This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

Transmission-Integrated HAIT Ray Tracing for Massive O2I Propagation Modeling with MTN Preprocessing and Anxel Beam Transformation

Yongwan Kim, *Member, IEEE*, Hooyoung Kim, *Member, IEEE,* Jonghyup Lee*, Member, IEEE,* Hyunho Cho*, Member, IEEE,* and Jungsuek Oh, Senior *Member, IEEE*

***Abstract*— Despite its computational efficiency and accuracy, the heterogeneous computing-based anxel beam shrinkage method-accelerated image theory (HAIT) ray tracing (RT) method has been limited to outdoor scenarios due to the lack of transmission modeling. Existing HAIT approaches face challenges in transmission-inclusive environments, demanding substantial computational acceleration. This paper proposes a transmission-integrated HAIT (TI-HAIT) RT framework for large-scale outdoor-to-indoor (O2I) propagation modeling. TI-HAIT introduces two key innovations: 1) a GPU-based minimum transmission number preprocessing (MTNP) algorithm for efficient transmission handling, and 2) an anxel beam transformation (ABT) method for optimized AZB matrix utilization. Leveraging CPU/GPU parallel computing, TI-HAIT achieves up to 6,051× speedup over the image theory (IT) solver in WinProp. MTNP accelerates simulations by up to 153× and reduces CPU memory usage by 78%, while ABT improves visibility tree generation by 4.1× and reduces CPU/GPU memory usage by 7% and 9%, respectively. The framework enables simulations of 1.1 × 1.1 km urban areas with large indoor structures, supporting up to five bounces, four transmissions, and hundreds of field observation points. TI-HAIT advances IT-based RT by enabling efficient and accurate transmission analysis in complex scenarios previously infeasible with conventional IT-based RT methods.**

***Index Terms*—Ray tracing, image theory, shooting and bouncing rays, central processing unit (CPU), graphical processing unit (GPU), heterogeneous computing**

I. Introduction

RT electromagnetic (EM) analysis is widely employed for deterministic channel modeling in both link- and system-level simulations [1], [2]. It supports a broad frequency range and diverse channel types, including millimeter-wave (mmWave), terahertz (THz), maritime, underwater acoustic, vehicle-to-vehicle (V2V), massive MIMO, and industrial IoT channels [1], [3]–[7]. RT is particularly important for indoor and O2I scenarios, where accurate modeling of signal penetration through obstacles is essential. It enables realistic indoor simulations in dense urban areas and smart buildings that utilize next-generation networks operating in the mmWave and THz bands. For example, RT has been applied to the ITU-R P.1238 indoor model [2], 140 GHz multipath analysis [8], 300 GHz path loss modeling [9], [10], and 3D MIMO channel characterization in O2I environments [11].

RT methods include the shooting-and-bouncing ray (SBR) method, which offers high computational efficiency but suffers from phase errors and ray-tube cutoff, and the IT method, which achieves higher accuracy but at the cost of efficiency due to shadow test and visibility tree generation [12].

The SBR method is widely adopted for its high computational efficiency and scalability. However, as modern communication systems demand higher data rates and broader bandwidths, the shift to higher frequencies poses challenges. At these frequencies, phase errors from ray-tube limitations [13]-[15] and ray-tube cutoff errors from inadequate resolution [16], [17] reduce accuracy, requiring more rays or advanced techniques, which increase computational complexity. Moreover, the lack of a systematic approach to optimize ray numbers adds ambiguity to high-frequency simulations [17].

In this context, The IT method offers a robust alternative, with accuracy relying solely on geometric mesh quality and floating-point precision rather than frequency [16]. Unlike SBR, it is also unaffected by beamwidth or environmental resolution, eliminating arbitrary ray increases and enhancing flexibility for high-frequency systems like 6G networks [17].

Despite its high accuracy, the IT method is rarely applied to transmission-inclusive scenarios requiring high-order interactions greater than three—such as channel modeling for indoor or O2I environments—due to its substantial computational cost in such complex settings [15], [18]. Thus, acceleration techniques are needed to enable its use in challenging propagation environments.

A. Related works

In [19], building on the 90% computation time reduction

Manuscript received Dec 16, 2024; This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the South Korea government (MIST) (No.2019-0-00098, Advanced and Integrated Software Development for Electromagnetic Analysis) and by HD Korea Shipbuilding and Offshore Engineering Co., Ltd. *(Corresponding author: Jungsuek Oh)*

Y. Kim, H. Kim, and J. Oh 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; kimhy3010@snu.ac.kr; jungsuek@snu.ac.kr).

J. Lee and H. Cho are with HD Korea Shipbuilding and Offshore Engineering Co., Ltd., Seongnam 13553, South Korea (e-mail: jonghyup.lee@hd.com; hyunho.cho@hd.com).

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

achieved in [20] using the angular-Z buffer (AZB) algorithm for outdoor IT-based RT, a transmission-integrated AZB-enhanced IT method was proposed. This approach demonstrated effective acceleration but incurs high memory usage in parallel computing environments due to its DAZB concept [17]. In [15], the GPU-based kD-tree-accelerated beam tracing (GKBT) method achieved a tenfold speedup over earlier transmission-integrated RT techniques, highlighting significant progress in practical performance. However, its hybrid SBR-IT design introduces ray-tube cutoff errors and lacks support for reflections of diffracted fields, limiting its applicability in O2I scenarios.

In [17], the HAIT RT framework was proposed, combining the anxel beam shrinkage (ABS) method with CPU/GPU heterogeneous computing, achieving a 651× speedup over the IT solver in WinProp. It was the first IT-based framework to simulate a 1 km × 1 km dense urban environment with six ray bounces and 10,000 FOPs. However, HAIT is designed for outdoor scenarios and does not support transmission analysis, limiting its use in indoor and O2I environments.

B. Motivations

Although HAIT can handle transmission with minor adjustments, it remains inefficient due to reliance on conventional visibility preprocessing [21]-[26], which is inadequate for transmission analysis where shadowed primitives can still be ‘electromagnetically' visible. This results in prohibitively long computation times in transmission -inclusive scenarios requiring high-order interactions. (see Section VI). To date, no dedicated preprocessing algorithm for efficient transmission analysis has been proposed.

Another inefficiency in conventional HAIT arises from the ABS method's primitive search within an anxel beam, which requires basis transformations and angular bound calculations involving matrix operations and inverse trigonometric functions [27]. When repeated for many primitives in complex environments, this process significantly increases computation time.

AZB matrices, introduced in [28], [29], offer a solution to accelerate the search process by storing directional information between primitives [17]. They can filter out primitives at angles far from the anxel beam's direction, thereby avoiding basis transformations and angle calculations, and performing detailed computations only for nearby primitives. While promising for reducing search times, their full implementation and validation remain unrealized.

A key limitation of AZB matrices is that their angles are defined in the original basis system and cannot be directly applied to anxel beams in other systems using the ABS method. To use them, each anxel beam must be transformed to the original basis. However, differences in azimuth/elevation directions between basis systems and the curved side surfaces of anxel beams [17] make exact shape transformation fundamentally impossible.

C. Contributions

This study proposes TI-HAIT, a transmission-inclusive IT-based ray tracing framework designed for efficient analysis of high-order interactions required in applications such as channel modeling. The key contributions are summarized as follows:

1) A GPU-accelerated MTNP algorithm is proposed for transmission-inclusive IT-based RT, enabling efficient analysis of high-order interactions in complex O2I and indoor environments. Unlike conventional visibility preprocessing focused on reflection and diffraction, it incorporates transmission and improves efficiency using the remaining transmission number (RTN) and beam direction (BD) tree.

2) A novel ABT method is proposed to transform anxel beams between arbitrary basis systems, ensuring full enclosure of the original beam with minimal wavefront expansion. This enables efficient use of AZB matrices, accelerating primitive identification within anxel beams and improving computational performance in both conventional and TI-HAIT scenarios. The method is applicable to arbitrary facet-based 3D structures.

The paper is structured as follows: Section II validates the practicality of HAIT through performance comparisons with benchmark RT software, highlighting the research value of further advancement. Section III introduces the GPU-based MTNP technique and RTN/BD tree. Section IV presents the ABT method for basis transformations. Section V details the integration of these methods into the TI-HAIT framework. Section VI evaluates efficiency and accuracy of TI-HAIT in O2I scenarios, and Section VII concludes the paper.

II. Performance Comparison Between Conventional HAIT and Benchmark RT Software

In this section, we compare the simulation performance of conventional HAIT with two benchmark RT tools—Wireless InSite’s X3D module (SBR method) and Sionna’s exhaustive solver (IT method)—to demonstrate its practicality. All tools employ geometrical optics (GO) and uniform theory of diffraction (UTD) with CPU/GPU parallel computing, ensuring a fair and unbiased comparison of algorithmic performance. In comparing with X3D, we demonstrate HAIT’s robustness against systemic errors inherent to SBR-based RT methods, such as phase errors and ray-tube cutoffs. In comparing with Sionna’s exhaustive solver, we highlight HAIT’s computational efficiency over conventional IT-based RT methods for high bounce orders.

The test scenario, identical to the massive scenario described in [17] where HAIT’s accuracy was validated against WinProp, consists of a 1 km × 1 km dense urban environment with 10,000 FOPs uniformly distributed at 1.5 meters height and a single omnidirectional transmitter operating at 28GHz. All geometry materials are set as perfect electric conductors (PEC) to eliminate discrepancies in electric field (E-field) simulation results due to differences in heuristic diffraction models [17]. An identical CAD file is used for all simulations, ensuring that differences between conventional HAIT and X3D result solely from X3D’s systemic errors and floating-point inaccuracies. Differences between HAIT and Sionna’s IT solver are attributed only to floating-point inaccuracies.

All simulations were conducted in the same computational environment: Intel(R) Xeon(R) CPU E5-2687 v4 @ 3.00 GHz (2 processors), 512GB RAM (16 threads), and an NVIDIA RTX A6000 GPU.

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

For comparison with X3D, we analyze direct rays and multiple reflections (R) combined with a single diffraction (D), setting the maximum number of reflections before and after diffraction to half of the total maximum reflections, rounded up (e.g., 3 reflections → 2 before and after), following X3D's configuration. For Sionna's exhaustive method, only direct rays and multiple reflections are considered.

Table I compares the performance of conventional HAIT and X3D. For X3D, the ray spacing was set to 0.11°, 0.3°, 0.2°, 0.025°, and 0.006° for maximum reflection orders of 1 to 5, respectively, to match HAIT's computation time, including preprocessing (0.25° was used as the default spacing for zero bounce). Although X3D required slightly more computation time, HAIT identified approximately 8%, 36%, 48%, 52%, and 74% more rays. Fig. 1 shows the cumulative distribution functions (CDFs) of E-field differences between HAIT and X3D for FOPs. The difference grows with bounce order, surpassing 10 dB for 20% of FOPs and 5 dB for 44% in NLOS regions at 5 bounces. These results confirm that HAIT achieves higher accuracy under comparable computation times, although SBR can reduce computation time by increasing ray spacing at the expense of greater accuracy degradation.

Table II compares the computation time of conventional HAIT and Sionna's exhaustive method. As HAIT's accuracy was validated against WinProp in [17] under the same scenario, this comparison focuses solely on computation time. Sionna achieves approximately 32× faster performance at a maximum reflection order of 1; however, its exhaustive testing of all 3D primitive combinations [30] limits scalability beyond two reflections. In contrast, HAIT efficiently supports higher-order reflections by eliminating unnecessary combinations through visibility preprocessing and the ABS method.

As shown above, HAIT demonstrates superior accuracy over SBR and greater computational efficiency over conventional IT methods for high-order analysis. These results underscore the research value of advancing beyond conventional HAIT, as discussed in the following sections.

TABLE I
PERFORMANCE COMPARISON OF CONVENTIONAL HAIT AND X3D

Max. Bounce #	Computation time [sec]		Total rays found	
	HAIT	X3D (ray spacing)	HAIT	X3D
Preprocessing	136	-	-	-
R: 0, D: 0	4	5 (0.25°)	1,065	1,065
R: 1, D: 1	11	148 (0.11°)	358,073	330,155
R: 2, D: 1	18	157 (0.3°)	606,026	447,159
R: 3, D: 1	207	353 (0.2°)	1,534,279	1,036,337
R: 4, D: 1	1,662	1,943 (0.025°)	1,903,685	1,250,631
R: 5, D: 1	33,328	34,449 (0.006°)	3,732,054	2,146,868

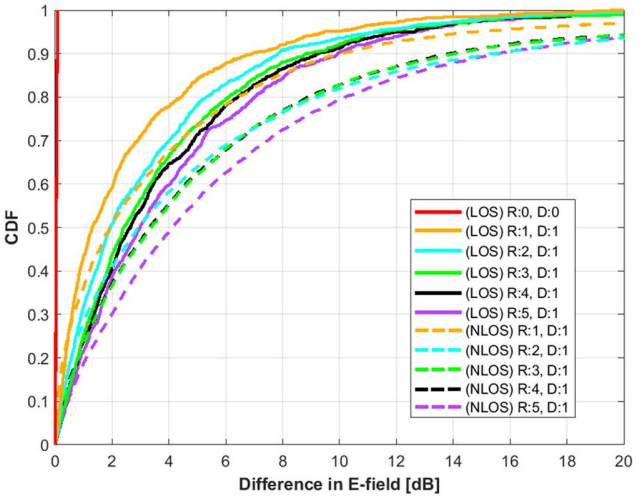


Fig. 1. CDFs of E-field simulation differences between conventional HAIT and X3D.

TABLE II
COMPUTATION TIME COMPARISON OF CONVENTIONAL HAIT AND SIONNA'S EXHAUSTIVE METHOD

Max. Reflection #	Computation time [sec]	
	HAIT	Sionna
Preprocessing	88	-
1	7	3
2	8	Resource exhausted error
3	10	Resource exhausted error

III. GPU-BASED MTNP AND RTN/BD TREE

Direct visibility from general visibility preprocessing is inefficient in transmission-inclusive scenarios, as repeated checks for obscured primitives are needed to track transmission counts. This redundancy across visibility tree nodes leads to significant computational overhead, as analyzed in Section VI.

To address these limitations, the proposed TI-HAIT method incorporates GPU-based MTNP, extending HAIT's visibility preprocessing to include 'electromagnetic' visibility with transmission, applicable to both transmission-inclusive and general outdoor scenarios. Also, this section introduces the RTN/BD tree concept to enhance MTNP integration efficiency.

A. MTNP algorithm

The proposed MTNP algorithm consists of two steps: 1) generating sampling points for primitives, and 2) performing shadow test to derive minimum transmission numbers (MTNs) between primitives, Tx, and FOPs. The process for deriving MTNs between primitives is explained first, followed by simpler methods for other cases. Fig. 2 provides a detailed flowchart of the MTNP process.

1) Sampling points generation

The first step generates sampling points on facets and edges with spacing defined by an arbitrary distance. Poisson disk sampling is applied to facets to ensure uniform distribution, maintaining a minimum 'Poisson radius' between points (Fig. 3(a)).

Edge sampling points are placed at uniform intervals equal to the Poisson radius (Fig. 3(b)), applied only to diffraction edges. Non-diffraction edges, such as intersections between walls and the middle of ceilings, are excluded.

2) Deriving MTNs between primitives, Tx, and FOPs through shadow test among sampling points, Tx, and FOPs

After generating sampling points on each facet and edge, shadow test is performed between every pair of points to calculate the number of blocking facets between them using a simple RT algorithm. With theoretically maximum shadow test number $Q = \binom{M}{2} = M(M-1)/2$ with M sampling points, e.g., about 5e+11 shadow tests for a million points, large-scale simulations demand efficient computation. In the proposed MTNP, GPU parallel computing with NVIDIA OptiX resolves this issue, significantly outperforming CPU processing.

Before the main shadow test process, one-dimensional arrays

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

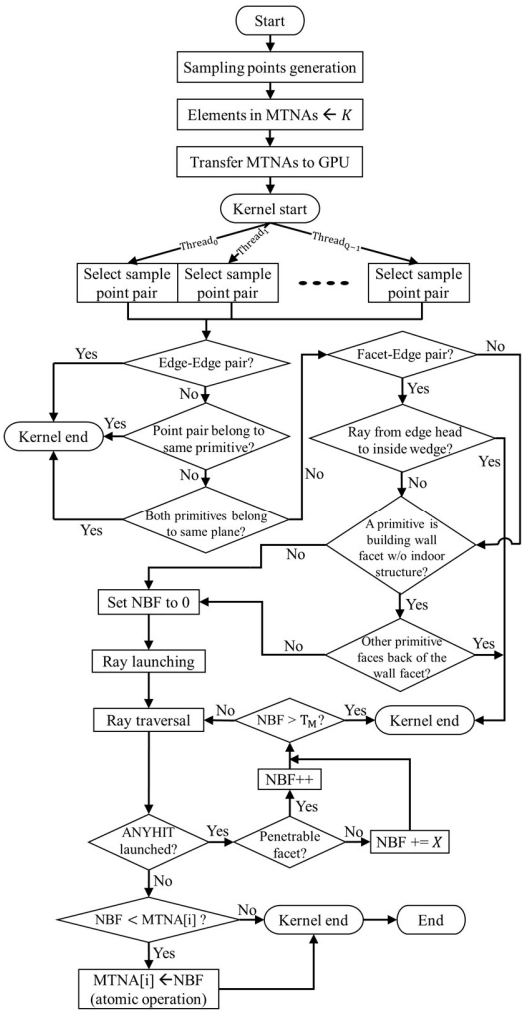
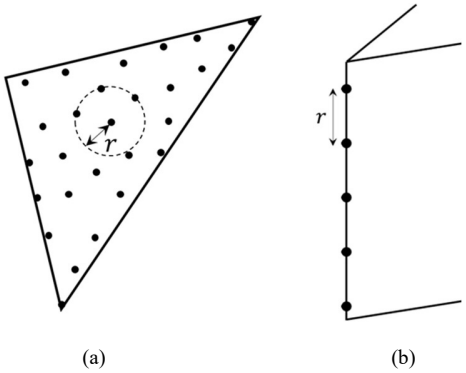


Fig. 2. Flowchart of MTNP between primitives. (NBF: number of blocking facets)



(a) (b)

Fig. 3. Sampling points: (a) Poisson disk sampling on a facet with Poisson radius r, (b) Uniform sampling on an edge with gap r.

called MTN arrays (MTNAs) are allocated on the GPU to store MTN values between primitives. Four types of MTNAs are defined: 1) MTNA_FF: Records MTN values between facets pairs. For k facets, the array size is k^2, with the $(i \cdot k + j)^{th}$ element representing the MTN from the i^{th} facet to the j^{th} facet along the i^{th} facet's normal vector direction half-space. 2) MTNA_FF_O: Similar to MTNA_FF but considers the opposite half-space with i^{th} facet's normal vector direction space. 3) MTNA_FE: Stores MTN values between facets and edges. For k facets and l edges, the array size is $k \cdot l$. The $(i \cdot l + j)^{th}$ element represents the MTN from the i^{th} facet to the j^{th} edge along the i^{th} facet's normal vector direction half-space. 4) MTNA_FE_O: Similar to MTNA_FE but considers the opposite half-space with i^{th} facet's normal vector direction space. Initially, all MTNA elements are set to a very large value K.

Each sampling point pair is assigned to a GPU thread for parallel shadow test operations. While OptiX internally redistributes shadow test workloads among warps and streaming multiprocessors for load balancing, minimizing branch divergence at the controllable algorithmic level remains important. To reduce branch divergence, pairs with similar spatial positions (e.g., within the same primitive pair) are mapped to adjacent OptiX thread indices. This is achieved by sequentially indexing sampling points per primitive and concatenating them, grouping adjacent points together. A one-dimensional OptiX launch is then used, with each thread handling a point pair (p_i, p_j) indexed as: $p_i = N_p - 2 - \alpha$, $p_j = N_p - 1 - (L_i - \alpha(\alpha+1)/2)$, $\alpha = \lfloor(-1 + \sqrt{1 + 8 \cdot L_i})/2\rfloor$ where N_p is the total number of sampling points and L_i is the OptiX's launch index. This indexing strategy ensures adjacent threads in a warp perform similar shadow tests, reducing divergence. Although not applied in this work due to the use of OptiX 7, we note that shader execution reordering, introduced in OptiX 8 for NVIDIA Ada Lovelace GPUs, can offer further optimization potential by on-the-fly thread reordering [31].

The kernel terminates if the pair involves edge sampling points (no multiple diffraction considered), points on the same primitive, or facet-facet or facet-edge pairs on the same plane (no sequential bounces possible). For facet-edge pairs, the kernel also terminates if the ray from the edge point to the facet point is directed inside the wedge, as this study does not analyze diffraction waves propagating within wedges.

If the building lacks indoor structures, rays from primitives located at the interior side of a building wall are considered unreachable to the wall facet, and vice versa. This is because such rays in the main program must traverse the entire building, causing significant attenuation. The kernel terminates early in this case.

Next, a ray is launched between each pair of sampling points, and the number of blocking facets between them is determined using OptiX's ANYHIT program, which is triggered upon ray–facet collisions within the ray segment. The number of blocking facets is initialized to 0 and is incremented by 1 for collisions with penetrable facets (e.g., indoor structures and their exterior walls), or by a large value X for impenetrable ones (e.g., ground, building walls without indoor structure). If the number of blocking facets exceeds the user-defined maximum transmission number T_M, the kernel terminates early to reduce computation.

After computing the number of blocking facets for a point pair, the ray's orientation relative to the facet normal determines which MTNA to update. The MTNA element is

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

atomically updated only if the new number of blocking facets is smaller than the existing value. For facet–facet pairs, two elements in MTNA_FF or MTNA_FF_O may be updated per thread; for facet–edge pairs, one element in MTNA_FE or MTNA_FE_O is considered. After all point pairs are processed, MTNAs store the final MTN values between primitives. This approach avoids storing or transferring all shadow test results to the CPU, retaining only necessary values. Consequently, major CPU–GPU data transfer occurs only in three stages: (1) transferring fundamental information (e.g., geometry and sampling points) to the GPU, (2) sending initial MTNAs to GPU, and (3) retrieving final MTNAs to CPU.

The final preprocessing step transfers MTNA data from the GPU to CPU, optimizing memory usage in the main program. C++ STL vector containers, termed MTN vectors (MTNVs), are used for this purpose. Five MTNV types are generated: MTNV_FF, MTNV_FF_O, MTNV_FE, MTNV_FE_O, and MTNV_EF.

MTNV_FF stores data from MTNA_FF in a 3D structure of size $(\mathrm{T_M}+1, k, l_{mn})$, where k is the number of facets in the scene, and $l_{mn}$is the number of facets reachable from the n^{th} facet at MTN m. The first dimension represents the MTN, the second the reference facet, and the third the list of reachable facets. Fig. 4 shows an example where facet $k-1$ reaches facets 2, 18, and 22 at MTN 1.

MTNV_FF_O, MTNV_FE, and MTNV_FE_O share the same structure as MTNV_FF, storing data from MTNA_FF_O, MTNA_FE, and MTNA_FE_O, respectively. MTNV_EF combines data from MTNA_FE and MTNA_FE_O, with its dimensions representing the MTN, reference edges, and reachable facets.

This concludes the MTNP algorithm for primitives. Similar procedures are applied to Tx–primitive, primitive–FOP, and Tx–FOP pairs, treating Tx and FOP as single points. The MTN for Tx/FOP–primitive pair is the minimum number of blocking facets from shadow tests between Tx/FOP point and primitive sampling points, while Tx–FOP MTNs are computed from a single shadow test. The resulting MTN data are stored in MTNV_TF, MTNV_TE, MTNV_FR, MTNV_FR_O, MTNV_ER, and MTNV_TR for Tx–facet, Tx–edge, facet–FOP, edge–FOP, and Tx–FOP pairs.

B. RTN and BD tree

In TI-HAIT, two additional trees are generated alongside the general visibility tree (Fig. 5(a)), with nodes in one-to-one correspondence. The RTN tree (Fig. 5(b)) stores RTNs for each node to support upward propagation. The BD tree (Fig. 5(c)) stores BD values for facet nodes: 1 if the beam follows the facet normal, 0 otherwise. Edge nodes are excluded, as diffraction beams are assumed to always propagate outward from the wedge in this study.

For example, in the branch marked by green arrows in Fig. 5, the ray from the Tx to facet 1 must penetrate at least $\mathrm{T_M}-5$ facets, as given by MTNV_TF. Thus, 5 is recorded in the first-level RTN node. Since the Tx lies in the half-space aligned with facet 1's normal, the reflected beam follows the normal, and 1 is recorded in the BD tree. The reflected beam then reaches facet 7 with minimally 3 transmissions, as given by

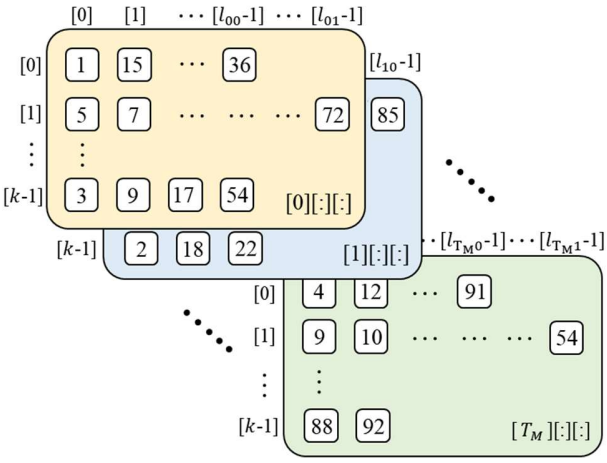


Fig. 4. Example of MTNV_FF.

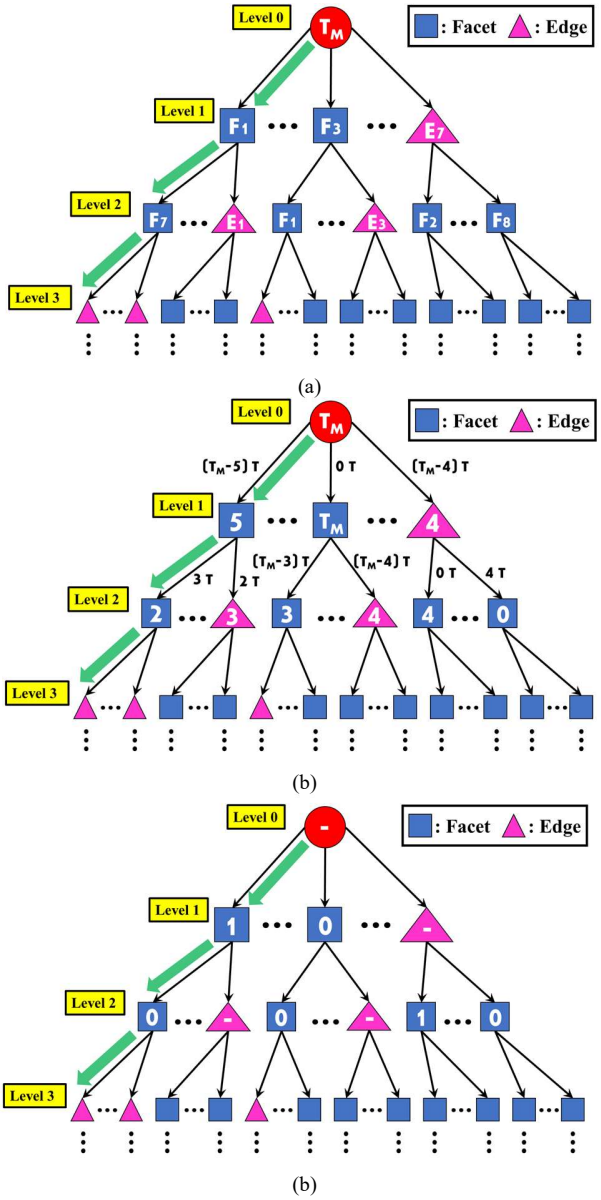


Fig. 5. Three tree concepts in the proposed TI-HAIT: (a) visibility tree, (b) RTN tree, (c) BD tree.

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

MTNV_FF, so $5-3=2$ is recorded in the second-level RTN node. As this second reflected beam propagates opposite to facet 7's normal, 0 is recorded in the BD tree's second level node.

The RTN/BD trees, combined with MTNP results, accelerate visibility tree generation and shadow test by reducing primitives tested for anxel beam inclusion and minimizing FOPs requiring shadow tests. Detailed examples are provided in Section V.

IV. ABT between Different Basis Systems

This section introduces the ABT method for handling different basis systems, addressing propagation paths that end with reflection and may include diffraction—such as multiple reflections, diffraction–reflection, and reflections–diffraction–reflection. For cases involving diffraction without subsequent reflection (e.g., first-order diffraction or reflection–diffraction paths), AZB matrices can be directly applied without ABT, with slight modifications, as detailed in Section V.

The ABT method covers two cases: (A) multiple reflections and (B) diffraction–reflection, with case (B) addressing all diffraction inclusive paths that end in reflection. Anxel beams are defined in the standard spherical coordinate system [15].

A. Multiple reflections

Fig. 6 shows seven possible anxel beam shapes in the multiple reflection case, defined by ϕ' and θ', the ϕ- and θ-angles in the arbitrary $x'y'z'$-basis. Figs. 6(a)–(c) depict common shapes with $\Delta\phi'$ and $\Delta\theta'$ less than π, corresponding to upward, downward, or $x'y'$-plane-inclusive beams. Figs. 6(d) and (e) show beams including the $\pm z'$-axis ($\theta'=0$ or π), where $\Delta\phi'=2\pi$, occurring when the reflecting facet lies in the $\pm z'$-directions relative to the source $\mathbf{S}$ (Tx or image Tx). Figs. 6(f) and (g) illustrate beams with $\Delta\phi'=2\pi$ excluding the $\pm z'$-axis, arising from consecutive reflecting facets in the $\pm z'$-directions at each bounce.

In Figs. 6(d) and (e), $\Delta\theta'$ can exceed $\pi/2$, creating an anxel beam wavefront larger than 2π steradians. In Figs. 6(f) and (g), where ${\theta'}_{min}$ and ${\theta'}_{max}$ denote the beam's angular bounds, ${\theta'}_{min}<\pi/2$ and ${\theta'}_{max}>\pi/2$ may occur simultaneously, making projection onto a plane impossible. Such beams are not transformed in this study, as their transformed shapes would cover excessively large angular regions, reducing efficiency.

In this study, azimuth and elevation angles are denoted as ϕ, θ in the original basis, and as ϕ', θ' for arbitrary systems. The proposed ABT generates a $\phi-\theta$ anxel beam in the original basis that fully encloses the wavefront of a $\phi'-\theta'$ beam while minimizing its wavefront area. For beams like Fig. 6(a), the transformed angular area can be determined using two methods: computing ϕ_{min}^{max} and θ_{min}^{max} (1) over the entire wavefront of the $\phi'-\theta'$ beam or (2) for its four edges. The first is complex and time-consuming, involving multivariate differential equations and extreme value analysis. The second is simpler but still requires solving nonlinear equations. Moreover, the curved side surfaces of anxel beams introduce ambiguity—identical ϕ_{min}^{max}, θ_{min}^{max} values can yield different transformed beam shapes (e.g., Fig. 6(a) vs. 6(d)), adding further complexity.

To simplify the transformation of the beam shape in Fig. 6(a),

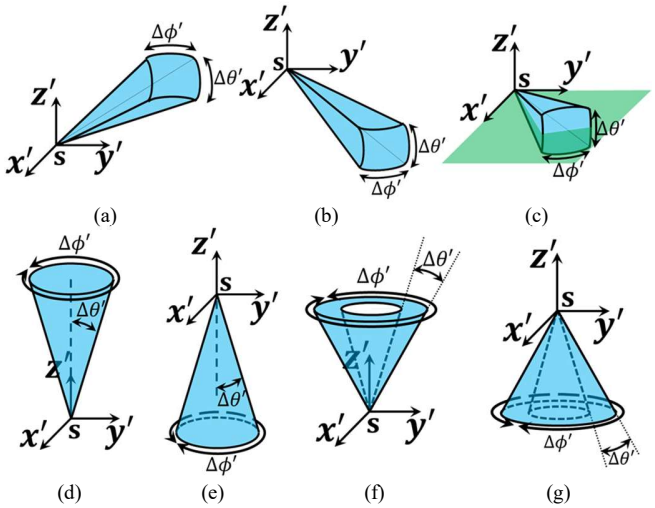


Fig. 6. Seven possible shapes of multiple-reflection anxel beams.

the proposed method defines a quadrangle with four vertices $\mathbf{V_1'}, \mathbf{V_2'}, \mathbf{V_3'}$ and $\mathbf{V_4'}$ that minimally encloses the cut face of the $\phi'-\theta'$ beam, avoiding complexities from its curved sides. These vertices are then transformed into the original basis as $\mathbf{V_1}, \mathbf{V_2}, \mathbf{V_3}$, and $\mathbf{V_4}$ using the following equation [32]:

$$[\mathbf{V_1}\quad \mathbf{V_2}\quad \mathbf{V_3}\quad \mathbf{V_4}] = [\boldsymbol{x'}\quad \boldsymbol{y'}\quad \boldsymbol{z'}][\mathbf{V_1'}\quad \mathbf{V_2'}\quad \mathbf{V_3'}\quad \mathbf{V_4'}] \tag{1}$$

A $\phi-\theta$ anxel beam can then be created to minimally enclose the quadrangle using methods from [33] and [34], ensuring full coverage of the $\phi'-\theta'$ beam wavefront.

The quadrangle is generated by enclosing the cut face of the $\phi'-\theta'$ beam while minimizing its area to improve AZB matrix filtering efficiency. Although infinitely many cutting planes are possible, a simple approach uses a horizontal plane at an arbitrary z', as shown in Fig. 7(a). This yields the same transformed beam as using a plane perpendicular to the $\phi'-\theta'$ beam direction.

The cut face is bounded by two arcs, $\mathbf{ARC_i}(\phi')$ and $\mathbf{ARC_m}(\phi')$, formed by the intersection of the horizontal plane $z'=z_a'$ with the cones defined by $\theta'={\theta'}_{min}$ and ${\theta'}_{max}$. Points on these arcs are expressed as:

$$\mathbf{ARC_i}(\phi') = \mathbf{S} + z_a' \sec{\theta'}_{min}\,(\sin{\theta'}_{min}\cos\phi', \sin{\theta'}_{min}\sin\phi', \cos{\theta'}_{min}) \tag{2}$$

$$\mathbf{ARC_m}(\phi') = \mathbf{S} + z_a' \sec{\theta'}_{max}\,(\sin{\theta'}_{max}\cos\phi', \sin{\theta'}_{max}\sin\phi', \cos{\theta'}_{max}) \tag{3}$$

Two quadrangle vertices, $\mathbf{V_1'}$ and $\mathbf{V_2'}$, lie on $\mathbf{ARC_i}(\phi'_{max})$ and $\mathbf{ARC_i}(\phi'_{min})$, respectively, as shown in Fig. 7(a). However, the other two vertices, $\mathbf{V_3'}$ and $\mathbf{V_4'}$, are not on $\mathbf{ARC_m}(\phi')$ due to the beam's curved sides but instead lie on $\mathbf{ARC_o}(\phi')$, defined as:

$$\mathbf{ARC_o}(\phi') = \mathbf{S} + z_a' \sec{\theta'}^{eff}_{max}\,(\sin{\theta'}^{eff}_{max}\cos\phi', \sin{\theta'}^{eff}_{max}\sin\phi', \cos{\theta'}^{eff}_{max}) \tag{4}$$

This arc represents the intersection of the cone $\theta'={\theta'}^{eff}_{max}$ with the horizontal plane $z'=z_a'$. Setting $\mathbf{V_3'}=\mathbf{ARC_o}(\phi'_{max})$ and $\mathbf{V_4'}=\mathbf{ARC_o}(\phi'_{min})$ makes $\overline{\mathbf{V_3'}\mathbf{V_4'}}$ tangent to $\mathbf{ARC_m}(\phi')$, forming a minimum-area enclosing quadrangle. The value of ${\theta'}^{eff}_{max}$ is calculated using ρ and ρ^{eff}, which are illustrated in Fig. 7(a),

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

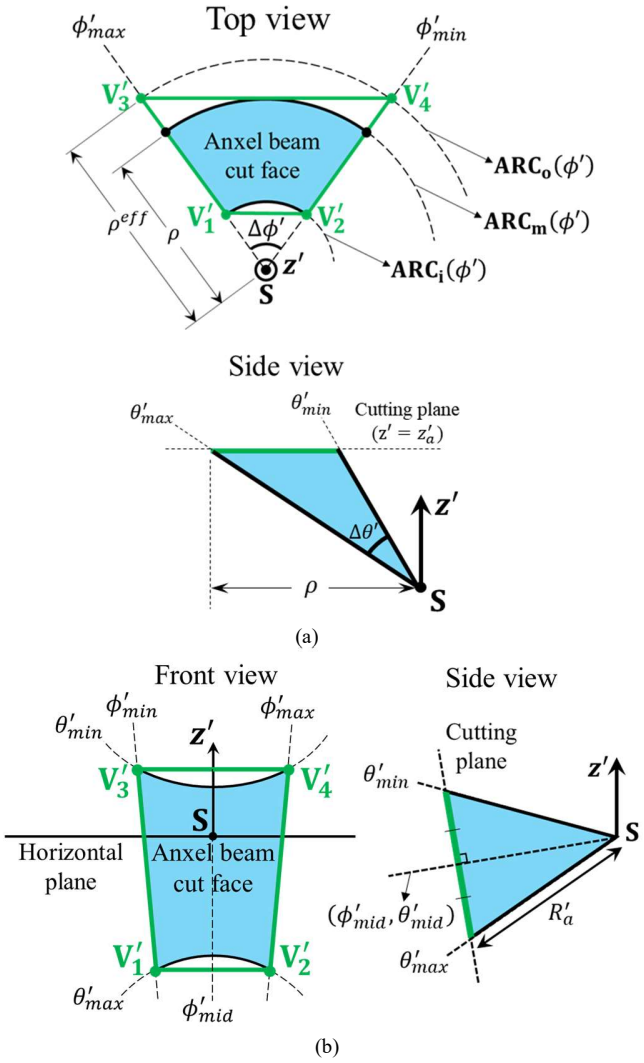


Fig. 7. Enclosing quadrangles for (a) Fig. 6(a) and (b) Fig. 6(c) cases.

and is expressed as:

$$\rho = z'_a \tan\theta'_{max} \tag{5}$$

$$\rho^{eff} = \rho \sec\frac{\Delta\phi'}{2} \tag{6}$$

$$\theta'^{eff}_{max} = \tan^{-1}\frac{\rho^{eff}}{z'_a} = \tan^{-1}\left(\tan\theta'_{max}\sec\frac{\Delta\phi'}{2}\right) \tag{7}$$

For the case shown in Fig. 6(b), the quadrangle can be derived by leveraging the symmetry of the case in Fig. 6(a).

For the anxel beam in Fig. 6(c), the quadrangle's four vertices can align with the beam's corners due to its concave side surfaces, defined by $\theta' = \theta'^{max}_{min}$ (Fig. 7(b)). When the anxel beam is intersected by a plane perpendicular to the $(\phi'_{mid}, \theta'_{mid})$ direction—where $\phi'_{mid} = (\phi'_{min} + \phi'_{max})/2$ and $\theta'_{mid} = (\theta'_{min} + \theta'_{max})/2$—the vertices are expressed as:

$$\mathbf{V'}_{1,2,3,4} = \mathbf{S} + R'_a\left(\sin\theta'^{max}_{min}\cos\phi'^{max}_{min}, \sin\theta'^{max}_{min}\sin\phi'^{max}_{min}, \cos\theta'^{max}_{min}\right) \tag{8}$$

where R'_a is an arbitrary positive value.

For $\phi' - \theta'$ anxel beams shaped like Fig. 6(d) or (e), an enclosing quadrangle is unnecessary. The transformed $\phi - \theta$ beam's exact ϕ^{max}_{min} and θ^{max}_{min} values can be directly derived using the circular symmetry of the $\phi' - \theta'$ beam. Fig. 8 shows the transformation process for a Fig. 6(d)-shaped beam. In Fig. 8(a), where $\Delta\theta' \le \theta_{z'} \le \pi - \Delta\theta'$, $\theta_{z'}$ is the θ-angle of the z'-axis relative to the original basis system, and $\Delta\theta'$ is the θ'-span of the beam (Fig. 6(d)). Using circular symmetry, θ^{max}_{min} is $\theta_{z'} \pm \Delta\theta'$. Projecting the $\phi' - \theta'$ beam onto the xy-plane, ϕ^{max}_{min} becomes $\phi_{z'} \pm \Delta\phi$, where $\phi_{z'}$ is the ϕ-angle of the z'-axis, and $\Delta\phi$ is calculated as:

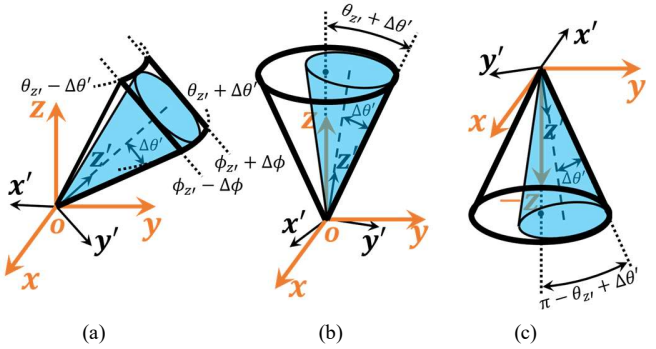


Fig. 8. ABT for the Fig. 6(d) case: (a) $\Delta\theta' \le \theta_{z'} \le \pi - \Delta\theta'$, (b) $\theta_{z'} < \Delta\theta'$, (c) $\theta_{z'} > \pi - \Delta\theta'$.

$$\Delta\phi = \tan^{-1}\left(\frac{\tan\Delta\theta'}{\sin\theta_{z'}}\right) \tag{9}$$

Fig. 8(b) shows the case $\theta_{z'} < \Delta\theta'$, where the $\phi' - \theta'$ anxel beam includes the z-axis. The transformed beam corresponds to Fig. 6(d), with $\theta_{max} = \theta_{z'} + \Delta\theta'$. Conversely, Fig. 8(c) illustrates $\theta_{z'} > \pi - \Delta\theta'$, where the beam includes the $-z$-axis. In this case, the transformed beam matches Fig. 6(e), with $\theta_{min} = \theta_{z'} - \Delta\theta'$.

Beams in Fig. 6(e) are transformed using the symmetry of Fig. 6(d). Beams in Fig. 6(f) and (g) follow the same principles as Fig. 6(d) and (e), ignoring cavities within the $\phi' - \theta'$ beam.

B. Diffraction-reflection

In HAIT, the diffraction-reflection mechanism simplifies to a single diffraction [17], where a beam from the image Tx hits the image edge, producing a diffracted beam toward the reflecting facet. Fig. 9 shows the five primary $\phi' - \beta'$ beam shapes transformable into the original basis system.

Fig. 9(a) illustrates a $\phi' - \beta'$ anxel beam with $\Delta\phi' < \pi$ and $\beta'^{tmax}_{tmin} < \pi/2$, where $\Delta\phi' = \phi'_{max} - \phi'_{min}$ and β'^{tmax}_{tmin} is the angle between the z'-axis and diffraction rays from edge points t_{min} and t_{max}. The normalized edge length t ranges from 0 to 1, representing the start and end of the edge. The z'-axis aligns with the edge direction vector $\overrightarrow{P_1P_2}$, though they may point in opposite directions during ABS reflection. For simplicity, this study assumes z'-axis aligns with $\overrightarrow{P_1P_2}$, handling the opposite case through symmetry.

Fig. 9(b) illustrates $\Delta\phi' < \pi$ and $\beta'^{tmax}_{tmin} > \pi/2$. Fig. 9(c) depicts $\Delta\phi' < \pi$, $\beta'_{tmin} > \pi/2$ and $\beta'_{tmax} < \pi/2$. In Fig. 9(d), $\Delta\phi' > \pi$ and $\beta'^{tmax}_{tmin} < \pi/2$. Finally, Fig. 9(e) shows $\Delta\phi' > \pi$ and $\beta'^{tmax}_{tmin} > \pi/2$.

In all other cases, such as when $\Delta\phi' > \pi$, $\beta'_{tmin} > \pi/2$, and $\beta'_{tmax} < \pi/2$, the beam cannot be projected onto a 2D plane. In such scenarios, ABT is avoided, as the transformed beam

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

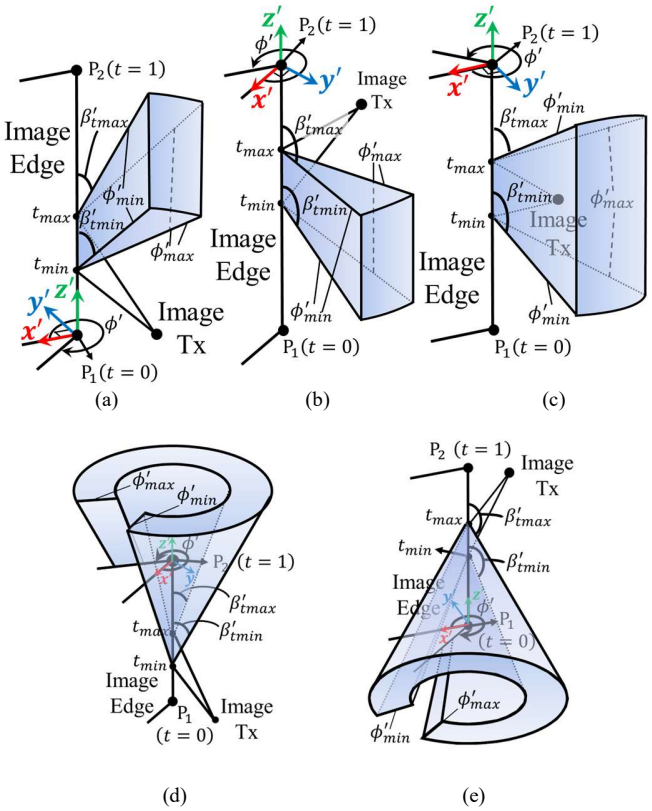


Fig. 9. Five possible shapes of diffraction-reflection anxel beams that can be transformed in the original basis system.

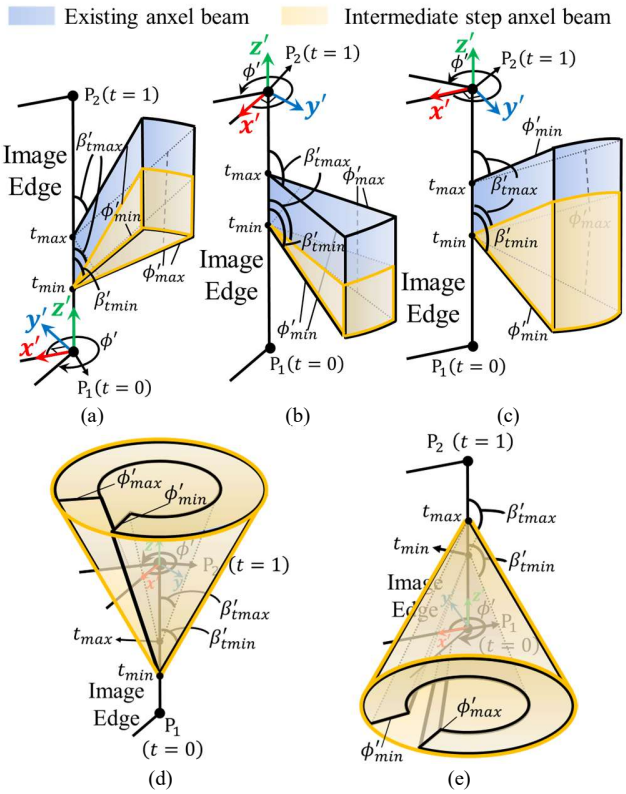


Fig. 10. Intermediate step anxel beams with a point source covering all angular areas of anxel beams shown in Fig. 9.

would likely encompass an excessively large solid angle, resulting in inefficiency.

In the diffraction–reflection mechanism, the anxel beam is emitted from an edge rather than a point, unlike in multiple reflections. However, the source's shape or location is not critical, as the transformed beam is used to identify primitives within its angular region relative to the reflecting facet, not the source. In other words, using AZB matrices, primitives are identified within $\phi_{min} < \phi < \phi_{max}$ and $\theta_{min} < \theta < \theta_{max}$, with angles measured from any point on the reflecting facet. Thus, the key requirement is only that the transformed beam fully encloses the angular extent of the $\phi' - \beta'$ beam while minimizing wavefront area.

To achieve this, an intermediate step involves constructing a $\phi' - \beta'$ anxel beam with a point source that fully covers the angular region of the existing edge-originating beam, as illustrated in Fig. 10. For Figs. 9(a)-(c), the source is set at $\mathbf{P}_{tmin}$, and the beam's corners are defined as $(\phi', \beta') = (\phi'^{max}_{min}, \beta'^{tmax}_{tmin})$, corresponding to Figs. 6(a)-(c). For Figs. 9(d) and (e), point-source anxel beams corresponding to Figs. 6(d) and (e) are created by setting the source at $\mathbf{P}^{tmax}_{tmin}$ and using $\theta'^{max}_{min} = \beta'^{tmax}_{tmin}$, as shown in Fig. 10(d) and (e).

All intermediate beams in Fig. 10 can be transformed into the original basis using the same method as in the multiple reflection case, with the θ'-domain replaced by the β'-domain.

V. Application of MTNP and AZB Matrices in TI-HAIT Using RTN/BD Trees and ABT

This section describes how MTNP results, AZB matrices, and related implementation details are integrated into the TI-HAIT framework using RTN/BD trees and ABT. Updated workflows are presented for each main component of TI-HAIT compared to general HAIT. As in general HAIT, TI-HAIT consists of three components: (A) visibility tree generation, (B) shadow test, and (C) field calculation. MTNP results and RTN/BD trees are used in both (A) and (B), while ABT and AZB matrices are applied specifically in (A).

In this study, the definition of AZB matrices is slightly modified. In TI-HAIT, the anxel beam's angular area narrows exponentially with increasing bounce order due to the ABS method. To search primitives within an anxel beam effectively, AZB matrices would require narrow angular regions per anxel, increasing memory usage as the same primitives are stored across multiple anxels. Additionally, extracting candidate primitives from adjacent anxels requires duplication checks, and defining anxel's suitable angular areas introduces ambiguity. To address these issues, this study avoids subdividing anxels in AZB matrices. Instead, exact visible angles between electromagnetically visible primitives are directly recorded as ϕ^{max}_{min} and θ^{max}_{min} for facet references, and β'^{max}_{min} for edge references (Section V-A).

A. Visibility tree generation

The TI-HAIT visibility tree generation algorithm shares its structure with general HAIT but has a key difference. In TI-HAIT, MTNP results and AZB matrices, integrated through RTN/BD trees and ABT, enhance visibility tree generation efficiency. This is demonstrated in Fig. 11 for attaching child

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

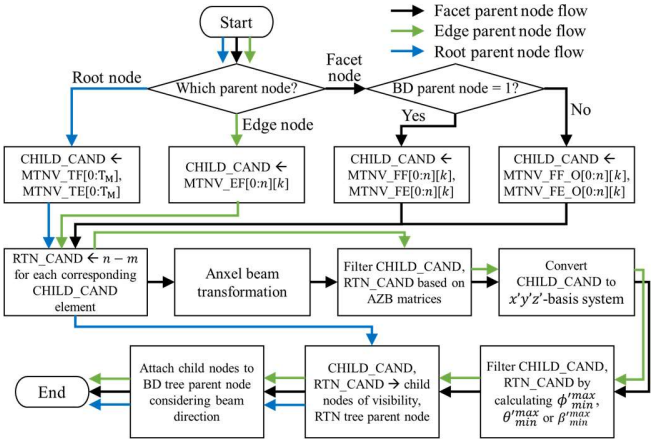


Fig. 11. Flowchart for attaching child nodes to the visibility, RTN, and BD trees.

nodes to arbitrary parent nodes.

For a facet parent node, electromagnetically visible primitives are retrieved using MTNP results and RTN/BD tree values. For parent facet k with RTN n, primitives are fetched from MTNV_FF[0:n][k][:] and MTNV_FE[0:n][k][:] if the BD tree value is 1, or from MTNV_FF_O[0:n][k][:] and MTNV_FE_O[0:n][k][:] if it is 0. These candidates are stored in the array CHILD_CAND. Without RTN and BD trees, all primitives from MTNV_FF, _FE, _FF_O, _FE_O[0:$\mathrm{T_M}$][k][:] (where $\mathrm{T_M} \geq n$) would need to be accessed, leading to inefficiency. RTN and BD trees thus significantly reduce candidate primitives, accelerating visibility tree generation.

An array, RTN_CAND, is also created to store RTN values for each primitive in CHILD_CAND. For primitives from MTNV[m][k][:], $n - m$ is stored in RTN_CAND.

Next, the reflected anxel beam from facet k, defined in the $x'y'z'$-basis, is transformed into the xyz-basis using ABT (Section IV). Primitives in CHILD_CAND and their RTN_CAND values are filtered by comparing the transformed beam's angular boundaries with the angular regions in the AZB matrices generated during preprocessing.

Primitives outside the existing $\phi' - \theta'$ anxel beam are further filtered by converting their coordinates to the $x'y'z'$-basis and verifying their angular bounds (ϕ'^{max}_{min}, θ'^{max}_{min}, or β'^{max}_{min}) relative to the image source or edge.

Filtered primitives in CHILD_CAND are added as child nodes in the visibility tree, with their RTN_CAND values added to the RTN tree. In the BD tree, facet child nodes are assigned 1 or 0 based on the reflected beam direction, while edge child nodes are assigned NAN.

For edge parent nodes, child nodes are attached similarly to facet parent nodes but use MTNV_EF instead of MTNV_FF, MTNV_FE, MTNV_FF_O, and MTNV_FE_O (Fig. 11). Notably, ABT is unnecessary for edge parent nodes, as AZB matrices can be directly applied with minor modifications.

In the basis system from Fig. 9, defined by the ABS method in the HAIT, non-reflected diffraction beams propagate throughout the entire ϕ'-domain outside the wedge and within an arbitrary β'-domain. At this point, MTNP results filter out all primitives outside the ϕ'-domain, leaving AZB matrices to filter only those outside the β'-domain. To eliminate the need for ABT, we construct AZB matrices for edge references using β'-angle ranges in the fixed $x'y'z'$-basis (Fig. 9), instead of $\phi - \theta$ angles in the original basis. These matrices encode the β' ranges where diffraction rays from the edge can reach other primitives. Primitives outside the β' domain are efficiently excluded by comparing the β' region of the non-transformed anxel beam with the AZB matrices.

For root (Tx) nodes, all MTNV_TF and MTNV_TE elements are directly assigned as child nodes without filtering (Fig. 11). Their corresponding RTN and BD child nodes are determined in the same manner as described above.

B. *Shadow test*

The shadow test algorithm in TI-HAIT is similar to general HAIT but with two key differences:

1) *Incorporation of MTNP results using the RTN/BD tree*

Unlike general HAIT, which assigns FOPs to primitive nodes based on conventional visibility preprocessing, TI-HAIT leverages MTNP results and RTN/BD trees to assign only electromagnetically visible FOPs, including transmission paths. For a parent node with facet index k and RTN n, only FOPs from MTNV_FR[0: n][k][:] (BD = 1) or MTNV_FR_O [0: n][k][:] (BD = 0) are assigned. Without MTNP and RTN/BD trees, all FOPs would need to be attached, as there would be no way to determine the number of transmissions required for each FOP to be reached from the primitive or the RTN and BD for each node.

2) *Counting transmission numbers in the shadow test*

In TI-HAIT, the shadow test accounts for transmissions. Unlike general HAIT, where shadow test passes if the ray path to the FOP is unobstructed, TI-HAIT passes the shadow test if the ray avoids impenetrable facets and crosses no more than $\mathrm{T_M}$ penetrable facets. The ANYHIT program in NVIDIA OptiX monitors the number of penetrable facets along the ray path. A variable COUNT, initialized to zero, increments with each collision with a penetrable facet. If ANYHIT detects an impenetrable facet, the shadow test terminates immediately for efficiency.

C. *Field calculation*

Field calculation in TI-HAIT differs from general HAIT in two key ways:

1) *Recalculating transmission facets during the field calculation phase in the CPU environment*

In TI-HAIT, unlike general HAIT, identifying transmission-causing facets along ray paths is required. Due to GPU memory constraints, these indices are not stored during shadow test, so rays that pass the shadow test are retraced on the CPU during field calculation. If memory permits, transferring the indices from GPU to CPU can improve performance. Otherwise, the authors recommend using Intel's Embree library for efficient CPU-based ray-facet intersection tests.

2) *Field calculation theory*

After identifying transmission facets, fields at the FOP are calculated as in general HAIT, using geometrical optics (GO) [35]-[38] for reflections and uniform geometrical theory of diffraction (UTD) [39], [40] for diffractions, with an additional step to incorporate transmission coefficients when a ray passes

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

through a facet. The generalized transmission coefficients are detailed in [35, equations (5-90a)–(5-92k)].

D. Time and space complexity

The general IT-based RT algorithm has a time complexity of $O(N^k)$, where N is the number of primitives and k is the number of ray bounces. This arises from the exponential growth of the visibility tree, expanding as N^1, N^2, N^3, … at each level [28]. TI-HAIT mitigates this growth through three key techniques—MTNP, ABS, and ABT/AZB matrices—that reduce the visibility tree size by average factors W, X, and Y, respectively. MTNP restricts child nodes to electromagnetically visible primitives only. ABS method further filters out primitives outside the shrunken anxel beam at each tree node. ABT and AZB matrices complement ABS by eliminating remaining out-of-beam primitives that ABS alone cannot filter (see Section VI), enabling additional pruning. The combined effect yields a reduced complexity of $O[\{N/(WXY)\}^k]$. In addition, ABT and AZB matrices provide a linear acceleration factor Z by speeding up primitive filtering at each tree node, resulting in a final time complexity of $O[\{N/(WXY)\}^k/Z]$. Here, W, X, Y, $Z > 1$, and their values depend on simulation parameters such as geometry and transmitter location.

The space complexity is $O[\{N/(WXY)\}^k]$, as it is solely determined by the size of the constructed visibility tree and is not affected by the linear acceleration factor Z.

VI. Validation

This section evaluates TI-HAIT under three configurations: without MTNP, without ABT, and using the IT solver from the commercial ray tracer WinProp, across both simple and complex O2I scenarios. Other well-known IT-based ray tracing tools—Sionna, newFASANT, Eigenray, and MATLAB—were considered but ultimately deemed unsuitable as baselines due to certain limitations. Sionna's exhaustive solver, while supporting GPU acceleration, does not handle ray transmission or reflection–diffraction combinations, which are essential for this study. newFASANT's GTD module allows up to two transmissions, limiting its applicability in deep indoor scenarios. Eigenray supports up to three reflections and diffractions, which restricts its use in dense urban environments. MATLAB's IT solver supports only second-order reflections and does not model transmission or diffraction. In contrast, WinProp supports up to six reflections and transmissions, two diffractions, and all reflection–diffraction combinations. Despite lacking GPU acceleration, it was the only tool that supported the same high number of bounces and transmissions as TI-HAIT, making it suitable for accuracy evaluation.

Implementing TI-HAIT without ABT omits AZB matrix generation in the preprocessor and skips primitive filtering by ABT and AZB matrices in the main program.

When MTNP and the RTN tree are omitted, conventional visibility preprocessing is used for transmission analysis. (To mitigate excessive performance degradation, the BD tree is employed in this configuration, although it is not ideally suited for this case.) Candidate primitives within the anxel beam are identified through an iterative process. First, List 0 is constructed using general visibility preprocessing to include primitives directly visible from the source primitive. Next, List 1 includes primitives that are potentially reachable with one transmission, identified as those directly visible from the primitives in List 0 along the beam direction using visibility preprocessing results. Duplicate entries within and across lists are removed to avoid redundancy. This process is repeated up to T_M times, resulting in multiple lists (List 0 to List T_M) that collectively define the candidate set for the search process.

This method requires recursive processing and duplication checks, and produces a larger visibility tree than with the MTNP and RTN tree, since primitives in List n may have actual MTNs from the parent exceeding n. Moreover, the combined candidate set may include many primitives whose MTNs exceed the RTN in the RTN tree. Consequently, the time complexity becomes $O[\alpha \cdot \{\beta N/(WXY)\}^k/Z]$ where $\alpha > 1$ accounts for linear degradation from recursion and duplication checks at each visibility tree node, and $\beta > 1$ for exponential degradation due to increased child nodes. The space complexity is given by $O[\{\beta N/(WXY)\}^k]$, excluding linear time factors.

Without MTNP and RTN trees during the shadow test phase, identifying FOPs reachable with specific transmission numbers is infeasible. As a result, all FOPs within the half-space defined by a facet's normal must be attached to facet nodes, while those outside the wedge are assigned to edge nodes.

Without MTNP, AZB matrix generation requires angle calculations for all primitives, increasing complexity and resource usage, whereas MTNP limits processing to primitives with MTNs below T_M, improving efficiency.

WinProp computes fields using GO and UTD, similar to TI-HAIT, but simplifies transmission by applying two coefficients—one for slab entry and one for exit—assuming a single attenuation path without phase delay. For fair comparison, TI-HAIT is configured to omit internal reflections and phase delays inside the slab, typically modeled using detailed coefficients in [35, equations (5-90a)–(5-92k)]. The resulting transmission coefficients for validation are as follows:

$$T_{eff(TE,TM)} = T_{(TEi,TMi)} \cdot T_{(TEo,TMo)} \cdot e^{-\alpha_s d_s} \tag{10}$$

where $T_{(TEi,TMi)}$ and $T_{(TEo,TMo)}$ are the transmission coefficients for TE and TM polarizations when entering the slab from air and exiting the slab to air, respectively. α_s denotes the slab's attenuation constant, and d_s its thickness. These parameters are defined as follows [36]:

$$T_{(TEi,TMi)} = \frac{2(\mu_s, \varepsilon_s) j\beta_0 \cos\xi_0}{(\mu_s, \varepsilon_s) j\beta_0 \cos\xi_0 + (\mu_0, \varepsilon_0)(\alpha_s + j\beta_s \cos\xi_s)} \tag{11}$$

$$T_{(TEo,TMo)} = \frac{2(\mu_0, \varepsilon_0)(\alpha_s + j\beta_s \cos\xi_s)}{(\mu_0, \varepsilon_0)(\alpha_s + j\beta_s \cos\xi_s) + (\mu_s, \varepsilon_s) j\beta_0 \cos\xi_0} \tag{12}$$

Here, ε_0 and μ_0 denote the permittivity and permeability of air, while ε_s and μ_s represent those of the slab. ξ_0 and ξ_s are the incidence and refraction angles of the phase vector at the air–slab interface. The intrinsic phase constant of air is β_0, and the slab's effective attenuation (α_s) and phase constants (β_s) are given as:

$$\alpha_s = \sqrt{(|\gamma_{0t}|^2 + \mathrm{Re}(\gamma_{0s}^2) + |\gamma_{0t}^2 - \gamma_{0s}^2|)/2} \tag{13}$$

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

$$\beta_s = \sqrt{(|\gamma_{0t}|^2 - \mathrm{Re}(\gamma_{0s}^2) + |\gamma_{0t}^2 - \gamma_{0s}^2|)/2} \quad (14)$$

where $\gamma_{0t} = j\beta_0 \sin\xi_0$ and $\gamma_{0s} = \sqrt{j\omega\mu_s(\sigma_s + j\omega\varepsilon_s)}$. Here, σ_s represents the conductivity of the slab material [37].

To ensure consistency, the same CAD files were used for both TI-HAIT and WinProp simulations. As a result, E-field results would be nearly identical, with minor differences arising only from floating-point precision.

All simulations were conducted in the same computational environment as in Section II.

All in-house codes—TI-HAIT, its variants without MTNP and ABT—were implemented in C/C++ and OptiX.

The first simulation scenario, shown in Fig. 12 and detailed in Table III, represents a relatively simple environment: Teheran-ro in Seoul, South Korea, including an indoor structure. The AutoCAD model contains 409 triangular facets—269 for outdoor and 140 for indoor structures.

In this scenario, the E-field at 28 GHz is analyzed for a Hertzian dipole Tx radiating 1 W, positioned at (-50, 0, 10) m. 100 FOPs are evenly distributed along a line from (-42, 85, 11.5) m to (-23, 85, 11.5) m. The environment includes concrete, glass, and PEC materials (details in Table IV [17]). PEC is assigned to the ground, building exteriors without indoor structures, and wedge facets to address differences between the diffraction models: the heuristic model in [40] for TI-HAIT and the model in [41] for WinProp. Glass facets (blue in Fig. 12(b)) are 2 cm thick, while other facets are 20 cm thick concrete. The simulation considers up to six reflections and one diffraction, accounting for all reflection–diffraction combinations, and up to four transmissions. Poisson radii of 1 m (outdoors) and 0.1 m (indoors) are used for MTNP, yielding approximately 120,000 primitive sampling points.

Table V compares TI-HAIT, its variants, and WinProp in a simple scenario. All TI-HAIT variants yield zero RMSPE, as MTNP and ABT only exclude unnecessary primitives without affecting accuracy. Computation times are similar up to three bounces, but TI-HAIT becomes significantly faster at higher orders—2.8×, 10.3×, and 19.1× faster than without MTNP at four to six bounces, and 1.4–1.7× faster than without ABT at five and six bounces. Preprocessing times remain low (3–5 seconds) for all cases.

WinProp simulations are limited to four bounces due to excessive computation time at higher orders. RMSPEs between TI-HAIT and WinProp are only 0.83, 0.65, 1.12, and 1.95% for maximum bouncing order of 1, 2, 3, and 4, likely due to floating-point precision differences. Fig. 13 shows E-field results for TI-HAIT, its variants, and WinProp at four bounces. The near-constant E-field between $x = -32$ and -28 results from the region being mostly enclosed by PEC facets, which significantly limits transmission paths and leaves only one dominant propagation path.

TI-HAIT is significantly faster—2× faster than WinProp at one and two bounces, and 15× and 6,051× faster at three and four bounces, respectively. This speedup is primarily due to the ABS method, CPU/GPU parallelism, and a heterogeneous algorithm that accelerate shadow test and field calculations [17]. MTNP with RTN/BD trees and AZB matrices with ABT further improve efficiency.

Fig. 14 shows the second simulation scenario, a massive urban and indoor environment in Gangnam, Seoul, with 6,764 triangular facets: 5,924 for outdoor structures and three identical indoor structures with 280 facets each. Spanning 1.1 × 1.1 km, it is more complex than the simple scenario. Indoor blue facets (Fig. 14(b)) are 1 cm thick glass, building walls are 20 cm thick concrete, and the ground is wet earth with a relative

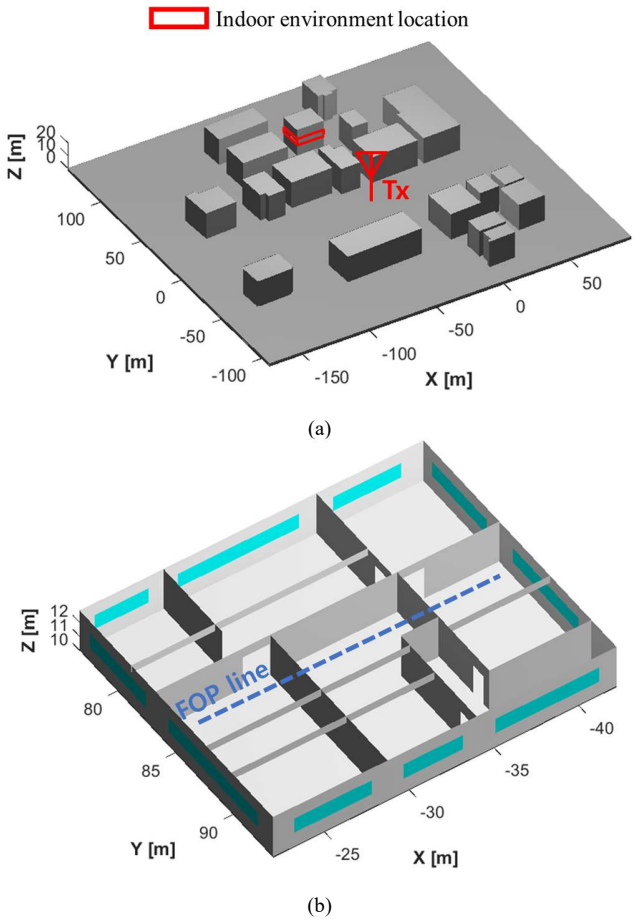


Fig. 12. Simulation environment of the first scenario (simple scenario) describing Teheran-ro in Seoul, South Korea, and locations of Tx and FOPs: (a) Outdoor and (b) Indoor structures.

TABLE III
SIMULATION PARAMETERS OF SIMPLE SCENARIO

Frequency	Tx antenna	Tx radiation power
28 [GHz]	Hertzian dipole (z-directed)	1 [W]
Tx location	Max. Bouncing #	Number of FOP
(-50, 0, 10) [m]	1, 2, 3, 4, 5, 6	100
Poisson radius	T_M	Location of FOP
Outdoor: 1[m] Indoor: 0.1 [m]	4	Uniformly distributed on line between (-42 to -23, 85, 11.5) [m]

TABLE IV
ELECTRICAL PROPERTIES OF CONCRETE, GLASS, AND PEC

	ε_r	μ_r	σ
Concrete	6.5	1	0.668
Glass	4.7	1	0.022
PEC	1	1	10^{18}

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

TABLE V
PERFORMANCE FOR SIMPLE SCENARIO

Max. Bounce #	Computation time [sec]			
	TI-HAIT	TI-HAIT w/o MTNP	TI-HAIT w/o ABT	WinProp
Preprocessing	4	5	3	-
1	5	5	5	19
2	5	5	5	20
3	5	6	5	136
4	6	17	6	60,514
5	19	195	26	-
6	140	2,674	234	-

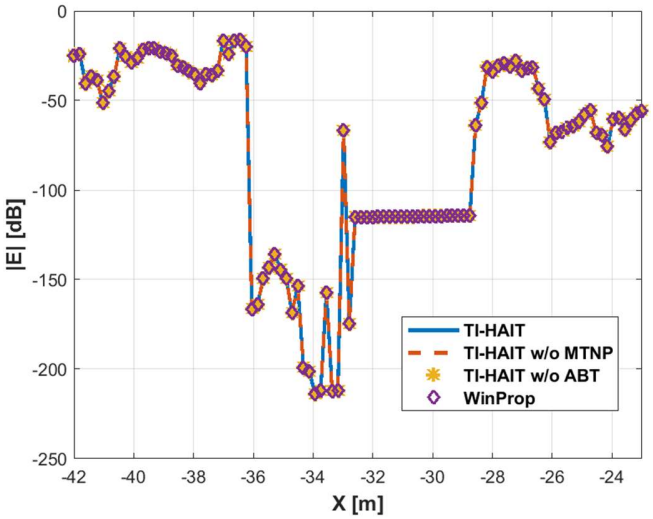


Fig. 13. E-Field simulation results for TI-HAIT, TI-HAIT without MTNP, TI-HAIT without ABT, and WinProp with a maximum bouncing number of four in the simple scenario

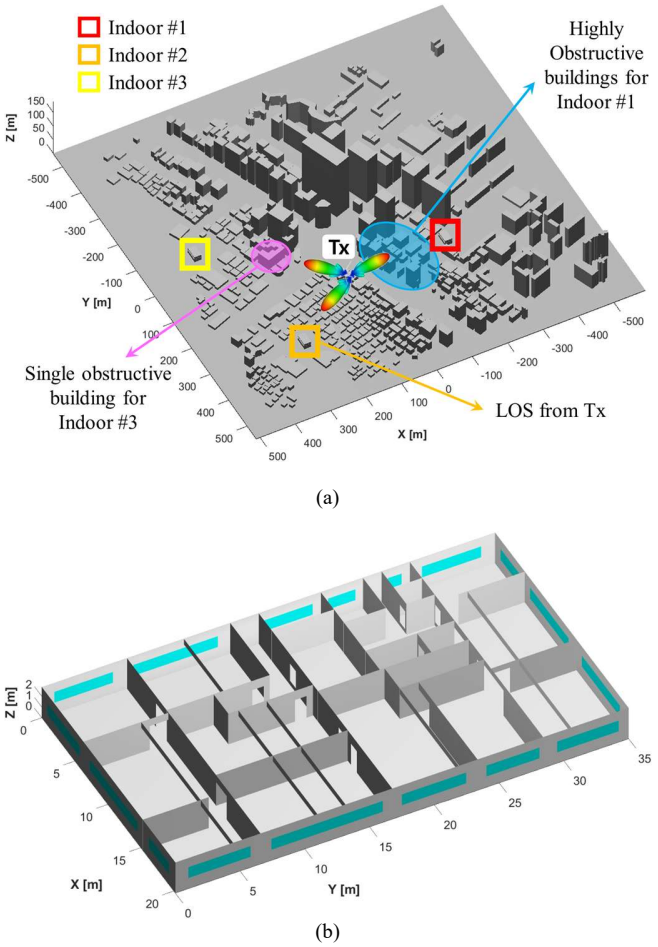


Fig. 14. Simulation environment of the second scenario (massive scenario) describing Gangnam in Seoul, South Korea: (a) Outdoor and (b) Indoor structure.

TABLE VI
SIMULATION PARAMETERS OF MASSIVE SCENARIO

Frequency	Tx antenna	Tx radiation power
28 [GHz]	12 dB directive antenna	50 [W] per each Tx
Tx location [m]	Max. Bouncing #	Number of FOP
#1: (37, 167.2, 12.5) #2: (55.4, 185.9, 12.5) #3: (55.4, 165.6, 12.5)	1, 2, 3, 4, 5	735 per indoor structure
Poisson radius	T_M	Location of FOP
Outdoor: 1 [m] Indoor: 0.1 [m]	4	Uniformly distributed at 1.5 [m] height within each indoor structure

permittivity of 15, permeability of 1, and conductivity of 1.336 [42]. Other material properties remain as in Table IV.

Simulation parameters are in Table VI. The E-field is analyzed at 28 GHz using a base station with three 12 dB directive Tx antennas (each radiating 50 W), positioned at (37, 167.2, 12.5) m, (55.4, 185.9, 12.5) m, and (55.4, 165.6, 12.5) m. Their main beams are directed toward the centers of indoor structures #1, #2, and #3, respectively. Indoor #1 is heavily obstructed by multiple buildings from the Tx, indoor #2 has a clear line-of-sight (LOS) to the Tx, and indoor #3 is slightly obstructed by a single building. Each structure contains 735 FOPs uniformly placed at 11.5 m height (1.5 m above the 10 m indoor floor). Fields from antennas targeting other structures are excluded due to negligible impact. The simulation includes up to five reflections and one diffraction, accounting for all reflection–diffraction combinations, and up to four transmissions. This setting achieves a large-scale fading error within 5% compared to six-bounce results [17]. To manage GPU memory, the visibility tree is partitioned into eight parts at five bounces [17]. MTNP uses Poisson radii of 1 m (outdoor) and 0.1 m (indoor), generating approximately two million primitive sample points.

This massive scenario was simulated using TI-HAIT and its variants (without MTNP and ABT), excluding WinProp due to excessive computation time. Tables VII and VIII summarize computation times and memory usage. We note that, like the simple scenario, all methods achieved zero RMSPE, confirming MTNP and ABT do not impact accuracy.

Table VII presents preprocessing times: on average, 236 s for TI-HAIT, 566 s without MTNP, and 190 s without ABT. TI-HAIT is 2.4× faster than the version without MTNP but 1.2× slower than without ABT. The longer time without MTNP results from generating AZB matrices for all primitives, whereas with MTNP, matrices are created only for electromagnetically visible primitives, significantly reducing preprocessing time.

TI-HAIT without MTNP was limited to a maximum bouncing order of 2 due to excessive computation times (e.g.,

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

TABLE VII
COMPUTATION TIME PERFORMANCE FOR MASSIVE SCENARIO. (VTG: VISIBILITY TREE GENERATION, ST: SHADOW TEST, FC: FIELD CALCULATION)

Max. Bounce #		Computation time [sec]								
		TI-HAIT			TI-HAIT w/o MTNP			TI-HAIT w/o ABT		
		Tx #1	Tx #2	Tx #3	Tx #1	Tx #2	Tx #3	Tx #1	Tx #2	Tx #3
preprocessing		238	235	234	571	567	561	189	190	192
1	Total	30	30	31	145	147	145	15	17	16
	VTG	5	4	5	8	9	9	3	4	4
	ST/FC	2	2	2	2	2	2	1	2	1
2	Total	30	32	32	164	172	171	15	18	18
	VTG	4	4	5	24	27	26	3	4	4
	ST/FC	2	4	3	7	9	8	2	4	3
3	Total	47	89	70	>7,200	>7,200	>7,200	38	87	67
	VTG	6	9	8	-	-	-	14	18	16
	ST/FC	13	51	35	-	-	-	13	53	37
4	Total	365	863	629	-	-	-	752	1,561	1,361
	VTG	143	259	210	-	-	-	542	923	854
	ST/FC	167	575	388	-	-	-	189	621	489
5 (Tree partition)	Total	7,554	14,415	10,298	-	-	-	19,579	26,724	23,558
	VTG	4,616	4,695	4,176	-	-	-	16,098	15,602	16,191
	ST/FC	2,885	9,661	6,063	-	-	-	3,439	11,075	7,319

TABLE VIII
MEMORY CONSUMPTION PERFORMANCE FOR MASSIVE SCENARIO

Max. Bounce #	Memory [GB]																	
	TI-HAIT						TI-HAIT w/o MTNP						TI-HAIT w/o ABT					
	Tx #1		Tx #2		Tx #3		Tx #1		Tx #2		Tx #3		Tx #1		Tx #2		Tx #3	
	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
1	1.8	12.9	1.9	12.9	1.9	12.9	6.1	12.9	6.1	12.9	6.1	12.9	0.6	12.9	0.6	12.9	0.6	12.9
2	1.8	12.9	2.7	12.9	2.0	12.9	8.2	12.9	9.6	12.9	8.9	12.9	0.6	12.9	2.0	12.9	1.3	12.9
3	7.8	13.0	22.3	13.0	19.4	13.0	-	-	-	-	-	-	7.5	13.0	21.6	13.0	19.3	13.0
4	24.6	14.8	25.0	15.2	25.0	15.1	-	-	-	-	-	-	24.1	15.1	24.6	15.4	24.5	15.4
5 (Tree partition)	48.5	33.7	47.9	32.8	46.9	31.9	-	-	-	-	-	-	51.8	36.4	51.3	35.5	50.0	35.0

over two hours for order 3). On average, TI-HAIT is 4.8 and 5.4 times faster than without MTNP for orders 1 and 2, respectively. For order 3, where simulations without MTNP were infeasible, TI-HAIT showed an average efficiency increase of at least 105 times, highlighting MTNP's importance for efficient transmission analysis.

As shown in Table VIII, both methods use similar GPU memory (~12.9 GB) due to FOP indices, branch indices, and shadow test buffers [17]. In contrast, TI-HAIT reduces CPU memory by 3.3× and 4.1× for bounce orders 1 and 2, respectively, owing to the compact AZB matrices from MTNP.

To compare TI-HAIT with a pre-HAIT AZB-based IT ray-tracer, we conducted simulations using the GTD module of newFASANT, which efficiently employs conventional AZB with the SVP and A^* heuristic search [28], [29], without GPU acceleration. However, simulation results for three-bounce cases—which exceeds the supported limit of TI-HAIT without MTNP—(with up to two transmissions, the maximum supported) could also not be obtained due to long computation times. The performance advantages of TI-HAIT over existing AZB approaches arise from both algorithmic and hardware-level enhancements. Algorithmically, TI-HAIT combines two AZB-based visibility tree approaches: the source-dependent tree from conventional HAIT, which reduces tree size but requires reconstruction when the Tx location changes, and the geometry-only-dependent tree using AZB matrices [28], which avoids reconstruction but can grow large in complex scenarios. These are integrated through the proposed ABT method, which applies AZB matrices within the HAIT framework to accelerate the search for illuminated primitives while maintaining a compact tree structure. In addition, MTNP reduces candidate primitives within the anxel beam, further enhancing efficiency. On the hardware side, unlike CPU-only approaches of newFASANT, TI-HAIT employs CPU parallelism via OpenMP for visibility tree generation and field computation, along with GPU acceleration using NVIDIA OptiX for shadow tests.

TI-HAIT is on average 1.9×, 1.8×, and 1.1× slower than without ABT at bounce orders 1–3 due to AZB matrix import and formatting overhead. However, at higher orders, ABT accelerates visibility tree generation, making TI-HAIT 2× and 2.2× faster (up to 2.6×), on average, at orders 4 and 5, respectively. ABT has little effect at lower orders but significantly accelerates visibility tree generation from order 3 onward. TI-HAIT achieves 2.1×, 3.8×, and 3.6× visibility tree generation speedups on average (up to 4.1×) at orders 3–5, indicating that ABT becomes increasingly important for total computation time as the number of FOPs in the scene decreases.

Notably, ABT reduces shadow test computation time by 13% and 15% at orders 4 and 5, respectively, demonstrating its role in both accelerating tree generation and optimizing tree size. Fig. 15 illustrates this effect: an image Tx above a blue ground facet emits an anxel beam toward a green reflecting facet with angular margins ϕ_{min}^{max}, θ_{min}^{max}. To compute the next bounce, primitives within the beam must be identified. Although the ground facet appears visible from the reflecting facet, it lies outside the beam. With AZB matrices and ABT, this is easily verified by comparing the beam's angular margins with the stored relative angles. In contrast, without ABT and AZB, the ABS method alone is used. It emits an anxel beam from the image Tx enclosing the ground facet and checks for intersection

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

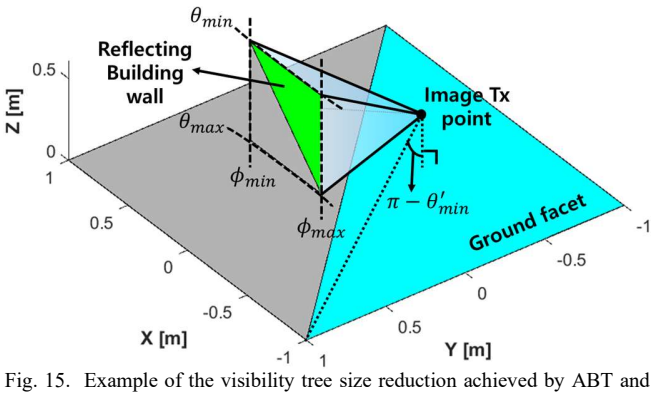


Fig. 15. Example of the visibility tree size reduction achieved by ABT and AZB matrices

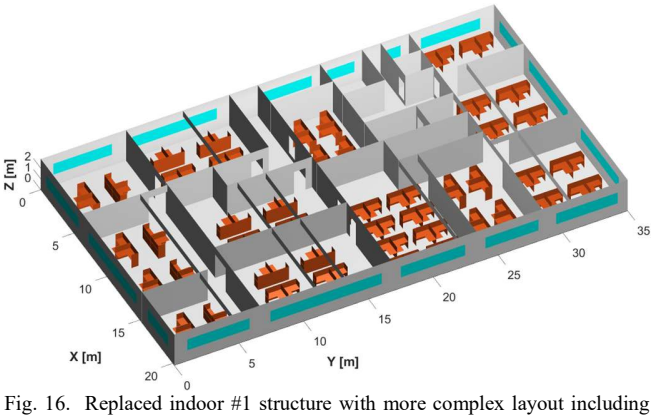


Fig. 16. Replaced indoor #1 structure with more complex layout including desks.

with the reflecting beam [17]. If $\theta'_{min} < \theta_{max}$, the method falsely identifies an intersection. This example highlights how AZB matrices with ABT prevent such false positives, enabling efficient primitive filtering, reduced tree size, and improved shadow test performance.

GPU memory usage for TI-HAIT and TI-HAIT without ABT is similar up to three bounces. From order 4 onward, ABT reduces visibility tree size (Fig. 15), lowering GPU memory usage by 2% and 8% on average at orders 4 and 5, respectively. TI-HAIT's CPU memory usage is higher at lower orders due to AZB matrix storage and formatting, averaging 3.1×, 1.7×, 1.02×, and 1.02× more at orders 1–4. However, at order 5, visibility tree reduction allows TI-HAIT to use 6% less CPU memory compared to the version without ABT.

To evaluate a more complex scenario, we replaced the Indoor #1 structure in Fig. 14 with a denser indoor layout including desks (1,108 facets) having a facet thickness of 4 cm, a dielectric constant of 1.99, and a conductivity of 0.167 [43], as shown in Fig. 16. The updated scene contains 7,592 facets in total, including the outdoor environment and three indoor structures (Indoor #1: modified as in Fig. 16; Indoor #2 and #3: as in Fig. 14(b)). E-field distribution was analyzed in the new Indoor #1 at 1 m height above indoor ground level (30 cm above desk level) using Tx #1 with up to five bounces. All other simulation settings remained the same. This scenario was tested using TI-HAIT and TI-HAIT without ABT, while TI-HAIT without MTNP was excluded due to excessive computation time. Results are summarized in Table IX. Although TI-HAIT required 1.4× more preprocessing time than without ABT, it achieved 2.4×, 3.2×, and 1.3× speedups in total simulation time, visibility tree generation, and shadowing test/field calculation, respectively. It also reduced CPU and GPU memory usage by 6% and 9% (Smaller memory usage than in Table VIII results from a more balanced load across tree partitions). As expected, the RMSPE between the two configurations was zero.

Fig. 17 presents the E-field simulation results using TI-HAIT for both the original scenario and the modified indoor #1 scenario with five bounces. To the best of the authors' knowledge, TI-HAIT is the first IT-based ray tracer capable of simulating such a large-scale environment, comprising complex indoor structures within a dense 1.1 × 1.1 km outdoor area and hundreds of FOPs. This capability is achieved through: (1) MTNP-based transmission analysis via the RTN/BD tree, (2) acceleration using ABT and modified AZB matrices, and (3) the high-performance HAIT core leveraging ABS and CPU/GPU heterogeneous computing. These advancements

TABLE IX
PERFORMANCE FOR SCENARIO WITH REPLACED INDOOR #1

Computation time & memory consumption	TI-HAIT	TI-HAIT w/o ABT
Preprocessing [sec]	304	218
Total [sec]	9,025	21,597
VTG [sec]	5,328	16,902
ST/FC [sec]	3,611	4,638
CPU [GB]	45.9	48.9
GPU[GB]	31.7	34.9

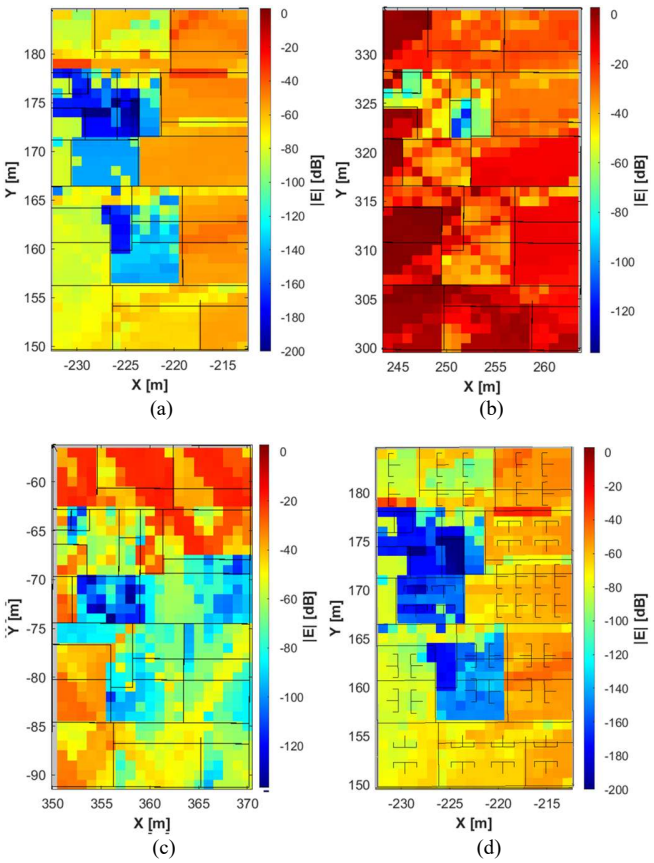


Fig. 17. E-field simulation results of TI-HAIT (Maximum five bounces) for the original and replaced Indoor #1 scenarios: (a) Indoor #1 (original), (b) Indoor #2, (c) Indoor #3, (d) Indoor #1 (replaced).

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

overcome key limitations of existing IT ray tracers.

TI-HAIT's MTNP and first-order path analysis take several seconds due to thorough electromagnetic visibility checks between scene primitives and the overhead from data import and formatting. For real-time applications, such as radar simulations where higher-order paths are less critical, TI-HAIT may be inefficient. However, the proposed MTNP/ABT enable significant acceleration in high-order ray analysis covering all possible combinations of reflection and diffraction, enabling IT-based ray tracing for accurate LOS and NLOS channel modeling in large-scale dense urban environments.

VII. Conclusion

This study proposed the TI-HAIT RT framework for large-scale O2I propagation modeling, featuring two key innovations: 1) MTNP for transmission analysis, integrated via the RTN/BD tree, and 2) ABT leveraging AZB matrices to accelerate visibility tree generation and shadow tests.

MTNP reduced computation time and memory usage by precomputing electromagnetic visibility, enabling over 100× faster simulations and 78% lower CPU memory usage in large scenarios. ABT transformed anxel beams into the original basis, enabling AZB matrix application and achieving up to 4.1× faster visibility tree generation, 1.3× faster shadowing tests, and 7% and 9% reductions in CPU and GPU memory, respectively.

The proposed TI-HAIT framework successfully simulated complex indoor structures within a dense 1.1 × 1.1 km urban environment, supporting five ray bounces, four transmissions, and hundreds of FOPs. This demonstrates its feasibility and accuracy without any systemic errors for large-scale O2I scenarios, while also being applicable to fully indoor environments.

However, due to the inherent exponential time complexity of IT-based ray tracing, TI-HAIT may still require more simulation time than SBR methods which suffer from systemic errors. Future work should explore further acceleration techniques, including algorithmic improvements and hardware-based approaches such as cluster computing and multi-GPU parallelization for visibility tree generation, shadow tests, and field calculations. Incorporating effects like multiple diffraction and diffuse scattering also presents promising directions.

References

[1] C. -X. Wang et al, "6G Wireless Channel Measurements and Models: Trends and Challenges," in *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 22-32, Dec. 2020.

[2] H. Yi et al., "Ray Tracing Meets Terahertz: Challenges and Opportunities," in *IEEE Communications Magazine*, vol. 62, no. 2, pp. 40-46, Feb. 2024.

[3] J. Huang et al, "5G Millimeter Wave Channel Sounders, Measurements, and Models: Recent Developments and Future Challenges," in *IEEE Communications Magazine*, vol. 57, no. 1, pp. 138-145, Jan. 2019.

[4] A. A. Khuwaja et al, "A Survey of Channel Modeling for UAV Communications," in *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804-2821, Fourthquarter 2018.

[5] J. Wang et al., "Wireless Channel Models for Maritime Communications," in *IEEE Access*, vol. 6, pp. 68070-68088, 2018.

[6] Y. Liu, C. -X. Wang and J. Huang, "Recent Developments and Future Challenges in Channel Measurements and Models for 5G and Beyond High-Speed Train Communication Systems," in *IEEE Communications Magazine*, vol. 57, no. 9, pp. 50-56, Sep. 2019.

[7] R. He et al., "Propagation Channels of 5G Millimeter-Wave Vehicle-to-Vehicle Communications: Recent Advances and Future Challenges," in *IEEE Vehicular Technology Magazine*, vol. 15, no. 1, pp. 16-26, Mar. 2020.

[8] Y. Chen et al., "Channel Measurement and Ray-Tracing-Statistical Hybrid Modeling for Low-Terahertz Indoor Communications," in *IEEE Trans. Wireless Commun.*, vol. 20, no. 12, 2021, pp. 8163–76.

[9] S. Priebe and T. Kürner, "Stochastic Modeling of THz Indoor Radio Channels," in *IEEE Trans. Wireless Commun.*, vol. 12, no. 9, 2013, pp. 4445–55.

[10] B. K. Jung and T. K̈urner, "Automatic Planning Algorithm of 300 GHz Backhaul Links Using Ring Topology," in *2021 15th European Conference on Antennas and Propagation (EuCAP)*, Dusseldorf, Germany, 2021, pp. 1-5.

[11] X. -Y. Wang et al, "Elevation angle research in three-dimension channel model using ray-tracing," in *2014 XXXIth URSI General Assembly and Scientific Symposium (URSI GASS)*, Beijing, China, 2014, pp. 1-4

[12] Lai, Z, et al, "Intelligent ray launching algorithm for indoor scenarios," in *Radioengineering*, 20(2), 2011, pp.398-408.

[13] 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.

[14] M. M. Taygur and T. F. Eibert, "Determination of Exact Ray Paths by Bidirectional Ray-Tracing," *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*, Aug. 2020, pp. 1-4.

[15] 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.

[16] S. Kasdorf et al, "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.

[17] Y. Kim et al, "Anxel Beam Shrinkage Method and Heterogeneous Computing-Accelerated Full-Image Theory Method Ray Tracing Enabling Massive Outdoor Propagation Modeling," in *IEEE Trans. Antennas Propag*, vol. 72, no. 7, pp. 5935-5949, Jul. 2024.

[18] Z. Yun and M. F. Iskander, "Ray Tracing for Radio Propagation Modeling: Principles and Applications," in *IEEE Access*, vol. 3, pp. 1089-1100, 2015.

[19] F. Saez de Adana et al, "Propagation model based on ray tracing for the design of personal communication systems in indoor environments," in *IEEE Transactions on Vehicular Technology*, vol. 49, no. 6, pp. 2105-2112, Nov. 2000.

[20] M. F. Catedra et al, "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.

[21] 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.

[22] 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.

[23] 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.

[24] R. Hoppe et al, "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.

[25] 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.

[26] 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.

[27] A. Ukil, V. H. Shah and B. Deck, "Fast computation of arctangent functions for embedded applications: A comparative analysis," in *2011*

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Antennas and Propagation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAP.2025.3595956

IEEE International Symposium on Industrial Electronics, Gdansk, Poland, 2011, pp. 1206-1211

[28] F. Catedra et al, "Efficient techniques for accelerating the ray-tracing for computing the multiple bounce scattering of complex bodies modeled by flat facets," in *Appl. Comput. Electromagn. Soc. J.*, vol. 25, no. 5, pp. 395–409, May 2010.

[29] L. Lozano et al, "Efficient combination of acceleration techniques applied to high frequency methods for solving radiation and scattering problems," in *Comput. Phys. Commun.*, vol. 221, pp. 28–41, Dec. 2017.

[30] M. Zhu et al, "Toward Real-Time Digital Twins of EM Environments: Computational Benchmark for Ray Launching Software," in *IEEE Open Journal of the Communications Society*, vol. 5, pp. 6291-6302, Sep. 2024.

[31] (2025). NVIDIA OptiX 8.1-programming Guide. Accessed: Apr. 28, 2025. [Online]. Available: https://raytracing-docs.nvidia.com/optix8/guide/index.html#preface#

[32] S. H. Friedberg, A. J. Insel, and L. E. Spence, *Linear Algebra*, 5th ed., London, England, U.K.: Pearson, 2018.

[33] J. Lyu et al, "An Improved Triangular Facets based Angular Z-Buffer Algorithm for IM Ray Tracing Channel Modeling," in *Proc. IEEE/CIC ICCC'22*, 2022, pp. 1050-1056.

[34] C. Saeidi and F. Hodjatkashani, "Modified Angular Z-Buffer as an Acceleration Technique for Ray Tracing," in *IEEE Trans. Antennas Propag*., vol. 58, no. 5, pp. 1822-1825, May. 2010.

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

[36] 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.

[37] 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.

[38] Y. Kim, H. Yang and J. Oh, "Critical Angle Formulation of Nonuniform Plane Waves for Determining Correct Refraction Angles at Planar Interface," in *IEEE Trans. Antennas Propag*., vol. 71, no. 3, pp. 2861-2866, Mar. 2023.

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

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

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

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

[43] ITU-R, "Effects of building materials and structures on radiowave propagation above about 100 MHz," *Reccommendation ITU-R P.2040-3*, Aug. 2023

Yongwan Kim received a B.S. degree in electrical engineering from Chungnam National University, Korea, in 2019. He is currently pursuing a Ph.D. degree with Seoul National University, Korea. His current research interests include ray-tracing EM analysis technique, EM theory, parallel computing, heterogeneous computing, wireless communication system, and EMI/EMC.

Hooyoung Kim received the B.S. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2024. He is currently pursuing an integrated master's and Ph.D. degree in the Department of Electrical and Computer Engineering at Seoul National University, Seoul, Korea. His current research interests include ray-tracing EM analysis technique for 5G communication.

Jonghyup Lee received his B.S. degree in 2017 and his M.S. degree in 2019 from Ajou University, Korea. He is currently a researcher at HD Korea Shipbuilding and Offshore Engineering Co., Ltd., Republic of Korea. His current research interests include electromagnetic interference (EMI) and electromagnetic compatibility (EMC) in maritime systems, Navy systems, and construction equipment systems.

Hyunho Cho received his B.S. degree in electrical engineering from Korea Military Academy, in 2016 and M.S. degree in electrical and computer engineering from University of Florida, in 2021. From 2016 to 2022, he served as an information communication officer in the Korean Army. He is currently working as a researcher at Korea Shipbuilding Offshore Engineering(KSOE). His research areas include electromagnetic interference compatibility (EMC) and remote monitoring services for vessel commissioning with communication Tech.

Jungsuek Oh (S'08) received his B.S. and M.S. degrees from Seoul National University, Korea, in 2002 and 2007, respectively, and a Ph.D. degree from the University of Michigan at Ann Arbor in 2012. From 2007 to 2008, he was with Korea Telecom as a hardware research engineer, working on the development of flexible RF devices. In 2012, he was a postdoctoral research fellow in the Radiation Laboratory at the University of Michigan. From 2013 to 2014, he was a staff RF engineer with Samsung Research America, Dallas, working as a project leader for the 5G/millimeter-wave antenna system. From 2015 to 2018, he was a faculty member in the Department of Electronic Engineering at Inha University in South Korea. He is currently an Associate Professor in the School of Electrical and Computer Engineering, Seoul National University, South Korea. His research areas include mmWave beam focusing/shaping techniques, antenna miniaturization for integrated systems, and radio propagation modeling for indoor scenarios. He is the recipient of the 2011 Rackham Predoctoral Fellowship Award at the University of Michigan. He has published more than 50 technical journal and conference papers, and has served as a technical reviewer for the IEEE Transactions on Antennas and Propagation, IEEE Antenna and Wireless Propagation Letters, and so on. He has served as a TPC member and as a session chair for the IEEE AP-S/USNC-URSI and ISAP. He has been a senior member of IEEE since 2017.

Authorized licensed use limited to: Seoul National University. Downloaded on August 14,2025 at 09:52:53 UTC from IEEE Xplore. Restrictions apply.
© 2025 IEEE. All rights reserved, including rights for text and data mining and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.