the number of paths decreases, but at the price of a loss in accuracy. For example, if only 5 paths are considered for each receiver, the path computation is almost instantaneous but with an important loss in accuracy. Limiting to 50 paths provides almost the same simulation results in terms of accuracy as for all paths computation, but in about 0.59 s instead of 6.78 s. Consequently, the default value for the optimal number of propagation paths is empirically set to 50 in the rest of this article.

One might ask about the difference between limiting the number of interactions and limiting the number of paths for a fixed number of interactions. It is not exactly the same thing, even if both procedures limit the number of paths. Increasing the number of interactions while limiting the number of

TABLE 2. Limitation to the most significant paths: accuracy and computation time.

Maximum number of paths	5	10	20	30	50	100	150	200	All
Average number of paths	4.59	9.12	17.99	26.53	42.99	79.63	108.81	130.56	261.34
Absolute mean error (dB)	11.35	7.63	4.44	3.04	1.75	0.63	0.48	0.37	0.00
Paths computation, all R_{xs} (s)	0.081	0.125	0.25	0.36	0.59	1.20	1.70	2.25	6.78

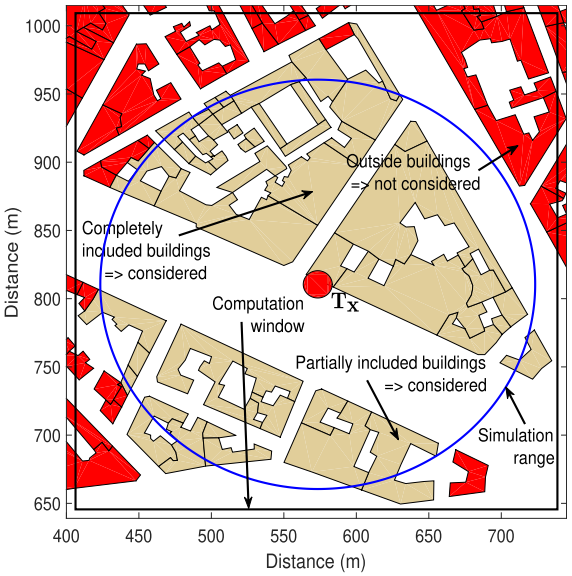

FIGURE 9. Principle of simulation scene limitation.

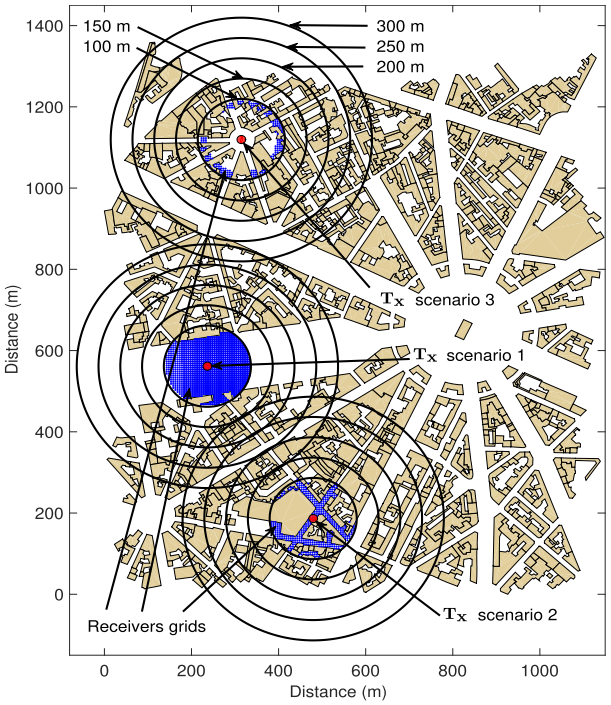

FIGURE 10. Scenarios for simulation scene limitation.

propagation paths has the advantage of achieving a higher coverage rate, i.e. distant receivers need more interactions to be reached.

B. SIZE LIMITATION OF THE SIMULATION ENVIRONMENT

As presented in Section II-C, the complexity of the simulation environment has a direct impact on the computation time for any RT model, VG included: the lower the complexity, the faster the computation. Furthermore, in the context of WSN, the transmitting power is usually limited for supply autonomy reasons. As an example, nominal transmitting power for the most commonly used protocols based on the IEEE 802.15.4 physical layer is 1 mW (0 dBm) [69], giving a radio range of at most a few hundred meters. Thus, limiting the simulation environment size to the order of magnitude of the distance separating the sensors should reduce the computation time while limiting the loss of precision.

The developed algorithm is designed as follows: starting from the transmitter, the buildings completely within a given range are considered in the simulation (blue circle in Fig.9). What exists outside this range is completely ignored (red buildings). Partially included buildings are also considered by adjusting the computation window's size (black rectangle) without including any new buildings that may exist inside the modified window. Consequently, the VG is only computed for the window of interest, which is much smaller than the whole propagation scene.

In order to determine the optimal simulation range, three urban test scenarios were defined on Fig.10 as follows:

- **Scenario 1 - an open area:** a uniform grid of 1,061 receivers (the blue points in Fig.10) located around the transmitter at distances of 0 - 100 m in an open space. The receivers in this scenario are mainly in LOS;
- **Scenario 2 - a narrow street:** a uniform grid of 347 receivers attached around the transmitter in a narrow street. This environment is rich in multi-paths, and the receivers are mainly in non-line-of-sight (NLOS);
- **Scenario 3 - an intersection:** a uniform grid of 162 receivers placed at a distance between 80 - 100 m (where the error values are higher than those of the near receivers due to the scene limitation, in order to assess the worst-case impact due to the area limitation technique). The receivers are both in LOS and NLOS.

From these scenarios, the simulation ranges (represented in Fig.10 as circles around the transmitter) vary from 100 m to 300 m with 50 m steps.

TABLE 3. Simulation range limitation: performance evaluation.

Scenario 1: Open area						
Range (m)	**100**	**150**	**200**	**250**	**300**	**Whole scene**
Absolute mean error (dB)	0.47	0.14	0.10	0.10	0.07	-
***VG computation* + *mapping* (s)**	0.09	0.84	1.96	3.03	3.55	12.14
Path computation time, all R_{xs} (s)	0.81	3.81	5.61	6.65	7.05	15.48
Scenario 2: Narrow street						
Range (m)	**100**	**150**	**200**	**250**	**300**	**Whole scene**
Absolute mean error (dB)	0.69	0.31	0.03	0.01	0.01	-
***VG computation* + *mapping* (s)**	0.90	1.34	1.78	2.06	2.39	4.03
Path computation time, all R_{xs} (s)	0.99	1.11	1.20	1.29	1.29	1.36
Scenario 3: Intersection						
Range (m)	**100**	**150**	**200**	**250**	**300**	**Whole scene**
Absolute mean error (dB)	1.34	0.11	0.10	0.01	0.004	-
***VG computation* + *mapping* (s)**	1.50	2.56	3.06	4.26	4.97	7.80
Path computation time, all R_{xs} (s)	0.28	0.45	0.50	0.59	0.59	0.59

Table 3 shows both the absolute mean error and the computation time (*VG computation* + *mapping* and multi-paths computation for all R_{xs}) of the three test scenarios for 4R1D at 1.8 GHz, using the whole scene as a self-reference for evaluating the impact of the simulation range. All three scenarios showed the same behavior: limiting the simulation range (∼150 m for these test scenarios) has a negligible impact on accuracy (from 0.11 to 0.84 dB) with a significant gain in time (gain factor from 2.2 to 5.9).

C. VISIBILITY GRAPHS PRE-PROCESSING

The VG computation depends on a given emitter location. However, smart cities contain many mobile sensors. A new VG needs to be computed at each location. In [66], Hussain *et al.* proposed computing a dynamic visibility table according to the emitters' linear displacement. Although it reduces the pre-processing time with regard to the computation of a visibility image tree according to each emitter's location, first its computation remains very time consuming and second, it only allows linear emitter displacements. The pre-process time for the emitter's mobile route in Fig.8 composed of 76 locations and computed for 5R0D is 42,715 s [66]. In comparison, the computation time for all our 76 exact VGs for 5R0D is 3 s. However, for the more complex VG considering 4R1D, the computation time is 371 s. Although this time is significantly less than the pre-process in [66], it remains too long to treat the high number of mobile sensors involved in an urban WSN. A significantly faster solution is proposed in the next section.

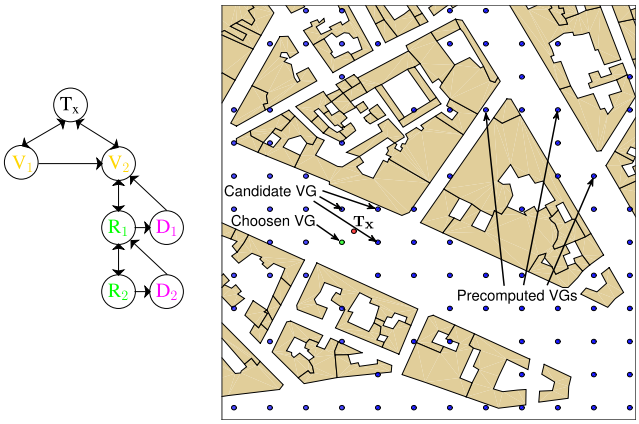

FIGURE 11. (a) Example of VG, (b) nearest VG search principle.

1) DESCRIPTION OF THE ALGORITHM

The VG structure presented in Section III models the propagation channels between an emitter and a set of receivers in a very efficient way, especially when the path computation is limited to the most significant ones, and the propagation environment is limited with regard to the radio range of the sensor. Consequently, the idea proposed here is to save on hard disk a set of VGs computed for some virtual transmitter locations. Then, from a given transmitter location, the VG pre-computed on the nearest virtual transmitter is directly used to compute the propagation paths, allowing a considerable time gain. The introduced approximation does not greatly affect accuracy if the distance between the position of the pre-computed VG and the current transmitter is within a limited range. To achieve this goal effectively and efficiently, three problems have to be addressed: firstly, the best data structure for saving the VGs in order to reduce both the writing/reading processing times and the storage size; secondly, the positioning strategy of the transmitters for which the VGs are calculated; and finally, how to find the best VG to use for a given transmitter location and the impact on accuracy.

a: DATA STRUCTURE

Let us assume that the VG in Fig.11(a) needs to be saved. The proposed data structure is a 2D integer array whose size equals the number of the VG's nodes. For each entry, i.e. node, the number of rows is linked to the node's depth in the VG (the number of levels from that node to the root) as illustrated in Table 4. Visible nodes are coded by 6 integer values corresponding to the Cartesian coordinates of the 3 zone vertexes (triangle). Simply reflected/diffracted nodes are represented by 12/8 integer values: 8/6 values for the zone' vertexes for reflected/diffracted zones respectively; 1 integer value coding the type of zone (Type R/D); 2 integer values for the coordinates of the reflection zone' source only; and

TABLE 4. Data structure for pre-computed VG file, storing the VG depicted on Fig.11(a).

V_1	V_2	R_1	D_1	R_2	D_2
x_1-V_1	x_1-V_2	Type R	Type D	Type R	Type R
y_1-V_1	y_1-V_2	IDFace	IDEdge	IDFace	IDFace
x_2-V_1	x_2-V_2	$x-S_{R_1}$	x_1-D_1	$x-S_{R_2}$	$x-S_{D_2}$
y_2-V_1	y_2-V_2	$y-S_{R_1}$	y_1-D_1	$y-S_{R_2}$	$y-S_{D_2}$
x_3-V_1	x_3-V_2	x_1-R_1	x_2-D_1	Type R	Type D
y_3-V_1	y_3-V_2	y_1-R_1	y_2-D_1	IDFace	IDEdge
		x_2-R_1	x_3-D_1	$x-S_{R_2}$	x_1-D_2
		y_2-R_1	y_3-D_1	$y-S_{R_2}$	y_1-D_2
		x_3-R_1		x_1-R_2	x_2-D_2
		y_3-R_1		y_1-R_2	y_2-D_2
		x_4-R_1		x_2-R_2	x_3-D_2
		y_4-R_1		y_2-R_2	y_3-D_2
				x_3-R_2	
				y_3-R_2	
				x_4-R_2	
				y_4-R_2	

TABLE 5. Computation time per link according to interactions number.

Nb of interactions	Time per link
1R1D	< 1 ms
2R1D	2 ms
3R1D	6 ms
4R1D	10 ms

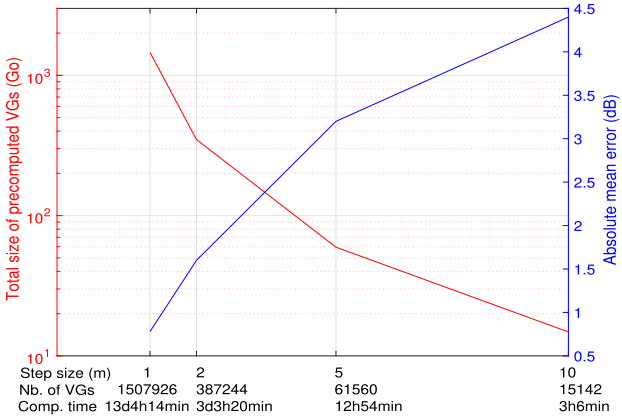

FIGURE 12. Absolute mean error and storage size *vs* step size, and the corresponding number of VGs and computation time.

1 integer value corresponding to the index of the face/edge where reflection/diffraction has occurred. The choice of integer values to code coordinates instead of floating ones allows the memory size of the data structure to be reduced. Hence, all the coordinates are rounded in centimeters, and so are easily coded with integer values. For nodes of depth > 2, all zones from depth 1 are saved sequentially, except that only the vertexes of the last level one are saved (*e.g.* R_2 and D_2 in Table 4). The last-level zone is needed to perform the inclusion test with the receiver, whereas the upper zones are no longer needed.

b: PRE-COMPUTED VG LOCATIONS

The virtual transmitters' locations are set using two strategies:

- A regular grid of NxM equally spaced virtual transmitters, with a chosen space step;
- When some data on the sensors' mobility is known, the user may provide a list of predefined locations corresponding to potential sensors' locations, allowing the number of pre-computed VGs to be limited.

c: VG SEARCH PROCESS

The principle is illustrated in Fig.11(b) where blue points correspond to virtual transmitters of pre-computed VGs. An intelligent search strategy is implemented by using a naming format that contains the position of the graph as VG_xLocation_yLocation. This means that there is no need to read all the saved VGs to obtain their locations, which can instead be directly revealed from the name of the file. Finally, the distance between the transmitter and the candidate VGs is calculated to select only the nearest VG (green point). Hence, only one VG file is read and processed. From this file, the propagation paths are built very quickly, as presented in Section III-B.

2) PERFORMANCES

Table 5 shows the computation time evolution according to the interactions number for selecting the nearest pre-computed VG file, reading it, reconstructing the 3D propagation paths, and calculating the field strength. It is averaged over thousands of sensors distributed in the *Charles de Gaulle - Étoile* environment (*cf.* Fig.5), and is thus given per link. It shows that our proposed model can perform very high-speed simulations even for quite a large number of interactions, in the order of 1 to 10 ms per link. The impact on accuracy of the VG pre-processing depends on the distance between the location of the used VG's root and the real transmitter. With our regular grid of virtual transmitters, it is thus directly linked to the step size. Another important parameter is the storage space needed to back up the VGs. It depends both on the mean VG size, which relies on the configuration of the surrounding environment, the number of interactions and the radio range considered, and on the number of VGs to be stored, which is again directly linked to the step size of the regular grid.

To illustrate the trade-off between accuracy and disk space consumption, Fig.12 shows the evolution of the size of all the precomputed VGs (red curve) and the corresponding absolute mean error (blue curve) according to the step size, along with the corresponding number of precomputed VGs

and computation time. These results were computed in the *Charles de Gaulle - Étoile* scene for a 4R1D configuration. VGs were pre-computed using step sizes of 10 m, 5 m, 2 m, and 1 m. Then, 150 transmitters were distributed through a route of 500 m length. Each transmitter picked the nearest pre-computed VG to reconstruct the paths. The absolute mean error was estimated over a grid of 150 receivers regularly distributed inside a range of 150 m around each transmitter. Of course, Fig.12 shows that the error decreases with the step size while the disk space requirement increases. The error goes from 0.78 dB for a 1 m step to 4.4 dB for a 10 m step. It remains relatively modest and of the order of the estimation error of conventional RT models. Both the required disk space and the computation time of the VGs pre-process reveal a square dependency to the grid step size. Hence the required amount of data to be stored goes from 14.8 Go for a 10 m step to 1.5 To for a 1 m step, while the corresponding computation time goes from about 3 hours to 13 days. The best trade-off would be fixed by the accuracy requirement of the application considered and the storage capacity of the used computer. But in a first approach, a 10 m step could be a standard solution for conventional computers.

V. OVER ROOFTOP PROPAGATION SOLUTION

As mentioned in section II, communication between two sensors separated by a large distance in a dense urban area, as for Lora or Sigfox sensors, is often established based on ORT propagation. Models for this type of contribution must first extract the geometry involved in the vertical profile between the two sensors to compute the propagation paths. Then a suitable physical model has to be used to predict the corresponding electrical field.

A. COMPUTATION OF VERTICAL PROPAGATION PATHS

As presented in Section III-A, all the horizontal building edges were mapped into a 2D discrete grid during the *scene2D mapping* (*cf.* pink pixels in Fig.13). In order to extract the vertical profile between the transmitter and the receiver, the 2D straight segment of line is first drawn between these two points. Next the segment of line is discretized according to the super-cover scheme (*cf.* blue pixels in Fig.13). Then its pixels are scanned, and all the contained edges intersecting the segment are added to the vertical profile. The above-mentioned algorithm was used to extract the vertical profile between a transmitter and a receiver in the dense urban environment of Fig.14(a). The corresponding profile (buildings height as a function of distance) is shown in Fig.14(b). From this profile, the propagation path with the minimum number of diffractions is considered to evaluate the electrical field, enhanced with a ground reflection between the last building and the receiver. These two paths are depicted in blue on Fig.14(b).

B. ELECTRICAL FIELD PREDICTION

As mentioned in Section II, propagation paths involved in ORT configuration can present a large number of successive

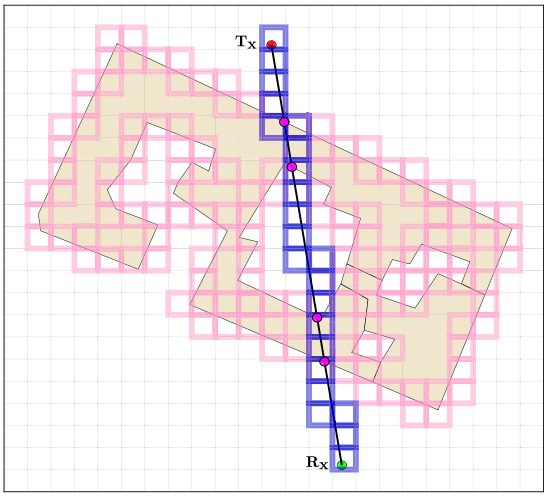

FIGURE 13. Principle of vertical profile extraction.

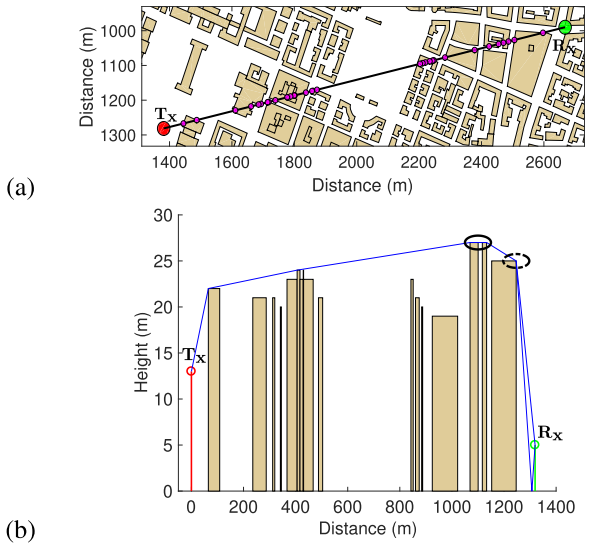

FIGURE 14. (a) Considered link in Munich, (b) extracted vertical profile.

diffracted rays on edges in each other's transition zones (*cf.* black circle in full line in Fig.14(b)). Conventional UTD diffraction coefficients [53] fail to accurately predict the electrical field in this case. Our solution relies on the closed-form solution for high-frequency double diffraction problems with a perfectly-conducting thick screen, provided by Capolino and Albani [57]. This was first formulated in a 2D geometry illuminated by a line source. It was validated against results obtained from the method of moment, and is still applicable when the thickness becomes vanishingly small, which is the case when a ray diffracts near any roof corner. Although it needs the computation of generalized Fresnel integral, an efficient solution was provided in [70]. Finally, it was extended in [58] to handle 3D diffraction problems. This solution is used in this work, in combination with classical UTD coefficients when single wedge diffraction occurs (*cf.* black circle in dotted line in Fig.14(b)).

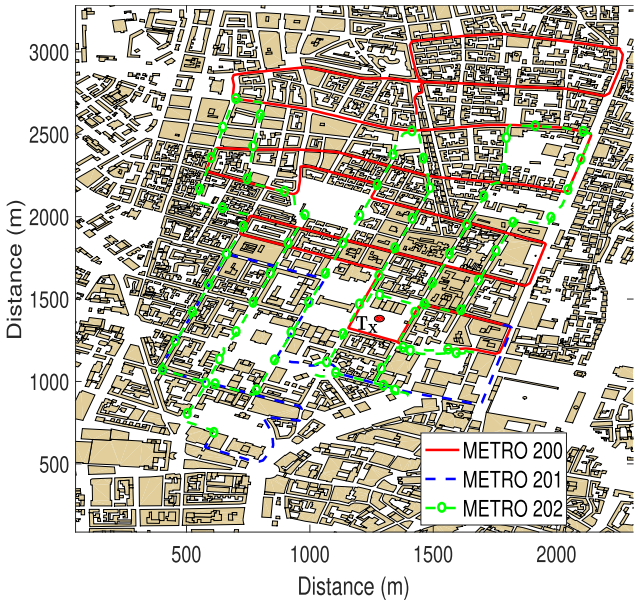

FIGURE 15. COST 231 measurement routes in Munich downtown.

VI. OVERALL PERFORMANCES

Our model was evaluated in comparison with the well-known measurement data realized in the framework of the COST231 European project [71]. Path loss measurement campaigns were conducted at 947 MHz in downtown Munich over three routes (the total length of the routes was about 23 km) including different receiver heights (from 1.2 to 1.9 m), as illustrated in Fig.15. The transmitter was located 13 m above the ground and had an omnidirectional radiation pattern. These three routes had receiver locations both distant and close to the transmitter, with significant paths propagated in both lateral and vertical planes. Hence, simulation results were obtained with the full model, i.e. the 2.5D model (*cf.* Section III) configured with 3R1D for the METRO200 and METRO202 routes and 3R2D for the METRO201 route (sufficient for convergence in this environment), in addition to the ORT model. Both geometrical data and measurements were provided by the Mannesmann Mobilfunk company.

Fig.16(a, b, and c) compare our simulation results to the measurements. The pink areas indicate the portions of the routes mainly covered by ORT contributions. They show good agreement with the measurements. Quantitative comparison values in Table 6 are of the same order as those in other published models [27]. Notice that both the mean errors and the standard deviations given in Table 6 are those obtained according to the full model, i.e. ORT+ 2.5D model.

One important point is that despite the good agreement between simulation and measurements, some larger errors were noticed at certain receiver locations (black circles in Fig.16(b and c)) where the simulation underestimated the path loss. This observation is similar to other models' results [63], [64]. These points are located near the transmitter as shown in Fig.16(d) (bottom scheme). The upper scheme of

TABLE 6. Computation time performances in *Munich* environment. The pre-processing time given in the last row ([66]) is given for order 2 with a maximum of 1 diffraction (i.e. 1R1D + 2R0D), for the small part of *Munich* environment presented in Fig.8.

Measurement routes		Metro 200 970 R_{xs}	Metro 201 355 R_{xs}	Metro 202 1,031 R_{xs}
Mean error (dB)		-2.29	-1.95	2.25
Standard deviation of the error (dB)		8.66	7.77	7.81
scene2D mapping + *DG computation*		85 s		
ORT model	All R_{xs}	580 ms	180 ms	480 ms
	Per link	0.60 ms	0.51 ms	0.47 ms
2.5D model	*VG comp.* + *map.*, 3R1D	11 s		
	All R_{xs} (all paths)	2.1 s	128 ms	12 s
	All R_{xs} (50 paths)	100 ms	35 ms	150 ms
[27]	Pre-proc.	1 hour		
	All R_{xs}	123.5 ms	45.20 ms	131.28 ms
[64], [66]	Pre-proc.	1,871 s		
	All Paths, All R_{xs}, order 2	1,400 s	540 s	1,560 s

Fig.16(d) shows a Google Earth map in this area, with a lot of vegetation (green circles) around the buildings. These trees are not considered in the GIS data. These missing obstacles that obstruct the propagation path between the transmitter and the receivers could explain the highlighted errors. Some vegetation models recommended by the ITU could be integrated to model the vegetation-induced losses [72]. The two buildings surrounded by purple rectangles were built after the measurement campaign and are therefore not modeled in the GIS data.

Table 6 also presents the overall model performance in terms of computation time. First, the *scene2D mapping* plus the *DG computation* takes 85 s. These computations are performed only once, whatever the transceivers locations. Then, it shows that the ORT model is very fast: it takes less than 1 ms per link to extract the vertical profile, to find the vertical paths, and to estimate the electrical field. Consequently, the computation time for all the receivers in each of the three measurement routes is less than 600 ms.

The 2.5D horizontal model was also calculated for *Munich* environment to show that the overall model is fast even for large scenarios. Table 6 shows that the *VG computation* according to the combination 3R1D takes about 11 s, whereas the paths computation takes 2.1 s, 128 ms, and 12 s for routes 200, 201, and 202 respectively, when all paths are computed. When only the 50 most significant paths are computed, according to the method presented in Section IV-A,

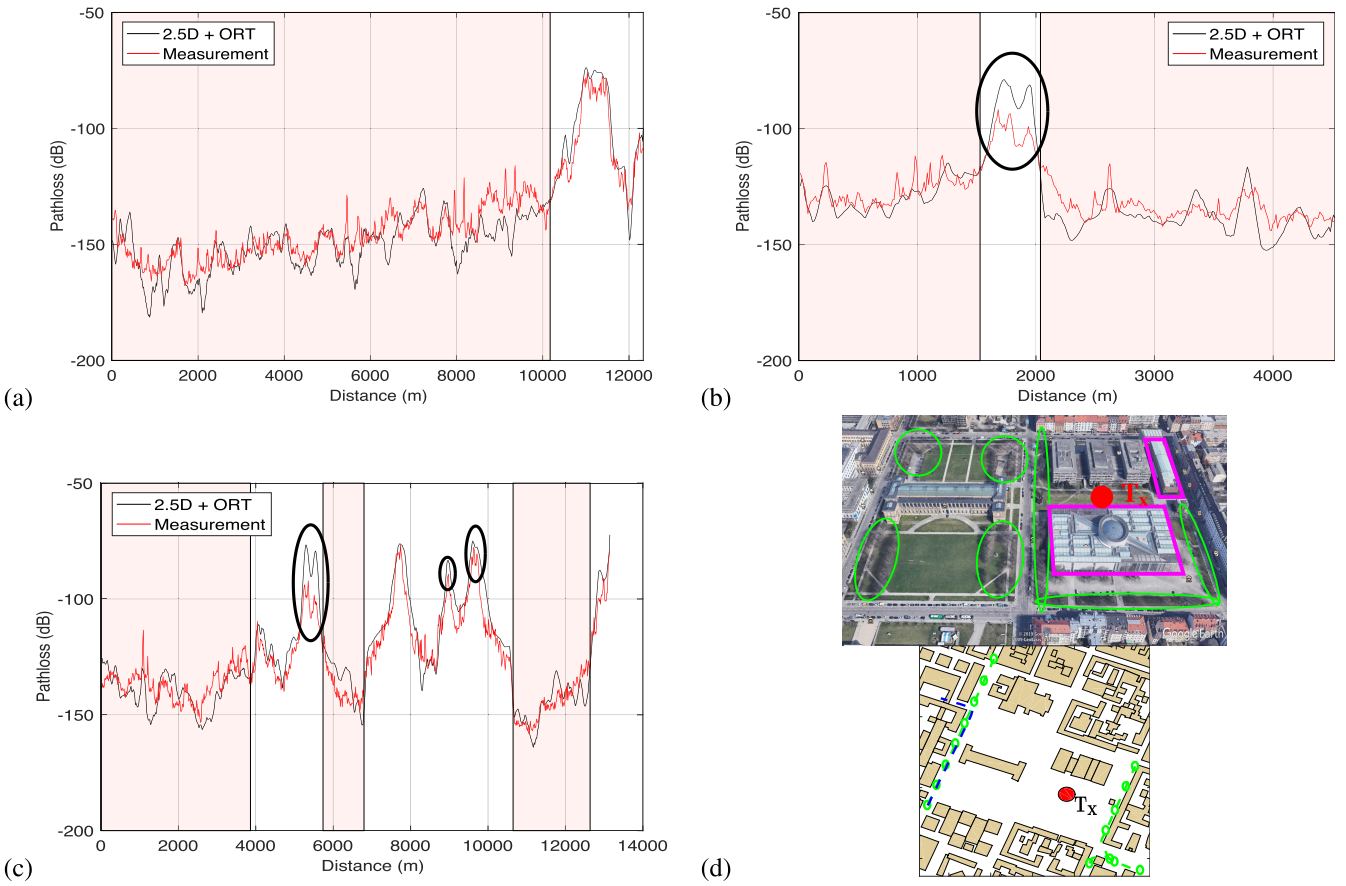

FIGURE 16. Performance evaluation: (a) route Metro 200, (b) route Metro 201, (c) route Metro 202. (d) Vegetation impact and missing buildings.

the computation time is drastically reduced to 100, 35, and 150 ms for the three routes. It should be noted that it is still possible to obtain very fast performance with the VG pre-processing mode (*cf.* section IV-C). The computation time per link for the 2.5D model is not included because it is not particularly dependent on the number of receivers.

Finally, Table 6 compares computation times obtained from the proposed model with those of other published solutions. The solution presented in [27] demands a pre-processing step for the environment model, taking about 1 hour with a CORE i5 running at 2.8 GHz. Then the computation of the propagation paths and the corresponding electrical field is very fast but needs some measurements to calibrate the model, which is a significant constraint because measurements will be needed for any new environment in which one wants to deploy a WSN. In [64], [66], no measurement is needed, as in our solution, but computation times are very high (obtained with a CORE i5 running at 3.3 GHz). The visibility pre-process for 2R0D + 1R1D takes 142,200 / 76 = 1,871 s on the small part of the *Munich* environment presented in Fig.8 [66]. Unfortunately, the pre-process time for the whole Munich environment is not given in any papers, but should be much greater. To compute all the paths for all R_{xs} locations, the RT-process takes 1,400 s, 540 s, and 1,560 s for routes Metro 200, 201, and 202 respectively [64].

VII. INTEGRATION INTO THE WSN SIMULATOR

Having demonstrated the performance of our model in terms of both accuracy and efficiency, this last section presents its integration into the CupCarbon WSN simulator developed in the framework of the PERSEPTEUR project, funded by the French national research agency. An illustration of the advantage of using a realistic channel model instead of a conventional statistical ones is also presented.

A. CUPCARBON

CupCarbon [73]–[75] is a Smart City and Internet of Things WSN simulator for both scientific and educational purposes. Its objective is to provide reliable simulations for WSNs, mainly in terms of propagation and interference of signals. It is very useful for designing, visualizing, validating, and debugging distributed algorithms for real projects such as environmental data monitoring. It also enables engineers and researchers to test their wireless topologies, protocols, etc. in a 3D urban environment.

Networks can be easily designed with CupCarbon's user interface by deploying sensors directly on the map. CupCarbon uses the OpenStreetMap (OSM) framework, which is the geometry source for deterministic radio propagation models. Two simulation environments are offered by CupCarbon. The first one enables the design of mobility scenarios and the

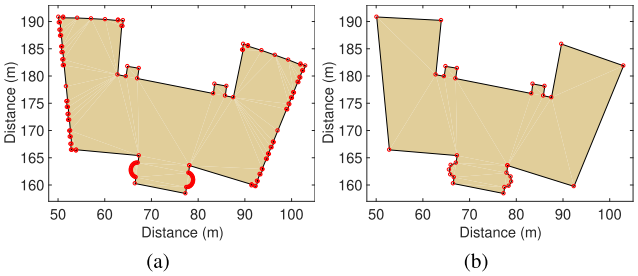

FIGURE 17. (a) Initial OSM building (138 faces), (b) simplified building (30 faces) with $\epsilon = 0.3$.

generation of events. The second enables discrete event simulation, and takes into consideration the mobility scenarios defined in the first environment.

CupCarbon also includes many interesting features such as: the ability to simulate the interference of signals; an energy consumption model; an easy script language (SenScript) to program sensors; intelligent mobility; a user-friendly graphical interface; the ability to split sensors into separate networks; and clear visualization of the network and the working environment, etc. It also includes some WSN protocols such as ZigBee, LoRa, and WiFi.

B. PROCESSING OF GEOMETRICAL DATA

Radio propagation models were integrated through an Application Programming Interface (API), designed to send only the required simulation parameters and the geometry database, to launch the computation and deliver back the required channel estimations without having access to the operations occurring in the background. The end-user can define (through CupCarbon interface) the simulation parameters for the API; as for example, the combination of electromagnetic interactions.

The polygons defined in the geometry database contain a large number of points, many of them having been generated by the measurement process and thus having no physical meaning. These unnecessary details considerably reduce the performance of any RT model while consuming a huge amount of memory. Therefore, in order to ensure the best performance of the API, it is necessary to simplify the buildings' outlines before using them as an input for the API. We used the Douglas-Peucker method [76] to do this, as illustrated in Fig.17, where its parameter ϵ was fixed at 0.3. This process allows a drastic reduction of the number of faces composing the building (138 to 30 in this example) while preserving its overall shape.

Finally, some problems remain with the OSM geometric database. First, a large number of existing buildings are not modeled, even in the dense urban environments of the city centers of major cities, such as Paris in France. Second, the heights of the modeled buildings are often very poorly documented or missing. However, these problems are likely to be corrected in the near future with the continuous improvement of GIS.

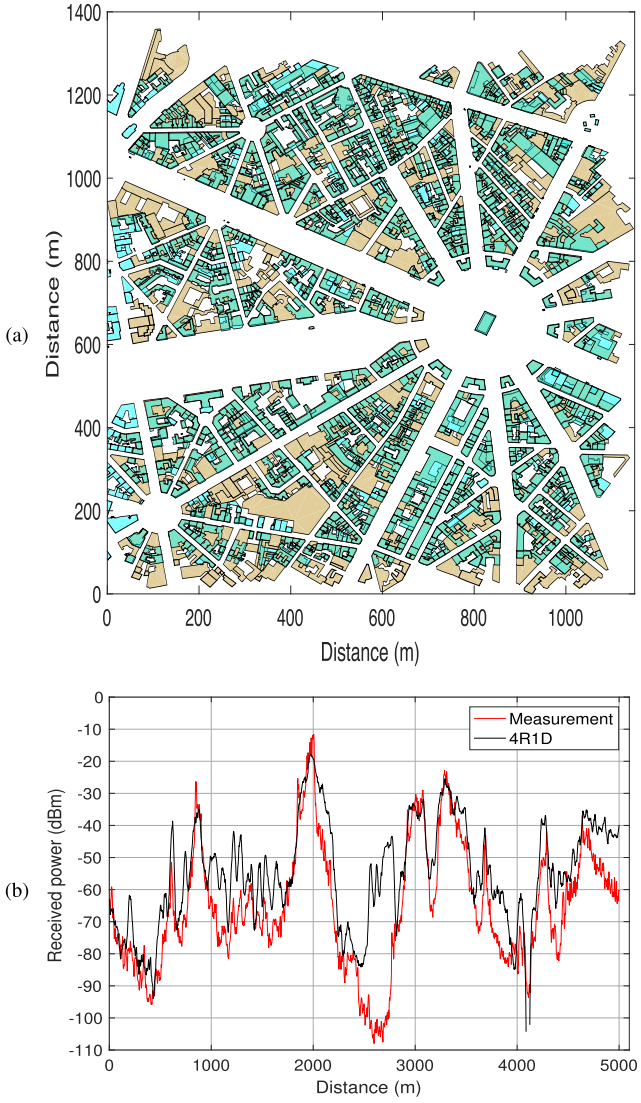

FIGURE 18. (a) Scene extracted from OSM (in blue) superposed to the one provided by the IGN (in brown, *cf.* Fig.5); some buildings are missing in OSM; (b) simulation result from OSM scene.

C. API VALIDATION

We validated the API with the scene *Charles de Gaulle - Étoile*, Paris, already used in section III-C. But here, its geometry was taken directly from the CupCarbon user interface, and so from the OSM database. Obviously, the building outline simplification based on Douglas-Peucker algorithm was applied. The result is shown in Fig.18(a). As mentioned in Section VII-B, some buildings are missing in comparison to those shown in Fig.5.

Fig.18(b) compares the simulation result of the OSM scene to the measurements, by considering 4R1D. Relatively good agreement can be seen even though the mean and standard deviation errors are higher than those obtained in section III-C (10.87 dB and 12.83 dB respectively), as expected with the large number of missing buildings. This result shows the importance of accurate geometrical data in achieving good field predictions in urban areas.

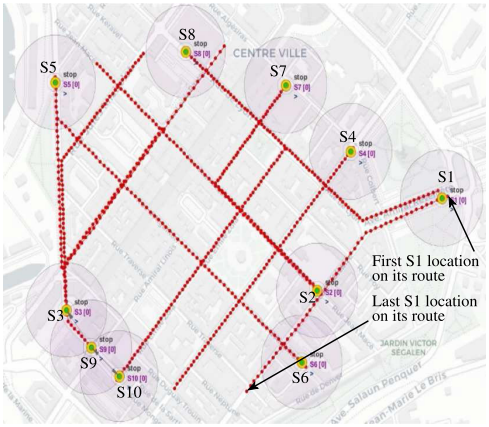

FIGURE 19. WSN scenario.

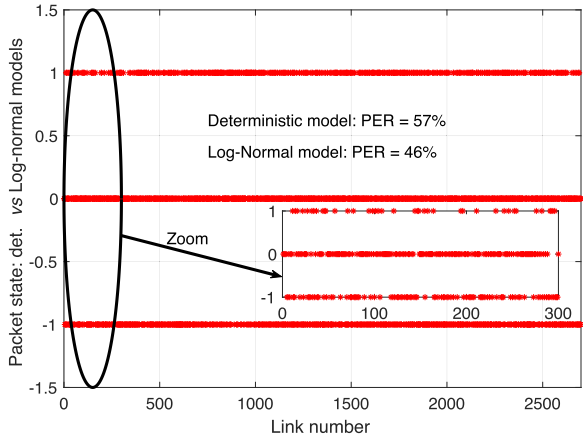

FIGURE 20. Impact of channel model on the PER. The results for the 300 first links are detailed in the zoomed part.

D. IMPACT OF DETERMINISTIC MODEL ON WSN PERFORMANCES

This section describes a case study carried out in Brest city, France. The test scenario consisted of a small WSN with 10 sensors (s1, s2, ..., s10). All the sensors are moving along predefined trajectories represented by the red-dotted paths in Fig.19. The sensors move at a constant speed along these paths and reach the end of their trajectories in 59 seconds; therefore, it is convenient to trace the behavior in steps of 1 second, resulting in 60 snapshots for each sensor.

The main simulation parameters are presented below:

Configuration: Urban;
Location: Downtown Brest;
Simul. area: 800 m x 800 m;
Protocol: ZigBee;
Transmit power: 20 dBm;
Frequency: 2.4 GHz;
Sensitivity level: -100 dBm;
Channel model: 2.5D;
Interactions number:4R1D.

However, it should be noted that the transmitter's power and R_x's sensitivity values are not the theoretical values defined by the standard; rather, they came from a commercial ZigBee module (Ember EM357 Transceiver - ZICM357P2) in order to extend the link budget.

In order to show the importance of using a realistic channel model that takes the propagation environment into account, we conducted a comparison of a network parameter: the Packet Error Rate (PER). PER values were obtained by using firstly our deterministic model and then a statistical channel model classically used in WSN simulators, i.e. the Log-Normal shadowing model. To obtain comparable results, the different parameters of the Log-Normal model were fitted to measurements obtained in the relevant environment, which led to the following model:

$$L_{dB}(d) = -14.27 + 10 \times 5.7 \times \log(d) + X_{\sigma=16dB}, \quad (6)$$

where d is the distance between the sensors and X_σ is a normal distributed random variable with σ standard deviation equal to 16 dB. Thus, random data were generated and formatted as ZigBee Payloads, then they were added to the ZigBee preamble to form a ZigBee frame as defined by the physical layer of the protocol. All the other physical layer parameters such as center frequency, sampling frequency, bit rate, chip rate, etc., were also considered. The ZigBee frames were then sent to the receiver sensor, via the radio channel: first with the proposed deterministic model and then with the adapted Log-Normal one. The same white Gaussian noise was added to the two channels. At the receiver side, the received signal was decoded with a maximum of three retransmission requests, as indicated by the ZigBee protocol.

The PER values of the two models were computed over the 2,700 links of the considered mobility scenario in Fig.19. Our deterministic model led to a mean PER of 57.10^{-2} whereas the Log-normal model overestimated the transmission performances with a mean PER of 46.10^{-2}.

In order to highlight their specific impact, Fig.20 compares the PER obtained by our deterministic model to the Log-normal one, radio link per radio link. More specifically, the three y-axis values in the figure were analyzed as follows:

- 1: the packet was correctly received/not received by considering the deterministic model and the Log-normal one respectively;
- 0: the packet was received or not received for both models;
- -1: the packet was not received/correctly received by considering the deterministic model and the Log-normal one respectively.

From this figure, it appears that 51% of packets were considered as received by the Log-normal model when they were not received correctly by the deterministic model, and vice versa. This difference confirms that the parametrized Log-Normal is not accurate enough to characterize the WSN performances in a realistic specific environment.

VIII. CONCLUSION

This article has presented a study of the radio wireless channel used in a dense urban environment for the communication between a set of mobile sensors composing a WSN. The proposed solution is a ray-tracing based model able to efficiently simulate large networks. It consists of two approaches linked to the main mode of wave propagation - horizontal or

vertical - depending on the relative heights of the antennas and buildings, and the distance between the sensors.

The first approach uses a 2D exact visibility graph, which is a point-to-zone structure allowing the fast calculation of exact lit zones based on the super-cover model. The exploitation of this structure permits very efficient computation of the 3D propagation paths between a transmitter and a set of receivers. As a first contribution, a new implementation of the VG computation allows a reduction of the number of nodes, and so both the memory consumption and the final ray-paths computation time. After validating the model by comparison with measurements, and as a second contribution, we showed that the specific VG structure naturally led to classifying the ray-paths contributions in an increasing order of attenuation. It is thus easy to limit the calculation to the main significant paths so as to decrease the computation time while controlling the loss of precision in the simulation. A third contribution uses the theoretical radio range of sensors to limit the size of the simulation area. The results show that this drastically reduces the computation time without decreasing the accuracy level of the simulation. As a fourth contribution, we proposed an algorithm dedicated to the management of the dynamic behavior of the sensors. This involves precomputing a set of VGs according to a set of virtual transmitter locations. Then, the one computed for the nearest location to the considered sensor is used on demand, instead of the exact VG, to simulate the propagation paths. The results showed that this process leads to mean computation times of less than 10 ms per link, which is significantly better than other published models, and compatible with WSNs composed of thousands of sensors. Nevertheless, a tradeoff has to be found between computation time gain and disk space consumption for VG storage.

The second approach is dedicated to ORT transmission for distant sensors. First we used the super-cover of the T_x-R_x segment to quasi-instantaneously extract the vertical profile, and compute the main significant ray-paths contributions. Then, we avoided the field divergence problem for multiple successive diffractions on the building roof tops, by the joint use of conventional UTD coefficients for single edge diffraction and the Albani and Capolino's coefficients for double diffraction on a thick screen. The simulation results were validated by comparison with measurements and gave a level of accuracy of the same order as the literature results. Computation times were, in turn, significantly better.

As a global validation, we integrated the proposed propagation model into the CupCarbon WSN simulator. Some problems relating to the accuracy of the geometric database were highlighted, such as the level of detail of the building outlines, and solutions were proposed to remedy them by adapting the available data in an automatic way. Finally, we demonstrated the importance of considering a realistic channel model instead of a conventional statistical one, by simulating a WSN evolving in a real environment. Results showed that a statistical model can lead to the mis-estimation of more than 50% of the links.

Further research could address the environment geometry. First, new algorithms could be developed to simplify the 2D outline of the buildings, for instance by grouping a set of related buildings together, in order to further improve the VG computation time. Second, the geometry of roof tops could be enhanced by introducing the real shape of the roof, i.e. not a flat surface as considered in this study. The impact on the ORT model could be significant. Finally, new algorithms will have to be developed to deal with future 5G networks, taking into account the diffusion phenomena for millimeter waves or even light waves. In this case, particular attention should be paid to the vegetation and accuracy of 3D urban models, which will have a great impact on predictions.

As the parametrization space for this type of channel propagation modelling approach is large, future research may consider machine learning approaches to adaptively determine suitable parameters, for any properly defined urban context. This would allow the WSN designer to more productively focus on the task at hand with less knowledge needed on the intricate radio model components.

REFERENCES

[1] R. Lea. *An Overview of the Technology Trends Driving Smart Cities*. Accessed: Mar. 2017. [Online]. Available: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-industry-advisory-board/ieee-smart-cities-trend-paper-2017.pdf

[2] A. Ersoy, "Smart cities as a mechanism towards a broader understanding of infrastructure interdependencies," *Regional Stud., Regional Sci.*, vol. 4, no. 1, pp. 26–31, Jan. 2017, doi: 10.1080/21681376.2017.1281154.

[3] F. Al-Fayez, A. Abuarqoub, M. Hammoudeh, and A. Nisbet, "Wireless sensor network simulation : The current state and simulation tools," *Sensors Transd. J.*, vol. 18, pp. 145–155, Jan. 2013. [Online]. Available: https://www.sensorsportal.com/HTML/DIGEST/P_SI_301.htm

[4] K. Tan, D. Wu, A. Chan, and P. Mohapatra, "Comparing simulation tools and experimental testbeds for wireless mesh networks," in *Proc. IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2010, pp. 1–9, doi: 10.1109/WOWMOM.2010.5534917.

[5] A. Abuarqoub, F. Al-Fayez, T. Alsboui, M. Hammoudeh, and A. Nisbet, "Simulation issues in wireless sensor networks : A survey," in *Proc. SENSORCOMM*, Rome, Italy, 2012, pp. 222–228.

[6] A. Nayyar and R. Singh, "A comprehensive review of simulation tools for wireless sensor networks (WSNs)," *J. Wireless Netw. Commun.*, vol. 5, no. 1, pp. 19–47, 2015, doi: 10.5923/j.jwnc.20150501.03.

[7] A. Abuarqoub, M. Hammoudeh, F. Al-Fayez, and O. Aldabbas, "A Survey on Wireless Sensor Networks Simulation Tools and Testbeds," in *Sensors, Transducers, Signal Conditioning and Wireless Sensors Networks Advances in Sensors: Reviews*, vol. 3, S. Y. Yurish, Ed. Barcelona, Spain: IFSA, 2016, ch. 14.

[8] *The Network Simulator NS2*. Accessed: Nov. 2017. [Online]. Available: https://www.isi.edu/nsnam/ns/

[9] *Radio Propagation Models Implemented in NS2*. Accessed: Nov. 2017. [Online]. Available: https://www.isi.edu/nsnam/ns/doc/node216.html

[10] R. M. Pereira, L. B. Ruiz, and M. L. A. Ghizoni, "Mannasim: A NS2 extension to simulate wireless sensor network," presented at the 14th Intern. Conf. Netw., Apr. 2015. [Online]. Available: https://www.researchgate.net/publication/283502095_MannaSim_A_NS-2_Extension_to_Simulate_Wireless_Sensor_Network

[11] *NS3 Model Library*. Accessed: Oct. 2016. [Online]. Available: https://www.nsnam.org/docs/release/3.26/models/ns-3-model-library.pdf

[12] *Tossim Simulator*. Accessed: Nov. 2017. [Online]. Available: http://tinyos.stanford.edu/tinyos-wiki/index.php/TOSSIM/

[13] *Building a Network Topology for Tossim*. Accessed: Nov. 2017. [Online]. Available: https://github.com/tschmid/tinyos-2.x/blob/master/doc/html/tutorial/usc-topologies.html/

[14] A. Fraboulet, G. Chelius, and E. Fleury, "Worldsens: Development and prototyping tools for application specific wireless sensors networks," in *Proc. 6th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2007, pp. 176–185.

[15] *The Models Included in the Wsnet Distribution*. Accessed: Nov. 2017. [Online]. Available: https://wsnet.gforge.inria.fr/models.html/

[16] *QualNet 8.0 Model Library Index, Scalable Network Technologies Inc., Japan*. Accessed: May 2017. [Online]. Available: https://vdocuments.mx/download/qualnet-8-80-model-library

[17] A. Boulis. (Mar. 2011). *Castalia User's Manual : A Simulator for Wireless Sensor Networks and Body Area Networks*. Castalia Version 3.2. [Online]. Available: http://cpham.perso.univ-pau.fr/ENSEIGNEMENT/PAU-UPPA/INGRESM1/Castalia%20-%20User%20Manual.pdf

[18] A. Köpke, M. Swigulski, K. Wessel, D. Willkomm, P. T. K. Haneveld, T. E. V. Parker, O. W. Visser, H. S. Lichte, and S. Valentin, "Simulating wireless and mobile networks in OMNeT++ the MiXiM vision," in *Proc. 1st Int. ICST Conf. Simulation Tools Techn. Commun. Netw. Syst.*, 2008, pp. 1–8, doi: 10.4108/ICST.SIMUTOOLS2008.3031.

[19] *Mixim Simulator Website*. Accessed: Nov. 2017. [Online]. Available: http://mixim.sourceforge.net

[20] A. Schmitz, M. Schinnenburg, J. Gross, and A. Aguiar, "Channel modelling," in *Modeling and Tools for Network Simulation*, 1st ed. Berlin, Germany: Springer, 2010, ch. 11.

[21] T. K. Sarkar, Z. Ji, K. Kim, A. Medouri, and M. Salazar-Palma, "A survey of various propagation models for mobile communication," *IEEE Antennas Propag. Mag.*, vol. 45, no. 3, pp. 51–82, Jun. 2003, doi: 10.1109/MAP.2003.1232163.

[22] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, Jul. 2015, doi: 10.1109/ACCESS.2015.2453991.

[23] L. Aveneau, P. Combeau, R. Vauzelle, and M. Meriaux, "Efficient computation of radio coverage zone using a spatial partitionment approach," in *Proc. IEEE 58th Veh. Technol. Conf. VTC -Fall*, Oct. 2003, pp. 65–68.

[24] P. Combeau, L. Aveneau, R. Vauzelle, and Y. Pousset, "Efficient 2-D ray-tracing method for narrow and wideband channel characterisation in microcellular configurations," *IEE Proc. Microw., Antennas Propag.*, vol. 153, no. 6, pp. 502–509, Dec. 2006, doi: 10.1049/ip-map:20045142.

[25] T. Alwajeeh, P. Combeau, R. Vauzelle, and A. Bounceur, "A high-speed 2.5D ray-tracing propagation model for microcellular systems, application: Smart cities," in *Proc. 11th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2017, pp. 3515–3519.

[26] N. Noori, "Radio propagation and channel modeling aspects of 4G and beyond networks," in *Advances in Mobile Computing and Communications: Perspectives and Emerging Trends in 5G Networks*, 1st ed. Boca Raton, Florida, USA: CRC Press, 2016, ch. 2, sec. 2.3, pp. 34–35.

[27] D. He, G. Liang, J. Portilla, and T. Riesgo, "A novel method for radio propagation simulation based on automatic 3D environment reconstruction," *Radioengineering*, vol. 21, no. 4, pp. 985–992, Dec. 2012.

[28] A. Schmitz, T. Rick, T. Karolski, L. Kobbelt, and T. Kuhlen, "Beam tracing for multipath propagation in urban environments," in *IEEE Proc. EUCAP*, Berlin, Germany, Mar. 2009, pp. 2631–2635.

[29] S. Y. Seidel and T. S. Rappaport, "Site-specific propagation prediction for wireless in-building personal communication system design," *IEEE Trans. Veh. Technol.*, vol. 43, no. 4, pp. 879–891, Nov. 1994, doi: 10.1109/25.330150.

[30] G. Liang and H. L. Bertoni, "A new approach to 3-D ray tracing for propagation prediction in cities," *IEEE Trans. Antennas Propag.*, vol. 46, no. 6, pp. 853–863, Jun. 1998, doi: 10.1109/8.686774.

[31] F. S. de Adana, O. G. Blanco, I. G. Diego, J. P. Arriaga, and M. F. Catedra, "Propagation model based on ray tracing for the design of personal communication systems in indoor environments," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2105–2112, Nov. 2000, doi: 10.1109/25.901882.

[32] F. A. Agelet, F. P. Fontan, and A. Formella, "Fast ray tracing for microcellular and indoor environments," *IEEE Trans. Magn.*, vol. 33, no. 2, pp. 1484–1487, Mar. 1997, doi: 10.1109/20.582541.

[33] R. P. Torres, L. Valle, M. Domingo, S. Loredo, and M. C. Diez, "CINDOOR: An engineering tool for planning and design of wireless systems in enclosed spaces," *IEEE Antennas Propag. Mag.*, vol. 41, no. 4, pp. 11–22, Aug. 1999, doi: 10.1109/74.789733.

[34] Z. Yun, M. F. Iskander, and Z. Zhang, "Fast ray tracing procedure using space division with uniform rectangular grid," *Electron. Lett.*, vol. 36, no. 10, pp. 895–897, May 2000, doi: 10.1049/el:20000653.

[35] Z. Yun, M. F. Iskander, and Z. Zhang, "A fast indoor/outdoor ray tracing procedure using combined uniform rectangular and unstructured triangular grids," in *Proc. IEEE Antennas Propag. Soc. Int. Symp. Transmitting Waves Prog. Next Millennium. Digest. Held Conjunct, USNC/URSI Nat. Radio Sci. Meeting*, Jul. 2000, pp. 1134–1137.

[36] Z. Yun, Z. Zhang, and M. F. Iskander, "A ray-tracing method based on the triangular grid approach and application to propagation prediction in urban environments," *IEEE Trans. Antennas Propag.*, vol. 50, no. 5, pp. 750–758, May 2002, doi: 10.1109/TAP.2002.1011243.

[37] M. F. Iskander and Z. Yun, "Propagation prediction models for wireless communication systems," *IEEE Trans. Microw. Theory Techn.*, vol. 50, no. 3, pp. 662–673, Mar. 2002, doi: 10.1109/22.989951.

[38] I. Wald and V. Havran, "On building fast kd-trees for ray tracing, and on doing that in O(N log N)," in *Proc. IEEE Symp. Interact. Ray Tracing*, Sep. 2006, pp. 61–69.

[39] Y.-S. Feng, L.-X. Guo, P. Wang, and Z.-Y. Liu, "Efficient ray-tracing model for propagation prediction for microcellular wireless communication systems," in *Proc. ISAPE*, Oct. 2012, pp. 432–435.

[40] W. He, M. Zhang, and Y. Bo, "A fast ray-tracing software package for field distribution predictions," in *Proc. IEEE Int. Workshop Electromagn., Appl. Student Innov. Competition (iWEM)*, May 2016, pp. 1–3.

[41] F. Weinmann, "Adaptive and automated multilevel uniform space division for acceleration of high-frequency electromagnetic simulations [EM Programmer's Notebook]," *IEEE Antennas Propag. Mag.*, vol. 59, no. 1, pp. 130–138, Feb. 2017, doi: 10.1109/map.2016.2629186.

[42] C. Luo, X. Mei, and H. Lin, "Localized KD-tree accelerated beam tracing method for modelling complex urban environment propagation," in *Proc. IEEE Int. Conf. Comput. Electromagn. (ICCEM)*, Mar. 2018, pp. 1–2.

[43] G. Wölfe, R. Hoppe, and F. Landstorfer. (1999). *A Fast and Enhanced Ray Optical Propagation Model for Indoor and Urban Scenarios, Based on an Intelligent Preprocessing of the Database*. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.100&rep=rep1&type=pdf

[44] C. Saeidi, A. Fard, and F. Hodjatkashani, "Full three-dimensional radio wave propagation prediction model," *IEEE Trans. Antennas Propag.*, vol. 60, no. 5, pp. 2462–2471, May 2012, doi: 10.1109/tap.2012.2189692.

[45] J. Tan, Z. Su, and Y. Long, "A full 3-D GPU-based beam-tracing method for complex indoor environments propagation modeling," *IEEE Trans. Antennas Propag.*, vol. 63, no. 6, pp. 2705–2718, Jun. 2015, doi: 10.1109/tap.2015.2415036.

[46] E. M. Kenny and E. O. Nuallain, "Convex space building discretization for ray-tracing," *IEEE Trans. Antennas Propag.*, vol. 65, no. 5, pp. 2578–2591, May 2017, doi: 10.1109/tap.2017.2676726.

[47] C.-H. Teh, B.-K. Chung, and E.-H. Lim, "An accurate and efficient 3-D shooting-and- bouncing-polygon ray tracer for radio propagation modeling," *IEEE Trans. Antennas Propag.*, vol. 66, no. 12, pp. 7244–7254, Dec. 2018, doi: 10.1109/tap.2018.2874519.

[48] K. Rizk, R. Valenzuela, S. Fortune, D. Chizhik, and F. Gardiol, "Lateral, full-3D and vertical plane propagation in microcells and small cells," in *Proc. VTC 98. 48th IEEE Veh. Technol. Conf. Pathway Global Wireless Revolution*, May 1998, pp. 998–1003.

[49] J. Deygout, "Multiple knife-edge diffraction of microwaves," *IEEE Trans. Antennas Propag.*, vol. 14, no. 4, pp. 480–489, Jul. 1966, doi: 10.1109/tap.1966.1138719.

[50] C. L. Giovaneli, "An analysis of simplified solutions for multiple knife-edge diffraction," *IEEE Trans. Antennas Propag.*, vol. 32, no. 3, pp. 297–301, Mar. 1984, doi: 10.1109/tap.1984.1143299.

[51] L. E. Vogler, "An attenuation function for multiple knife-edge diffraction," *Radio Sci.*, vol. 17, no. 6, pp. 1541–1546, Nov. 1982, doi: 10.1029/rs017i006p01541.

[52] J. Beyer, "An approximate approach to predict multiple screen diffraction in the case of grazing incidence," *Radio Sci.*, vol. 39, no. 4, pp. 1–6, Aug. 2004, doi: 10.1029/2003rs002940.

[53] R. G. Kouyoumjian and P. H. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *Proc. IEEE*, vol. 62, no. 11, pp. 1448–1461, Nov. 1974, doi: 10.1109/PROC.1974.9651.

[54] P. D. Holm, "UTD-diffraction coefficients for higher order wedge diffracted fields," *IEEE Trans. Antennas Propag.*, vol. 44, no. 6, pp. 879–888, Jun. 1996, doi: 10.1109/8.509892.

[55] J. B. Andersen, "Transition zone diffraction by multiple edges," *IEE Proc. Microw., Antennas Propag.*, vol. 141, no. 5, pp. 382–384, Oct. 1994, doi: 10.1049/ip-map:19941407.

[56] K. Rizk, R. Valenzuela, D. Chizhik, and F. Gardiol, "Application of the slope diffraction method for urban microwave propagation prediction," in *Proc. VTC 98. 48th IEEE Veh. Technol. Conf., Pathway Global Wireless Revolution*, May 1998, pp. 1150–1155.

[57] M. Albani, F. Capolino, S. Maci, and R. Tiberio, "Diffraction at a thick screen including corrugations on the top face," *IEEE Trans. Antennas Propag.*, vol. 45, no. 2, pp. 277–283, Feb. 1997, doi: 10.1109/8.560346.

[58] F. Capolino, I. M. Albani, S. Maci, and R. Tiberio, "Double diffraction at a pair of coplanar skew edges," *IEEE Trans. Antennas Propag.*, vol. 45, no. 8, pp. 1219–1226, Aug. 1997, doi: 10.1109/8.611240.
[59] K. Rizk, J.-F. Wagen, and F. Gardiol, "Two-dimensional ray-tracing modeling for propagation prediction in microcellular environments," *IEEE Trans. Veh. Technol.*, vol. 46, no. 2, pp. 508–518, May 1997, doi: 10.1109/25.580789.
[60] G. E. Athanasiadou and A. R. Nix, "A novel 3-D indoor ray-tracing propagation model: The Path generator and evaluation of narrow-band and wide-band predictions," *IEEE Trans. Veh. Technol.*, vol. 49, no. 4, pp. 1152–1168, Jul. 2000, doi: 10.1109/25.875222.
[61] G. E. Athanasiadou, A. R. Nix, and J. P. McGeehan, "A microcellular ray-tracing propagation model and evaluation of its narrow-band and wide-band predictions," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 322–335, Mar. 2000, doi: 10.1109/49.840192.
[62] F. Aguado Agelet, A. Formella, J. M. Hernando Rabanos, F. Isasi de Vicente, and F. Perez Fontan, "Efficient ray-tracing acceleration techniques for radio propagation modeling," *IEEE Trans. Veh. Technol.*, vol. 49, no. 6, pp. 2089–2104, Nov. 2000, doi: 10.1109/25.901880.
[63] S. Hussain and C. Brennan, "An image visibility based pre-processing method for fast ray tracing in urban environments," in *Proc. 10th Eur. Conf. Antennas Propag. (EuCAP)*, Apr. 2016, pp. 1–5.
[64] S. Hussain and C. Brennan, "An efficient ray tracing method for propagation prediction along a mobile route in urban environments," *Radio Sci.*, vol. 52, no. 7, pp. 862–873, Jul. 2017, doi: 10.1002/2017rs006275.
[65] S. Hussain and C. Brennan, "An intra-visibility matrix based environment pre-processing for efficient ray tracing," in *Proc. 11th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2017, pp. 3520–3523.
[66] S. Hussain and C. Brennan, "Efficient preprocessed ray tracing for 5G mobile transmitter scenarios in urban microcellular environments," *IEEE Trans. Antennas Propag.*, vol. 67, no. 5, pp. 3323–3333, May 2019, doi: 10.1109/tap.2019.2896706.
[67] S. Hussain and C. Brennan, "A dynamic visibility algorithm for ray tracing in outdoor environments with moving transmitters and scatterers," in *Proc. 14th Eur. Conf. Antennas Propag. (EuCAP)*, Mar. 2020, pp. 1–5.
[68] R. Wahl and G. Wolfle, "Combined urban and indoor network planning using the dominant path propagation model," in *Proc. 1st Eur. Conf. Antennas Propag.*, Nov. 2006, pp. 1–6.
[69] *IEEE Standard for Information technology–Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs): Amendment to add alternate PHY*, Standard P802.15.4a/D7, 2007.
[70] F. Capolino and S. Maci, "Simplified closed-form expressions for computing the generalized fresnel integral and their application to vertex diffraction," *Microw. Opt. Technol. Lett.*, vol. 9, no. 1, pp. 32–37, May 1995, doi: 10.1002/mop.4650090113.
[71] E. Damasso and L. M. Correia, *COST Action 231 : Digital Mobile Radio Towards Future Generation Systems : Final Report*. Brussels, Belgium: European Commission, 1999.
[72] *Attenuation in Vegetation*, Interface Telecom. Union, Geneva, Switzerland, 2016. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/p/RREC-P.833-9-201609-I!!PDF-E.pdf
[73] *PERSEPTEUR Project*. Accessed: Oct. 2014. [Online]. Available: http://pagesperso.univ-brest.fr/ bounceur/anr/persepteur/
[74] A. Bounceur, O. Marc, M. Lounis, J. Soler, L. Clavier, P. Combeau, R. Vauzelle, L. Lagadec, R. Euler, M. Bezoui, and P. Manzoni, "CupCarbon-lab: An IoT emulator," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–2.
[75] A. Bounceur, L. Clavier, P. Combeau, O. Marc, R. Vauzelle, A. Masserann, J. Soler, R. Euler, T. Alwajeeh, V. Devendra, U. Noreen, E. Soret, and M. Lounis, "CupCarbon: A new platform for the design, simulation and 2D/3D visualization of radio propagation and interferences in IoT networks," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–4.
[76] Z. Xie, H. Wang, and L. Wu, "The improved douglas-peucker algorithm based on the contour character," in *Proc. 19th Int. Conf. Geoinform.*, Jun. 2011, pp. 1–5.

T. ALWAJEEH was born in Sana'a, Yemen, in 1985. He received the B.S. degree in electrical engineering from Science and Technology Jordanian University, Irbid, Jordan, in 2009, and the M.S. degree in telecommunications networks, multimedia and automatic, and the Ph.D. degree in electronics, micro-electronics, nano-electronics and micro-waves from the University of Poitiers, France, in 2014 and 2018, respectively.

Since 2019, he has been a Research Engineer with ALTAIR group. His research interests include the study of electromagnetic wave propagation for wireless communication systems in the radio frequency domain.

P. COMBEAU was born in Angoulême, France, in 1978. He received the B.S. and M.S. degrees in information processing: computer science, images, automatic, and the Ph.D. degree in electronics, optronics and systems from the University of Poitiers, in 1999, 2001, and 2004, respectively.

Since 2005, he has been an Associate Professor with the XLIM Laboratory, Optical and Radio Communication System and Network Department, Poitiers University. He is the author of one book chapter, 21 articles in international journals, and more than 60 conference papers. His research interests include the study of electromagnetic wave propagation for wireless communication systems in radio and optical frequency domains.

L. AVENEAU was born in Angoulême, France, in 1969. He received the B.S. and M.S. degrees in computer science from the University of Bordeaux, France, in 1992 and 1996, respectively, and the Ph.D. degree in computer science from the University of Poitiers, France, in 1999.

Since 2000, he has been an Associate Professor with the University of Poitiers within the XLIM Laboratory, the Computer Graphics Department, and the Optical and Radio Communication System and Network Department. He is the author of three book chapters, 18 articles in international journals and more than 50 conference papers. His research interests include visibility characterization in *n*-dimensional space, radio and light wave propagation, and the simulation of fluids.