Tracking of Interaction Points for Improved Dynamic Ray Tracing

Florian Quatresooz*, Simon Demey†, Claude Oestges‡
ICTEAM - Université catholique de Louvain - Louvain-la-Neuve, Belgium
*florian.quatresooz@uclouvain.be, †simon.demey@uclouvain.be, ‡claude.oestges@uclouvain.be

***Abstract*—Ray tracing is a powerful tool to obtain deterministic and dynamic descriptions of communication channels. However, performing ray tracing simulations at each discrete time instant is computationally expensive. Instead, a new approach is proposed to extrapolate results obtained from a single ray tracing simulation. It relies on the geometric tracking of interaction points (i.e. reflection or diffraction points), enabling analytical or numerical predictions of the evolution of any ray identified during an initial ray tracing simulation. The performance of this new approach is studied on several canonical vehicle-to-vehicle configurations, as part of a statistical study. Focus is also given to the time horizon during which dynamic ray tracing is possible, related to the lifetime of the main rays. This time horizon can be directly estimated based on the knowledge of the geometry and its evolution. It is found to be the main parameter influencing the accuracy and the computational gain of the presented approach.**

***Index Terms*—Ray Tracing, Dynamic Ray Tracing, V2V Communications, Millimeter Waves**

I. Introduction

A. Background

Deterministic models of communication channels, such as Ray Tracing (RT), are expected to assist the design of future millimeter wave wireless systems. Indeed, working at higher frequencies makes the ray-optics assumption more valid, improving the accuracy of the results obtained by RT [1]. Current trends in research about RT focus on two main topics [2]: the level of details required to accurately model an environment is studied (such as modelling trees, scattering [3], etc.), and new software approaches to reduce the computation time are developed [4]. However, these two goals are rather antagonist, as adding more details to improve the accuracy of the models also makes them more complex and more computationally intensive, leading here to an important trade-off. Recent illustrations of such a trade-off between accuracy and complexity for millimeter-wave communication can be found in [5]–[7].

Furthermore, future communication channels will become more dynamic, with fast motion (e.g. for V2V communications [8] or for train communications [9]) and shadowing due to vehicle up to 15 dB [10-11]. Real-time beamforming or channel estimation may then become more challenging, advocating for new solutions, such as directly embedding RT simulations inside wireless communication systems [1-2]. This requires RT models to be more dynamic, with stronger timing constraints.

In the present work, a possible approach to perform Dynamic Ray Tracing (DRT), with a very low computation time and a high accuracy, is investigated. The goal is to avoid performing RT simulations at each discrete time instant (i.e. Monte Carlo approach, repeating classical static RT simulations), while still being able to estimate the communication channel (the rays) at these instants. We rely on a single RT simulation and on a perfect knowledge of the environment and its time-evolution. Doing so, it is possible to predict the evolution of any ray identified by the RT simulation, thanks to the tracking of the interaction points.

B. Related Works

The idea of extrapolating results from a single RT simulation is not new. In [12], a spatial extrapolation method is presented in order to determine the electric field in the vicinity of the receiver. Regarding moving scenarios, some methods computing the evolution of the visibility relationships prior to the RT simulations have been applied. This is the case in [13], where the geometry and the visibility are updated based on the mobility of the transmitter, the receiver and the objects in the environment. After this update, classical RT simulations are performed at several snapshots. The same principle has also been presented for game engine RT of dynamic scenes, using distinct bounding volume hierarchies for static and dynamic objects [14]. Similarly, a visibility table is constructed in [15] based on the trajectory of the moving transmitter. With such a table, the complexity of the following RT simulations is reduced.

Furthermore, [16] tackles the receiver mobility using the concept of ray entity. A ray entity is a set of rays undergoing the same propagation phenomena on the same objects, as the receiver is moving. The main idea of [16] is to store the ray entities instead of all the rays at several snapshots, thus reducing the memory needed and enabling continuous interpolation of RT.

In [17], moving transmitter and receiver are considered in a static environment. Performing RT simulations separated by a given time interval, complex-valued coefficients of all multipath components of the channel are interpolated based on the displacement vectors of the transmitter and the receiver. Discussions about the time resolution at which RT simulation must be performed are also briefly introduced.

Finally, dynamic ray tracing has already been investigated,

Manuscript received October 13, 2020; revised February 26, 2021; accepted May 10, 2021. Digital Object Identifier 10.1109/TVT.2021.3081766

namely to obtain Doppler shift profiles [8-18-19]. However, most of these works do not take into account the possible sliding of interaction points on surfaces [20]. In [20], a new extrapolation model based on a single RT simulation is presented. It provides a description of the delay and the Doppler shift of all identified rays, during the lifetime of the main multipath components of the channel. However, the main drawback of the approach is related to the creation of new rays due to the dynamic evolution of the environment.

Here, the following contributions are added:

- The fully deterministic (and analytic) characterization of the motion (i.e. the sliding) of any interaction point previously identified, giving all the necessary mathematical expressions. The framework presented here enables to support fully dynamic environments, with moving transmitter and receiver as well as moving obstacles. The presented methodology for characterizing the motion of interaction points is tool agnostic and can use the outputs of any initial RT simulation.
- The generalization to multiple interactions, enabling to find directly the new rays with all information about the communication channel (delay, phase, power, angles, etc.). Those outputs are similar to the ones obtained using the ray entity concept presented in [16] but the approach is different and can handle more than two interactions and scenarios where not only the receiver is moving.
- The characterization of the extrapolation time between RT simulations thanks to a statistical study. Going further than in [17], some simple rules giving the extrapolation time for a specified dynamic environment are presented and discussed. The trade-off between the accuracy of our DRT approach and its computational gain with respect to classical RT simulations is also explored.

Thereby, the DRT algorithm presented here offers a geometric computational gain (i.e. comparing the simulation time needed to identify all rays at a given instant by RT or by using the DRT approach previously initialised with a RT simulation) of three orders of magnitude with negligible error (of the order of nanometer) on the positions. It is also shown that the real computational gain arising from the spared RT simulations and the loss of accuracy due to missing rays in DRT are strongly linked to the extrapolation time.

C. Structure of the paper

In Section II, the DRT approach based on the tracking of interaction points is presented. Then, Section III focuses on possible implementations of the new approach, inside a RT tool, and Section IV validates those implementations on simple examples. Finally, a study of the extrapolation time is performed in Section V.

Notation: Throughout this work, $\boldsymbol{a}$ denotes a vector, having for components $(a_x, a_y, a_z)^T$.

II. Description of DRT

To extrapolate the geometry of a given ray, one needs to characterize the time-evolution of the positions of any of the interaction points constituting this ray. From the geometry of the problem, the following quantities are known, at any time:

- the position and velocity of the transmitter and the receiver;
- the position and velocity of any object (buildings, vehicles, etc.) in the environment.

All velocities can be time-varying, even though it will be often assumed in this work that all velocities are constant during the channel stationarity time, i.e. the time during which the main multipath components (MPCs) are related to the same rays. Furthermore, rigid-body rotations or deformations are not allowed.

Then, from a unique RT simulation at time t_0, it is possible to extract the positions of all interaction points, including all reflection and diffraction points. Based on their initial positions, their motion can be fully characterized. The equations stating the positions and the velocities of reflection points for a horizontal plane are first derived, before being generalized to oblique and multiple planes. Next, a similar approach is applied to diffraction points. The main contributions in this section are the presented generalization approaches, in Sections II-A3 and II-B3, as well as the computation of the image point velocities.

A. Tracking of reflection points

1) Horizontal plane: A simple 3D case is considered, with a horizontal plane Π, a transmitter and a receiver located at different heights. The transmitter and receiver points are located respectively in $(x_{TX}, y_{TX}, z_{TX})^T$ and $(x_{RX}, y_{RX}, z_{RX})^T$. The plane equation is given by $\Pi : y = y_w$ with $x, z \in \mathbb{R}$ and y_w being a given (real) constant denoting the height (along y-axis) of the horizontal plane with respect to the origin. The geometry is depicted in Figure 1. Using the image method, the reflection point (IX) is determined and is located at the intersection between the plane and the line passing through the receiver (RX) and the image of the transmitter (TX'). The equation of this line is:

$$d: \ \frac{x - x_{TX'}}{x_{RX} - x_{TX'}} = \frac{y - y_{TX'}}{y_{RX} - y_{TX'}} = \frac{z - z_{TX'}}{z_{RX} - z_{TX'}} \quad (1)$$

with the transmitter image point position given by $(x_{TX'}, y_{TX'}, z_{TX'})^T = (x_{TX}, 2y_w - y_{TX}, z_{TX})^T$. Since the reflection point IX belongs to the plane, the expression of its position is:

$$\begin{cases} x_{IX} &= x_{TX} + \left(\frac{x_{RX}-x_{TX}}{y_{RX}-2y_w+y_{TX}}\right)(y_{TX} - y_w) \\ y_{IX} &= y_w \\ z_{IX} &= z_{TX} + \left(\frac{z_{RX}-z_{TX}}{y_{RX}-2y_w+y_{TX}}\right)(y_{TX} - y_w) \end{cases} \quad (2)$$

Deriving these last expressions with respect to time gives the instantaneous velocity of the reflection point. Assuming a purely vertical motion for the plane (with speed $v_{w,y}$) and constant velocities for the transmitter and the receiver ($\boldsymbol{v}_{TX}$ and $\boldsymbol{v}_{RX}$), the x-velocity of IX is given by the chain rule:

$$v_{IX,x} = \frac{\partial x_{IX}}{\partial x_{TX}} v_{TX,x} + \frac{\partial x_{IX}}{\partial y_{TX}} v_{TX,y} + \frac{\partial x_{IX}}{\partial x_{RX}} v_{RX,x} + \frac{\partial x_{IX}}{\partial y_{RX}} v_{RX,y} + \frac{\partial x_{IX}}{\partial y_w} v_{w,y} \quad (3)$$

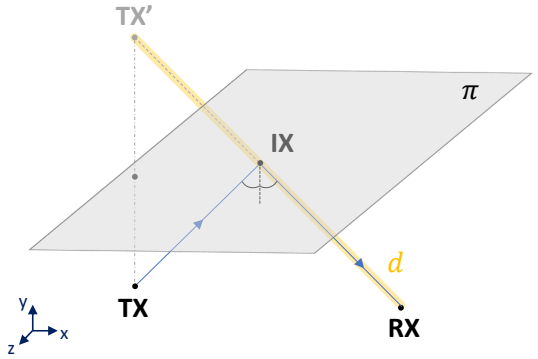


Fig. 1: Horizontal plane - TX *and* RX *are below the plane.*

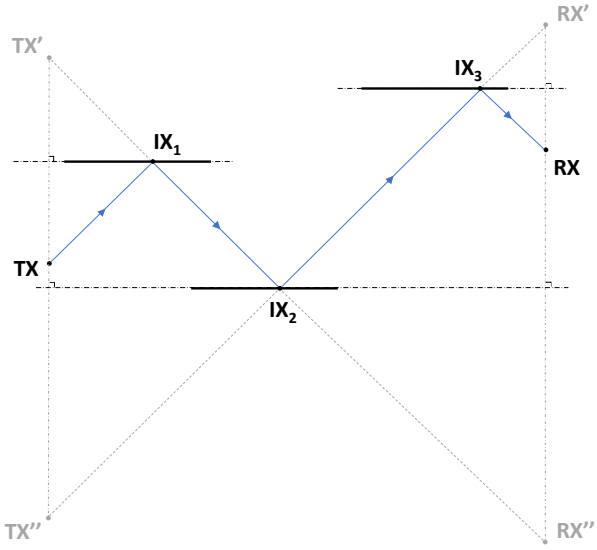


Fig. 2: Example of generalization to three reflections, in 2D

with all derivatives with respect to the positions that can be computed from (2). Similarly, the z-velocity is computed, whereas the y-velocity of IX is given by $v_{w,y}$.

Therefore, the instantaneous velocity of the reflection point depends on the velocities of the transmitter, receiver and the plane, as well as on their instantaneous positions. *This velocity is thus not constant even if the TX, RX or the plane velocities are assumed to be constant.*

2) Oblique plane: Oblique planes can be handled using changes of coordinate systems thanks to rotation matrices, so as to end up in the local frame associated to the plane (i.e. the frame where the normal is aligned with the y-axis, as considered in the previous section).

3) Generalization to multiple reflections: In order to compute the motion of each interaction point independently (applying (2)), we suggest to use the corresponding images of the transmitter and the receiver as TX and RX for the plane under consideration. This means, for example, that TX' and RX' are used in Figure 2 to characterize the motion of IX_2. The instantaneous positions (and velocities) of TX and RX images is found thanks to the knowledge of TX and RX initial positions and velocities, as well as the velocities of all planes

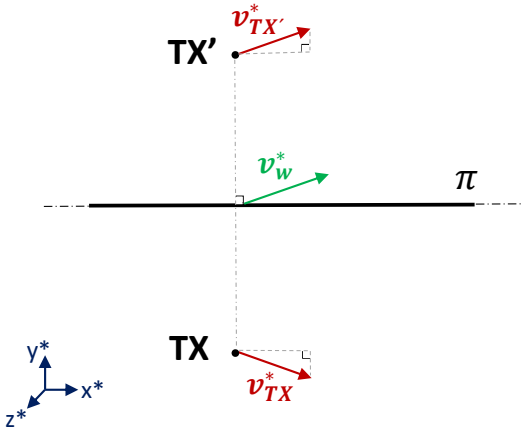


Fig. 3: Velocity of TX image point

used to construct the image points.

With the assumption of constant velocities, the velocities of the image points are also constant and can be computed once, during the first RT iteration. If not, they must be recomputed at any instant. In both cases, using geometrical considerations illustrated in Figure 3 and reasoning in the local frame associated with the plane (i.e. after having applied the rotation matrix $\mathbf{M}$ of the plane), the velocity of the TX image point is given by:

$$\begin{cases} v^*_{TX',x^*} &= v^*_{TX,x^*} \\ v^*_{TX',y^*} &= 2v^*_{w,y^*} - v^*_{TX,y^*} \\ v^*_{TX',z^*} &= v^*_{TX,z^*} \end{cases} \tag{4}$$

with $[.]^*$ denoting the local frame coordinates ($\boldsymbol{v}^*_{TX} = \mathbf{M}\boldsymbol{v}_{TX}$ and $\boldsymbol{v}^*_w = \mathbf{M}\,\boldsymbol{v}_w$). After that, the inverse rotation matrix is applied to $\boldsymbol{v}^*_{TX'}$ to get back to the initial frame. Similarly, velocities of RX image points are found and this approach can be used successively if several planes are considered.

Finally, the positions of the image points at a given instant t are obtained from the initial positions using the laws of linear motion.

B. Tracking of diffraction points

In a similar way to what has been done for the tracking of reflection points, the equations describing the positions of diffraction points are derived.

1) Canonical configuration: Following Keller's law of diffraction [21], the diffraction by an edge is depicted in Figure 4. In this canonical configuration, the edge vector $\hat{\boldsymbol{e}}$ is assumed to be aligned with the y-axis, i.e. $\hat{\boldsymbol{e}} = (0, 1, 0)^T$. TX and RX positions are known, as well as the x- and z-position of the diffraction point IX, since the diffraction point must be located on the edge. Therefore, we define $x_{IX} = x_e$ and $z_{IX} = z_e$, both known. Only y_{IX} remains to be computed, using the theory of similar triangles.

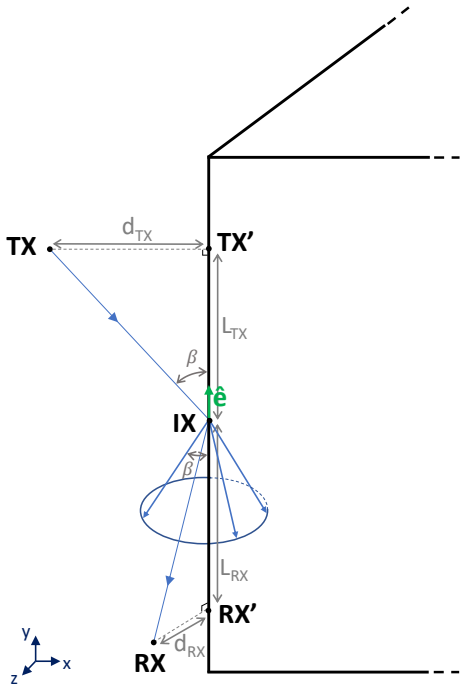


Fig. 4: Diffraction by an edge

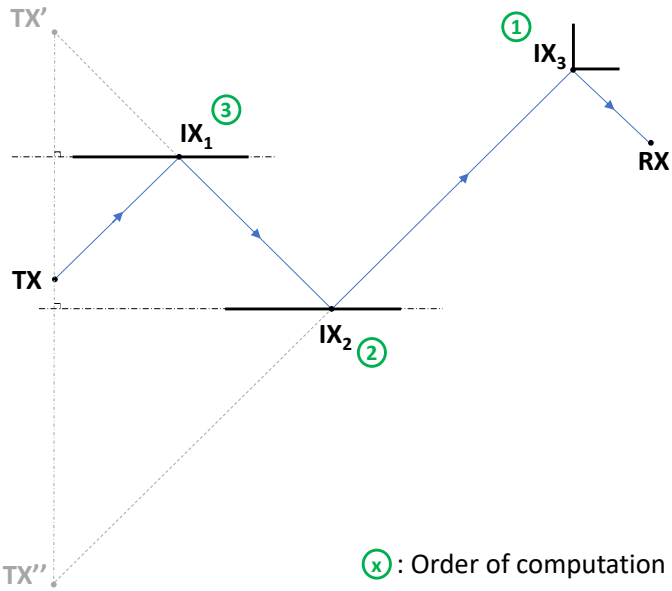


Fig. 5: Generalization for diffraction - *First,* IX_3 *is determined thanks to* RX *and* TX". *Then,* IX_2 *can be found using* TX' *and* IX_3. *Finally,* IX_1 *is obtained with* TX *and* IX_2.

Indeed, the two triangles formed in Figure 4 are similar and share the following property in terms of distance ratios (distances are defined in Figure 4):

$$c := \frac{d_{TX}}{d_{RX}} = \frac{L_{TX}}{L_{RX}} = \frac{y_{TX} - y_{IX}}{y_{IX} - y_{RX}} \tag{5}$$

with $d_{TX} = \sqrt{(x_{TX} - x_e)^2 + (z_{TX} - z_e)^2}$ and d_{RX} found similarly. Hence, the y-position of the diffraction point is given by, with c computed knowing d_{TX} and d_{RX}:

$$y_{IX} = \frac{y_{TX} + c\, y_{RX}}{c+1} \tag{6}$$

Time derivation of (6) to obtain the instantaneous velocity of the diffraction point is not trivial and has not been further studied.

2) Oblique edge: As for reflections, if the edge is not oriented along the y-axis, a rotation matrix is first applied in order to end up in the local coordinate system considered previously.

3) Generalization: It is assumed that diffraction is only possible at the last interaction, as this drastically reduces the complexity of UCLouvain RT software [22-23]. In this case, image points of the transmitter can only come from reflections and there are no receiver image points. Hence, the computation of all interaction points is performed successively, as depicted in Figure 5. This approach can also be generalized to one diffraction at any interaction, by recursively computing the TX and RX image points up to the diffraction, finding the diffraction point and then working backwards. For example, for a ray with three interactions in which there is a diffraction at the second interaction, the images of the transmitter and the receiver are first computed (TX' and RX'). Then these latter are used to determine the position of the diffraction point, i.e. IX_2. Knowing IX_2 and the TX-RX image points, the reflection points IX_1 and IX_3 are found.

However, generalization to multiple diffraction in a single ray remains an open question. Generalization to more complex geometries as well as to other propagation phenomena will also be part of future work. Similar approaches to the ones presented here can be used, relying on the characterization of the motion of interaction points.

III. Implementation

Dynamic ray tracing is considered as a post-processing approach of an initial RT simulation. Indeed, based on the previous section, it is possible to track the positions of the interaction points, if, during an initial RT simulation, all the following quantities are recorded for each ray:

- Position and velocity of TX;
- Position and velocity of RX;
- For each wall/edge i interacting with the ray:
  - Reflection/diffraction point position (IX_i);
  - Velocity of the wall/edge ($\boldsymbol{v}_w$);
  - Position and velocity of corresponding image of TX;
  - Position and velocity of corresponding image of RX.

Then, for each wall/edge, relations from Section II-A can be applied *independently* (for reflection points) or *recursively* (for diffraction and reflection points).

In this section, two different approaches have been identified to evaluate the positions of the interaction points: either a step-by-step (numerical) integration, or a fully analytical integration. Both rely on different hypotheses.

A. Step-by-step (numerical) integration

This approach is particularly suited for geometries where the velocities are time-varying and especially when one does not have access to analytical expressions of the temporal variations of the velocities, but rather access to the velocity values at some discrete instants.

1) Closed-form for reflection only: If only reflections are considered, one can directly work on the positions of the reflection points IX and compute $\boldsymbol{x}_{IX}(t+\Delta t) = \boldsymbol{x}_{IX}(t) + \boldsymbol{v}_{IX}(t)\,\Delta t$ with $\boldsymbol{v}_{IX}(t)$ computed thanks to (3), using the velocities of TX, RX and the obstacles at time t. This leads to an iterative algorithm, starting from the results of the RT simulation and updating the positions of the interaction points every Δt.

However, this implementation cannot be used with diffraction, the main reason being that the presented generalization for diffraction with multiple interactions does not allow independent computations for interaction points. Moreover, a closed-form for the velocity of a diffraction point has not been derived.

2) With analytical substitution, for reflection and diffraction: Here, the new positions of the TX, RX or their image points are first computed by numerical integration. For example, using $\boldsymbol{x}_{TX}(t+\Delta t) = \boldsymbol{x}_{TX}(t) + \boldsymbol{v}_{TX}(t)\,\Delta t$ for TX position. Then, these positions are substituted in (2) and (6) to determine the position of the interaction points.

The numerical integration can be applied even if the velocities are not constant. However, for geometries with constant velocities, this approach is equivalent to the analytical approach explained next.

B. Analytical approach

If analytical expressions of the time-variations of the velocities are known, the time dependency can be made explicit in (2) and (6) using, for example, $\boldsymbol{x}_{TX}(t) = \boldsymbol{x}_{TX}(t_0) + \int_{t_0}^{t} \boldsymbol{v}_{TX}(t')dt'$ for the transmitter. Hence, if constant velocities are assumed, the integration is straightforward.

C. Creation and suppression of rays

Since the DRT approach is based on an initial RT simulation for identifying the existing rays, rays that are not present in this preliminary RT simulation will not be found by the DRT algorithm. This happens especially when new rays are created due to the dynamic evolution of the geometry. The effect of this creation of rays is further studied in Section IV-B.

Furthermore, all equations obtained in Section II assumed the planes or the edges to be infinite, which is not the case in practice. Therefore, the computation of the new IX positions must be compared with the evolution of the geometry. Indeed, the computed IX positions may end up falling out of the surface on which a reflection was possible before, or out of the edge where there was a diffraction. In this case, the whole ray should be suppressed since it does not exist anymore and will not be found by the classical RT algorithm. This verification is implemented as a ray-suppression mechanism.

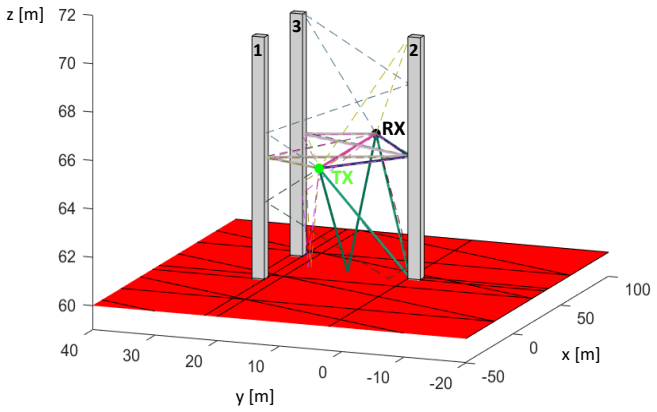


Fig. 6: Three-obstacle configuration: rays at initialisation - *diffraction rays are in dashed lines*

Finally, rays may also disappear because of blockages, i.e. when moving obstacles start blocking rays that previously existed. In the current approach, this is not verified dynamically. This means that not only newly created rays but also blocked rays will impact the accuracy of the DRT approach. This impact is studied on some channel metrics in Section V.

D. Electric field values

The presented DRT approach is performed at the geometry-level, having for output the geometry of the rays at an given instant. This means that equations from geometrical optics (GO) and the uniform theory of diffraction (UTD) must be used afterwards to compute the received electric fields [24]. This is no different than the last part of any RT software, that first identifies the rays geometrically and then computes their respective electric field values.

IV. Validation on simple examples

The presented approach is validated on two main examples and in both cases, constant velocities for all obstacles in the geometry are assumed. This validation has been performed by comparing the outputs of UCLouvain original RT software [22-23] and the outputs of the new DRT software. Outputs of RT simulations are considered as ground truth for the comparison with the DRT outputs in order to establish the performances of the DRT methodology (in terms of accuracy and complexity) and validate it.

A. Three-obstacle configuration

UCLouvain RT software has been used [22-23], with the maximal number of interactions set to three. The studied configuration, along with the rays identified at initialisation, is depicted in Figure 6. The transmitter is given a velocity $(0.1, -0.2, 0)^T$, and the receiver $(0, 0.5, 0)^T$. The obstacles in the environment are also moving, having for velocities $\boldsymbol{v}_1 = (1, 1, 0)^T$, $\boldsymbol{v}_2 = (0, -0.5, 0)^T$ and $\boldsymbol{v}_3 = (0, 0, 1)^T$, all in metre per second. The ground is depicted in red.

Based on this initial configuration, the DRT approach is used to characterize the ray geometry every $\Delta t = 0.1$ s, up to

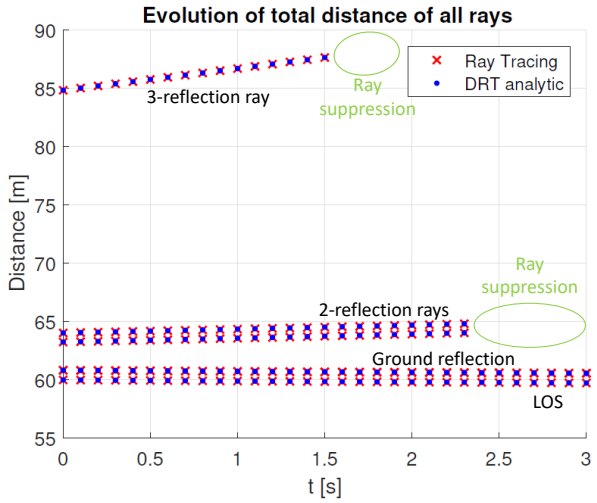


Fig. 7: Evolution of total distance of all reflection rays (for three-obstacle configuration)

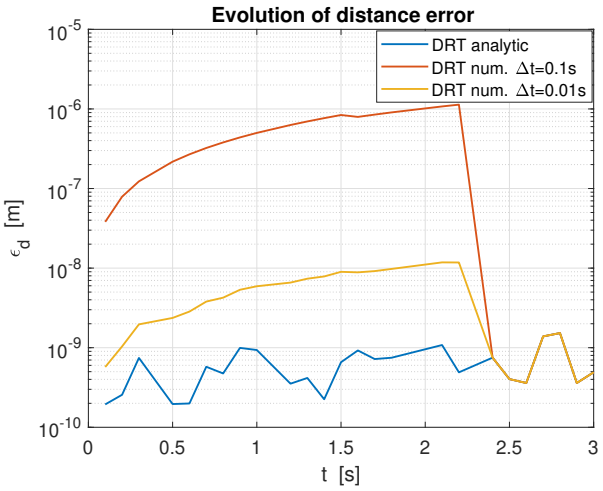


Fig. 8: Evolution of distance error, considering only reflection rays (for three-obstacle configuration)

a time horizon of 3 seconds. Results are compared with the ones obtained by RT simulations at these instants, in terms of total distance travelled by each ray, as given in Figure 7 (only reflection rays). There seems to be a total agreement between the distances computed based on RT simulations and the ones from the DRT approach since the points are superimposed. This agreement is quantified using the mean of the absolute distance errors on all rays, defined by

$$\epsilon_d(t) \;=\; \frac{1}{N_{\text{ray}}} \sum_{n=1}^{N_{\text{ray}}} \left| d_n^{RT}(t) - d_n^{DRT}(t) \right|, \tag{7}$$

which is depicted in Figure 8. N_{ray} is the total number of rays (which also depends on time), d_n^{RT} (resp. d_n^{DRT}) is the total distance of ray n computed from RT simulations (resp. from DRT simulations). Two possible implementations of DRT are compared: the analytical approach and the numerical integration using the closed-form (since only reflection rays are considered here).

In Figure 8, an expected error propagation mechanism for the numerical integration approach is observed. This comes from the assumption of constant velocity for the reflection points used in the numerical integration approach ($\boldsymbol{v}_{IX}(t)$ assumed constant during Δt) which is not verified, as already stated at the end of Section II-A1. Moreover, the higher the value of Δt, the less accurate is this assumption and therefore the higher the error.

One can also notice that most of the error is coming from the rays having a reflection on the second obstacle, as the error drops drastically when these rays are suppressed. This is also a consequence of the constant velocity assumption for the reflection points, which is not verified for the two-reflection rays. However, in this particular example, the application of (3) for the ground reflection point leads to a constant velocity v_{IX}. This explains why, at the end of the simulation (i.e. when only the ground reflection and the LOS rays remain), results between numerical and analytical DRT are superimposed, for any value of Δt.

In addition, the best achievable accuracy is obtained thanks to the analytical approach and is close to the nanometer. Infinite accuracy is not reached, owing to the accuracy level of the RT software (10^{-10} meter). Nevertheless, this nanometer accuracy is, on average, the ground level for all rays. Indeed, even when some rays are suppressed, the accuracy remains close to the nanometer with the analytical approach.

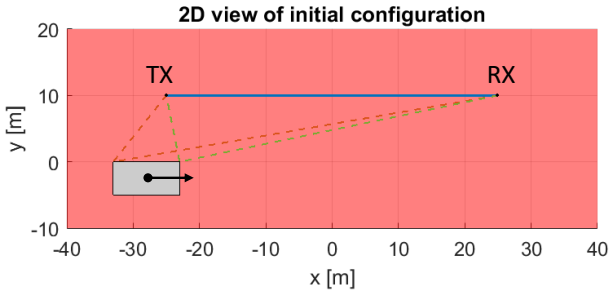
Considering diffraction rays, only the analytical approach can be used[1]. We found that the measured accuracy is again close to the nanometer, the average value of ϵ_d being 5.51×10^{-10} m, whereas the maximum observed value is 8.97×10^{-10} meter.

Nonetheless, the presented example does not involve the creation of new rays. This explains why DRT results will always be superimposed to RT results, during a theoretically infinite extrapolation time.

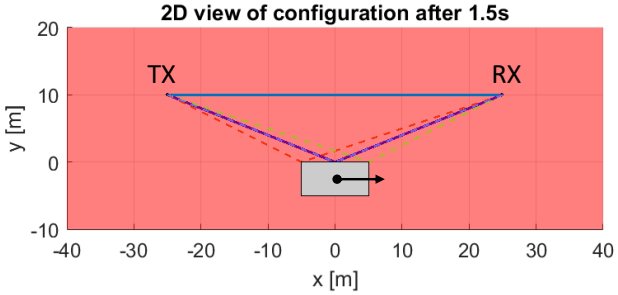
B. One moving obstacle and ray creation

On the contrary, the simplest example involving the creation of new rays is considered in this section, its geometry being given in Figure 9 (RT simulation outputs). The transmitter is located in $(-25, 10, 5)^T$ and the receiver is in $(25, 10, 5)^T$. Both are kept fixed. The ground level is at $z = 0$ m and there is only one moving obstacle (of a height of 10 meters) in the environment. The obstacle is initially located 10 meters away from the transmitter in the y-direction and is moving towards the receiver in the x-direction (its velocity vector is $(20, 0, 0)^T$). After 3 seconds, the obstacle is now located next to the receiver (symmetric situation w.r.t. the initial configuration) and the simulation is completed. *Analytical DRT* is performed every 0.02 second, always based on the RT

[1]Since constant velocities are assumed in the geometry, the numerical integration with analytical substitution falls back to the analytical approach, whereas the closed-form numerical integration can only be used with reflection rays.



(a) Rays at initialisation



(b) Rays after 1.5 seconds, with a new reflection ray on the obstacle (in purple)

Fig. 9: Geometry and rays configuration for the example with one moving obstacle *(diffraction rays are in dashed lines)*

simulation at time $t = 0$ s. For the comparison, RT simulations are performed every 0.1 s and channel quantities are compared. The chosen carrier frequency is 1.8 GHz, and the effective relative permittivity (of the ground and the obstacle) is set to $\epsilon_r^{eff} = 4.44 - j\,0.01$ [25-26].

As expected, discrepancies between channel metrics computed using RT or DRT are noticed, owing to the new rays. This is observed in Figure 10, depicting the evolution of the RMS delay spread. The new reflection rays appear between 1.25 and 1.75 seconds of simulation time, drastically impacting the delay spread.

Therefore, an important quantity taking into account the impact of the forgotten rays must be defined: the extrapolation time. This latter corresponds to the maximum time interval during which one can perform DRT simulations and still get a sufficiently *good knowledge* of the channel. After the extrapolation time, it will be recommended to reperform a classical RT simulation in order to find new rays that have been created or to suppress rays that have been blocked. This new RT simulation can then be used as a new initialisation for the DRT approach.

In this simple example with one moving obstacle, the extrapolation time is directly related to the lifetime of the new rays. This lifetime is given by the ratio between the width d_w and the velocity v_w of the moving obstacle. Numerical values for $d_w = 10$ meters and $v_w = 20$ m/s lead to a ratio $\frac{d_w}{v_w}$ of 0.5 second. This corresponds to the duration of the time interval between 1.25 and 1.75 seconds of simulation time where the new rays are present. Taking an extrapolation time

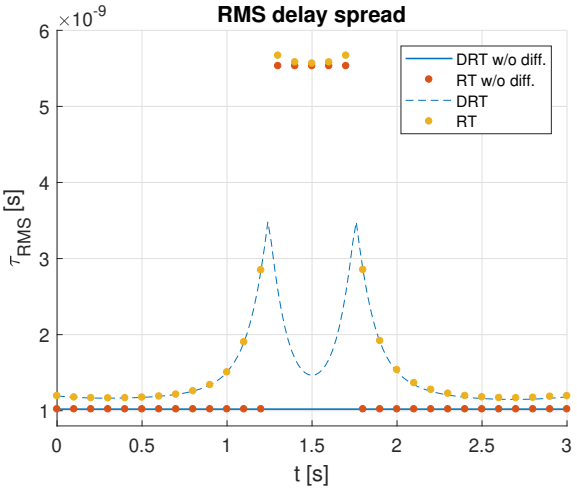


Fig. 10: Evolution of RMS delay spread (for one moving obstacle configuration)

of 0.5 second means that the new reflection rays will at least be found at one given instant, since a RT simulation will be performed in the time interval between 1.25 and 1.75 seconds.

V. Statistical Study of Extrapolation Time

For our DRT approach to be complete, one must know beforehand the extrapolation time that must be used. We know that this time is related to the lifetime of the rays, especially the strongest rays (usually reflection rays). However, in the case of complex geometries involving several moving obstacles, it is not easy to obtain an expression of the accurate lifetime of the rays. This is especially true because there will be multiple reflection rays, having their own lifetime. Therefore, the following rule has been suggested as an approximation of an upper bound for the extrapolation time:

$$t_{ext} \leq \frac{d_{min}}{v_{max}} \tag{8}$$

where d_{min} is the minimal length (or width or height, i.e. the smallest dimension) of the obstacles in the geometry on which a reflection may occur during the dynamic evolution of the geometry, and v_{max} is the maximal amplitude of all velocities in the geometry.

Equation (8) leads to two important observations:

- It does not provide a real upper bound for the extrapolation time, or equivalently a lower bound for all (reflection) rays lifetime. Indeed, one must actually use the *relative velocity* (with respect to the wall) of the reflection points at the denominator and not an absolute velocity. In a general case, finding these relative velocities or even an upper bound is difficult and not practical since it implies to take vectorially into account the direction of motion of all obstacles and of TX and RX. However, in a scenario with 1D motion for example, it is possible to upper bound the maximal relative velocity by twice the maximal velocity v_{max}, i.e. $v_{rel,max} \leq 2\,v_{max}$.
- Upper bounding the extrapolation time ensures that all reflection rays will be found at least at one instant by

using a new RT initialisation. Nevertheless, it does not give any confidence about the instant where the new RT initialisation will occur: it can be at the beginning of the apparition of the new rays, or at the end, and this strongly depends on the initial configuration of the geometry. A possible solution to make sure to find the new rays at the beginning of their apparition would be to use 10% of the value given by (8) for example, so that it can be expected that each ray will be found for at least around 90% of their lifetime. However, this is a rather pessimistic and strong constraint since few or no rays will have a lifetime equals to or lower than $\frac{d_{min}}{v_{max}}$.

Following these observations, some simple rules giving the extrapolation time based only on the knowledge of the geometry and its evolution have been suggested and are grouped in Table I.[2]

	Rule A	**Rule B**	**Rule C**	**Rule D**
t_{ext}	$\frac{1}{10}\frac{d_{min}}{v_{max}}$	$\frac{1}{2}\frac{d_{min}}{v_{max}}$	$\frac{d_{min}}{v_{max}}$	$\frac{1}{4}\frac{d_{min}}{v_{max}}$

TABLE I: Considered rules for the extrapolation time

These are *rules of thumb* easy to apply and that do not require a deep analysis of the geometry. Hence, they provide a coarse approximation of the ray lifetime. In order to determine which rules to use, a statistical study on 60 RT simulations has been conducted. Three typical dynamic environments have been considered: a *city street*, a *city crossroad* and a *highway*. These environments are depicted in Figures 11 to 13, where the velocity vectors have been added. In each of these configurations, several parameters have been modified in order to perform 20 different RT simulations per configuration. This involves modifications in the transmitter, receiver, vehicle and wall initial positions and velocities.

The simulation parameters used in all configurations are given in Table II. Vehicles have been modeled by rectangular cuboids (5 meters length, 2 meters width and 2 meters height for cars; 10 meters length, 2 meters width and 4 meters height for trucks/buses). They have been assumed to be made of metal, having an effective relative permittivity of $\epsilon_r^{eff} = 4.5 - j\,4 \times 10^8$ [8]. The receiver antenna is a dipole, whereas the transmitting antenna is assumed isotropic. The time interval of simulation ranges from 0 to 3.5 seconds for all configurations. RT and DRT simulations are performed and compared every 0.01 second.

Carrier frequency [GHz]	1.8
Effective relative permittivity of bricks	4.44 - j 0.01
Effective relative permittivity of ground	3 - j 0.021
Maximal number of interactions	2

TABLE II: Simulation parameters for statistical study *(Permittivities based on [25]–[27])*

A. Study of channel-based metrics

There is no point at comparing geometric quantities obtained by RT or DRT since some rays will be missing. Instead,

[2]Rule D is an extra rule that was found to be useful for configurations dominated by diffraction.

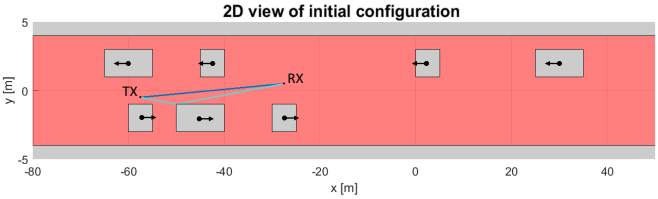


Fig. 11: Upper view of one example of the *city street* configuration (with all reflection rays, diffraction rays not depicted) - *scales in x- and y-axis are different*

channel metrics are computed based on RT or DRT rays and compared. The considered channel metrics are the delay spread, the azimuth and elevation angle spreads as well as the K-factor and the total received power. Given a certain rule for the extrapolation time, the percentage of simulation time where the relative error on a given channel metric is below 20% has been recorded. For example, regarding the delay spread, the percentage of time where

$$\left| \frac{\tau_{RMS}^{RT}(t) - \tau_{RMS}^{DRT}(t)}{\tau_{RMS}^{RT}(t)} \right| \leq 0.2 \tag{9}$$

holds is computed, with $\tau_{RMS}^{RT}(t)$ the delay spread obtained using RT and $\tau_{RMS}^{DRT}(t)$ obtained using DRT, at time t. As stated previously, RT and DRT simulations have been performed every 0.01 second (t), therefore the time percentage results from counting the number of discrete time instants t where (9) is true.

On the one hand, the choice of using a relative error is motivated by the dependency of the values of the channel quantities with the carrier frequency. Indeed, the relative power of the different rays depends on the carrier frequency since this latter influences the relative permittivity that itself modifies reflection and diffraction coefficients [27]. On the other hand, fixing the acceptable relative error level at 20% is mostly based on observations of the variations of the channel metrics. As a matter of fact, it has been observed that the creation of important rays, such as the LOS or main reflection rays, has a strong impact on the metrics, whereas the creation of diffraction rays or weak reflections has a lower impact, usually below 20%. Moreover, since RT is based on the geometrical optics assumption, it is in itself an approximation of the reality with some errors. Therefore, allowing an extra 20% variation is considered acceptable.

The mean and the standard deviation of this time percentage have been computed and are given in Tables IV to VIII, in Appendix. For all cases, *Rule A* led to extrapolation times less than or equal to 0.01 second, which is the timestep at which RT and DRT simulations are performed. This explains why this rule provides a perfect accuracy since each RT simulation is used to feed the DRT approach. However, reducing the timestep to study *Rule A* more accurately was not computationally affordable, i.e. this rule is too constraining.

Notwithstanding this observation, the general trend is the following: the longer the extrapolation time, the lower the

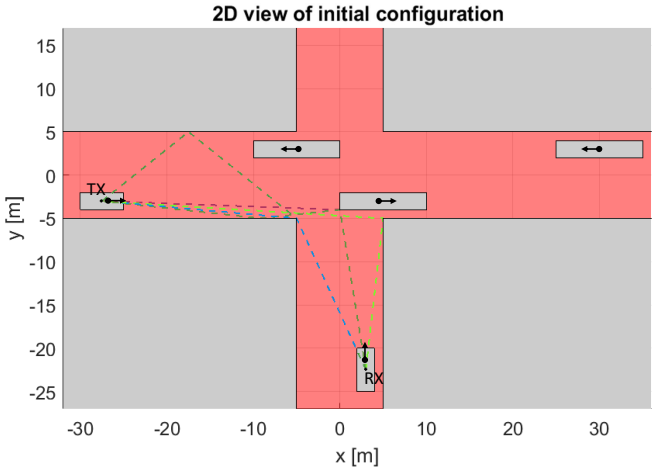


Fig. 12: Upper view of one example of the *city crossroad* configuration (no reflection rays, diffraction rays depicted)

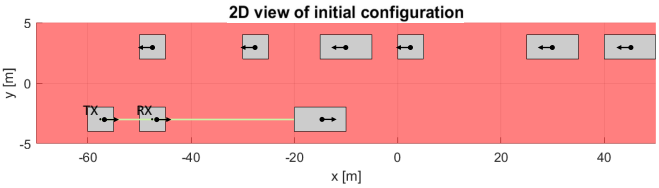


Fig. 13: Upper view of one example of the *highway* configuration (with all reflection rays, diffraction rays not depicted) - *scales in x- and y-axis are different*

mean percentage of time where the relative error is below 20% and the larger the standard deviation.

Moreover, regarding the *city street* and *highway* configurations, *Rule B* seems to achieve the best trade-off between having the longest possible extrapolation time while ensuring accurate results for more than 90% of the time. It also provides the lowest standard deviation (among the rules studied and excepting *Rule A*). However, for the *city crossroad* configuration, *Rule B* does not offer the 90% of time accuracy but this latter can be achieved by following *Rule D*. This means that the extrapolation time should be further reduced, probably owing to the prevalence of diffraction in this configuration. Indeed, there are mainly diffraction rays linking TX and RX in the *city crossroad* configuration, with rare LOS or reflection rays (see Figure 12). Since the presented rules have been deduced based on the lifetime of reflection rays, it is not surprising that they must be slightly adapted in geometries dominated by diffraction.

As a result, *Rule B seems to be the most promising rule, except in geometries dominated by diffraction* where it is suggested to further reduce the extrapolation time.

B. Computational gain

For all simulations, the execution times for RT and DRT have been recorded. They were found to vary from one simulation to another, depending on the current load of the shared computer used and the complexity of the geometry.[3]

	City street		City crossroad		Highway	
	Mean	Std	Mean	Std	Mean	Std
DRT gain	1405	510	1546	439	5401	2044

TABLE III: Geometric computational gain C_G thanks to DRT

The geometric computational gain C_G compares only the execution times needed to obtain only the geometry of the rays from a RT simulation or from a DRT simulation (assuming that a RT initialisation has been performed previously but not taking into account the time for this RT initialisation), hence its name. By definition, this gain C_G is thus independent of the extrapolation time. Its mean and standard deviation are given in Table III. It can be observed that this gain reaches on average three orders of magnitude and mainly comes from the saving of the polygon visibility computation and the reflection tree that is part of the RT. Regarding the *highway* configuration, the gain is even larger owing to the small number of rays. In this case, very few computations are performed by DRT (since the number of computations in DRT is proportional to the number of rays), whereas whatever the number of rays, RT needs to construct the reflection tree and evaluate the visibility [23]. Modifying slightly the initial configuration leads to the creation of new rays (e.g. apparition of a LOS), explaining why the standard deviation can be quite large.

Nevertheless, the real, and effective, computational gain C_R also takes into account the computation time for the RT initialisations as part of the DRT execution time, as well as the computation of the electric fields based on the rays (using GO). This gain is expected to increase with the extrapolation time since a longer extrapolation time leads to fewer RT initialisations, reducing the overall time of the DRT simulations. In fact, assuming that the DRT computation time is negligible compared to the RT execution time, an upper bound for the real computational gain (noted C_R^{th}) is:

$$C_R < C_R^{th} = \frac{t_{ext}}{\Delta t}, \tag{10}$$

with Δt the time interval between two simulations where one wants to know the channel. Equation (10) results from the number of spared RT simulations between performing RT simulations at each Δt or at every t_{ext} seconds. A similar discussion is presented in [17].

In practice, the theoretical C_R has not been reached. Given the large geometric gain C_G, neglecting the computation time for the ray geometry, as assumed in (10), is not an issue. However, the time necessary for the recomputation of the electric field values based on the DRT ray geometry was not found to be negligible. This is a consequence of the diffraction model used (Guevara's coefficients, based on [28]), involving the computation of Fresnel integrals. Hence, to fully benefit from the DRT computational gain, further improvements or

[3]For example, in the *city street* configuration, the computation time for the RT simulations is above 6000 seconds whereas the DRT simulations (without RT initialisation) are performed in close to 4 seconds (using an Intel Xeon E5-2630 v3 @2.40GHz).

modifications to the diffraction model in our current DRT model must be carried out. Neglecting the computation time of the diffracted fields, the measured computational gain is close to the theoretical bound (see Table IX in Appendix).

VI. Conclusion

An approach to perform dynamic ray tracing, i.e. to predict the evolution of the rays in a geometry evolving with time, has been presented. It relies on a single ray tracing simulation and enables to track the motion of interaction points constituting any ray up to a given time horizon.

Two solutions to implement the tracking have been proposed, either a numerical approach particularly suited for geometries with time-varying velocities, or an analytical method, faster and more accurate but only possible if analytical expressions of the time for the velocities are available. Both approaches have been validated geometrically as well as at the channel-level on a few examples.

Focus has also been given to the time horizon during which DRT can be applied, i.e. the extrapolation time. Few rules have been derived in order to determine this time beforehand, based only on the knowledge of the geometry and its evolution. The validity of those rules has been established thanks to a statistical study on several RT simulations in typical urban environments. Based on this study, it is suggested to upper-bound the extrapolation time by $\frac{1}{2}\frac{d_{min}}{v_{max}}$ (***Rule B***) to ensure a relative error smaller than 20% on most channel metrics for more than 90% of the simulation time, in environments that are not dominated by diffraction (see Section V).

All in all, the presented approach offers a geometric accuracy close to the nanometer (i.e. the accuracy level of UCLouvain RT software) as well as a geometric computational gain of three orders of magnitude, only depending on the number of rays in the geometry. However, the overall gain, that involves the recomputation of the electric field quantities based on the known ray geometry, has been found to be strongly limited by the diffraction model used.

Future work can focus on some limitations of the presented DRT approach, such as the addition of more propagation phenomena, the addition of new types of motions (e.g. rotations), the handling of more complex geometries or the generalization to multiple diffraction in a single ray. The validity of the presented *Rules* for the extrapolation time can then be evaluated on more complex scenarios, as well as the computational gain. Finally, future work will also assess the validity of the constant velocity assumption in real scenarios.

References

[1] F. Fuschini, E. M. Vitucci, M. Barbiroli, G. Falciasecca, and V. Degli-Esposti, "Ray tracing propagation modeling for future small-cell and indoor applications: A review of current techniques," *Radio Science*, vol. 50, no. 6, pp. 469–485, 2015.

[2] Z. Yun and M. F. Iskander, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE Access*, vol. 3, pp. 1089–1100, 2015.

[3] E. Vitucci, J. Chen, V. Degli-Esposti, J. Lu, H. Bertoni, and X. Yin, "Analyzing radio scattering caused by various building elements using millimeter-wave scale model measurements and ray tracing," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 1, pp. 665–669, 2018.

[4] T. K. Geok, F. Hossain, M. Kamaruddin, N. Z. A. Rahman, S. Thiagarajah, A. T. W. Chiat, and C. Liew, "A comprehensive review of efficient ray-tracing techniques for wireless communication," *Int. J. Commun. Antenna Propag*, vol. 8, pp. 123–136, 2018.

[5] M. Lecci, P. Testolina, M. Giordani, M. Polese, T. Ropitault, C. Gentile, N. Varshney, A. Bodi, and M. Zorzi, "Simplified ray tracing for the millimeter wave channel: a performance evaluation," in *2020 Information Theory and Applications Workshop (ITA)*, pp. 1–6, IEEE, 2020.

[6] H. Mi, D. He, K. Guan, B. Ai, C. Liu, T. Shui, L. Zhu, and H. Mei, "Implementation and evaluation of ray-tracing acceleration methods in wireless communication," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, IEEE, 2020.

[7] D. Yan, K. Guan, D. He, B. Ai, Z. Li, J. Kim, H. Chung, and Z. Zhong, "Channel characterization for vehicle-to-infrastructure communications in millimeter-wave band," *IEEE Access*, vol. 8, pp. 42325–42341, 2020.

[8] L. Azpilicueta, C. Vargas-Rosales, and F. Falcone, "Intelligent vehicle communication: Deterministic propagation prediction in transportation systems," *IEEE Vehicular Technology Magazine*, vol. 11, no. 3, pp. 29–37, 2016.

[9] S. K. Kalyankar, Y. H. Lee, and Y. S. Meng, "Two-slope path loss model for curved-tunnel environment with concept of break point," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[10] H. Nguyen, X. Xiaoli, M. Noor-A-Rahim, Y. L. Guan, D. Pesch, H. Li, and A. Filippi, "Impact of big vehicle shadowing on vehicle-to-vehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6902–6915, 2020.

[11] K. Guan, D. He, B. Ai, D. W. Matolak, Q. Wang, Z. Zhong, and T. Kürner, "5-ghz obstructed vehicle-to-vehicle channel characterization for internet of intelligent vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 100–110, 2018.

[12] J. Pascual-García, J.-M. Molina-Garcia-Pardo, M.-T. Martínez-Inglés, J.-V. Rodríguez, and L. Juan-Llácer, "Fast and accurate electric field estimation from a single ray tracing simulation.," *Applied Computational Electromagnetics Society Journal*, vol. 30, no. 6, 2015.

[13] D. He, B. Ai, K. Guan, L. Wang, Z. Zhong, and T. Kürner, "The design and applications of high-performance ray-tracing simulation platform for 5g and beyond wireless communications: A tutorial," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 10–27, 2018.

[14] P. Moreau, M. Pharr, and P. Clarberg, "Dynamic many-light sampling for real-time ray tracing," in *High-Performance Graphics 2019 - Short Papers* (M. Steinberger and T. Foley, eds.), Eurographics - European Association for Computer Graphics, 2019.

[15] S. Hussain and C. Brennan, "Efficient preprocessed ray tracing for 5g mobile transmitter scenarios in urban microcellular environments," *IEEE Transactions on Antennas and Propagation*, vol. 67, no. 5, pp. 3323–3333, 2019.

[16] N. Mataga, R. Zentner, and A. K. Mucalo, "Ray entity based postprocessing of ray-tracing data for continuous modeling of radio channel," *Radio Science*, vol. 49, no. 3, pp. 217–230, 2014.

[17] J. Nuckelt, M. Schack, and T. Kürner, "Geometry-based path interpolation for rapid ray-optical modeling of vehicular channels," in *2015 9th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, IEEE, 2015.

[18] F. Wiffen, L. Sayer, M. Z. Bocus, A. Doufexi, and A. Nix, "Comparison of otfs and ofdm in ray launched sub-6 ghz and mmwave line-of-sight mobility channels," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 73–79, IEEE, 2018.

[19] S. O. Wald and F. Weinmann, "Ray tracing for range-doppler simulation of 77 ghz automotive scenarios," in *2019 13th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–4, IEEE, 2019.

[20] D. Bilibashi, E. M. Vitucci, and V. Degli-Esposti, "Dynamic ray tracing: Introduction and concept," in *2020 14th European Conference on Antennas and Propagation (EuCAP)*, 2020.

[21] J. B. Keller, "Geometrical theory of diffraction," *Josa*, vol. 52, no. 2, pp. 116–130, 1962.

[22] F. Mani, *Improved ray-tracing for advanced radio propagation channel modeling*. PhD thesis, ICTEAM, UCL Belgium, 2012.

[23] Q. Gueuning, *Inhomogeneous plane-wave spectrum based Physical Optics for the simulation of urban radio propagation*. PhD thesis, UCL-Université Catholique de Louvain, 2019.

[24] D. McNamara, C. Pistorius, and J. Malherbe, "The uniform geometrical theory of diffraction," *Artech House, London*, 1990.

[25] G. E. Athanasiadou and A. R. Nix, "Investigation into the sensitivity of the power predictions of a microcellular ray tracing propagation model," *IEEE transactions on Vehicular Technology*, vol. 49, no. 4, pp. 1140–1151, 2000.

[26] F. S. De Adana, O. G. Blanco, I. G. Diego, J. P. Arriaga, and M. F. Cátedra, "Propagation model based on ray tracing for the design of personal communication systems in indoor environments," *IEEE transactions on vehicular technology*, vol. 49, no. 6, pp. 2105–2112, 2000.
[27] C. Oestges, *Propagation Modelling of Low Earth-Orbit Satellite Personal Communication Systems*. PhD thesis, UCL Belgium, 2000.
[28] D. Tami, C. G. Rego, D. Guevara, A. Navarro, F. J. Moreira, J. Giménez, and H. G. Triana, "Analysis of heuristic uniform theory of diffraction coefficients for electromagnetic scattering prediction," *International Journal of Antennas and Propagation*, vol. 2018, 2018.

APPENDIX

t_{ext}	City street		City crossroad		Highway	
	Mean	Std	Mean	Std	Mean	Std
Rule A	100.00	0.00	100.00	0.00	100.00	0.00
Rule B	95.12	3.44	83.02	6.01	92.83	3.75
Rule C	91.79	5.18	70.14	9.16	86.72	6.29
Rule D	-		90.47	3.58	-	

TABLE IV: Percentage of simulation time where relative error on *delay spread* is below 20%, depending on the chosen extrapolation time rule (*all results are in percent*)

t_{ext}	City street		City crossroad		Highway	
	Mean	Std	Mean	Std	Mean	Std
Rule A	100.00	0.00	100.00	0.00	100.00	0.00
Rule B	93.89	5.04	83.45	7.15	91.26	3.63
Rule C	89.62	7.77	73.58	10.66	84.26	6.58
Rule D	-		91.17	3.84	-	

TABLE V: Percentage of simulation time where relative error on *azimuth spread* is below 20%, depending on the chosen extrapolation time rule (*all results are in percent*)

t_{ext}	City street		City crossroad		Highway	
	Mean	Std	Mean	Std	Mean	Std
Rule A	100.00	0.00	100.00	0.00	100.00	0.00
Rule B	95.51	3.83	88.77	5.47	90.48	6.84
Rule C	91.71	6.40	80.67	8.81	85.11	8.65
Rule D	-		93.80	3.31	-	

TABLE VI: Percentage of simulation time where relative error on *elevation spread* is below 20%, depending on the chosen extrapolation time rule (*all results are in percent*)

t_{ext}	City street		City crossroad		Highway	
	Mean	Std	Mean	Std	Mean	Std
Rule A	100.00	0.00	100.00	0.00	100.00	0.00
Rule B	93.51	6.18	84.05	7.93	91.87	3.51
Rule C	90.17	7.63	75.13	11.49	84.68	6.60
Rule D	-		90.66	4.50	-	

TABLE VII: Percentage of simulation time where relative error on *K-factor* is below 20%, depending on the chosen extrapolation time rule (*all results are in percent*)

t_{ext}	City street		City crossroad		Highway	
	Mean	Std	Mean	Std	Mean	Std
Rule A	100.00	0.00	100.00	0.00	100.00	0.00
Rule B	95.51	2.88	85.47	4.82	92.29	3.05
Rule C	91.72	5.02	76.15	7.48	86.10	5.41
Rule D	-		92.39	3.53	-	

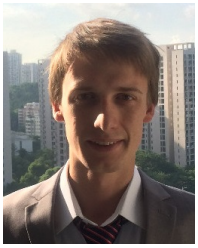
TABLE VIII: Percentage of simulation time where relative error on *total received power* is below 20%, depending on the chosen extrapolation time rule (*all results are in percent*)

City street		
Th.	Mean	Std
5 (*Rule B*)	2.25 (4.98)	0.49 (0.01)
10 (*Rule C*)	2.97 (9.93)	0.92 (0.03)
20	3.56 (19.7)	1.44 (0.11)

City crossroad		
Th.	Mean	Std
5 (*Rule D*)	2.08 (4.98)	0.26 (0.02)
10 (*Rule B*)	2.63 (9.93)	0.408 (0.02)
20 (*Rule C*)	3.04 (19.72)	0.54 (0.09)

Highway		
Th.	Mean	Std
4 (*Rule B*)	2.56 (3.99)	0.34 (0.002)
8 (*Rule C*)	3.80 (7.99)	0.72 (0.01)
16	5.06 (15.94)	1.23 (0.03)

TABLE IX: Real computational gain C_R *(theoretical value from equation (10) with the specified rule for computing the extrapolation time and assuming a* Δt *of 0.01 second - mean and std based on simulation results - results in parenthesis neglect diffraction)*



Florian Quatresooz received the B.Sc. and the M.Sc. degrees in Electrical Engineering from the Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium, respectively in 2018 and 2020. He is currently pursuing a Ph.D. degree at the Institute for Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM) at UCLouvain. His research interests include ray tracing propagation for channel characterization, wave propagation and optical communications.

Simon Demey received the M.Sc. degree in Electrical Engineering from the Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium, in 2018 and then joined the Institute for Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM) at UCLouvain as a Ph.D. student. His research interests lie in the area of channel characterization for emerging networks, investigating the trade-offs between deterministic ray tracing and stochastic methods for small cells.

Claude Oestges received the M.Sc. and Ph.D. degrees in Electrical Engineering from the Université catholique de Louvain (UCLouvain), Louvain-la-Neuve, Belgium, respectively in 1996 and 2000. In January 2001, he joined as a post-doctoral scholar the Smart Antennas Research Group (Information Systems Laboratory), Stanford University, CA, USA. From January 2002 to September 2005, he was associated with the Microwave Laboratory UCLouvain as a post-doctoral fellow of the Belgian Fonds de la Recherche Scientifique (FRS-FNRS). Claude Oestges is presently Full Professor with the Electrical Engineering Department, Institute for Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), UCLouvain. He is the author or co-author of four books and more than 200 journal papers and conference communications, and he was the chair of COST Action CA15104 IRACON (2016-2020), as well as the recipient of the 1999-2000 IET Marconi Premium Award and of the IEEE Vehicular Technology Society Neal Shepherd Award in 2004 and 2012.