Progress In Electromagnetics Research C, Vol. 91, 129–142, 2019

Investigation of Massive MIMO Scenarios Involving Rooftop Propagation by Bidirectional Ray-Tracing

Mehmet M. Taygur*, Ilya O. Sukharevsky, and Thomas F. Eibert

Abstract—The average downlink data-rate in massive Multiple Input Multiple Output (MIMO) networks within realistic urban environments is characterized by means of ray-tracing simulations. The links between the receivers and transmitters are mostly established through rooftop propagation, which requires special treatment due to multiple diffractions near the optical boundaries. The bidirectional ray-tracing method is utilized in order to simulate these effects accurately. The average downlink data-rate is also calculated according to an empirical rooftop propagation model and the differences as well as the similarities with the bidirectional ray-tracing results are demonstrated. Additionally, an iterative Shooting and Bouncing Rays (SBR) algorithm, which improves the computational efficiency of the bidirectional ray-tracing, is introduced. The algorithm aims to maximize the number of rays, which contribute to the result, by setting specific launch directions. The results show that noticeable improvements in the computation time are possible.

1. INTRODUCTION

Ray-tracing has emerged as a valuable simulation method to analyze massive Multiple Input Multiple Output (MIMO) networks. As the simulations provide accurate data with much less effort than real measurements, ray-tracing simulations have been utilized for analyzing numerous massive MIMO scenarios where geometry-specific data is needed [1].

A typical simulation involves the calculation of the relevant ray paths between two points with the Shooting and Bouncing Rays (SBR) method [2], and the calculation of the field expressions according to Geometrical Optics and Uniform Theory of Diffraction (GO/UTD) principles [3].

Although ray-tracing can be successfully applied to many problems, some scenarios may not be accurately simulated due to certain limitations of the method. In particular, traditional UTD-based diffraction computations are usually inaccurate when multiple diffractions occur and the diffraction paths are near the optical boundaries [4]. A typical example of such a case is rooftop propagation, which can emerge frequently in urban cellular networks. The issue has been addressed in various studies with analytical or empirical models [5, 6], which rely on certain assumptions about the scenario (e.g., limited number of ray interactions, limited frequency range). However, in many realistic problems, these assumptions might be easily violated and the model may yield inaccurate results. Consequently, such approaches have not been widely adopted in general-purpose simulation tools, hence, the performance of the massive MIMO networks in realistic rooftop propagation scenarios has still not been fully recognized.

In this study, the performance of an urban massive MIMO network, where rooftop propagation is dominant, is investigated by means of a bidirectional ray-tracing algorithm [7, 8]. Ray launching is performed from both transmitter and receiver sites and the rays are captured on an interaction surface. The antenna transfer function is then obtained by evaluating the reciprocity integral over this surface, in

Received 27 December 2018, Accepted 7 March 2019, Scheduled 21 March 2019

* Corresponding author: Mehmet Mert Taygur (mehmet.taygur@tum.de).

The authors are with the Technical University of Munich, Department of Electrical and Computer Engineering, Chair of High-Frequency Engineering, Germany.

a similar way as for Physical Optics (PO). The method can accurately handle multiple diffractions near the optical boundaries and no strict requirements are imposed on the scenario in general, in contrast to other methods in the literature.

Additionally, an iterative SBR algorithm is presented in order to address issues regarding the computational efficiency of the traditional SBR method. Instead of launching a large number of rays in a single step, multiple ray-launching processes are executed with a smaller number of rays and the launch directions, which contribute to the result (i.e., the launch directions which lead to ray hits), are identified. In each step, the new launch directions are chosen according to the launch directions which lead to ray hits in the previous iterations. It is shown that the algorithm converges quickly and the total computation time is smaller than for the traditional SBR algorithm.

2. CALCULATION OF ANTENNA TRANSFER FUNCTION

Let R and T be two antennas and Γ be a surface which is closed around R. According to the reciprocity theorem, the induced open circuit voltage at the terminals of R, when T is in transmit, can be given by [9]

$$\oiint_{\Gamma} \left[(\mathbf{H}_T \times \mathbf{E}_R) - (\mathbf{H}_R \times \mathbf{E}_T)\right] \cdot d\mathbf{S} = v_R^{oc} i_R, \tag{1}$$

where v_R^{oc} is the induced open-circuit voltage at antenna R on receive, and i_R is the port current at antenna R on transmit. $\mathbf{E}_R$, $\mathbf{E}_T$ and $\mathbf{H}_R$, $\mathbf{H}_T$ are the electric and magnetic fields for R and T, respectively. Thus, the transfer function for R and T can be found by computing the surface integral in Eq. (1), if the port voltage at T is also known. Here it is noted that we consider time-harmonic fields with a suppressed time dependence $e^{j\omega t}$, where ω is the angular frequency.

2.1. Evaluation of the Reciprocity Integral

The surface integral given in Eq. (1) is an oscillatory integral, which is evaluated as follows:

(i) The integrand is expressed as the multiplication of a magnitude and a phase function such that

$$\oiint_{\Gamma} \left[(\mathbf{H}_T \times \mathbf{E}_R) - (\mathbf{H}_R \times \mathbf{E}_T)\right] \cdot d\mathbf{S} = \oiint_{\Gamma} e^{j\beta\phi(\mathbf{x})} u(\mathbf{x}) dS, \tag{2}$$

with

$$\begin{aligned} \Psi(\mathbf{x}) &= \left[(\mathbf{H}_T \times \mathbf{E}_R) - (\mathbf{H}_R \times \mathbf{E}_T)\right] \cdot \hat{\mathbf{n}}, \\ u(\mathbf{x}) &= \operatorname{sgn}\left(\operatorname{Re}\left(\frac{e^{j\beta\phi(\mathbf{x})}|\Psi(\mathbf{x})|}{\Psi(\mathbf{x})}\right)\right)|\Psi(\mathbf{x})|, \\ \phi(\mathbf{x}) &= \arg(\Psi(\mathbf{x}))/\beta, \end{aligned} \tag{3}$$

where $\hat{\mathbf{n}}$ is the unit surface normal, $\mathbf{x}$ is the position in space, and β is the propagation constant.

(ii) The integration domain is divided into small triangular sub-domains such that u and ϕ can be effectively assumed linear in each sub-domain [10]. Hence, the original surface integral can be approximated by

$$\oiint_{\Gamma} e^{j\beta\phi(\mathbf{x})} u(\mathbf{x}) dS \approx \sum_{n=1}^{N} \iint_{\Gamma_n} e^{j\beta\tilde{\phi}_n(\mathbf{x})} \tilde{u}_n(\mathbf{x}) dS, \tag{4}$$

where N is the number of the subdomains (i.e., triangles), and $\tilde{u}_n$ and $\tilde{\phi}_n$ are the linear approximations of u and ϕ in the nth subdomain, respectively.

(iii) $\tilde{u}_n$ and $\tilde{\phi}_n$ can be computed according to the values of u and ϕ at each corner of each triangle.

(iv) According to Eq. (3), $\mathbf{E}_R$, $\mathbf{E}_T$, $\mathbf{H}_R$, $\mathbf{H}_T$ should be known at the triangle corners. Note that the ray-tracing simulation is utilized to obtain the field information at the intersection points of the rays and the surface. Nevertheless, it is highly unlikely for the rays to hit all the triangles exactly at the corner points, therefore, the available samples are interpolated to obtain the field expressions at the corners.

(v) Once the field expressions at the triangle corners are known and $\tilde{u}_n(\mathbf{x})$, $\tilde{\phi}_n(\mathbf{x})$ are obtained accordingly, the surface integral in each subdomain (right-hand side of Eq. (4)) can be computed analytically.

It can be noticed that the interaction surface Γ should be closed according to Eq. (1). However, the contributions near the stationary phase points or critical points of second kind are usually more prominent compared to the rest of the integration domain [11]. Thus, an open surface, which is located between the transmitter and receiver antennas, can be utilized to simulate the diffraction effects, as demonstrated in [8].

3. BIDIRECTIONAL RAY-TRACING SIMULATION

The bidirectional ray-tracing is mainly based on simultaneous ray launching from transmitter and receiver sites [12]. The rays are captured on an interaction surface, which is usually more flexible than reception spheres (commonly utilized in the traditional unidirectional ray-tracing) in shape and size. An illustrative comparison of traditional unidirectional ray-tracing and bidirectional ray-tracing methods is shown in Fig. 1. In [8], it was shown that the bidirectional ray-tracing is computationally more efficient than the traditional unidirectional ray-tracing, and also as accurate. The bidirectional ray-tracing can usually match the accuracy of unidirectional ray-tracing with a smaller number of ray launches, even though both antennas are utilized for ray launching. The reception spheres in the unidirectional ray-tracing simulations are usually no larger than a couple of wavelengths, hence, the number of ray launches should be large enough to hit such a small target, especially at large distances. In the bidirectional ray-tracing though, the target might be as large or even larger than the surrounding objects in the geometry and the number of the rays, which is necessary to perform interpolation and integration (described in Subsection 2.1), is relatively small. In other words, the distance between the adjacent rays can be significantly larger than the size of a typical reception sphere in bidirectional ray-tracing simulations whereas the same conditions could easily yield missed rays in the unidirectional ray-tracing. Furthermore, the elimination of UTD-based diffraction computations in bidirectional ray-tracing prevents the growth of the ray tree and the number of the rays in general. A typical simulation involves three main steps:

(i) *Computation of the ray-paths:* The feasible ray paths between the surface and antennas are computed. Ray-launching is performed according to the SBR method at each site where many rays are launched in random directions. Each ray is traced throughout the geometry until it reaches a maximum number of interactions or leaves the scene. A ray is discarded from memory, unless it has intersected with the surface at least once (multiple intersections are possible). Duplicate rays (i.e., the rays which have identical path history) are also eliminated after the ray paths are obtained.

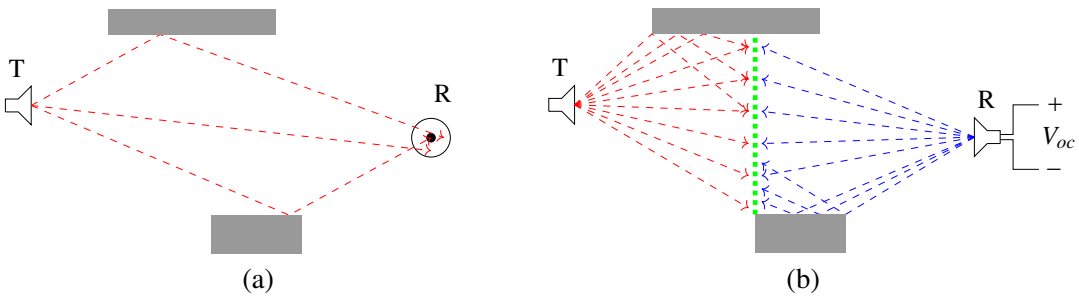

Figure 1. Illustration of unidirectional (a) and bidirectional ray-tracing (b).

The ray path history is characterized according to the launching antenna, the facets that the ray bounce from and the triangle on the interaction surface that the ray intersects with.

(ii) *Computation of the field expressions:* The electric and magnetic field expressions at the ray-surface intersection points are calculated according to GO/UTD principles. The path history of each ray, which was obtained during the first step, is utilized for the calculation.

(iii) *Computation of the antenna transfer function:* The antenna transfer function is calculated according to the procedure described in Section 2.

The computation time is mostly determined by the ray-tracing and integration processes. Specifying an explicit complexity expression for a typical ray-tracing algorithm is generally considered to be extremely challenging though, therefore, empirical evidence and comparative studies are preferable for a complexity estimation.

It is well known that the bidirectional ray-tracing method can provide considerable improvements in the computation time, compared to the conventional unidirectional ray-tracing approach. Especially the UTD-based diffraction calculations, which are commonly utilized in the unidirectional ray-tracing, is a major factor which inflates the number of the rays in the simulation and makes the ray-tree grow. A single ray, which is incident on an edge, may cause thousands of new rays to be generated on a Keller cone, depending on the ray density. It can be predicted that the situation may become even more severe if multiple diffractions occur in the scenario. A notable advantage of the proposed method is the feasibility of reducing the number of UTD diffractions (or even completely avoid them under certain conditions). Thus, the growth of the ray-tree can be repressed and the computation time might be reduced since a smaller number of rays would be involved in the simulation, as already stated. Previous studies have shown that the bidirectional ray-tracing method can give good results with a number of the rays 10 times smaller than required in a unidirectional ray-tracing simulation (only single diffractions were considered) [8]. It is possible to observe a similar reduction in the number of rays, even in scenarios without diffractions (e.g., 2-ray ground reflection model), although with less dramatic differences [7].

The complexity of the integration process is typically dependent on the number of the subdomains created on the interaction surface (see Subsection 2.1). Note that the number of the subdomains is a function of many different parameters such as wavelength, size of the interaction surface, validity of linear approximation within the subdomain, which themselves depend on other parameters. Nevertheless, it is possible to estimate the complexity under certain assumptions. Let us consider that the subdomains are identical in size in the entire integration domain such that the linear approximation is sufficiently accurate everywhere. The size of a single triangle is then proportional to λ^2. A key point is that the size of the interaction surface typically scales also with λ^2 in many scenarios since utilizing higher frequencies (and therefore small wavelengths) might be infeasible in geometrically large scenarios due to excessive path loss (reduced aperture size, to be precise). Hence, both effects compensate each other and the number of integration triangles as well as the related computational effort for performing the integrations remain relatively constant within practically relevant propagation scenarios.

All computations are executed in parallel on a GPU by means of NVIDIA OptiX$^{\text{TM}}$ and Thrust CUDA libraries, whenever possible [13, 14].

3.1. Iterative Shooting and Bouncing Rays Algorithm

The SBR method is widely utilized for the simulation of complex scenarios (i.e., many objects, many interactions), when the alternatives (e.g., image method) are prohibitively time-consuming [15, 16]. A large number of ray launches is usually needed in the SBR method for accurate results but a significant portion of the rays may not contribute anything as they leave the geometry without interacting with the target (i.e., the interaction surface) [17]. Thus, the process might become very inefficient since the computational resources might be utilized for operations which do not affect the result. The issue has previously been addressed for unidirectional ray-tracing simulations [18] but the proposed solution is not directly applicable to the bidirectional case. Therefore, a new iterative SBR algorithm, which accommodates the bidirectional ray-tracing properly, is introduced in this study. Unlike the unidirectional case, where the size of the target (i.e., reception sphere) should also be modified in each iteration, the new algorithm is exempt from such a requirement, therefore, more robust in general.

The algorithm relies on successive ray-launching processes where the launch directions are chosen according to the directions of the rays which hit the surface in the previous ray-launching steps. The pseudocode of the algorithm is shown in Fig. 2.

```
1: procedure ITERATIVE SBR (d_0; ε)        ▷ d_0[1:N][1:2]: initial ray directions in (θ,ϕ) in spherical coord., N: Number of ray launches, ε: iteration stop criteria
2:     d ← d_0;
3:     K_1 ← 0;                            ▷ K_1:No. of rays which hit in previous iteration
4:     while (true) do
5:         h ← rayTrace(d);                ▷ h[1:K][1:2]: launc directions of rays which hit
6:         if ((K − K_1)=K < ε) then
7:             break;
8:         end if
9:         if (K < N) then
10:            d[1:K][:] ← h[:][:];        ▷ Note that the ray directions, which previously yield hits, are re-committed into the buffer the rays are re-traced. The performance can further be improved by avoiding re-tracing.
11:            for i ← (K : N) do
12:                h[mod(i; K)][0] ← h[mod(i; K)][0] + Δθ;
13:                h[mod(i; K)][1] ← h[mod(i; K)][1] + Δϕ;   ▷ Δθ, Δϕ: uni. dist. random variables in [−π/2; π/2]. The new launch directions are strongly dependent on the selection of Δθ and Δϕ. A brief discussion about the selection is given in the following section.
14:                d[i][:] ← h[mod(i; K)][:];
15:            end for
16:        else
17:            break;
18:        end if
19:        K_1 ← K;
20:    end while
21: end procedure
```

Figure 2. Iterative SBR pseudocode.

3.1.1. Convergence Study

A key principle, which makes the iterative SBR algorithm convergent in relevant propagation scenarios, is the conservation of the ray directions which yield ray hits. Hence, for a fixed number of ray launches, the number of ray hits monotonically increases while the number of the missed rays declines in a similar manner at each successive iteration. Using the notation from Fig. 2, let N be the number of ray launches and $K \in \mathbb{Z}^+$ be the number of hits from the initially launched $N \in \mathbb{Z}^+$ rays. Then, the probability of collecting a ray on the designated target (p) can be given as

$$p = \frac{K}{N}, \quad N \geq K. \tag{5}$$

Assuming that the geometrical configuration, and therefore the value of p, are fixed at each iteration, the number of hits and misses can be roughly quantified. Let K_i and M_i be the number of hits and misses at the ith iteration, respectively. For the sake of simplicity, we do not consider the uniqueness of the ray paths, i.e., duplicate rays may exist. The number of the ray launches is constant at each iteration, hence, $N = N_i = K_i + M_i$, $\forall i \in \mathbb{Z}^+$. Then, the relation among the given quantities can be expressed as

$$K_0 = pN,\ M_0 = N - K_0 = N(1-p),\ 0 < p \leq 1, \tag{6}$$

where K_0 and M_0 indicate the number of hits and number of misses for the first ray launch process, respectively. The initial ray launch is omnidirectional, i.e., the ray launch directions denoted by $(\Theta_d,\ \phi_d)$ in spherical coordinate system can lie in $0 \leqslant \Theta_d \leqslant \pi$, $-\pi \leqslant \phi_d \leqslant \pi$, which is the complete unit sphere. Note that we define $p = K_0/N$ as a constant according to the initial launch. The probability of hit or miss at each iteration $(K_i/N,\ M_i/N)$ can be expressed in terms of p though, as it will be shown in the following. At each iteration, the ray directions, which have yield hits, are preserved. The missed rays

are re-launched with new launch directions and hit the target with a probability of p. Thus, the number of the ray hits at the ith iteration can be written as

$$K_i = K_{i-1} + pM_{i-1}. \tag{7}$$

Since the sum of the number of the hits and the number of the misses should be N (the number of ray launches)

$$M_i = N - K_i. \tag{8}$$

By plugging Eq. (8) into Eq. (7),

$$\begin{aligned} M_i &= N - K_{i-1} - pM_{i-1} = N - (N - M_{i-1}) - pM_{i-1},\ i \in \mathbb{Z}^+, \\ \frac{M_i}{M_{i-1}} &= (1-p) < 1 \longrightarrow M_i < M_{i-1}. \end{aligned} \tag{9}$$

Equation (9) thus demonstrates that the number of the missed rays decays in each iteration, and the algorithm can converge as the number of the ray hits becomes very close to N, that is the number of the ray launches. The convergence rate is strongly dependent on p, which was calculated according to a certain ray launching scheme where the unit vectors denoting the launch directions are distributed over the complete unit sphere, in other words, rays might be launched in every direction in every iteration without any consideration of direction optimization. In the suggested algorithm though, the new ray directions are obtained by randomly shifting the available direction vectors. It has been observed that a shifting interval of $[-\pi/2, \pi/2]$ provides accurate results as well as a decent convergence rate. Utilizing smaller intervals has usually resulted in clustered intersection points (i.e., intersections points were not well distributed over the entire surface) while larger intervals have delayed the convergence in general. Let us now consider a single ray which hit the target in a previous iteration. As many new ray launch directions will be generated by shifting the launch direction of this ray, a shift between $[-\pi/2, \pi/2]$ in ϕ and θ indicate that the new launch directions could cover only half of the surface of the unit sphere, i.e., a hemisphere whose symmetry axis coincides with the original ray launch direction. Therefore, the likelihood of a hit is two times the likelihood in an omnidirectional launch scheme in general (exceptions may occur if p is already large). Then the convergence ratio in Eq. (9) can accordingly be modified as

$$\frac{M_i}{M_{i-1}} = (1-2p). \tag{10}$$

The variables $\Delta\theta$ and $\Delta\phi$ might also be computed with many different approaches since there are no strict limitations as long as the new ray directions yield well-distributed intersection points on the target and a likelihood of hit which is larger than p (the suggested approach provides a likelihood of $2p$). In general, the larger the likelihood is, the faster the algorithm can converge.

3.1.2. Comparison with Traditional SBR

A ray launching problem is now considered where the objective is to collect a certain number of rays. By using the assumptions and simplifications made for Eqs. (5)–(10), it can be shown that the number of ray launches, which are needed to collect a certain amount of rays on the interaction surface, is smaller in the case of iterative SBR. Let $R \in \mathbb{Z}^+$ be the number of rays which should be collected on the surface and $R < N$ holds where N is the number of ray launches in a single iteration. Furthermore, $0 < p \ll 1$ is assumed. Now the convergence criterion is described as R becoming equal to the number of the ray hits at the ith iteration, in other words $R = K_i$. K_i can indirectly be obtained by calculating the number of missed rays with

$$K_i = N - M_i. \tag{11}$$

Using Eqs. (9) and (10), M_i and K_i can be obtained as

$$\begin{aligned} M_i &= M_0 \frac{M_1}{M_0}\frac{M_2}{M_1}\cdots\frac{M_{i-1}}{M_{i-2}}\frac{M_i}{M_{i-1}} = N(1-p)(1-2p)^i \approx N(1-2p)^i, \\ K_i &= N\left(1-(1-2p)^i\right) = R. \end{aligned} \tag{12}$$

The total number of rays, which were launched (and traced) during the execution of the entire iterative SBR process (T_{ISBR}), can be given as

$$(1-2p)^i = 1 - \frac{R}{N} \longrightarrow i = \frac{\log\left(1-\frac{R}{N}\right)}{\log\left((1-2p)^i\right)},$$

$$T_{\mathrm{ISBR}} = iN = N\frac{\log\left(1-\frac{R}{N}\right)}{\log\left((1-2p)^i\right)}. \tag{13}$$

Using Eq. (5), the number of ray launches which is needed to achieve R ray hits can be obtained as

$$T_{\mathrm{SBR}} = \frac{R}{p}. \tag{14}$$

With the auxiliary variable $a = \frac{R}{N}$, T_{ISBR} and T_{SBR} can be given by

$$T_{\mathrm{ISBR}} = N\frac{\log(1-a)}{\log\left((1-2p)^i\right)}, \quad T_{\mathrm{SBR}} = N\frac{a}{p}. \tag{15}$$

Note that N is not a parameter which is utilized in a conventional SBR simulation (N indicates the number of ray launches per iteration in iterative SBR) but it is given for comparative purposes in T_{ISBR}. The relative variation of both quantities is shown in Fig. 3 for $N = 200,000$ ray launches per iteration, $a = 0.5$ (i.e., half of the ray launches should hit the surface, hence $R = 100,000$) and $0 < p < 0.1$ (i.e., no more than 10% of the rays is expected to reach the surface when the ray launch is omnidirectional.)

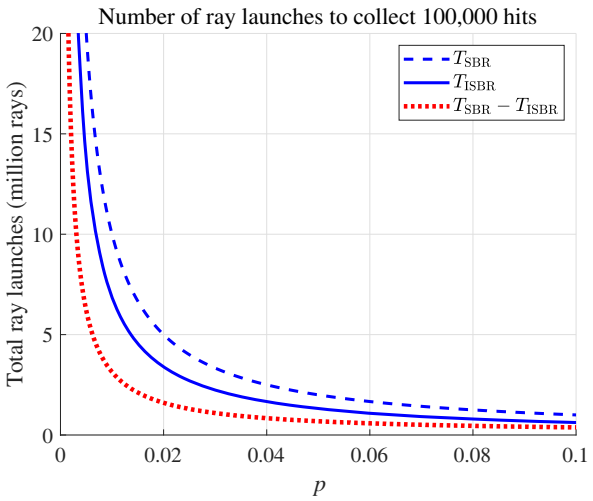

Figure 3. Comparison of number of rays required to collect 100,000 rays.

The results indicate that the number of the ray launches which are required to collect a certain number of rays on an interaction surface is generally smaller in the iterative SBR case. The difference between the T_{SBR} and T_{ISBR} curves, which is indicated by the dotted red curve, grows as the value of p drops. This means, the benefits of iterative SBR over traditional SBR are much more eminent when the target becomes more and more difficult to hit.

4. ROOFTOP DIFFRACTION SCENARIO

The considered scenario is based on a 3D urban grid with slightly different building heights, varying between 22 m–25 m. There are 25 buildings in total on a 250 m × 250 m area. The grid itself has

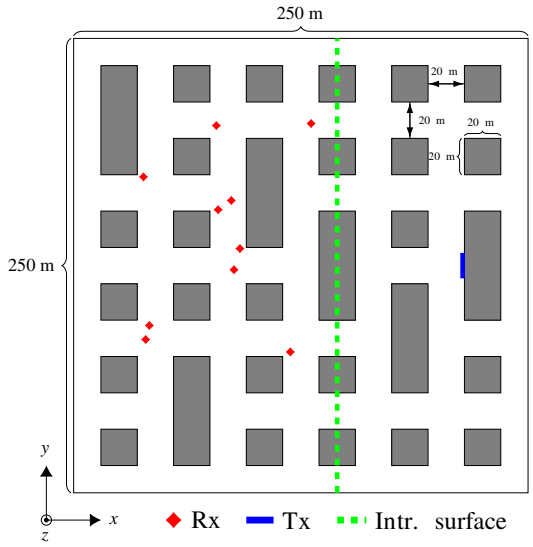

Figure 4. Urban scenario 2D top view illustration.

been arranged in a way that dominant paths through the street canyons are mostly prevented. Hence, propagation over the rooftops are usually the primal way to establish a link. 100 transmitter antennas are arranged as a 10×10 planar array and placed on top of a building while 10 single-antenna receivers are placed randomly on the ground level. All the antennas were assumed as half-wavelength dipoles which operate at 2.45 GHz. A rectangular interaction surface, which is $250\,\mathrm{m} \times 80\,\mathrm{m}$ large, is separating the transmitters and the receivers. An illustration of the scenario is shown in Fig. 4.

The simulations have been performed with maximum 4 reflections from both transmitter and receiver sites (combined paths up to 8 total reflections possible) and 1 UTD diffraction only from the receiver sites. The UTD diffractions were only allowed at the edges of the buildings which lie along the y-axis at the roof level. Note that the UTD diffractions are essential for the receiver sites since they are located at the shadow regions. On the other hand, the transmitter antennas are already located near the optical boundaries (i.e., roof level). Therefore, UTD contributions are not considered in that case (reflections are still feasible though). Illustrations of three potential propagation scenarios and the 2-D plot of the real part of the integrand $\Psi(\mathbf{x})$ (see Eq. (3)) over the surface are given in Fig. 5. Note that the surface plots depict $\mathrm{Re}(\Psi(\mathbf{x}))$ for small portions of the actual integration domains where certain critical points, which make the largest contributions into the integral result, exist.

Brief explanations for all three scenarios are given as follows:

- *Case 1:* Tall buildings are present between the surface and the transmitter & UTD locations. This yields two critical points of second kind where the level curve of the phase function ϕ (see Eq. (3)) is tangential to the domain boundary [19]. In other words, the partial derivatives of ϕ satisfy $\frac{\partial \phi}{\partial z} \neq 0$, $\frac{\partial \phi}{\partial y} = 0$ at the indicated points. However, most contributions comes from the point located at the bottom part of the surface, as the interaction surface is usually large enough to make the contributions from the upper boundary negligible where the magnitude function Ψ declines.
- *Case 2:* Transmitter and UTD locations are slightly above the surrounding buildings and line-of-sight (LOS) path exists through the interaction surface. The intersection point of the LOS path line and the interaction surface is a stationary phase point or a critical point of first kind. It can be noticed that two critical points of second kind are present at the boundaries since $\frac{\partial \phi}{\partial z} \neq 0$, $\frac{\partial \phi}{\partial y} = 0$ are satisfied at the indicated points. The contributions from the stationary phase points are generally dominant to those from the critical points of second kind though, especially when the critical points are sufficiently distant from each other and the wavenumber has a large value [11,19].
- *Case 3:* Multiple wavefront contributions come from the receiver site due to the reflections from the surrounding buildings. The resulting pattern is not intelligible due to the overlap of multiple

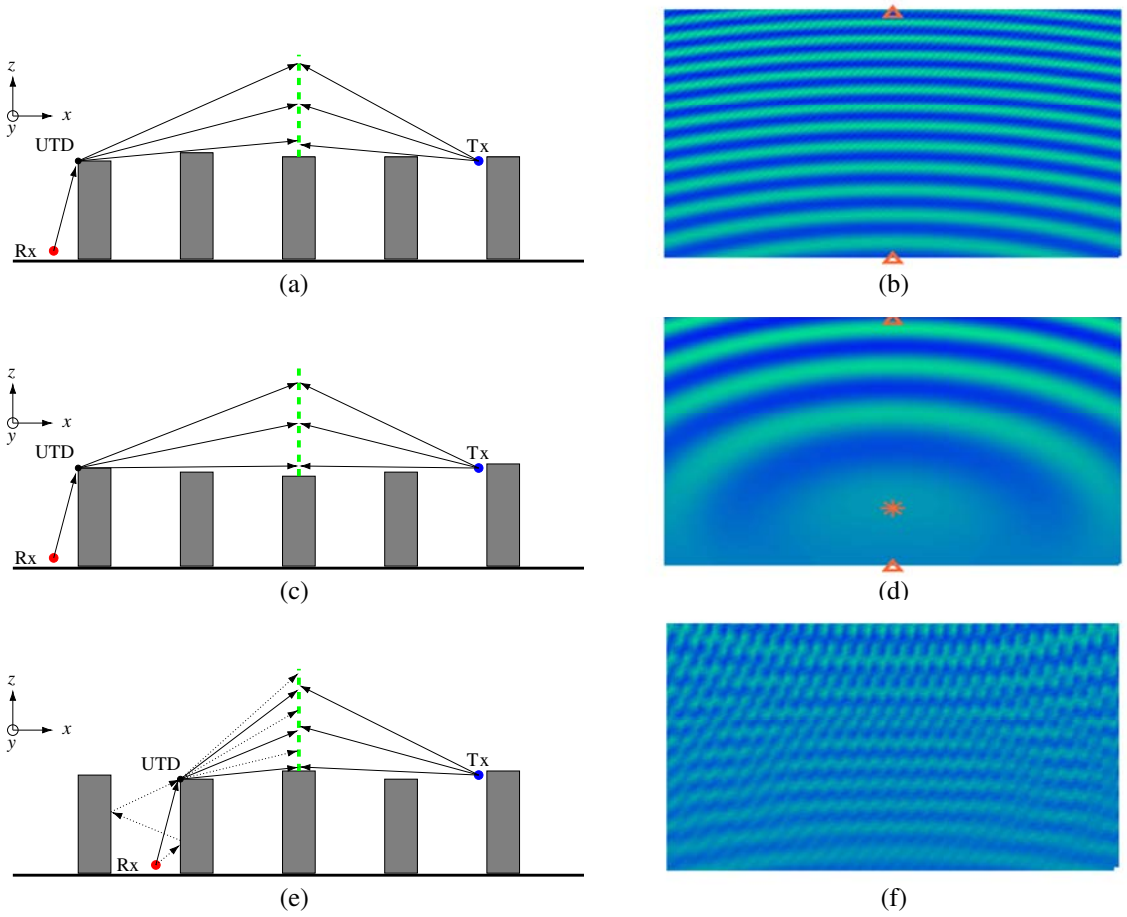

Figure 5. Illustration of 3 possible propagation scenarios ((a), (c), (e)) and corresponding $\mathrm{Re}(\Psi(x))$ (see Eq. (3)) on the interaction surface ((b), (d), (f)). The symbols "$*$" and "$\triangle$" denote the locations of the critical points of first and second kind, respectively.

contributions, therefore, the critical points may not be explicitly indicated. Nevertheless, individual wavefront pairs still create patterns similar to those in Case 1 and 2.

Note that the corners of the interaction surface may constitute critical points of third kind where the partial derivatives of ϕ might have discontinuities. However, the contributions from these points are usually negligible due to the significance of the critical points of first and second kind as well as the decline of the magnitude function Ψ at the corners when the surface is large.

5. MASSIVE MIMO

Massive MIMO is an improvement of the traditional multi-user MIMO where the number of the transmitter antennas is much larger than the number of the receivers. In order to serve every user simultaneously at the same frequency band, transmit beamforming is employed. In this study, a common beamforming method, the Regularized Zero-Forcing (RZF), is assumed and the network performance is investigated by computing the average downlink data rate among all the users. The channel matrix ($\mathbf{H}$) is obtained from the ray-tracing simulations and the individual elements of the matrix are expressed as

$$\mathbf{H}_{(i,j)} = \frac{v_i^{oc}}{v_j^{tx}}, \quad 1 \leq i \leq 10, \quad 1 \leq j \leq 100, \tag{16}$$

where $\mathbf{H}_{(i,j)}$ indicates the coefficient at the ith row and jth column of $\mathbf{H}$; v_j^{tx} is the port voltage of the jth transmitter; and v_i^{oc} is the induced open-circuit voltage of the ith transmitter, obtained from

Eq. (1). Then the beamforming matrix $\mathbf{g}$ can be given as

$$\mathbf{g} = \mathbf{H}^* \left(\mathbf{H}\mathbf{H}^* + \delta \mathbf{I}\right)^{-1}, \quad \delta > 0, \tag{17}$$

where $(^*)$ is the Hermitian transpose operator; $\mathbf{I}$ is the identity matrix; and δ is a regularization constant. The downlink data rate for user i can be found by

$$\begin{aligned} D_i &= W \log_2 \left(1 + \mathrm{SINR}_i\right), \\ \mathrm{SINR}_i &= \frac{P\left(\mathbf{H}_{(i,:)}\mathbf{g}_{(:,i)}\right)^2}{P\sum_{n=1}^{10}\left(\mathbf{H}_{(i,:)}\mathbf{g}_{(:,n)}\right)^2 + \sigma}, \end{aligned} \tag{18}$$

where W is the channel bandwidth; P is the total transmitter power; and σ is the noise power which is calculated as $\sigma = 10^{-20.4+\log_{10}(W)}$. Finally, the average data rate can be expressed as $D_{\mathrm{avg}} = \sum_{i=1}^{10} D_i/10$.

6. NUMERICAL RESULTS

6.1. Convergence of the Iterative SBR Algorithm

The convergence of the iterative SBR algorithm has been tested in the described urban environment with a single receiver-transmitter pair. At each iteration, a fixed number of rays were launched from both sites ($300,000 \times 2 = 600,000$ rays total) and the number of the unique rays (i.e., non-duplicate rays defined in Section 3), which hit the surface, has been used as the convergence criterion. The convergence plot is given in Fig. 6. The algorithm has been executed until the 10th iteration where the increase in the number of rays compared to the previous step was eventually less than 1% (red plot) and the number of the unique ray paths which reach the surface was around 200,000 (blue plot). In order to compare the proposed method with the classical techniques, the same scenario was simulated with the conventional SBR and the total number of ray launches, which yield 200,000 unique ray paths (as in the iterative SBR case), has been inquired. The investigations have revealed that 10,000,000 ray launches would be needed in total, in order to capture 200,000 unique rays on the surface. It can be noticed that the iterative SBR method consists of 11 steps (1 initial + 10 iterations) with 600,000 ray launches, which indicates 6,600,000 ray launches in total for the entire simulation. Compared to the conventional SBR (6,600,000 v 10,000,000), the iterative SBR method yields a decent improvement in terms of computational effort.

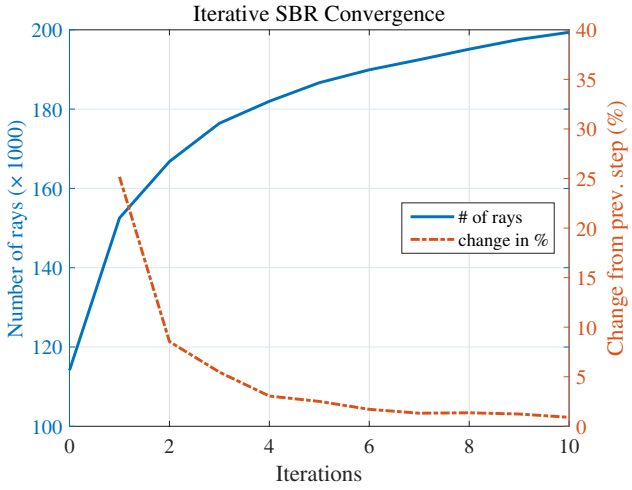

Figure 6. Convergence of the iterative SBR algorithm.

6.2. Validation of the Bidirectional Ray-Tracing for Propagation Near the Optical Boundaries

In order to validate the accuracy of the bidirectional ray-tracing and to demonstrate the breakdown of UTD-based diffraction computations in unidirectional ray-tracing simulations, a multiple knife-edge diffraction scenario has been simulated, and the results have been compared to the Vogler method [20], which can be considered as a reference. The knife-edges are identical in height and are separated by identical distances as well. A receiver and a transmitter are located at the optical boundary, and an interaction surface exists between them. Two different scenarios have been investigated with 2 and 4 knife-edges, respectively. The illustrations of the scenarios are given in Fig. 7.

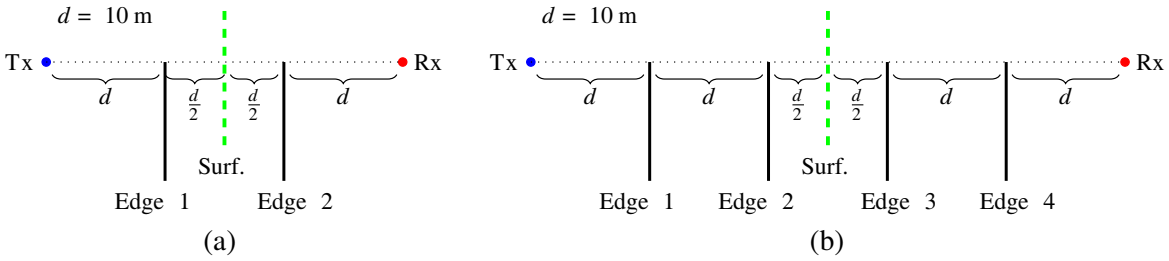

Figure 7. Illustration of multiple knife-edge diffraction scenarios with (a) 2 edges (b) 4 edges.

The path gain has been used for comparison. The results are shown in Fig. 8.

The results indicate that the bidirectional ray-tracing is in good agreement with the Vogler method. The difference between the bidirectional ray-tracing and the Vogler method is around 3 dB for 2 edges and around 7 dB for 4 edges. It has been observed that the bidirectional ray-tracing simulation is insensitive to the changes in the number of the knife-edges as the decline in the path gain mostly occur due to the increased distance. Nevertheless, this would not affect the accuracy in realistic scenarios since the attenuation is usually determined by only a few (< 4) dominant edges (i.e., the edges which are slightly higher than the others). The failure of the unidirectional ray-tracing is evident especially in the scenario with 4 edges. Although the deviation from the reference is similar to what bidirectional ray-tracing exhibits for 2 edges, the difference grows very quickly and rises to 17 dB for 4 edges, which is almost 10 dB larger than what the bidirectional ray-tracing simulation shows.

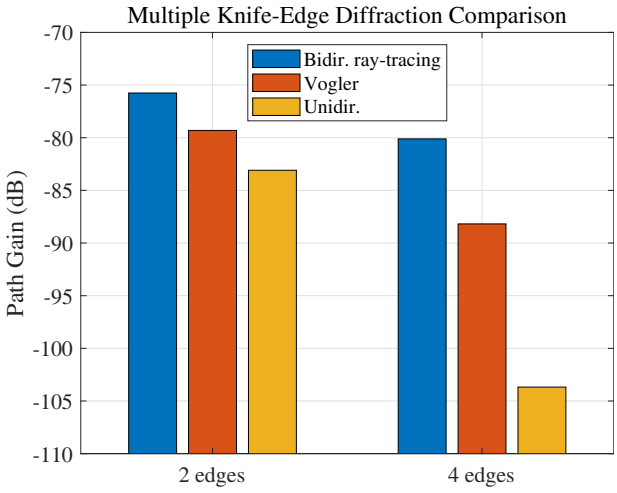

Figure 8. Comparison of bidirectional and unidirectional ray-tracing to Vogler method for two multiple knife-edge diffraction scenarios.

6.3. Network Performance

The performance of the massive MIMO network has been characterized by the average downlink data rate. In order to demonstrate the differences in the network performance under different channel models, the ITU-R P.1411 rooftop propagation model [21] (which is a significantly improved variation of the well known COST 231 Walfisch-Ikegami model) is utilized to generate another channel matrix, in addition to the one obtained by ray-tracing. The data rate has also been calculated for different numbers of transmitter antennas (i.e., some of the antennas in the array were not active) in order to illustrate how the data rate changes with respect to the number of the transmitters in the network. The total transmit power at the base station was assumed as 1 W and the channel bandwidth as 10 MHz. The results are shown in Fig. 9.

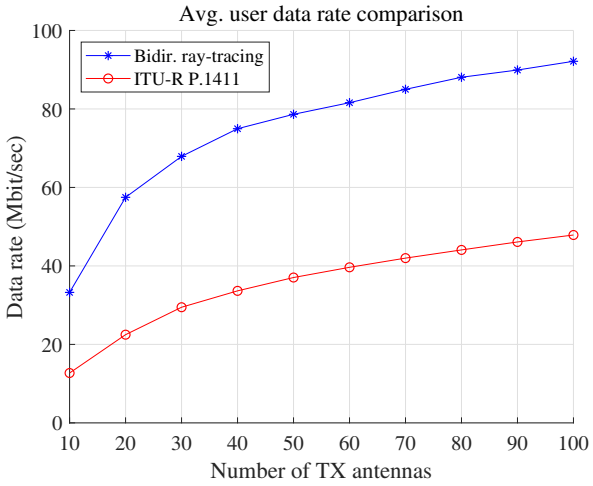

Figure 9. Comparison of average user data rate for ray-tracing and empirical channel models.

The results show that increasing the number of the transmitter antennas yields significant improvements in the performance, as both models indicate a similar behavior. However, the empirical model (ITU-R P.1411) underestimates the data rate by a large margin. In order to recognize the channel characteristics which lead to the difference, two parameters are investigated:

(i) *Channel Orthogonality*: Denoted by $\mathcal{M}$ and computed according to the inner product of the normalized user channels such that

$$\begin{aligned} &\forall (i,j) \in \{1,\ldots,10\} \times \{1,\ldots,10\}, \quad i \neq j, \\ &\mathcal{M} = A(|\hat{\mathbf{H}}_{(i,:)} \cdot \hat{\mathbf{H}}_{(j,:)}|), \end{aligned} \tag{19}$$

where $A(.)$ indicates the arithmetic average.

(ii) *Signal and Interference Power*: Denoted by $\mathcal{S}$ and $\mathcal{I}$, and computed according to the desired signal and interference powers at the receiver locations, respectively. $\mathcal{S}$ and $\mathcal{I}$ are given as

$$\begin{aligned} &\forall i \in \{1,\ldots,10\}, \quad \mathcal{S} = A\left((\mathbf{H}_{(i,:)}\mathbf{g}_{(:,i)})^2\right), \\ &\mathcal{I} = A\left(\sum_{j=1}^{10} (\mathbf{H}_{(i,:)}\mathbf{g}_{(:,j)})^2\right). \end{aligned} \tag{20}$$

Note that the noise power is assumed as constant for all receivers in both models as it only depends on the bandwidth.

The results are shown in Fig. 10.

The channel orthogonality results show that the directional properties of the user channels are accurately captured by the empirical model. As a result, both models are in good agreement. On the

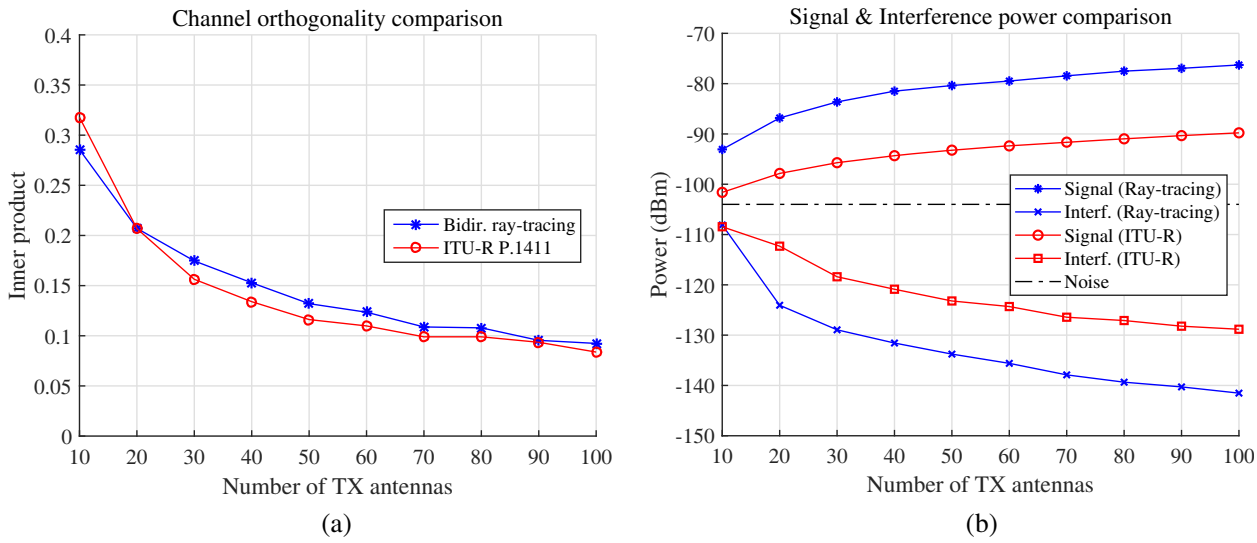

Figure 10. Comparison of bidirectional ray-tracing and ITU-R channel models with respect to (a) channel orthogonality (b) average signal and interference power at the receiver locations.

other hand, the signal and interference power at the receiver locations are predicted quite differently by the models. Compared to the ITU-R model, the signal power is estimated larger while the interference is estimated to be smaller in the bidirectional ray-tracing simulations. Additionally, the differences between the two models grow marginally when a larger number of transmitters is employed at the base station. Such a contrast can be attributed to the disparate directional properties of the channel vectors in the models and the effectiveness of the precoder over these channels as the number of the transmitters change. Even though the orthogonality of the channel vectors are similar in both models, the individual terms of the channel matrix presumably demonstrate a greater distinction according to the bidirectional ray-tracing model. The channel matrix might even be sparse (i.e., not every transmitter may establish a link with every receiver) in the ray-tracing simulations, hence, the massive MIMO precoder can demonstrate a better performance in boosting the signal and suppressing the interference. The key features of the power plots in Fig. 10 can clarify the difference of the data rate. It can be noticed that the discrepancy in the signal power is relatively small and the interference figures are almost identical when 10 transmitters are utilized, yielding only a modest difference in the data rate. The differences become larger when the number of the transmitters is increased, hence, a comparable growth in the difference of the data rate can be observed. Beyond 40 transmitter antennas though, the difference of data rate remains relatively stable where the variation of the data rate becomes mostly dependent on the variation of the signal power as the interference does not affect the SINR significantly due to the relative significance of the noise power over the interference. Since the difference of signal power does not change drastically, the data rate for both models also show similar variation, without any major fluctuation in the difference.

7. CONCLUSION

The average downlink performance in an urban massive MIMO network was characterized by ray-tracing simulations. The problem geometry mostly favors rooftop propagation where multiple diffractions near the optical boundaries might occur. In order to simulate these effects accurately, the bidirectional ray-tracing method was used. The results were compared to an empirical rooftop propagation model. It was found that the channel directionality can be modeled accurately with both approaches, but the path gain might be underestimated in the empirical model. Additionally, a new iterative SBR method, which addresses the inefficiencies of the traditional SBR, was presented. It was shown that the new algorithm can provide accurate results with a smaller number of ray launches, which indicates an improvement in the computation time.

REFERENCES

1. Fukuda, H., T. Nishimura, Y. Ogawa, and T. Ohgane, "Performance evaluation of full-dimension MIMO in indoor line-of-sight environments," *IEEE Wireless Communications and Networking Conference (WCNC)*, 159–164, Gran Canaria, Spain, 2015.
2. Ling, H., R. Chou, and S. Lee, "Shooting and bouncing rays: calculating the RCS of an arbitrarily shaped cavity," *IEEE Transactions on Antennas and Propagation*, Vol. 37, No. 2, 194–205, 1989.
3. Kouyoumjian, R. and P. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *Proceedings of the IEEE*, Vol. 62, No. 11, 1448–1461, 1974.
4. Holm, P. D., "UTD diffraction coefficients for higher order wedge diffracted fields," *IEEE Transactions on Antennas and Propagation*, Vol. 44, No. 6, 879–888, 1996.
5. Walfisch, J. and H. L. Bertoni, "A theoretical model of UHF propagation in urban environments," *IEEE Transactions on Antennas and Propagation*, Vol. 36, No. 12, 1788–1796, 1988.
6. Kanatas, A. G., I. D. Kountouris, G. B. Kostaras, and P. Constantinou, "A UTD propagation model in urban microcellular environments," *IEEE Transactions on Vehicular Technology*, Vol. 46, No. 1, 185–193, 1997.
7. Taygur, M. M., I. O. Sukharevsky, and T. F. Eibert, "Computation of antenna transfer functions with a bidirectional ray-tracing algorithm utilizing antenna reciprocity," *URSI Atlantic Radio Science Conference (AT-RASC)*, Gran Canaria, Spain, 2018.
8. Taygur, M. M., I. O. Sukharevsky, and T. F. Eibert, "A bidirectional ray-tracing method for antenna coupling evaluation based on the reciprocity theorem," *IEEE Transactions on Antennas and Propagation*, Vol. 66, No. 12, 6654–6664, 2018.
9. Collin, R. and F. Zucker, *Antenna Theory* , McGraw-Hill, New York, NY, 1969.
10. Sukharevsky, O., *Electromagnetic Wave Scattering by Aerial and Ground Radar Objects*, CRC Press Inc., Boca Raton, FL, 2014.
11. Gamkrelidze, R. V., *Analysis I: Integral Representations and Asymptotic Methods*, Springer, Berlin, 2011.
12. Xu. F. and Y.-Q. Jin, "Bidirectional analytic ray tracing for fast computation of composite scattering from electric large target over a randomly rough surface," *IEEE Transactions on Antennas and Propagation*, Vol. 57, No. 5, 1495–1505, 2009.
13. Mocker, M. S., M. Schiller, R. Brem, Z. Sun, H. Tazi, T. F. Eibert, and A. Knoll, "Combination of a full-wave method and ray tracing for radiation pattern simulations of antennas on vehicle roofs," *European Conference on Antennas and Propagation (EuCAP)*, Lisbon, Portugal, 2015.
14. Schiller, M., A. Knoll, M. S. Mocker, and T. F. Eibert, "GPU accelerated ray launching for high-fidelity virtual test drives of VANET applications," *Conference on High Performance Computing & Simulation (HPCS)*, Amsterdam, Netherlands, 2015.
15. Brem, R. and T. F. Eibert, "A shooting and bouncing ray (SBR) modeling framework involving dielectrics and perfect conductors," *IEEE Transactions on Antennas and Propagation*, Vol. 63, No. 8, 3599–3609, 2015.
16. Brem, R. and T. F. Eibert, "Multi-radiation center transmitter models for ray tracing," *IEEE Transactions on Antennas and Propagation*, Vol. 60, No. 7, 3382–3388, 2012.
17. Yun, Z. and M. Iskander, "Ray tracing for radio propagation modeling: principles and applications," *IEEE Access*, Vol. 3, 1089–1100, 2015.
18. Shi, D., N. Lv, N. Wang, and Y. Gao, "An improved shooting and bouncing ray method for outdoor wave propagation prediction," *ACES Journal*, Vol. 32, No. 7, 581–585, 2017.
19. Wong, R., *Asymptotic Approximations of Integrals*, SIAM, Philadelphia, PA, 2001.
20. Vogler, L. E., "An attenuation function for multiple knife edge diffraction," *Radio Science*, Vol. 17, No. 6, 1541–1546, 1982.
21. International Telecommunications Union, "Recommendation ITU-R P1411-9: Propagation data and prediction methods for the planning of short rnage outdoor radio communication systems and radio local area networks in the frequency range 300 MHz to 100 GHz," Report, 2017.