Anxel Beam Shrinkage Method and Heterogeneous Computing-Accelerated Full-Image Theory Method Ray Tracing Enabling Massive Outdoor Propagation Modeling

Yongwan Kim, *Member, IEEE*, Hyunjun Yang, *Member, IEEE*, Hooyoung Kim, *Member, IEEE*, Junpyo Jo, *Member, IEEE*, and Jungsuek Oh, *Senior Member, IEEE*

***Abstract*— Despite their accuracy, traditional image theory (IT) ray tracers were previously limited to basic simulation environments with fewer field observation points (FOPs) and lower ray bounce orders due to computational inefficiencies. In this study, we propose a novel full-3-D anxel beam shrinkage (ABS) method and heterogeneous computing-accelerated full-IT method ray-tracing (RT) framework enabling massive outdoor propagation modeling. The proposed framework is divided into three components: 1) visibility preprocessing; 2) visibility tree generation, which introduces a novel ABS method to expedite the creation process and minimize the visibility tree's size; and 3) shadow testing and field calculation, incorporating a heterogeneous computing algorithm designed to efficiently manage numerous FOPs. We also demonstrated that the proposed framework, utilizing both central processing unit (CPU) and graphical processing unit (GPU) parallel computing, is 651 times faster than the IT method solver of WinProp, which supports only CPU parallel computing. Furthermore, it is confirmed that the proposed RT framework can handle 1 × 1 km wide and dense urban outdoor simulation with up to the maximum ray bouncing order of 6 and thousands of FOPs. The proposed RT framework could serve as a foundation for future advancements in IT method RT techniques in complex and massive scenarios, which were previously exclusive to the shooting and bouncing rays method ray tracers.**

***Index Terms*— Asymptotic high-frequency techniques, graphical processing unit (GPU), image theory (IT) ray tracing (RT), RT, shooting and bouncing ray (SBR) techniques, wireless propagation modeling.**

I. Introduction

MILIMETER-WAVE (mmWave) communication systems that require sophisticated manipulation of line-of-sight (LOS) and non-line-of-sight (NLOS) propagation path channels are essential for characterizing deterministic electromagnetic (EM) channels based on channel data obtained from measurement campaigns and ray-tracing (RT) EM analyses [1], [2], [3], [4], [5]. In particular, RT is extensively utilized over mmWave frequencies and would be a preferable tool for 6G channel modeling and cell coverage analysis because of its convenience, whereas actual channel measurement on the 6G frequency band is extremely complex and time-consuming [6]. For outdoor channel modeling and cell-coverage analysis using RT, EM analysis of numerous field observation points (FOPs) in complex and massive simulation environments should be performed [1], [2], [3], [4], [5].

There are two representative RT methods. The first is the shooting and bouncing ray (SBR) method, which is the most widely used RT method that generally provides high computational efficiency; however, its accuracy is limited due to the phase error and cutoff of the ray tube [7]. The second method is the image theory (IT) method, which has high accuracy but relatively low computational efficiency compared to the SBR method due to its shadow testing (ST) and visibility tree generation algorithms [7].

Recently, the frequency range used for communication systems has been increasing to achieve a broad bandwidth, higher data rate, and low latency. If the analysis frequency increases, the SBR method generally produces a higher phase error, which is a byproduct of the ray-tube concept [8], [9], [10]. To maintain the accuracy of the SBR method when the analysis frequency increases, additional rays or additional optimization procedures should be launched, such as stationary point search [8], [9], resulting in larger computation tasks and algorithm complications. In addition, the SBR method has a ray-tube cutoff error that occurs when the wavefront size of the ray tube exceeds the resolution of the simulation environment [11]. As the operating frequency increases, the antenna's main beamwidth narrows to overcome the large path loss [6], resulting in a significant decrease in the number of important strong rays relative to the weak rays. If a ray-tube cutoff error occurs for these few important rays, the overall analysis results will be invalid. Therefore, the total number of source rays should be increased to achieve acceptable accuracy, resulting in a larger computational load [11]. Furthermore, there is ambiguity in determining the number of rays that should be launched to escape the ray-tube cutoff and

Manuscript received 1 October 2023; revised 5 May 2024; accepted 3 June 2024. Date of publication 14 June 2024; date of current version 9 July 2024. This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by the South Korea Government [Ministry of Science and ICT (MIST)] (Advanced and Integrated Software Development for Electromagnetic Analysis) under Grant 2019-0-00098. *(Corresponding author: Jungsuek Oh.)*

The authors are with the Institute of New Media and Communications (INMC) and the Department of Electrical and Computer Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: yongwankim@snu.ac.kr; guswns6217@snu.ac.kr; kimhy3010@snu.ac.kr; junpyo99@snu.ac.kr; jungsuek@snu.ac.kr).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAP.2024.3411793.

Digital Object Identifier 10.1109/TAP.2024.3411793

achieve sufficient accuracy for various analysis frequencies and simulation environments.

However, for the IT method, the accuracy is affected only by geometric mesh approximations and the floating-point number precision of computers [11]. Therefore, it presents zero systemic errors without increasing the computational load or complications of the algorithm, even if the analysis frequency increases. In addition, its accuracy is independent of the resolution of the simulation environment and the main beamwidth of the antenna, to ensure that explicit simulation is feasible without any ambiguity, such as the ray-tube cutoff and the number of rays to be launched to achieve sufficient accuracy [11]. Considering the aforementioned facts, the IT method is a potential RT method for analyzing 6G communication systems. However, the conventional IT method suffers from a prohibitively long computation time when the propagation scenario treats high-order maximum ray bounce, numerous FOPs, and facet modeling simulation environments; therefore, its utilization is limited to simple simulation environments [10], [12].

A. Related Works

To overcome the long computation time of the IT method RT, [13], [14], [15], [16], [17], [18], [19], [20], and [21] utilized the angular Z-buffer (AZB) algorithm in the IT method and achieved a computation time reduction of 90% [10]. Moreover, as indicated in [15] and [16], the adaptability of the AZB algorithm spans from outdoor to indoor environments. Additionally, in [17], it is illustrated that the AZB algorithm is not limited to flat facets but also accommodates curved facets. Furthermore, studies enhancing accuracy by rectifying the minimum and maximum elevation angles of facets are conducted in [18] and [19]. Alongside these studies on accuracy improvement, research on accelerating the AZB method through the combination of other acceleration techniques is introduced in [20] and [21]. In [20], a combination of AZB, space volumetric partitioning (SVP), and the depth-limited search method is presented to accelerate the radar cross-sectional analysis of complex targets. An acceleration method in [21] is based on combining AZB with the SVP algorithm and the A* heuristic search method to address multiple bounces in various propagation problems. The performance comparison between this approach and the proposed RT method will be presented in Section IV.

B. Motivations

The AZB-combined IT method divides the space into multiple angular spaces around a source, that is, Tx or image Tx. Subsequently, the source radiates the beam in the shape of anxel (which is the powerful concept of the AZB method), enclosing the reflection facet. The same procedure was performed for the facets inside the beam, up to the maximum ray-bouncing order. Here, the conventional AZB method creates an anxel beam enclosing the entire reflecting facet, including the portion that is not illuminated by the beam of the former bouncing order; that is, the portion of the facet that does not participate in multiple reflections. Consequently, the number of unnecessary computational tasks and memory usage increases exponentially as the ray-bouncing order increases. If a simulation scenario includes diffraction, this inefficiency worsens because the diffraction anxel beam generally illuminates exceedingly more facets than the reflection beam, resulting in the infeasibility of a massive analysis. Therefore, to avoid this inefficiency, the beam should be radiated only to the portion illuminated by the beam of the former bouncing order. Although techniques for discarding the unilluminated portion are presented in [22] and [23], these are only applicable to 2-D and 2.5-D structures, respectively, consisting solely of infinitely tall vertical planes with or without horizontal ground, feasible only when the base station and the mobiles are assumed to remain well below rooftop height. Under this assumption, eliminating the unilluminated portion becomes relatively straightforward, requiring no complex formulation and algorithm. However, when dealing with arbitrary 3-D structures, the algorithm becomes much more complex, requiring intricate formulations. These algorithms and formulations are not found in the existing literature.

The conventional AZB method creates DAZB matrices that store information on the facets contained in the angular region. Using DAZB matrices, the ST can be accelerated by considering only the facets inside the angular region to which the ray belongs. However, because DAZB matrices should be created for each Tx and image Tx when a high-order ray bouncing order is considered, they can occupy a prohibitively excessive memory, precluding acceleration through parallel computing. Although the parallel computing approach has been widely used to resolve the limited computational speed of RT, in most cases, it has been applied to the SBR method [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35]. On the other hand, in IT method RT, their utilization is infrequent, and the majority of these techniques are not geared toward general IT methods but instead toward hybrid methods, which have notable limitations. For example, in [10], a graphical processing unit (GPU)-based kD-tree-accelerated beam tracing (GKBT) method was presented. Although the GKBT method was approximately ten times faster than the RT techniques before the GKBT was developed, it cannot support the reflection of diffracted fields due to using the ray congruence concept for diffraction. The reflection of diffracted fields is crucial in outdoor wave propagation; hence, GKBT is unsuitable for analyzing outdoor environments. Moreover, although Shing-Min Liu and Tan [36] and Rainer et al. [37] presented the GPU-accelerated IT method RT technique, these studies mainly focused on visibility preprocessing and the diffuse scattering effect, respectively, and only limited information on the RT acceleration technique was provided. Given the aforementioned considerations, it is noteworthy that, to the best of the authors' knowledge, there is presently no existing literature offering a comprehensive procedure for applying central processing unit (CPU) and GPU parallel computing to general IT method RT, sufficient for program reproduction.

C. Contributions

In response to the aforementioned gap, we propose a novel full-IT method RT framework enabling massive outdoor propagation modeling. Our contributions consist of the following two aspects.

1) We propose the anxel beam shrinkage (ABS) method, which eliminates the unilluminated portion by the beam of the previous bouncing order, significantly enhancing the computational efficiency of the AZB method RT. Unlike the techniques in [22] and [23], the ABS method can be applied to arbitrary facet-based 3-D structures. It extends beyond realistic urban scenarios with rooftop heights to encompass other scenarios like ship and automobile structures, among others. Also, it can be seamlessly integrated into the conventional AZB method ray tracer.
2) We propose a novel heterogeneous computing-based ABS-accelerated IT (HAIT) method RT framework. This method offers a comprehensive guide for implementing heterogeneous computing, specifically CPU/GPU parallel computing, tailored to the general IT method RT. HAIT supports an unlimited number of reflections, combined with one order of diffraction, making it suitable for outdoor scenario analysis. Additionally, HAIT can efficiently handle thousands of FOPs in massive outdoor scenarios within a reasonable computation time.

The remainder of this article is organized as follows. In Section II, the numerical formulations of the ABS method for multiple reflections, reflection–diffraction, and diffraction–reflection propagation sequences are introduced. In Section III, the HAIT method RT framework is in detail. In Section IV, the accuracy and efficiency of the ABS and HAIT methods are validated using simple and complex outdoor urban scenarios. Finally, the conclusions are presented in Section V.

II. ABS Method

In this section, we propose the ABS method that reduces the size of the anxel beam as the ray bouncing order increases, leading to an exponential reduction of size and acceleration of generation speed of the visibility tree as maximum bouncing order increases, compared to the conventional AZB IT method. The ABS consists of three major cases: 1) multiple reflections; 2) reflection–diffraction; and 3) diffraction–reflection.

The general spherical coordinate system [10] was used for the ABS.

A. Multiple Reflections

A simple 2-D description of the second-order multiple reflections of the ABS is shown in Fig. 1, where the first and second reflecting facets are represented by the first and second facets, respectively. First, for the first-order reflection, we generate an image Tx for the first facet, which is represented as the first image Tx in Fig. 1. Subsequently, from the image Tx, we see the first facet and determine its $\phi - \theta$ AZB rectangle, which is a rectangle determined by four spherical angular margins ($\phi_{min}, \phi_{max}, \theta_{min}$, and θ_{max}) of the reflecting facet (see Fig. 2). The first anxel beam originates from the Tx image and extends toward the AZB rectangle, as illustrated in Fig. 1. Next, we determine the AZB rectangle and its corresponding anxel beam for the facet inside the first anxel beam, i.e., second facet, and determine the

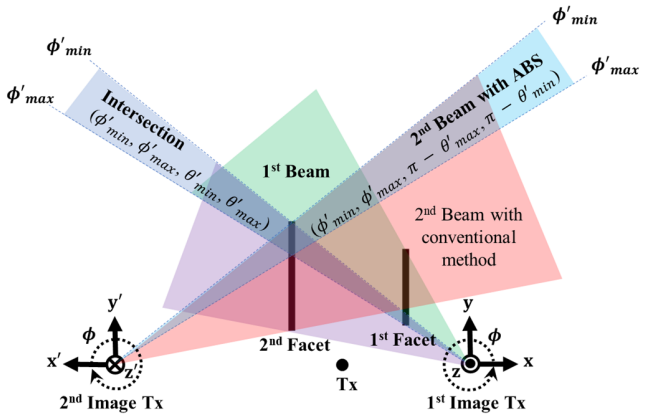


Fig. 1. Two-dimensional description of the second-order multiple reflections of ABS.

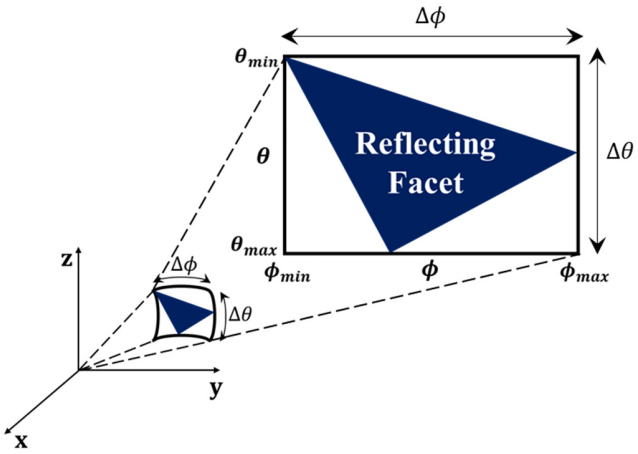


Fig. 2. Example of $\phi - \theta$ AZB rectangle.

intersection between two anxel beams, i.e., $\phi'_{min}, \phi'_{max}, \theta'_{min}$, and θ'_{max} in Fig. 1. Subsequently, to analyze the second-order reflection, we generated another image Tx of the second facet and transformed the basis as the second image Tx, as shown in Fig. 1. The first and second new bases, $\mathbf{x}'$ and $\mathbf{y}'$, are $\mathbf{x}$ and $\mathbf{y}$ mirrored to the second facet, respectively, and the last new basis, $\mathbf{z}'$, is $\mathbf{x}' \times \mathbf{y}'$. Subsequently, we can directly use the intersected anxel beam, that is, the reduced anxel beam, for the second bouncing order after replacing θ'_{min} and θ'_{max} with $\pi - \theta'_{max}$ and $\pi - \theta'_{min}$, respectively, which means that the second anxel beam is launched from the second image Tx to the AZB rectangle having angular margins of $\phi'_{min}, \phi'_{max}, \pi - \theta'_{max}$, and $\pi - \theta'_{min}$, as shown in Fig. 1. Subsequently, an identical procedure was performed up to the maximum ray-bouncing order.

Fig. 1 shows that the size of the second anxel beam with the ABS is significantly reduced compared with that of the conventional method, and this efficiency improvement increases exponentially as the bouncing order increases. Although only the 2-D case is illustrated in Fig. 1 for simplicity, the ABS method for multiple reflections can be applied to facets with any 3-D location and orientation.

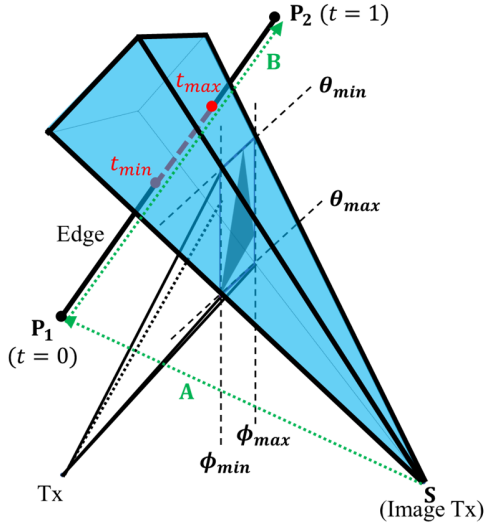


Fig. 3. Problem description of reflection–diffraction case of ABS.

B. Reflection–Diffraction

A simple problem description of the reflection–diffraction case of ABS is shown in Fig. 3. In Fig. 3, an AZB rectangle is formed for a facet and a reflection anxel beam, that is, a blue beam, is launched from an image Tx to the AZB rectangle. At this time, an edge penetrates the reflection anxel beam. At this moment, the conventional method creates the diffraction anxel beam extended from the entire edge. However, the portion of the edge that has the possibility of participation in the reflection–diffraction is only the portion inside the reflection anxel beam, which is represented as a red dotted line in Fig. 3 and we can shrink the anxel beam by considering only this portion. This is the portion between t_{min} and t_{max} where t is the normalized length of the edge, which implies the start point and endpoint of the edge are represented as $t = 0$ and $t = 1$, respectively, as shown in Fig. 3. Although the exact portion of the edge that participates the diffraction can be easily found using simple beam tracing algorithm in [10] when a first reflection–diffraction is considered, the beam tracing algorithm in [10] requires complex beam splitting procedure when multiple reflection–diffraction is considered, resulting in exponential increase of computational load and algorithm complexity. Rather, we use the AZB rectangle concept for reflection–diffraction and we can directly use the shrunk anxel beam through multiple reflections described in Section II-A for multiple reflection–diffraction. The t_{min} and t_{max} can be expressed as follows:

$$t_{min} = \max\left(t_{\phi min}, t_{\theta min}, 0\right) \tag{1}$$

$$t_{max} = \min\left(t_{\phi max}, t_{\theta max}, 1\right) \tag{2}$$

where t_ϕ and t_θ are t values that the edge intersects ϕ- and θ-axes sides of the reflection anxel beam, respectively.

First, to formulate t_ϕ, we represent the edge line as $\mathbf{E}$ in the following eqution:

$$\mathbf{E} = \mathbf{P_1} + t(\mathbf{P_2} - \mathbf{P_1}) \tag{3}$$

where $\mathbf{P_1}$ and $\mathbf{P_2}$ are the starting and ending points of the edge, respectively. If we represent the location of image Tx as $\mathbf{S}$, the vector from image Tx to a point on the edge line is represented as follows:

$$\mathbf{SE} = \mathbf{A} + t\mathbf{B} \tag{4}$$

where $\mathbf{A}$ and $\mathbf{B}$ represent $\mathbf{P_1} - \mathbf{S}$ and $\mathbf{P_2} - \mathbf{P_1}$, respectively. Currently, we formulate $\tan\phi$ of a point on the edge line considering the image Tx as the origin as follows:

$$\tan\phi = \frac{A_y + tB_y}{A_x + tB_x}. \tag{5}$$

After performing some algebra with (5), we obtain

$$t(\phi) = \frac{-A_x \tan\phi + A_y}{B_x \tan\phi - B_y} \tag{6}$$

which is the t value representing the edge point that is located on the angle of ϕ. Here, $t(\phi)$ equals $t(\phi + \pi)$ because $\tan\phi$ equals $\tan(\phi + \pi)$; therefore, a validation procedure of the t value should be performed by confirming the atan2 function of $\mathbf{A} + t(\phi)\mathbf{B}$ equals ϕ. There are five possible cases where the entire or a part of the edge line can be located inside the ϕ-domain of the reflection anxel beam, and these are shown in Fig. 4. Fig. 4(a) represents a case where the edge line is directed to the z-axis, and entire edge line is inside the ϕ-domain of the anxel beam, resulting in $t_{\phi min} \to -\infty$ and $t_{\phi max} \to \infty$. Fig. 4(b) shows the case where the edge line penetrates the entire ϕ range of the anxel beam to ensure that two t_ϕ, that is, $t(\phi_{Rmin})$ and $t(\phi_{Rmax})$ where ϕ_{Rmin} and ϕ_{Rmax} are ϕ-domain angular margins of anxel beam, are assigned to the $t_{\phi min}$ and $t_{\phi max}$, respectively, depending on their magnitude. Fig. 4(c) shows the case where one t_ϕ occurs at one ϕ-domain angular margin and the other t_ϕ occurs at the other angular margin $+\pi$, such that the first t_ϕ and $\pm\infty$ are allocated to the $t_{\phi min}$ and $t_{\phi max}$, respectively, depending on their magnitude. Fig. 4(d) shows the case where the projection of the edge to the xy plane (z = 0) passes through the $\mathbf{S}$ to ensure that $t_\phi = t(\phi_{Rmin}) = t(\phi_{Rmax})$ and this t_ϕ and $\pm\infty$ is allocated to $t_{\phi min}$ and $t_{\phi max}$, respectively, depending on their magnitude. Fig. 4(e) shows the case where one t_ϕ occurs at one ϕ-domain angular margin and the other angular margin is parallel with the edge line such that the t_ϕ and $\pm\infty$ are allocated to $t_{\phi min}$ and $t_{\phi max}$, respectively, depending on their magnitude. The pseudocode for the detailed determination procedure of $t_{\phi min}$ and $t_{\phi max}$ is shown in Algorithm 1. If the reflecting facet is located at $\pm$z-direction from the $\mathbf{S}$, that is, a ray with a direction of $\pm$z launched from $\mathbf{S}$ intersects the reflecting facet, we do not conduct Algorithm 1 and only set $t_{\phi min}$ and $t_{\phi max}$ to $-\infty$ and ∞, respectively, because ϕ_{Rmin} and ϕ_{Rmax} are 0 and 2π, respectively.

Next, to derive t_θ, we formulate $\cot\theta$ of a point on the edge line considering the $\mathbf{S}$ as the origin as follows:

$$\cot\theta = \frac{A_z + tB_z}{\sqrt{(A_x + tB_x)^2 + \left(A_y + tB_y\right)^2}}. \tag{7}$$

After performing some algebra with (7), we obtain

$$t(\theta) = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha} \tag{8}$$

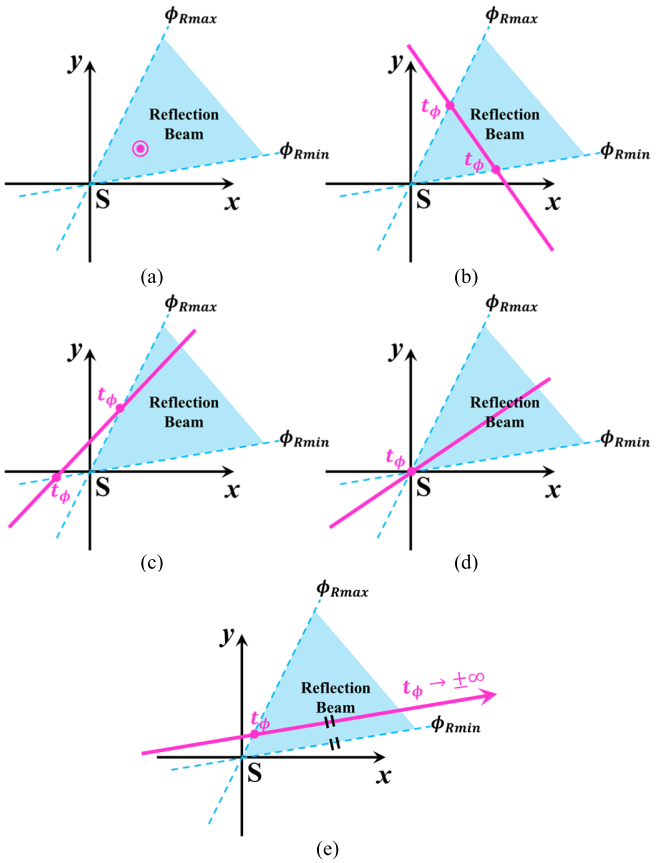


Fig. 4. Five possible cases where entire or part of edge line can be located inside the ϕ-domain of the reflection anxel beam.

where

$$\alpha = \left(B_x^2 + B_y^2\right)\cot^2\theta - B_z^2 \tag{9}$$

$$\beta = 2\left[\left(A_x B_x + A_y B_y\right)\cot^2\theta - A_z B_z\right] \tag{10}$$

$$\gamma = \left(A_x^2 + A_y^2\right)\cot^2\theta - A_z^2. \tag{11}$$

Here, $t(\theta)$ equals $t(\pi-\theta)$ because $\cot^2\theta$ equals $\cot^2(\pi-\theta)$; therefore, a validation procedure of the correct t value should be performed by substituting the calculated two t from (8) to the following equation:

$$\theta(t) = \cot^{-1}\frac{A_z + tB_z}{\sqrt{(A_x + tB_x)^2 + \left(A_y + tB_y\right)^2}} \tag{12}$$

There are three mathematical facts: 1) $t(\theta) = t(\pi-\theta)$; 2) $t(\theta)$ has maximally two solutions; and 3) the limit of (12) as t approaches infinity is expressed as follows:

$$\lim_{t\to\pm\infty}\theta(t) = \cot^{-1}\left(\pm B_z/\sqrt{B_x^2 + B_y^2}\right). \tag{13}$$

Based on the above mathematical facts, we derive two other mathematical facts: 1) $\theta(t)$ can have maximally one extreme value and 2) local minimum and local maximum values of $\theta(t)$ cannot exist in the area where $\theta(t) > \pi/2$ and $\theta(t) < \pi/2$, respectively. Based on these two mathematical facts, we derive a total of 20 possible cases, which is shown in Fig. 5, where the entire or a part of the edge line can be located inside the θ-domain of the reflection anxel beam. Fig. 5(a)–(c) depicts

Algorithm 1 Determination of $t_{\phi min}$ and $t_{\phi max}$

Input:
ϕ_{Rmin}, ϕ_{Rmax}, $\mathbf{P_1}$, $\mathbf{P_2}$, and S
Output:
$t_{\phi min}$ and $t_{\phi max}$.
function tPhiMinMax.
  if $P_{1x} - P_{2x} = P_{1y} - P_{2y} = 0$ **then**
    if atan2($P_{1y} - S_y$, $P_{1x} - S_x$) is between ϕ_{Rmin} and ϕ_{Rmax} **then**.
      $t_{\phi min}, t_{\phi max} \leftarrow -\infty, \infty$ ▷ Fig. 4(a)
    else
      $t_{\phi min}, t_{\phi max} \leftarrow$ NaN, NaN
        ▷ Edge line is located outside the reflection beam
    end if
    return $t_{\phi min}, t_{\phi max}$
  end if
  $\mathbf{A}, \mathbf{B} \leftarrow \mathbf{P_1} - \mathbf{S}, \mathbf{P_2} - \mathbf{P_1}$
  $t_{\phi 1}, t_{\phi 2} \leftarrow t(\phi_{Rmin}), t(\phi_{Rmax})$
  $\mathbf{d_1}, \mathbf{d_2} \leftarrow \mathbf{A} + t_{\phi 1}\mathbf{B}, \mathbf{A} + t_{\phi 2}\mathbf{B}$
  $\phi_1, \phi_2 \leftarrow$ atan2(d_{1y}, d_{1x}), atan2(d_{2y}, d_{2x})
  if ϕ_1=ϕ_{Rmin} and ϕ_2=ϕ_{Rmax} **then** ▷ Fig. 4(b)
    $t_{\phi min}, t_{\phi max} \leftarrow \min(t_{\phi 1}, t_{\phi 2}), \max(t_{\phi 1}, t_{\phi 2})$
  else if (ϕ_1=ϕ_{Rmin} and $\phi_2 = \phi_{Rmax} + \pi$) or
      ($\phi_1 = \phi_{Rmin} + \pi$ and $\phi_2 = \phi_{Rmax}$) **then** ▷ Fig. 4(c)
    $t_{\phi proper} \leftarrow$ The t value that matches ϕ_{Rmin} or ϕ_{Rmax} between $t_{\phi 1}$ and $t_{\phi 2}$
    $t_{\phi improper} \leftarrow$ The t value that matches $\phi_{Rmin} + \pi$ or $\phi_{Rmax} + \pi$ between $t_{\phi 1}$ and $t_{\phi 2}$
    if $t_{\phi proper} - t_{\phi improper} > 0$ **then**
      $t_{\phi min}, t_{\phi max} \leftarrow t_{\phi proper}, \infty$
    else
      $t_{\phi min}, t_{\phi max} \leftarrow -\infty, t_{\phi proper}$
    end if
  else if $t_{\phi 1} = t_{\phi 2}$ **then**
    $\mathbf{C} \leftarrow \mathbf{A} + (t_{\phi 1} + \alpha)\mathbf{B}$ ▷ α is arbitrary positive constant
    $\phi_0 \leftarrow$ atan2(C_y, C_x)
    if ϕ_0 is between ϕ_{Rmin} and ϕ_{Rmax} **then** ▷ Fig. 4(d)
      $t_{\phi min}, t_{\phi max} \leftarrow t_{\phi 1}, \infty$
    else if ϕ_0 is between $\phi_{Rmin} + \pi$ and $\phi_{Rmax} + \pi$ **then**
        ▷ Fig. 4(d)
      $t_{\phi min}, t_{\phi max} \leftarrow -\infty, t_{\phi 1}$
    else
      $t_{\phi min}, t_{\phi max} \leftarrow$ NaN, NaN
    end if
  else if (ϕ_1=ϕ_{Rmin} and $t_{\phi 2} \to \pm\infty$) or (ϕ_2=ϕ_{Rmax} and $t_{\phi 1} \to \pm\infty$) ▷ Fig. 4(e).
    $t_{\phi finite} \leftarrow$ Thefinite$t$ value between $t_{\phi 1}$ and $t_{\phi 2}$
    $\mathbf{C} \leftarrow \mathbf{A} + (t_{\phi finite} + \alpha)\mathbf{B}$
      ▷ α is arbitrary positive constant
    if atan2(C_y, C_x) is between ϕ_{Rmin} and ϕ_{Rmax} **then**
      $t_{\phi min}, t_{\phi max} \leftarrow t_{\phi finite}, \infty$
    else
      $t_{\phi min}, t_{\phi max} \leftarrow -\infty, t_{\phi finite}$
    end if
  else
    $t_{\phi min}, t_{\phi max} \leftarrow$ NaN, NaN
  end if
  return $t_{\phi min}, t_{\phi max}$
end function

cases where all t_θ, that is, $t(\theta_{Rmax})$ and $t(\theta_{Rmin})$ where θ_{Rmin} and θ_{Rmax} are θ-domain angular margins of anxel beam, are invalid. The edge line in the case of Fig. 5(a) and (b) is entirely lied within the θ-domain of the anxel beam, resulting in $t_{\theta min} \to -\infty$ and $t_{\theta max} \to \infty$, and does not intersect or intersects the **S**, i.e., the origin of the anxel beam, respectively. On the other hand, Fig. 5(c) represents a case where the edge line intersects the **S** but only one section based on **S** is inside the θ-domain of the anxel beam to ensure that the t value where the edge line intersects **S** and $\pm\infty$ is allocated to the $t_{\theta min}$ and $t_{\theta max}$, respectively, depending on their magnitude. Fig. 5(d) and (e) represents cases that only one t_θ, that is, $t(\theta_{Rmax})$ or $t(\theta_{Rmin})$ value is valid, such that the valid t_θ and $\pm\infty$ are allocated to the $t_{\theta min}$ and $t_{\theta max}$,

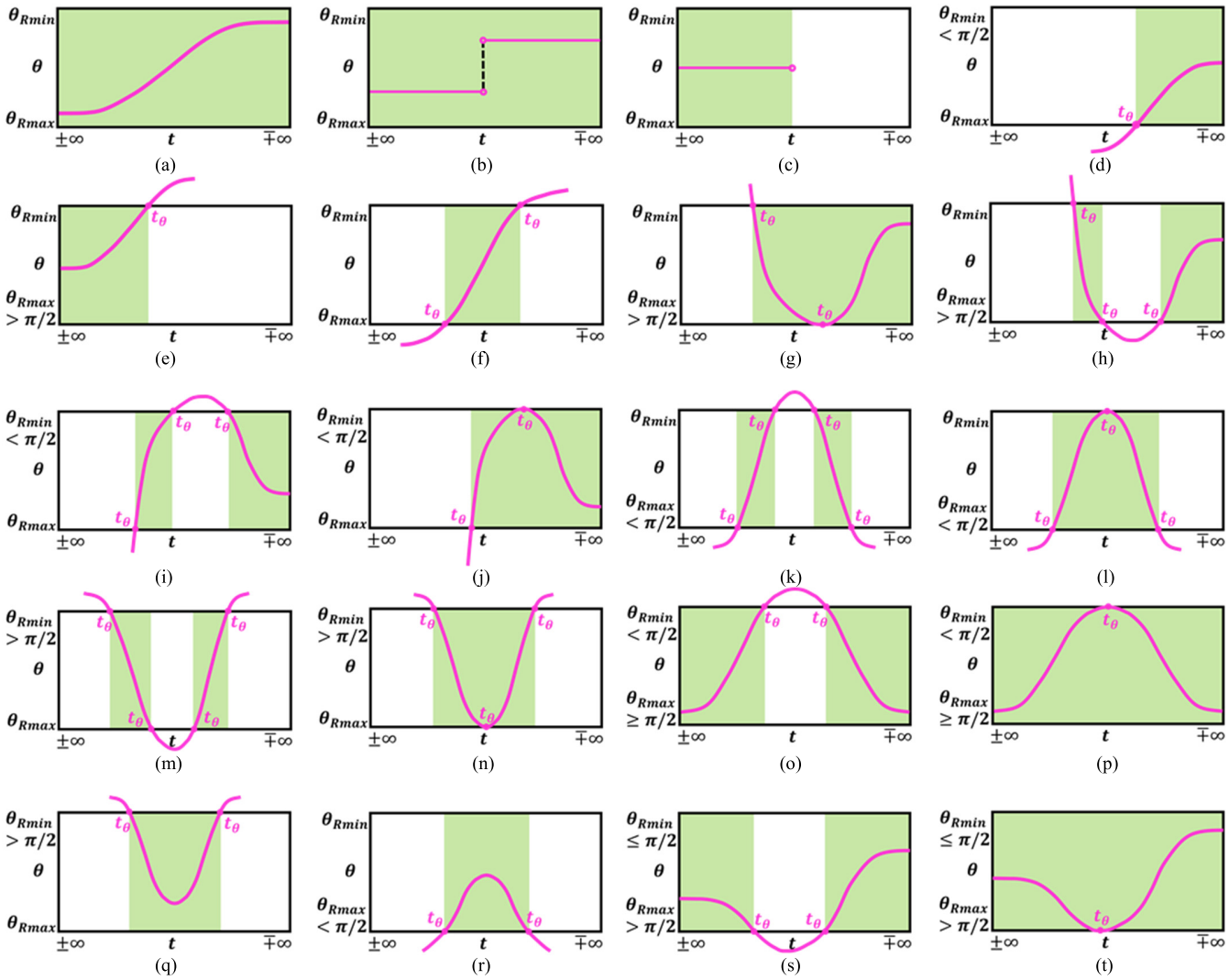


Fig. 5. Twenty possible cases where entire or a part of edge line can be located inside the θ-domain of the reflection anxel beam. The green part is range of t inside θ-domain of the reflection anxel beam.

respectively, depending on their magnitude. Fig. 5(f) shows the case where one $t(\theta_{Rmax})$ and one $t(\theta_{Rmin})$ are valid to ensure that the $t(\theta_{Rmax})$ and $t(\theta_{Rmin})$ are assigned to the $t_{\theta min}$ and $t_{\theta max}$, respectively, depending on their magnitude. Fig. 5(g)–(j) shows cases where only one of the t_θ values, among each two $t(\theta_{Rmax})$ and $t(\theta_{Rmin})$, is invalid, such that the smallest or largest valid t_θ and $\pm\infty$ are allocated to the $t_{\theta min}$ and $t_{\theta max}$, respectively, depending on their magnitude. We note that cases in Fig. 5(g) and (j) have equal roots on $t(\theta_{Rmax})$ and $t(\theta_{Rmin})$, respectively. Also, although cases like in Fig. 5(h) and (i) show beam splitting, for simplicity, we ignored it and merged the beams in this study. Fig. 5(k)–(n) depicts cases where all t_θ values are valid to ensure that the smallest and largest t_θ are allocated to the $t_{\theta min}$ and $t_{\theta max}$, respectively. Fig. 5(o)–(q) represents cases where all $t(\theta_{Rmin})$ are valid but all $t(\theta_{Rmax})$ are invalid. We note that cases in Fig. 5(o) and (p) are possible only when $\theta_{Rmin} < \pi/2$ and $\theta_{Rmax} \geq \pi/2$. In these cases, $-\infty$ and ∞ are assigned to $t_{\theta min}$ and $t_{\theta max}$, respectively. On the other hand, the case in Fig. 5(q) is possible only when $\theta_{Rmin} > \pi/2$ and the two $t(\theta_{Rmin})$ are allocated to the $t_{\theta min}$ and $t_{\theta max}$, respectively, depending on their magnitude. Fig. 5(r)–(t) shows cases where all $t(\theta_{Rmin})$ are invalid but all $t(\theta_{Rmax})$ are valid. The case in Fig. 5(r) is possible only when $\theta_{Rmax} < \pi/2$ and the two $t(\theta_{Rmax})$ are allocated to the $t_{\theta min}$ and $t_{\theta max}$, respectively, depending on their magnitude. On the other hand, cases in Fig. 5(s) and (t) are possible only when $\theta_{Rmin} \leq \pi/2$ and $\theta_{Rmax} > \pi/2$. In these cases, $-\infty$ and ∞ are assigned to $t_{\theta min}$ and $t_{\theta max}$, respectively.

The pseudocode for the detailed determination procedure of $t_{\theta min}$ and $t_{\theta max}$ is shown in Algorithm 2. As mentioned earlier, beam splitting occurs in some cases, for example, in Fig. 5(h) and (i); however, for simplicity, we ignored it and merged the beams also in Algorithm 2. Implementing beam splitting promises to enhance the algorithm's performance, representing a valuable direction for future work.

C. Diffraction–Reflection

First, we adopted the $\phi-t$ anxel beam system and transform basis as shown in Fig. 6 to conveniently handle the diffraction. For the first diffraction $\phi-t$ anxel beam, $\phi_{min} = 0$, $\phi_{max} = n\pi$, and $t_{min} = 0$ and $t_{max} = 1$ (first anxel beam). Subsequently,

Algorithm 2 Determination of $t_{\theta min}$ and $t_{\theta max}$

Input:
$\theta_{Rmin}, \theta_{Rmax}$, **A**, and **B**
Output:
$t_{\theta min}$ and $t_{\theta max}$

function tThetaMinMax
$t_{Rmin+}, t_{Rmin-} \leftarrow t(\theta_{Rmin})$ with $\pm$ sign
$t_{Rmax+}, t_{Rmax-} \leftarrow t(\theta_{Rmax})$ with $\pm$ sign
if all $t_{Rmin\pm}$ and $t_{Rmax\pm}$ invalid **then**
  if $\mathbf{A} \times \mathbf{B} \neq \mathbf{0}$ **then** ▷ Expanded edge line not intersects origin
    if $\theta_{Rmin} < \theta(\delta) < \theta_{Rmax}$ **then**
      ▷ Fig. 5(a), δ: arbitrary real value
      $t_{\theta min}, t_{\theta max} \leftarrow -\infty, \infty$
    else
      $t_{\theta min}, t_{\theta max} \leftarrow$ NaN, NaN
    end if
  else ▷ Expanded edge line intersects origin
    $t_0 \leftarrow value that satisfies \mathbf{A} + t_0\mathbf{B} = \mathbf{0}$
    if $\theta_{Rmin} < [\theta(t_0 + \varepsilon)$ and $\theta(t_0 - \varepsilon)] < \theta_{Rmax}$ **then**
      ▷ Fig. 5(b), ε: arbitrary positive value
      $t_{\theta min}, t_{\theta max} \leftarrow -\infty, \infty$
    else if $\theta_{Rmin} < \theta(t_0 + \varepsilon) < \theta_{Rmax}$ **then** ▷ Fig. 5(c)
      $t_{\theta min}, t_{\theta max} \leftarrow t_0, \infty$
    else if $\theta_{Rmin} < \theta(t_0 - \varepsilon) < \theta_{Rmax}$ **then** ▷ Fig. 5(c)
      $t_{\theta min}, t_{\theta max} \leftarrow -\infty, t_0$
    else
      $t_{\theta min}, t_{\theta max} \leftarrow$ NaN, NaN
    end if
  end if
else if only one of $t_{Rmin\pm,Rmax\pm}$ valid ▷ Fig. 5(d), (e)
  if $\theta_{Rmin} < \theta[($valid $t_{Rmin\pm,Rmax\pm}) + \varepsilon] < \theta_{Rmax}$ **then**
    $t_{\theta min}, t_{\theta max} \leftarrow$ valid $t_{Rmin\pm,Rmax\pm}, \infty$
  else
    $t_{\theta min}, t_{\theta max} \leftarrow -\infty,$ valid $t_{Rmin\pm,Rmax\pm}$
  end if
else if one of $t_{Rmin\pm}$ and $t_{Rmax\pm}$ valid ▷ Fig. 5(f)
    $t_{\theta min}, t_{\theta max} \leftarrow$ min. and max.of valid $t_{Rmin\pm,Rmax\pm}$
else if only one of $t_{Rmin\pm,Rmax\pm}$ invalid **then** ▷ Fig. 5(g)-(j)
    $t_{alone} \leftarrow$ valid solution of $t_{Rmin\pm}$ or $t_{Rmax\pm}$ pair containing invalid solution
    $t_{together} \leftarrow$ any solution of $t_{Rmin\pm}$ or $t_{Rmax\pm}$ pair containing two valid solutions
    if $t_{together} - t_{alone} > 0$ **then**
      $t_{\theta min}, t_{\theta max} \leftarrow t_{alone}, \infty$
    else
      $t_{\theta min}, t_{\theta max} \leftarrow -\infty, t_{alone}$
    end if
else if all $t_{Rmin\pm,Rmax\pm}$ valid **then** ▷ Fig. 5(k)-(n)
    $t_{\theta min}, t_{\theta max} \leftarrow$ min.andmax. of $t_{Rmin\pm,Rmax\pm}$
else if all $t_{Rmin\pm}$ and none of $t_{Rmax\pm}$ valid **then**
  if $\theta_{Rmin} < \pi/2$ **then** ▷ Fig. 5(o), (p)
    $t_{\theta min}, t_{\theta max} \leftarrow -\infty, \infty$.
  else ▷ Fig. 5(q)
    $t_{\theta min}, t_{\theta max} \leftarrow$ min. and max. of $t_{Rmin\pm}$
  end if
else if all $t_{Rmax\pm}$ and none of $t_{Rmin\pm}$ valid **then**
  if $\theta_{Rmax} < \pi/2$ **then** ▷ Fig. 5(r)
    $t_{\theta min}, t_{\theta max} \leftarrow$ min. and max. of $t_{Rmax\pm}$
  else ▷ Fig. 5(s), (t)
    $t_{\theta min}, t_{\theta max} \leftarrow -\infty, \infty$
  end if
else
  $t_{\theta min}, t_{\theta max} \leftarrow$ NaN, NaN
end if
  return $t_{\theta min}, t_{\theta max}$
end function

we determine the facet inside the $\phi - t$ anxel beam. Next, we determine the $\phi - t$ anxel beam (second anxel beam) from the edge, enclosing the facet located inside the first anxel beam. ϕ_{min} and ϕ_{max} of the second anxel beam can be easily determined by calculating ϕ of three vertices of the facet and comparing their magnitude. However, to determine the exact values of t_{min} and t_{max} enclosing the facet, a nonlinear equation must be solved, which is challenging and time-consuming. Rather, we calculated the values for the bounding region enclosing the facet, as shown in Fig. 6. This bounding region is z′-directed column-shaped top with two arcs whose center

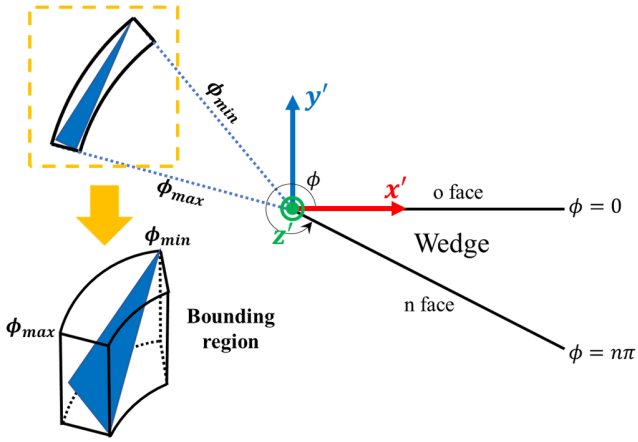


Fig. 6. New basis and bounding region for diffraction–reflection case of ABS.

is the diffracting edge. The radii of the outer and inner arcs are the horizontal ($x'y'$) furthest and closest distance from the edge to the facet, respectively, and the minimum and maximum z′ components of the bounding region are the minimum and maximum z′ components of the facet. We note that t_{min} and t_{max} of the bounding region are determined by four arcs: the lower and upper arcs. In addition, each point on the same arc is mapped to the same t value because the diffraction rays are spread in the shape of a Keller's cone [38]. To determine the t-value where the arc is mapped, we must solve two equations derived from Keller's law [39], that is, $\hat{\mathbf{TD}} \cdot \mathbf{B} = \hat{\mathbf{DR}} \cdot \mathbf{B}$ and $\mathbf{D} = \mathbf{P_1} + t\mathbf{B}$ where $\mathbf{T}, \mathbf{R}, \mathbf{D}, \mathbf{P_1}$, and $\mathbf{B}$ are locations of Tx, any point on the arc, diffraction point, edge start point, and vector from the edge start point to endpoint, respectively, and these are shown in Fig. 7. After performing algebra using the above equations, we formulate t where the arc is mapped as follows:

$$t_{arc} = \frac{D_{z'} - P_{1z'}}{P_{2z'} - P_{1z'}} \tag{14}$$

where

$$D_{z'} = \frac{r_T R_{z'} + r_R T_{z'}}{r_T + r_R} \tag{15}$$

and r_T and r_R, as shown in Fig. 7, are the horizontal distances between the edge and T, R, respectively. At this point, the proposed bounding region concept can also be applied to the facet located $\pm z'$-direction of the edge, that is, the bounding region has a cylindrical shape, by setting r_R of the inner arc to 0.

Using (14) and (15), we can calculate the t_{min} and t_{max} of the bounding region by calculating all four arcs of the bounding region and comparing their magnitudes. However, considering the relative position between the Tx and the bounding region, we can calculate t_{min} and t_{max} directly without a magnitude comparison by calculating t_{arc} for only two arcs. There are three cases for the relative position between the Tx and bounding region, as shown in Fig. 8. Fig. 8(a)–(c) shows the cases where $T_{z'}$ is larger than the maximum, larger than the minimum and smaller than the maximum, and smaller than the minimum z′ components of the bounding region, respectively. For each case, t_{min} and t_{max} occurred at the lower inner

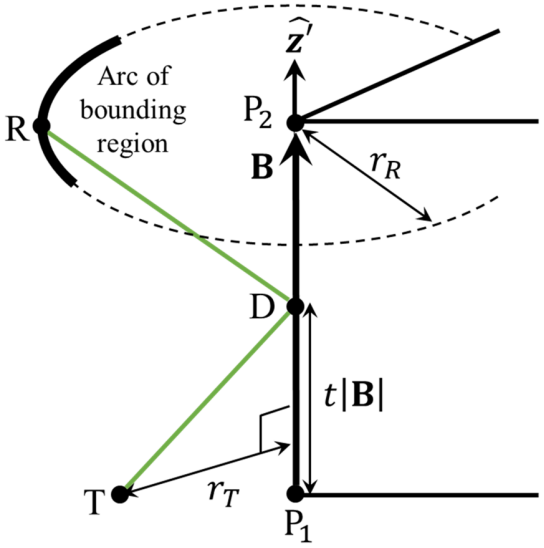


Fig. 7. Diffraction geometry for arc of bounding region of diffraction–reflection case of ABS.

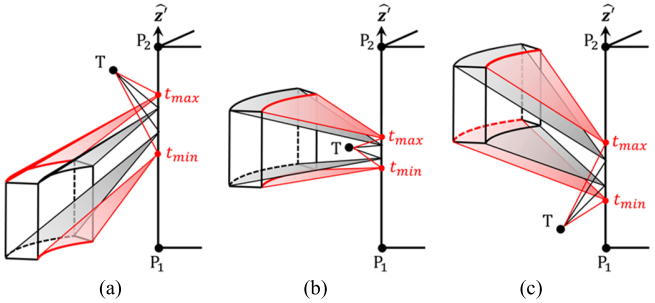


Fig. 8. Three cases for the relative position between Tx and bounding region of diffraction–reflection case of ABS: $T_{z'}$ is (a) larger than maximum, (b) larger than minimum and smaller than maximum, and (c) smaller than the minimum z′ component of the bounding region.

and upper outside, lower and upper inside, and lower outside and upper inside arcs, respectively. Ultimately, the proposed bounding region shape offers the advantage of simplifying the calculation of t_{min} and t_{max} values using (14) and (15), considering only two arcs of the bounding region. This is a superior characteristic compared to the popular axis-aligned bounding box with 12 corners, whose corners where the t_{min} and t_{max} values are located are difficult to predict, and the formulation of the t_{min} and t_{max} values of each corner is more complex than those of (14) and (15).

After determining the first and second $\phi - t$ anxel beam, we determine the intersection between the two. Subsequently, we generate images of the source and edge in the reflecting facet and transform basis similarly with the multiple reflection case, that is, the first and second new basis $\mathbf{x}''$ and $\mathbf{y}''$ are $\mathbf{x}'$ and $\mathbf{y}'$ mirrored to the reflecting facet, respectively, as shown in Fig. 9. Subsequently, the problem can be reduced to single diffraction of the image source and image edge on the image bounding region, that is, we can directly use the intersected anxel beam, that is, shrunk anxel beam with margins of $\phi_{min}, \phi_{max}, t_{min}$, and t_{max}, as shown in Fig. 9.

For diffraction–multiple reflections, the same procedure can be used to shrink the anxel beam because every diffraction–multiple reflection problem can be reduced to a single diffraction of the image source and image edge, as illustrated above. The sole distinction is that the two anxel beams (utilized for calculating the intersection) for each bouncing order consist of the shrunken anxel beam from the previous bouncing order and the one extended from the image edge and enclosing the bounding region of the reflecting facet of the subsequent bouncing order. Thus, the size of the anxel beam decreases exponentially as the bouncing order increases.

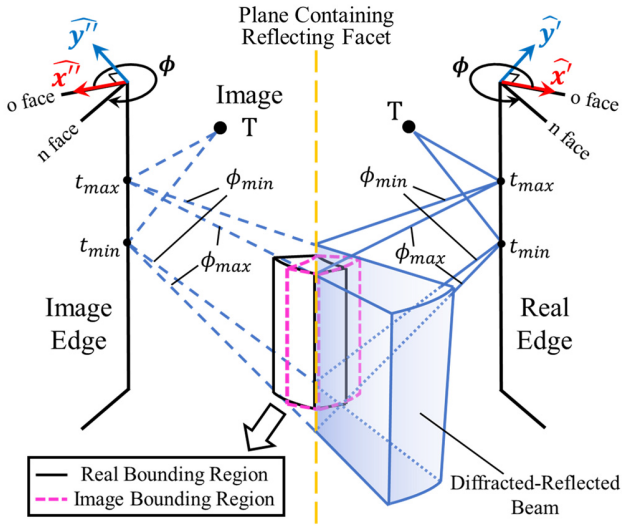


Fig. 9. Image source, edge, bounding region, and new basis for diffraction–reflection case of ABS.

This section describes the ABS method for multiple reflections, reflection–diffraction, and diffraction–reflections. The ABS method for other propagation sequences, for example, multiple reflection–diffraction, and reflection–diffraction–multiple reflection, can be formed by combinations of the abovementioned propagation sequences.

III. HAIT Method RT Framework

In this section, we describe our novel HAIT RT framework. The framework is divided into three components: 1) visibility preprocessing; 2) visibility tree generation using ABS and CPU parallel computing; and 3) ST and field calculations using CPU/GPU heterogeneous computing.

A. Visibility Preprocessing

The first part of HAIT is visibility preprocessing. In this section, the framework derives the visibility relationships between facet–facet, facet–edge, Tx–facet, Tx–edge, facet–FOP, and edge–FOP. The preceding four and the following relationships were used to accelerate visibility tree generation and ST, respectively. In this study, we did not describe our visibility preprocessing algorithm in detail, considering the size of this article. However, readers can refer to certain visibility preprocessing algorithms in [36], [40], [41], [42], [43], and [44].

B. Visibility Tree Generation Using ABS

The second part of HAIT is the visibility tree generation. In this section, we generate a visibility tree containing every possible sequence of primitives of a geometric object, that

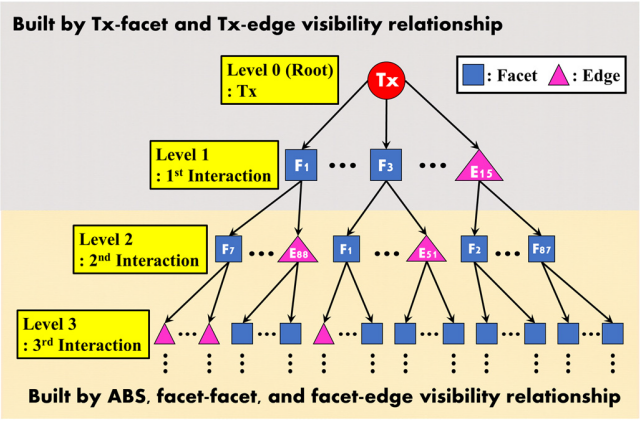


Fig. 10. Visibility tree without FOPs.

is, a facet or edge, that the ray can bounce, as shown in Fig. 10. Level 1 of the tree was directly constructed using the Tx–facet and edge visibility relationship obtained from visibility preprocessing. Higher level trees can be efficiently built using the ABS. Furthermore, the facet–facet and facet–edge visibility relationships obtained from the visibility preprocessing accelerate the visibility tree generation one more time by considering only the visible facets and edges from the facet or edge that produce the beam when we find facets or edges inside the beam. For this search, although not applied in this study, additional acceleration can be achieved using conventional AZB matrices in [20] and [21]. The conventional AZB matrices contain information about rough relative directions between facets. These are computed for each facet of the scenario and saved in a preprocessing step, allowing the matrices to be used very efficiently for any position of the Tx. However, if we generate the total visibility tree at once, as shown in Fig. 10, to achieve good CPU/GPU heterogeneous computing performance using only the conventional AZB matrices, an enormous amount of memory can be required. On the other hand, the way we create the visibility tree using ABS is very memory efficient because the subtrees of the visibility tree are for each position of the image Tx of different facets or edges of the scene, which have an angular visibility less than those of the facets or edges. However, each subtree should be recomputed for new position of the Tx, consuming significant CPU time. This tree generation time can be effectively reduced using the information about the rough relative direction between primitives, which is contained in conventional AZB matrices. If we use the information for the search of primitives inside the beam, resulting in examination for only primitives located nearby the beam direction, the search would be significantly accelerated.

We use open multiprocessing (OpenMP), a CPU parallel computing application programming interface (API), for parallel computing acceleration. Since each subtree of the root node is independent, we can parallelize the tree generation process. By allowing each OpenMP thread to construct different subtrees simultaneously, we achieve up to n times faster computation compared to standard serial computation, with n representing the number of CPU threads used.

We do not generate DAZB matrices which are used by the general AZB method to reduce the number of ray-facet intersection tests. This is because DAZB matrices should be generated for every image Tx and it results in the usage of a prohibitively large amount of memory; therefore, it is not suitable for parallel computing.

When we build the visibility tree, we do not attach FOPs, as shown in Fig. 10. This is because if we attach FOPs to the tree during tree construction, the tree size will be extremely large when an analysis scenario treats numerous FOPs, resulting in a prohibitively large amount of memory and precluding parallel computing acceleration. In addition, finding thousands of FOPs inside the anxel beam is a simple assembly of numerous similar operations, and it is more suitable for GPU parallel computing, which has thousands of cores, than for CPU parallel computing. Therefore, we attached FOPs bit by bit to the visibility tree in the following ST part and conducted the ST using GPU parallel computing. This trick, which ignores FOPs in the tree generation part, has the additional advantage that the completed visibility tree without FOPs can be reused when the locations of FOPs are changed because the tree is independent of their location. The SBR method RT does not retain this superior function because the general SBR method does not construct a visibility tree.

C. ST and Field Calculation

The final part of HAIT is the ST and field calculations. In this section, we introduce a GPU and CPU heterogeneous computing approach where the GPU and CPU concurrently execute ST and field calculations, respectively.

First, because HAIT does not generate DAZB matrices, we must use other acceleration structures such as the kD-tree and bounding volume hierarchy (BVH) structure. We adopted the BVH structure which is provided by NVIDIA[1] OptiX.[2] OptiX is an RT engine, which is a programmable system designed for NVIDIA GPUs and other highly parallel architectures. It offers a high-performance BVH structure, ray-facet intersection test algorithm, and high-quality optimization at low levels. In addition, new versions of OptiX are periodically released such that periodic performance improvement is feasible through slight adjustments in the program code. Detailed information regarding OptiX can be found in [45]. Using OptiX, we conducted ST in a GPU environment. The ST algorithm is described in detail in [13].

For field calculation, we use the geometrical optics (GO) [34], [39], [46], [47], [48], [49], [50] and uniform geometrical theory of diffraction (UTD) [51], [52] to calculate reflection and diffraction fields, respectively.

Fig. 11 shows a flowchart of the ST and field calculations. First, we developed a BVH structure using OptiX to accelerate the ST. Next, we attached FOPs to every node of the visibility tree. For the root node, we attached every FOPs, and for the other nodes at higher levels, we attached FOPs visible from the facet or edge corresponding to each node, considering the facet–FOP and edge–FOP visibility relationships obtained

[1]Registered trademark.
[2]Trademarked.

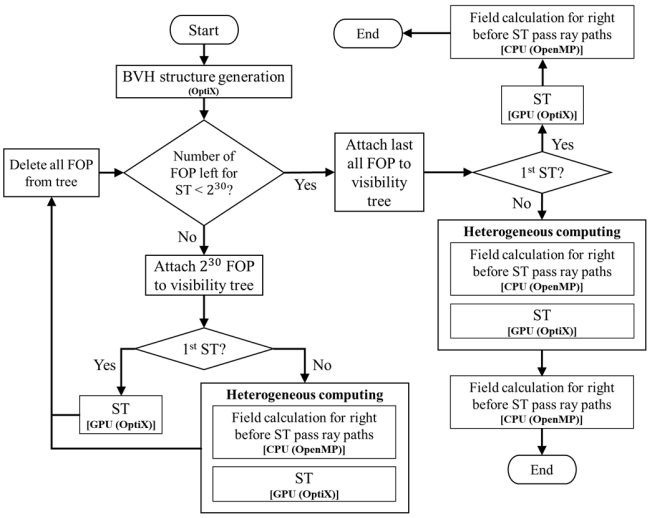


Fig. 11. Flowchart of the ST and field calculation part.

from visibility preprocessing. The maximum number of FOPs to be attached at once was 2^{30} because the maximum ray launch size of the OptiX 7 version that we used was 2^{30} [53]. This implies that if the total number of required STs is over 2^{30}, we cannot accomplish the ST in its entirety within a single iteration and the algorithm should be partitioned into multiple ST sets. Based on this, we propose a heterogeneous computing algorithm. This heterogeneous computing simultaneously performs nth ST set using OptiX, that is, GPU parallel computing, and field calculations for ST pass-ray paths of $(n-1)$th ST set using OpenMP, that is, CPU parallel computing, where each GPU/CPU thread performs ST/field calculations for each branch of the tree. This is possible because the compute unified device architecture (CUDA) returns control to the CPU immediately after invoking the CUDA kernel, without waiting for the GPU computation to be completed [54], increasing computation efficiency.

We note that FOP attachment/ST/field computation starts from the node of the low level to the high level of the visibility tree to avoid CUDA thread divergence as much as possible [55].

IV. Validation

In this section, we describe the performance of the proposed HAIT by comparing it with HAIT without ABS, and an IT solver of the commercial ray-tracer WinProp for two outdoor scenarios: simple and complex. We note that the field calculation theory of the IT solver in WinProp is based on GO and UTD, identical to that of HAIT. Additionally, we used the same CAD files for both HAIT and WinProp. Consequently, if there are no logical errors, the simulation results for electric field intensity (E-field) from HAIT and WinProp must be almost identical, with any very small differences attributable solely to the floating-point number precision of the computer. Also, we note that all simulations were conducted on an identical computation environment: Intel[1] Xeon[1] CPU E5-2687 v4 at 3.00 GHz (two processors) 512-GB RAM (16 threads were used) and NVIDIA RTX A6000 GPU. HAIT and HAIT without the ABS were implemented in the C/C++ language and OptiX.

TABLE I
Simulation Parameters of Simple Scenario

Frequency	Tx antenna	Tx radiation power
28 [GHz]	Hertzian dipole (z-directed)	1 [W]
Max. Bouncing #	Tx location	Number of FOP
1, 2, 3, 4, 5, 6	(-16.6, -94.3, 10) [m]	100
Location of FOP		
Uniformly distributed on line between (-143, -81, 1.5) and (242, 51, 1.5) [m]		

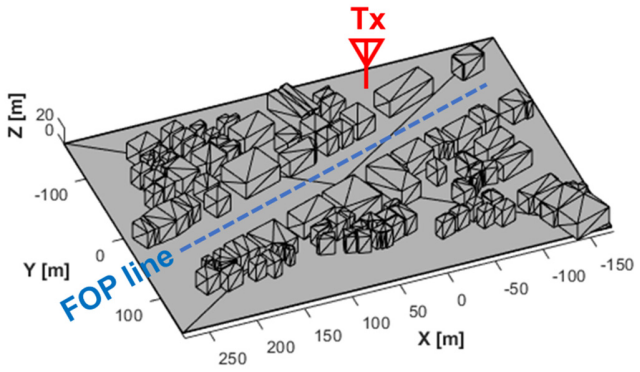


Fig. 12. Simulation environment of the first, i.e., simple, scenario describing Teheran-ro and locations of Tx and FOPs of simple scenario.

TABLE II
Performance for Simple Scenario

Max. bouncing #	RMSPE [%]		Computation time [sec]		
	HAIT vs HAIT w/o ABS	HAIT vs WinProp	HAIT	HAIT w/o ABS	WinProp
1	0	0.99	10		19
2	0	0.81	10	10	20
3	0	1.09	10	11	6,512
4	0	-	12	13	-
5	0	-	17	28	-
6	0	-	52	158	-

The first simulation scenario is simple, as shown in Fig. 12 and Table I. Fig. 12 shows the simulation environment describing Teheran-ro in Seoul, South Korea. It was constructed using AutoCAD and contained 1218 triangular facets. In this scenario, E-field is analyzed at 28 GHz for a hertzian dipole Tx radiating power of 1 W. Tx is located at $(-16.6, -94.3, 10)$ m and 100 FOPs are uniformly distributed on the line between $(-143, -81, 1.5)$ and $(242, 51, 1.5)$ m. Furthermore, the material of the simulation environment is set to perfect electric conductor (PEC) and it is because, for the diffraction of wedges with finite conductivity, our simulator and WinProp use different heuristic diffraction coefficient models, that is, models in [52] and [56], respectively,

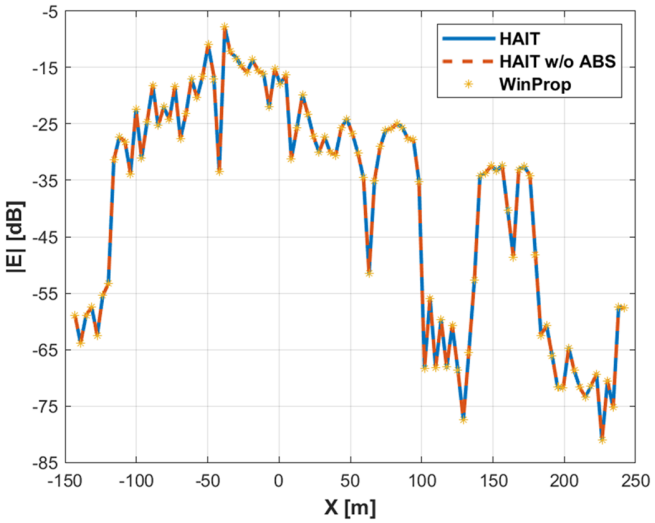


Fig. 13. *E*-field simulation results of simple scenario with maximum ray bouncing order of 3 for HAIT, HAIT without ABS, and WinProp.

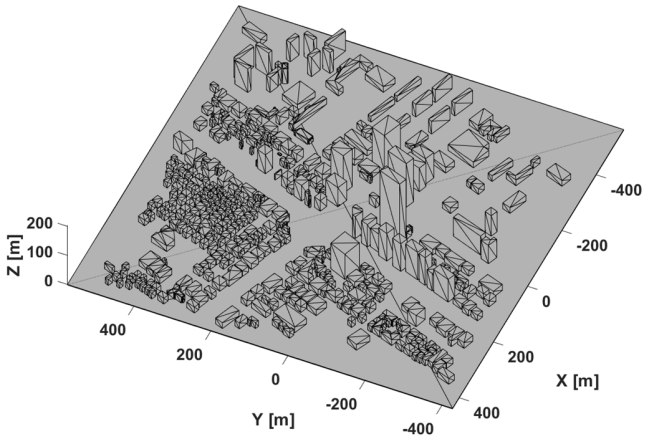


Fig. 14. Simulation environment of the second, i.e., complex, scenario describing Gangnam, Seoul.

TABLE III
SIMULATION PARAMETERS OF COMPLEX SCENARIO

Frequency	Tx antenna	Tx radiation power
28 [GHz]	Hertzian dipole (z-directed)	1 [W]
Max. Bouncing #	Tx location	Number of FOP
1, 2, 3, 4, 5, 6	(46,14,10) [m]	10,000
Location of FOP		
Uniformly distributed on area (x: -560~440, y: -430~570, z: 1.5) [m]		

resulting in different simulation results. Finally, one–six of the maximum numbers of reflections, combined with one order of diffraction, were considered. Table II shows the performances of HAIT, HAIT without the ABS, and WinProp for a simple scenario. Naturally, HAIT and HAIT without ABS show zero root-mean-square percentage error (RMSPE) because ABS removes only unnecessary facets and edges from the visibility tree. Although HAIT shows a similar computation time as HAIT without the ABS up to the maximum bouncing order of 4, HAIT shows approximately 1.6 and three times faster computation speeds than HAIT without the ABS at the maximum bouncing orders of 5 and 6, respectively. This phenomenon was expected because, as described in Section II, the effect of the ABS method increased exponentially as the bouncing order increased. Examining the simulation results of WinProp, we note that the simulation is conducted only for maximum bouncing orders of up to three because it consumes a tremendous amount of computation time at the maximum bouncing order of over three. The RMSPEs between HAIT and WinProp are extremely small, that is, under 1.5%, and this small error is probably because of the usage of different data types in each ray tracer, resulting in floating-point number precision error. Fig. 13 shows the *E*-field simulation results of HAIT, HAIT without ABS, and WinProp for a maximum bouncing order of three. Although HAIT has approximately two times faster computation time than WinProp at the maximum bouncing orders of one and two, a computation time approximately 651 times faster was confirmed at the maximum bouncing number of three. The reason for this considerable difference is difficult to explain because the exact WinProp algorithm is not open to the public. However, the authors suspect that the significant performance difference may be partly attributed to the fact that while the proposed HAIT framework utilizes both CPU and GPU parallel computing, the IT solver in WinProp relies solely on CPU parallel computing and does not support GPU parallel computing. Being a general IT ray tracer, the computation time of the IT solver in WinProp, like that of other similar ray tracers, exponentially increases with higher maximum bouncing orders and a larger number of FOPs. However, HAIT employs the ABS method and a heterogeneous computing-based ST and field calculation algorithm, optimized for high maximum ray bouncing orders and a large number of FOPs. This results in enhanced performance, significantly reducing and accelerating ST. Consequently, the authors expect that the ABS and heterogeneous computing algorithm will create a far superior performance compared to WinProp.

Fig. 14 shows the simulation environment of the second scenario, which is a complex scenario describing Gangnam, Seoul. It contains 4966 triangular facets, is broader (1 × 1 km), and has a more complex urban area than the simple scenario. Table III shows the simulation parameters. In this scenario, similar to the simple scenario, the *E*-field is analyzed at 28 GHz for a Hertzian dipole Tx radiating power of 1 W. The Tx is located at (46, 14, 10) m and 10 000 FOPs are uniformly distributed in the area (x: −560–440, y: −430–570, and z: 1.5) m. Additionally, for the same reason as in the simple scenario, the material of the simulation environment is set to PEC. Finally, 1–6 of the maximum orders of reflections, combined with one order of diffraction, were considered. Table IV presents the performance of the complex scenario. Similar to the simple scenario, HAIT and HAIT without the ABS exhibited zero RMSPE values. Although HAIT shows similar computation time and memory consumption as HAIT without ABS up to the maximum bouncing order of 3, HAIT shows approximately 1.5 times faster computation speed and

TABLE IV
PERFORMANCE OF COMPLEX SCENARIO

Max. Bounce #	RMSPE [%]		Computation time [sec]			Memory [GB]					
	HAIT vs HAIT w/o ABS	HAIT Vs WinProp	HAIT	HAIT w/o ABS	WinProp	HAIT		HAIT w/o ABS		WinProp	
						CPU	GPU	CPU	GPU	CPU	GPU
1	0	3.95	154		28	0.73	1.22	0.73	1.22	0.14	-
2	0	3.33	155	155	5,281	0.75	1.51	0.75	1.51	0.24	-
3	0	-	188	194	-	21.4	14.3	21.4	14.3	-	-
4	0	-	508	788	-	25.7	14.7	27.1	15.9	-	-
5	-	-	5,461	-	-	31.3	19.8	-	> 48	-	-
6	-	-	(Tree partition) 73,688	-	-	(Tree partition) 49.8	(Tree partition) 39.9	-	-	-	-

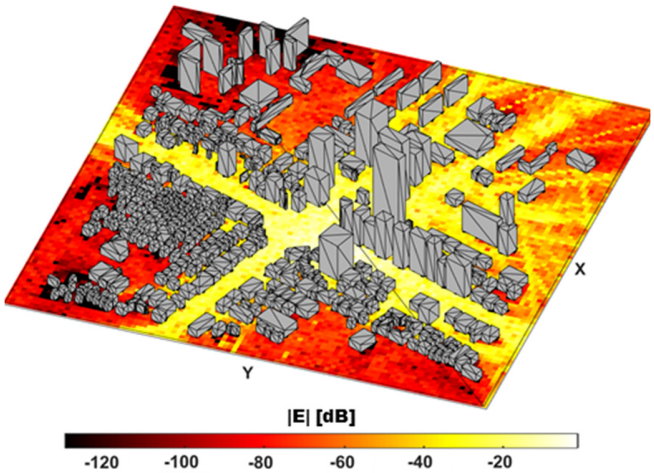


Fig. 15. *E*-field simulation results of the HAIT with maximum bouncing number of 6 for complex scenario.

1.4- and 1.2-GB less memory consumption of the CPU and GPU than HAIT without ABS at the maximum bouncing order of 4. Moreover, at the maximum bouncing order of five, the memory consumption of HAIT without the ABS exponentially increases and encounters memory errors in the GPU environment, that is, the GPU memory consumption is larger than 48 GB, whereas HAIT consumes only 19.8 GB in the GPU environment. At the maximum ray bouncing order of six, HAIT encounters a GPU out-of-memory error. To manage this error, we conducted a series of experiments by partitioning the visibility tree into six parts. Specifically, in the complex scenario, the root node of the full-visibility tree had 513 child nodes. We first completed 85 root node subtrees and conducted ST and field calculations for these 85 subtrees. Subsequently, the authors sequentially constructed another 85 or 86 subtrees and ST and field calculations five more times to handle every subtree of the root node. This visibility tree partitioning trick can be used to control the tradeoff between memory consumption and simulation time.

In Table IV, when comparing the performance of HAIT and HAIT without ABS, we notice distinct trends in the improvement of computation time and memory consumption as the maximum bouncing order increases. Specifically, we observe that memory efficiency improvement becomes evident after the improvement in computation time. This disparity is primarily attributed to the limitation of the maximum number of ST performed at once, set at 2^{30}. Were this limit unlimited, we would likely see parallel trends in both computation time and memory consumption. Here is the comprehensive analysis of the memory consumption: in Table IV, we observe a significant surge in memory usage from two bounces to three bounces for both HAIT and HAIT without ABS. This surge is primarily due to the notably larger size of level 3 of the visibility tree compared to level 2. Consequently, for both HAIT and HAIT without ABS, the scenario with three bounces requires ST for over 2^{30} FOPs, leading to the need for multiple computation sets of the ST/field calculations. At three bounces, GPU memory consumption for both HAIT and HAIT without ABS reaches 14.3 GB, primarily consisting of approximately 13 GB dedicated to storing information related to ST. This includes 2^{30} indices of FOPs, 2^{30} branch indices corresponding to FOPs, and a buffer for ST results with a size of 2^{30}. This accounts for the majority of GPU memory usage, with the remaining portion comprising miscellaneous information such as CAD file data. It is noteworthy that memory consumption related to ST does not increase further, as the maximum number of STs that can be performed at once is limited to 2^{30}. From the perspective of the visibility tree without FOPs, up to three bounces, the total size of the tree for both HAIT and HAIT without ABS remains relatively small compared to other data, such as ST information. Therefore, the difference in visibility tree size between HAIT and HAIT without ABS does not significantly impact total memory consumption. However, starting from four bounces, the size of the visibility tree without FOPs experiences an exponential increase, resulting in a more pronounced difference in memory consumption between HAIT and HAIT without ABS. Fig. 15 shows the *E*-field simulation results for HAIT with a maximum bouncing number of six. To the best of authors' knowledge, this is the first IT ray tracer that is feasible for simulating such a large scale, that is, 1×1 km wide, dense outdoor environment with a high maximum ray bouncing order of six and numerous FOPs (10 000). This achievement is attributed to the implementation of the ABS method and

several straightforward yet highly effective new approaches that facilitate efficient GPU parallel computing: 1) utilization of an innovative heterogeneous computing algorithm for ST/field calculations; 2) integration of FOPs during the ST phase, as opposed to during visibility tree generation; and 3) adoption of a BVH acceleration structure in place of the conventional DAZB matrix. Looking at the simulation results of WinProp, the simulation is conducted only for maximum bouncing orders of up to two because it is computationally intensive at the maximum bouncing order over two. Despite conducting simulations using another commercial IT method ray tracer, newFASANT (geometrical theory of diffraction (GTD) module), which employs a combination of conventional AZB with the SVP and the A$*$ heuristic search algorithm in [20] and [21] without GPU acceleration, to compare the simulation results between HAIT and an existing IT method ray tracer for bouncing orders exceeding 2, we were unable to obtain simulation results because newFASANT also required a significant amount of computation time at a maximum bouncing number of 3. The RMSPEs between HAIT and WinProp were very small, that is, under 4%. This slight error was probably due to the floating-point number precision error, as described in the simple scenario. Although HAIT has a slower computation time than WinProp at the maximum bouncing order of 1 because of the visibility preprocessing of HAIT, which takes 145 s, it was confirmed that HAIT is approximately 34 times faster than WinProp at the maximum bouncing number of 2.

Table V shows the comparison of the E-field simulation results of the HAIT for maximum bouncing orders from 1 to 5 with a reference model employing a maximum bouncing order of 6. The RMSPE and percent error (PE) are used to assess the disparities among simulation results for LOS, NLOS, and LOS + NLOS (total) regions, where PE is defined as follows:

$$\mathrm{PE} = \frac{1}{n}\sum_{i=1}^{n}\frac{y_i - \hat{y}_i}{\hat{y}_i} \times 100[\%] \tag{16}$$

where n, y_i, and $\hat{y}_i$ represent sample size, ith test value, and reference value, respectively. Additionally, we handled realistic simulation results by applying dielectric constant and conductivity of $\varepsilon_r = 6.5$ and $\sigma = 0.668\mathrm{S/m}$ to the objects as proposed in [57], rather than PEC. Notably, variations in electrical properties have minimal impact on the computation time and memory consumption of HAIT. As anticipated, the RMSPE and absolute value of PE decrease across all cases with the inclusion of more bouncing orders. However, there exists a noticeable difference in the degree of reduction between LOS and NLOS regions. In LOS regions, the RMSPE and absolute value of PE values exponentially decrease with an increase in bouncing orders, resulting in an RMSPE smaller than 5% with a maximum bouncing order of 5, and a PE smaller than 5% with a maximum bouncing order of 2. Conversely, for NLOS and total regions, the RMSPE never drops below 58%, even with a maximum bouncing order of 5, and the absolute value of PE falls below 5% only at the maximum bouncing order of 5. It is worth noting that the RMSPE value encompasses both large- and small-scale fading, while the PE value solely reflects large-scale fading. Thus, if we assume a permissible error margin of 5% and consider a maximum bouncing order of 6 as a reference model, employing a maximum bouncing order of 2 is sufficient for analyzing large-scale fading in LOS regions. Conversely, a maximum bouncing order of 5 proves adequate for examining small-scale fading. As for NLOS and total regions, utilizing a maximum bouncing order of 5 is sufficient for analyzing large-scale fading, but even at this maximum bouncing order, it falls short for analyzing small-scale fading.

TABLE V
RMSPE (PE) [%] BETWEEN SIMULATION DATA

Max. Bounce #	1	2	3	4	5
LOS	100.0 (14.29)	34.82 (1.682)	13.90 (0.1094)	5.964 (-0.1799)	3.157 (-0.09)
NLOS	96.88 (-54.62)	86.40 (-37.47)	74.66 (-22.19)	63.67 (-10.39)	62.13 (-2.16)
Total	97.22 (-47.22)	82.43 (-33.27)	70.68 (-19.80)	60.19 (-9.297)	58.71 (-1.94)

V. CONCLUSION

In this study, we proposed a HAIT RT framework for massive outdoor propagation modeling. HAIT is characterized by an ABS method, applicable to arbitrary 3-D facet-based structures, and CPU/GPU heterogeneous computing. The ABS method effectively accelerates generation times and reduces the size of the visibility tree by allowing the beam to radiate only to the portion that is illuminated by the beam of the former bouncing order. Compared to the case without the ABS method, a simulation time that is more than three times faster is confirmed in a simple scenario. In addition, it is confirmed that using the ABS, it is feasible to handle a complex scenario, which is infeasible without the ABS case due to the out-of-memory error, using less than half the memory. This efficiency improvement increases exponentially as the maximum ray-bouncing order, complexity of the simulation geometry, and number of FOPs increase.

The CPU/GPU heterogeneous computing accelerates the overall part of the algorithm by letting OpenMP CPU parallel computing handle the visibility tree generation and field computation task; OptiX GPU parallel computing handled the ST. At this moment, it utilized GPU parallel computing and handled thousands of FOPs using the simple new techniques. It was confirmed that HAIT was 651 times faster than the WinProp IT solver, which exclusively utilizes CPU parallel computing without GPU integration. Also, it was confirmed that the proposed HAIT could handle a 1 $\times$ 1 km wide complex urban environment with a maximum ray bouncing order of six and thousands of FOPs. Therefore, we expect that it could be used in areas that were exclusive to the SBR method ray tracers, such as channel modeling and coverage analysis, with better accuracy.

There are two potential avenues for future research. The first is the application of additional propagation mechanisms such as transmission and multiple diffraction. The second is to apply additional computing techniques such as cluster computing to the visibility tree generation part and multiple

GPU parallel computing to the ST/field calculation part for further computation acceleration.

ACKNOWLEDGMENT

The authors would like to extend their gratitude to Prof. Yong Heui Cho for his meticulous instruction on the use of NVIDIA[1] OptiX.[2]

REFERENCES

[1] S. Hur et al., "28 GHz channel modeling using 3D ray-tracing in urban environments," in *Proc. 9th Eur. Conf. Antennas Propag. (EuCAP)*, Apr. 2015, pp. 1–5.

[2] L. Tian, V. Degli-Esposti, E. M. Vitucci, and X. Yin, "Semi-deterministic radio channel modeling based on graph theory and ray-tracing," *IEEE Trans. Antennas Propag.*, vol. 64, no. 6, pp. 2475–2486, Jun. 2016.

[3] J.-H. Jung, J. Lee, J.-H. Lee, Y.-H. Kim, and S.-C. Kim, "Ray-tracing-aided modeling of user-shadowing effects in indoor wireless channels," *IEEE Trans. Antennas Propag.*, vol. 62, no. 6, pp. 3412–3416, Jun. 2014.

[4] G. Tiberi, S. Bertini, W. Q. Malik, A. Monorchio, D. J. Edwards, and G. Manara, "Analysis of realistic ultrawideband indoor communication channels by using an efficient ray-tracing based method," *IEEE Trans. Antennas Propag.*, vol. 57, no. 3, pp. 777–785, Mar. 2009.

[5] J.-H. Lee, J.-S. Choi, and S.-C. Kim, "Cell coverage analysis of 28 GHz millimeter wave in urban microcell environment using 3-D ray tracing," *IEEE Trans. Antennas Propag.*, vol. 66, no. 3, pp. 1479–1487, Mar. 2018.

[6] H. Yi et al., "Ray tracing meets terahertz: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 62, no. 2, pp. 40–46, Feb. 2024.

[7] Z. Lai et al., "Intelligent ray launching algorithm for indoor scenarios," *Radioengineering*, vol. 20, no. 2, pp. 398–408, Jun. 2011.

[8] M. M. Taygur and T. F. Eibert, "A ray-tracing algorithm based on the computation of (exact) ray paths with bidirectional ray-tracing," *IEEE Trans. Antennas Propag.*, vol. 68, no. 8, pp. 6277–6286, Aug. 2020.

[9] M. M. Taygur and T. F. Eibert, "Determination of exact ray paths by bidirectional ray-tracing," in *Proc. XXXIIIrd Gen. Assem. Sci. Symp. Int. Union Radio Sci.*, Aug. 2020, pp. 1–4.

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

[11] S. Kasdorf, B. Troksa, C. Key, J. Harmon, and B. M. Notaroš, "Advancing accuracy of shooting and bouncing rays method for ray-tracing propagation modeling based on novel approaches to ray cone angle calculation," *IEEE Trans. Antennas Propag.*, vol. 69, no. 8, pp. 4808–4815, Aug. 2021.

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

[13] M. F. Catedra, J. Perez, F. S. de Adana, and O. Gutierrez, "Efficient ray-tracing techniques for three-dimensional analyses of propagation in mobile communications: Application to picocell and microcell scenarios," *IEEE Antennas Propag. Mag.*, vol. 40, no. 2, pp. 15–28, Apr. 1998.

[14] M. Catedra and J. Perez, *Cell Planning for Wireless Communications*. Norwood, MA, USA: Artech House, 1999.

[15] O. Gutiérrez, F. S. de Adana, I. Gonzalez, J. Perez, and M. F. Cátedra, "Ray-tracing techniques for mobile communications," *Appl. Comput. Electromagn. Soc. J.*, vol. 15, no. 3, pp. 209–231, Nov. 2000.

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

[17] M. F. Cátedra, J. M. Gómez, F. S. de Adana, I. González, and O. Gutiérrez, "Application of ray tracing accelerating techniques for the analysis of antennas on complex platforms modelled by NURBS," in *Proc. IEEE Antennas Propag. Soc. Int. Symp.*, Jul. 2005, pp. 167–170.

[18] L. V. Jingyu, Y. Wang, J. Huang, Y. Li, J. Huang, and C. X. Wang, "An improved triangular facets based angular Z-buffer algorithm for IM ray tracing channel modeling," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2022, pp. 1050–1056.

[19] C. Saeidi and F. Hodjatkashani, "Modified angular Z-buffer as an acceleration technique for ray tracing," *IEEE Trans. Antennas Propag.*, vol. 58, no. 5, pp. 1822–1825, May 2010.

[20] F. Catedra, L. Lozano, I. Gonzalez, M. J. Algar, and E. García, "Efficient techniques for accelerating the ray-tracing for computing the multiple bounce scattering of complex bodies modeled by flat facets," *Appl. Comput. Electromagn. Soc. J.*, vol. 25, no. 5, pp. 395–409, May 2010.

[21] L. Lozano, M. J. Algar, E. García, I. González, and F. Cátedra, "Efficient combination of acceleration techniques applied to high frequency methods for solving radiation and scattering problems," *Comput. Phys. Commun.*, vol. 221, pp. 28–41, Dec. 2017.

[22] K. Rizk, J.-F. Wagen, and F. Gardiol, "Two-dimensional ray-tracing modeling for propagation prediction in microcellular environments," *IEEE Trans. Veh. Technol.*, vol. 46, no. 2, pp. 508–518, May 1997.

[23] G. E. Athanasiadou, A. R. Nix, and J. P. McGeehan, "A microcellular ray-tracing propagation model and evaluation of its narrow-band and wide-band predictions," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 322–335, Mar. 2000.

[24] S. Kasdorf, B. Troksa, J. Harmon, C. Key, and B. M. Notaroš, "Shooting-bouncing-rays technique to model mine tunnels: Algorithm acceleration," in *Proc. Int. Appl. Comput. Electromagn. Soc. Symp. (ACES)*, Jul. 2020, pp. 1–2.

[25] K. Williams, L. Tirado, Z. Chen, B. Gonzalez-Valdes, J. Á. Martínez, and C. M. Rappaport, "Ray tracing for simulation of millimeter-wave whole body imaging systems," *IEEE Trans. Antennas Propag.*, vol. 63, no. 12, pp. 5913–5918, Dec. 2015.

[26] C. Yun Kee, C.-F. Wang, and T. Tong Chia, "Optimizing high-frequency PO-SBR on GPU for multiple frequencies," in *Proc. IEEE 4th Asia–Pacific Conf. Antennas Propag. (APCAP)*, Kuta, Indonesia, Jul. 2015, pp. 132–133.

[27] C. Y. Kee and C.-F. Wang, "High-frequency PO-SBR code on GPU," in *Proc. IEEE Antennas Propag. Soc. Int. Symp. (APSURSI)*, Orlando, FL, USA, Jul. 2013, pp. 1890–1891.

[28] R. Felbecker, L. Raschkowski, W. Keusgen, and M. Peter, "Electromagnetic wave propagation in the millimeter wave band using the NVIDIA OptiX GPU ray tracing engine," in *Proc. 6th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2012, pp. 488–492.

[29] C. Y. Kee and C.-F. Wang, "Efficient GPU implementation of the high-frequency SBR-PO method," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 941–944, 2013.

[30] X. Niu, H. He, and M. Jin, "Application of ray-tracing method in electromagnetic numerical simulation algorithm," in *Proc. Int. Appl. Comput. Electromagn. Soc. (ACES-China) Symp.*, Chengdu, China, Jul. 2021, pp. 1–2.

[31] M. Pang, H. Wang, K. Lin, and H. Lin, "A GPU-based radio wave propagation prediction with progressive processing on point cloud," *IEEE Antennas Wireless Propag. Lett.*, vol. 20, no. 6, pp. 1078–1082, Jun. 2021.

[32] F. Zhang, C. Zhou, Y. Li, G. Xia, J. Zhu, and Z. Zhao, "A GPU acceleration algorithm for urban electromagnetic environments," in *Proc. 13th Int. Symp. Antennas, Propag. EM Theory (ISAPE)*, Zhuhai, China, Dec. 2021, pp. 1–3.

[33] J. Pyhtila et al., "3-D ray tracing based GPU accelerated field prediction radio channel simulator," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Kuala Lumpur, Malaysia, Apr. 2019, pp. 1–7.

[34] R. Brem and T. F. Eibert, "A shooting and bouncing ray (SBR) modeling framework involving dielectrics and perfect conductors," *IEEE Trans. Antennas Propag.*, vol. 63, no. 8, pp. 3599–3609, Aug. 2015.

[35] J. S. Lu et al., "A discrete environment-driven GPU-based ray launching algorithm," *IEEE Trans. Antennas Propag.*, vol. 67, no. 2, pp. 1180–1192, Feb. 2019.

[36] D. Shing-Min Liu and C.-M. Tan, "Visibility preprocessing suitable for virtual reality sound propagation with a moving receiver and multiple sources," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–6.

[37] B. Rainer, D. Löschenbrand, S. Zelenbaba, M. Hofer, and T. Zemen, "Towards a non-stationary correlated fading process for diffuse scattering in ray tracing," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–7.

[38] J. B. Keller, "Geometrical theory of diffraction," *J. Opt. Soc. Amer.*, vol. 52, no. 2, pp. 116–130, Jan. 1962.

[39] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York, NY, USA: Wiely, 1989.

[40] F. M. Landstorfer, "Wave propagation models for the planning of mobile communication networks," in *Proc. 29th Eur. Microw. Conf.*, Munich, Germany, Oct. 1999, pp. 1–6.

[41] R. Hoppe, G. Wolfle, and F. M. Landstorfer, "Accelerated ray optical propagation modeling for the planning of wireless communication networks," in *Proc. IEEE Radio Wireless Conf. (RAWCON)*, Aug. 1999, pp. 159–162.

[42] R. Hoppe, P. Wertz, G. Wolfle, and F. M. Landstorfer, "Wideband propagation modelling for indoor environments and for radio transmission into buildings," in *Proc. 11th IEEE Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, vol. 1, Sep. 2000, pp. 282–286.

[43] T. Rautiainen, G. Wolfle, and R. Hoppe, "Verifying path loss and delay spread predictions of a 3D ray tracing propagation model in urban environment," in *Proc. IEEE 56th Veh. Technol. Conf.*, Vancouver, BC, Canada, vol. 4, Sep. 2002, pp. 2470–2474.

[44] T. Rautiainen, R. Hoppe, and G. Wolfle, "Measurements and 3D ray tracing propagation predictions of channel characteristics in indoor environments," in *Proc. IEEE 18th Int. Symp. Pers., Indoor Mobile Radio Commun.*, Athens, Greece, Sep. 2007, pp. 1–5.

[45] S. G. Parker et al., "OptiX: A general purpose ray tracing engine," *ACM Trans. Graph.*, vol. 29, no. 4, p. 66, 2010.

[46] R. D. Radcliff and C. A. Balanis, "Modified propagation constants for nonuniform plane wave transmission through conducting media," *IEEE Trans. Geosci. Remote Sens.*, vol. GE-20, no. 3, pp. 408–411, Jul. 1982.

[47] J. E. Roy, "New results for the effective propagation constants of nonuniform plane waves at the planar interface of two lossy media," *IEEE Trans. Antennas Propag.*, vol. 51, no. 6, pp. 1206–1215, Jun. 2003.

[48] F. Frezza and N. Tedeschi, "On the electromagnetic power transmission between two lossy media: Discussion," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 29, no. 11, pp. 2281–2288, Nov. 2012.

[49] F. Frezza and N. Tedeschi, "Deeply penetrating waves in lossy media," *Opt. Lett.*, vol. 37, no. 13, pp. 2616–2618, Jul. 2012.

[50] Y. Kim, H. Yang, and J. Oh, "Critical angle formulation of nonuniform plane waves for determining correct refraction angles at planar interface," *IEEE Trans. Antennas Propag.*, vol. 71, no. 3, pp. 2861–2866, Mar. 2023.

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

[52] P. Bernardi, R. Cicchetti, and O. Testa, "A three-dimensional UTD heuristic diffraction coefficient for complex penetrable wedges," *IEEE Trans. Antennas Propag.*, vol. 50, no. 2, pp. 217–224, Feb. 2002.

[53] (2023). *NVIDIA OptiX 7.7—Programming Guide*. Accessed: Mar. 13, 2024. [Online]. Available: https://raytracing-docs.nvidia.com/optix7/guide/index.html#preface#

[54] J. Cheng, M. Grossman, and T. McKercher, *Professional CUDA C Programming*. Birmingham, U.K.: Wrox, 2014.

[55] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Reading, MA, USA: Addison-Wesley, 2010.

[56] R. Luebbers, "Finite conductivity uniform GTD versus knife edge diffraction in prediction of propagation path loss," *IEEE Trans. Antennas Propag.*, vol. AP-32, no. 1, pp. 70–76, Jan. 1984.

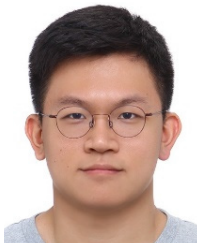
[57] S. Hur et al., "Proposal on millimeter-wave channel modeling for 5G cellular system," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 3, pp. 454–469, Apr. 2016.

Yongwan Kim (Member, IEEE) received the B.S. degree in electrical engineering from Chungnam National University, Daejeon, South Korea, in 2019. He is currently pursuing the Ph.D. degree with Seoul National University, Seoul, South Korea.

His current research interests include ray-tracing electromagnetic (EM) analysis technique, EM theory, parallel computing, heterogeneous computing, wireless communication system, and electromagnetic interference/electromagnetic compatibility (EMI/EMC).

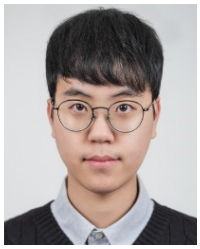
Hyunjun Yang (Member, IEEE) received the B.S. degree in electronic and electrical engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2020. He is currently pursuing the integrated master's and Ph.D. degree with the Department of Electrical Engineering and Computer Science, Seoul National University, Seoul, South Korea.

His current research interests include ray-tracing EM analysis technique, EM theory, wireless propagation channel measurement and modeling, and reconfigurable intelligent surface (RIS).



Hooyoung Kim (Member, IEEE) received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, South Korea, in 2024, where he is currently pursuing the integrated master's and Ph.D. degree with the Department of Electrical and Computer Engineering.

His current research interests include ray-tracing EM analysis technique for 5G communication.



Junpyo Jo (Member, IEEE) received the B.S. degree in electronic engineering from Kyung Hee University, Yongin, South Korea, in 2024. He is currently pursuing the integrated master's and Ph.D. degree with the Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea.

His current research interests include ray-tracing EM analysis technique, parallel computing, and RF/milimeter-wave (mmWave)/microwave integrated circuit.

Jungsuek Oh (Senior Member, IEEE) received the B.S. and M.S. degrees from Seoul National University, Seoul, South Korea, in 2002 and 2007, respectively, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 2012.

From 2007 to 2008, he was a Hardware Research Engineer with Korea Telecom, Seongnam, South Korea, working on the development of flexible RF devices. In 2012, he joined the Radiation Laboratory, University of Michigan, as a Post-Doctoral Research Fellow. From 2013 to 2014, he was a Staff RF Engineer with Samsung Research America, Dallas, TX, USA, working as a Project Leader of the 5G/millimeter-wave antenna system. From 2015 to 2018, he was a Faculty Member with the Department of Electronic Engineering, Inha University, Incheon, South Korea. He is currently an Associate Professor with the School of Electrical and Computer Engineering, Seoul National University. He has authored more than 50 technical journal articles and conference papers. His research areas include milimeter-wave (mmWave) beam focusing/shaping techniques, antenna miniaturization for integrated systems, and radio propagation modeling for indoor scenarios.

Dr. Oh was a recipient of the 2011 Rackham Predoctoral Fellowship Award from the University of Michigan. He has served as a TPC Member and a Session Chair for IEEE AP-S/USNC-URSI and ISAP. He has served as a Technical Reviewer for IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION, IEEE ANTENNAS AND WIRELESS PROPAGATION LETTERS, and so on.