BVH-Accelerated Ray Tracing for High-Frequency Electromagnetic Backscattering

Marco Pasquale, Andong Hu, Luca Pennati, Ivy Peng, and Stefano Markidis

KTH Royal Institute of Technology, Stockholm, Sweden
{marcopas,andonghu,pennati,bopeng,markidis}@kth.se

Abstract. As computational complexity in electromagnetics increases with frequency, full-wave solvers become computationally infeasible for electrically large problems. A shooting and bouncing rays (SBR) method for modeling electromagnetic backscattering from large metallic objects is presented, coupling multi-reflection geometrical optics with a physical optics discretized surface integral. To handle the massive intersection search space, we implement a bounding volume hierarchy (BVH) in a trace–integrate pipeline. Numerical accuracy is controlled through an incident-ray sampling rule to mitigate spatial aliasing. The method is validated against Mie solutions for a perfectly electrically conducting sphere, and applied to a complex aircraft for monostatic radar cross-section prediction, exhibiting robust parallel scaling on NVIDIA and AMD GPUs.

Keywords: Shooting and bouncing rays · Bounding volume hierarchy · Electromagnetic scattering · Radar cross section · GPU acceleration

1 Introduction

Computational electromagnetics (CEM) comprises numerical and asymptotic methods for predicting electromagnetic fields across applications such as antenna design, radar, and channel modeling [1, 2]. Full-wave methods are preferred when resonance or strong coupling dominate. However, when a structure is electrically large, meaning its characteristic dimension D spans many wavelengths λ ($D/\lambda \gg 1$), full-wave discretizations require exceptionally fine meshes that push memory and runtime beyond practical limits. This scaling pressure is increasingly prominent in emerging radar and 6G scenarios extending toward sub-terahertz frequencies [3]. In the high-frequency (HF) asymptotic limit (often mathematically denoted as $\lambda \to 0$), wave propagation behaves locally like plane waves and can be accurately approximated by ray optics. This motivates asymptotic methods that replace global field solves with a transport-and-accumulation procedure: tracing ensembles of rays through the geometry and coherently accumulating the scattered field produced at the surface [4]. Shooting and bouncing rays (SBR) is a representative Geometrical Optics plus Physical Optics (GO+PO) method introduced to enable radar cross-section (RCS) predictions for complex geometries [5, 6].

While recent work has leveraged GPU-accelerated ray tracing for channel modeling [7, 8], the application of these techniques to high-fidelity, multi-bounce RCS prediction requires careful treatment of the PO accumulation step to avoid phase aliasing. The primary novelty and contribution of this work is the development of a computationally efficient SBR algorithm tailored explicitly for CEM precision. Specifically,

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30

we: (i) formulate a ray-tube discretization driven by a strict sampling rule to guarantee phase accuracy; (ii) map the SBR procedure to a parallel trace–integrate pipeline that minimizes GPU thread divergence; and (iii) provide a comprehensive performance and scaling analysis across differing architectures (NVIDIA A100 vs. AMD MI250X), highlighting trade-offs between single (`FP32`) and double (`FP64`) precision arithmetic. The method is implemented in the open-source code `SagittaSBR` [9].

2 Problem Formulation

To bypass the memory constraints of global field solvers, we employ the SBR method. In the following equations, ε and μ are the permittivity and permeability of the medium, and the subscript "r" indicates the relative values compared to a vacuum. Time-harmonic fields in the high-frequency limit admit an eikonal formulation [10]:

$$\mathbf{E}(\mathbf{r}) = \mathbf{e}(\mathbf{r})e^{-jk_0L(\mathbf{r})}, \qquad \mathbf{H}(\mathbf{r}) = \mathbf{h}(\mathbf{r})e^{-jk_0L(\mathbf{r})},$$

where $L(\mathbf{r})$ is the optical path length and $k_0 = 2\pi/\lambda_0$ is the free-space wavenumber. The phase is $\Phi(\mathbf{r}) = k_0L(\mathbf{r})$. Substituting this ansatz into Maxwell's equations and extracting the leading order terms under the HF assumption yields the eikonal equation:

$$|\nabla L(\mathbf{r})|^2 = n^2(\mathbf{r}), \qquad n(\mathbf{r}) = \sqrt{\mu_r(\mathbf{r})\,\varepsilon_r(\mathbf{r})}.$$

Rays are curves $\mathbf{r}(s)$ normal to the wavefronts. In homogeneous media ($\nabla n = 0$), these rays propagate as straight lines and undergo specular reflection at boundaries. Geometrical optics provides these trajectories, but the scattered field is obtained by accumulating contributions over the illuminated surface using physical optics (PO). For a perfect electric conductor (PEC), the induced surface current is approximated as:

$$\mathbf{J}_s \approx \begin{cases} 2\,\hat{n} \times \mathbf{H}_{\text{inc}}, & \text{illuminated region,} \\ 0, & \text{shadowed region,} \end{cases}$$

where $\hat{n}$ is the unit outward surface normal. In the far-field, the scattered electric field follows the Stratton–Chu formulation [11]:

$$\mathbf{E}_s(\mathbf{r}) \approx \frac{-j\omega\mu_0}{4\pi r} e^{-jk_0r} \int_S \left[\mathbf{J}_s(\mathbf{r}') - \left(\mathbf{J}_s(\mathbf{r}') \cdot \hat{r}\right)\hat{r}\right] e^{jk_0\mathbf{r}'\cdot\hat{r}}\, dS'.$$

From a computational perspective, this continuous integral is discretized using ray tubes: each ray represents an area element ΔA on the launch grid. It should be noted that it is not strictly necessary for the mesh triangles to be larger than the ray tube cross-section ΔA; ray tubes serve as discrete sample points of the surface fields, and as long as the surface is adequately sampled, the validity of the discrete integral holds. For the monostatic case, the total far-field amplitude is the coherent sum over the valid rays:

$$A = \sum_{i=1}^{N_{\text{rays}}} \frac{jk\Delta A}{4\pi}\, 2\,(\hat{n}_i \cdot -\hat{k}_{\text{inc}})\, \Gamma^{N_i}\, e^{-j2kR_i}, \tag{1}$$

where $\hat{n}_i$ is the surface normal at the first interaction, $\hat{k}_{\text{inc}}$ is the incident direction, R_i is the accumulated optical path length including reflections, N_i is the number of reflections, and Γ is the reflection coefficient. The monostatic RCS is $\sigma = 4\pi|A|^2$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30

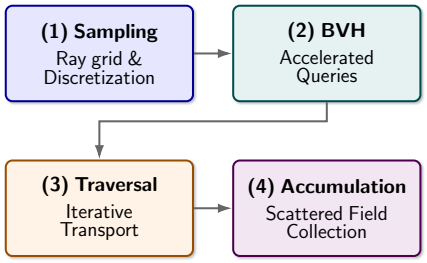


Fig. 1: `SagittaSBR` pipeline. Rays are launched from an orthographic grid. BVH accelerates traversal, and scattered field is accumulated from valid hits.

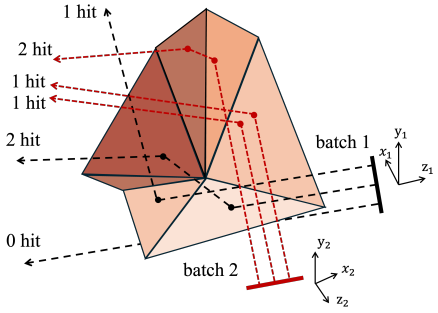


Fig. 2: Independent multi-bounce transport. At the final valid hit, path length, first-hit normal, and reflection count are recorded for the PO integral.

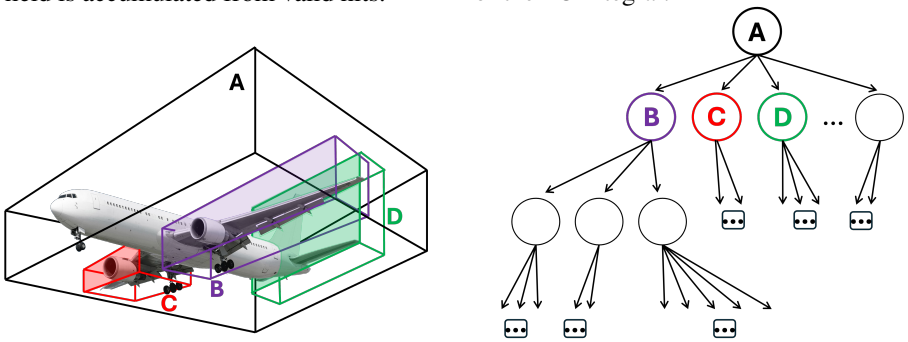


Fig. 3: Illustration of a BVH built with the binned SAH method. Bounding volumes are subdivided recursively, partitioning geometry into increasingly tight AABBs that reduce the number of intersection tests during traversal, accelerating the SBR pipeline.

3 Methodology and Architecture

The SBR evaluation is organized into a *trace-integrate* pipeline (Fig. 1). For each incident direction, we approximate the incident field using an orthographic bundle of rays originating from a virtual aperture spanning the target's bounding box.

Ray Grid Resolution. The PO integral (Eq. 1) is evaluated as a discrete sum. If the ray spacing Δs is too large, the sum undersamples the phase variation across the surface, producing spatial aliasing that manifests as nonphysical oscillations in the RCS pattern. To enforce numerical stability and act as a Nyquist-like criterion for the electromagnetic phase, we enforce a strict sampling rule: $\Delta s \leq \lambda/5$. This dictates that as frequency increases, computational workload scales *quadratically*.

BVH Acceleration. In complex geometries, an exhaustive ray-triangle intersection search is computationally prohibitive. We mitigate this by constructing a Bounding Volume Hierarchy (BVH) over the mesh. While a BVH does not reduce the intrinsic arithmetic cost of a ray-triangle intersection test, it drastically reduces the *search*

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30

Algorithm 1
BVH construction (selectable split)

Require: Triangles T, threshold N_{leaf}, split rule (MEDIAN/SAH)
Ensure: Linearized arrays `nodes`, `tris`
Compute centroids and global AABB; initialize root with T
BUILD(root)
Linearize nodes (preorder); store triangle ranges
function BUILD(n)
  if $|T(n)| \leq N_{\text{leaf}}$ **then**
    Mark n as leaf; **return**
  end if
  Choose split axis/position; partition $T \to T_L, T_R$
  Create children L, R with AABBs
  Parallel: BUILD(L), BUILD(R)
end function

Algorithm 2
Multi-reflection SBR traversal

Require: BVH `nodes`, `tris`, ray $(\mathbf{o}, \hat{d})$, max bounces $B_{\max}$
Ensure: Valid flag, $\hat{n}_0$, path length R, bounces N
$R \leftarrow 0$, $N \leftarrow 0$; `valid` $\leftarrow$ **false**
for $b = 1$ to $B_{\max}$ **do**
  $(\texttt{hit}, t, \hat{n}) \leftarrow$ INTERSECT($\mathbf{o}, \hat{d}$, `nodes`, `tris`)
  if $\neg$`hit` **then**
    break
  end if
  $\mathbf{x} \leftarrow \mathbf{o} + t\hat{d}$
  $R \leftarrow R + t$, $N \leftarrow N + 1$
  if $N = 1$ **then**
    $\hat{n}_0 \leftarrow \hat{n}$; `valid` $\leftarrow$ **true**
  end if
  $\hat{d} \leftarrow \hat{d} - 2(\hat{d} \cdot \hat{n})\hat{n}$
  $\mathbf{o} \leftarrow \mathbf{x} + \varepsilon\hat{n}$
end for
return (`valid`, $\hat{n}_0$, R, N)

space by choosing large portions of the geometry using Axis-Aligned Bounding Boxes (AABBs). To optimize tree traversal, we implement two partitioning options: a fast *median split*, which halves the volume along its longest axis, and a binned *Surface Area Heuristic* (SAH). The SAH estimates the expected cost of splitting a parent node P into children L and R. In Eq. 2, $SA(\cdot)$ denotes the surface area of the bounding box, N_L and N_R are the triangle counts in the respective children, and C_T and C_I are the constant computational costs of traversing a node and intersecting a triangle. Algorithm 1 outlines this recursive construction process represented in Fig. 3.

$$C_{\text{SAH}}(P \to L, R) = C_T + C_I \left(\frac{SA(L)}{SA(P)} N_L + \frac{SA(R)}{SA(P)} N_R \right) \tag{2}$$

3.1 GPU Optimizations and Pipeline

Adapting the SBR pipeline for high-throughput GPU execution requires carefully balancing memory locality and warp divergence, diverging from standard graphics ray tracing. We address these computational bottlenecks through different design choices.

Trace-Integrate Decomposition. To mitigate thread divergence, the workload is decoupled into two GPU kernels. The *trace* kernel (Algorithm 2, Fig. 2) iteratively traverses the BVH, processing up to $B_{\max}$ reflections per ray. To prevent self-intersection from floating-point inaccuracies, secondary ray origins are shifted by a normal-aligned ε-offset [12]. Instead of accumulating the physical optics integral in-flight, which would severely unbalance warp execution times, the trace kernel writes a state record (first-hit normal, path length, bounce count, and validity flag). Rays failing to return to the

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30

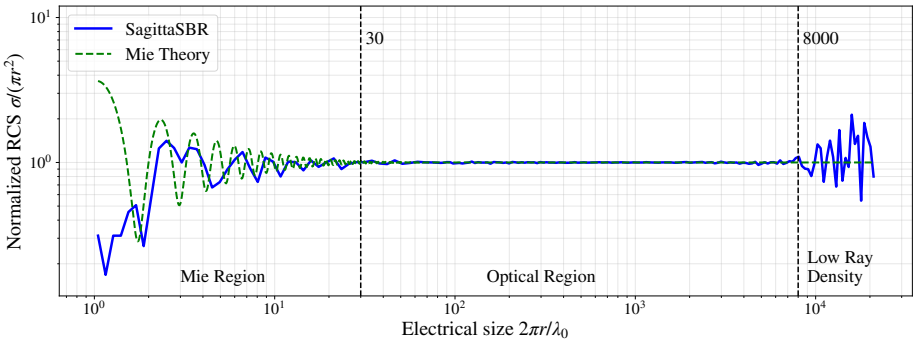


Fig. 4: Frequency scan of monostatic RCS for a triangulated PEC sphere. Here, the ray grid was held constant. The method achieves excellent agreement in the optical region until the shrinking wavelength violates the spatial sampling criterion, causing aliasing.

monostatic receiver are marked invalid. Next, the *integrate* kernel performs the parallel reduction strictly over valid records. This separates the divergent memory-gather operations of ray transport from the arithmetic-heavy PO summation.

State Management and Occupancy. Although orthographic rays initially provide excellent spatial locality and L1 cache utilization, post-reflection rays rapidly diverge. To robustly handle deep, multi-bounce BVH traversal without recursion, we utilize a carefully sized explicit memory stack. By keeping this local state bounded, we minimize per-thread register pressure, which prevents register spilling and maximizes active warps per Streaming Multiprocessor (SM).

MPI Distribution. Because a monostatic angular sweep is intrinsically decoupled, we distribute the incident angles across MPI ranks. This coarse-grained parallelization allows each GPU to autonomously compute a subset of the full spherical sweep, reducing communication to zero during the trace-integrate execution.

4 Numerical Results and Performance

4.1 Validation with PEC Sphere

To validate the physical accuracy of the solver, we computed the RCS of a PEC sphere against analytical Mie scattering theory (Fig. 4). Electrical size is defined as $kr = 2\pi r/\lambda$. At $kr < 30$ (the Mie resonance region), ray-based GO+PO approximations are fundamentally inapplicable. However, entering the optical region ($kr \geq 30$), our method accurately captures the scattering physics with a standard deviation of $\sim 2.5\%$. It is important to clarify the breakdown at extreme frequencies ($kr \approx 8{,}000$). To isolate the effect of frequency across this sweep, the simulation utilized a *constant* launch grid of $22,000 \times 22,000$ rays. Consequently, as the frequency increased (and λ decreased), the fixed grid eventually violated the $\Delta s \leq \lambda/5$ criterion. The undersampling of the wavelength causes the aliasing seen at the far right of the plot.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30

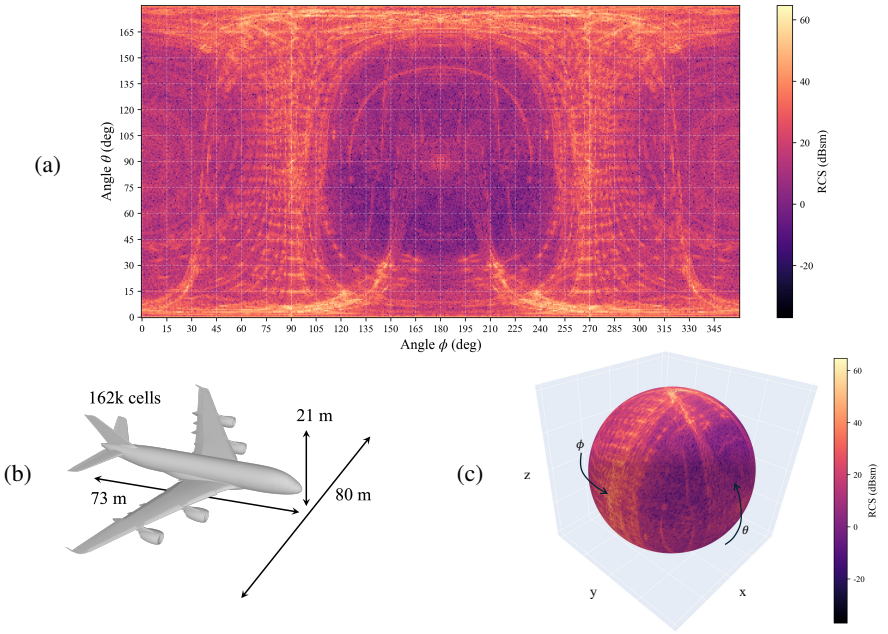


Fig. 5: RCS calculation and geometry for an A380 aircraft. (a) Full angular RCS scan (500 samples in ϕ, 250 in θ at 10 GHz) generated via a $30{,}000 \times 30{,}000$ ray grid. (b) Simulated aircraft model ($80 \times 73 \times 21$ m, 162k triangular elements). (c) Visualization of the RCS mapped onto a bounding spherical surface.

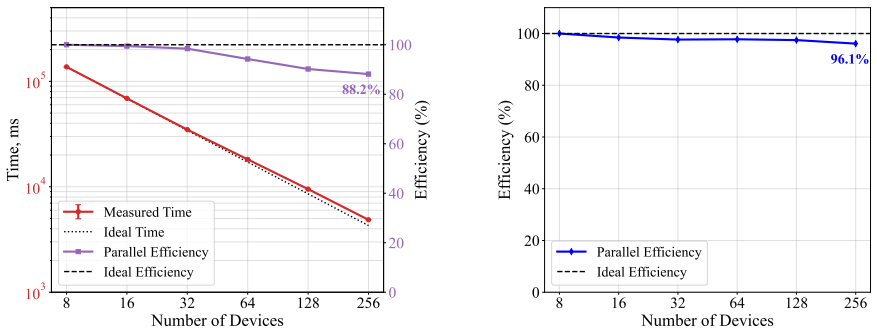
4.2 Radar Cross Section of Complex Geometries

To evaluate solver performance on electrically large, complex geometries, an A380 aircraft ($80 \times 73 \times 21$ m) was modeled as a PEC object at 10 GHz ($\lambda \approx 3$ cm). Ensuring the $\lambda/5$ criterion required a dense $30{,}000 \times 30{,}000$ launch grid. The full spherical monostatic scan ($125{,}000$ observation points, max 100 reflections per ray) processed 1.12×10^{14} rays in $\sim$27 minutes using 8 nodes of the LUMI supercomputer (32 AMD MI250X GPUs). The resulting heatmap (Fig. 5) clearly identifies high-intensity broadside specular reflections from the fuselage and wings.

4.3 Performance, Precision, and Scaling Analysis

The ray generation and intersection kernel dominates the computational workload, accounting for the vast majority of execution time relative to the PO integral reduction. Depending on the incident angle relative to the object's profile, the percentage of rays that actually intersect the target can be as low as 5-10%, meaning standard SBR performs significant intersection searches on invalid rays. This raises an interest for potential early filtering strategies, which could speed-up the process, but would require a simplified hit-detection mechanism, potentially disturbing final accuracy. Table 1 compares this kernel performance across NVIDIA A100 and AMD MI250X architectures.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30



(a) Strong Scaling (b) Weak Scaling

Fig. 6: Scaling performance on LUMI using `FP32`. The x-axis shows the total number of GCDs utilized on AMD MI250X GPUs. Strong scaling maintains an efficiency of $\sim 88\%$ at 256 GCDs, while weak scaling retains $\sim 96\%$ efficiency.

Table 1: Kernel-time breakdown for A380 scattering simulations ($18{,}000 \times 18{,}000$ rays) using one MPI process. Statistics evaluated over 10 independent trials.

Device	Kernel	time `FP32`, ms	time `FP64`, ms	Δ-time
NVIDIA A100 (Full)	Ray Launch	98.96 ± 1.18	151.30 ± 1.75	$+52.9\%$
	PO Integral	6.43 ± 0.09	6.01 ± 0.06	-7.0%
AMD MI250X (1 GCD)	Ray Launch	249.81 ± 0.52	198.70 ± 0.31	-20.5%
	PO Integral	4.83 ± 0.01	7.41 ± 0.01	$+53.4\%$

Unlike `FP64`, in `FP32` a slight accumulation error ($\sim 3\%$ deviation) was observed after 100 reflections due to the microscopic phase terms (kR_i). While the NVIDIA A100 shows the expected massive penalty ($\sim$53% slower) when switching to `FP64`, the AMD MI250X executed the ray launch kernel $\sim$20% *faster* in `FP64`. This indicates that optimal precision choices in modern SBR solvers must be strictly architecture-aware. Parallel scalability was assessed on the LUMI supercomputer by distributing angular sweeps across MPI ranks (Fig. 6). Strong scaling maintained an excellent efficiency of $\sim$88% at 256 GCDs. Weak scaling, where the base angular workload was scaled proportionally with the number of nodes, retained $\sim$96% efficiency, proving that MPI communication overhead is negligible compared to multi-bounce ray tracing computations.

5 Discussion and Conclusion

We presented an efficient SBR method for evaluating the monostatic RCS of electrically large objects, implemented in the open-source `SagittaSBR`. By enforcing an incident-ray sampling rule, we mitigated the phase aliasing common in discretized PO integrals. The method demonstrated robust scalability across heterogeneous GPU architectures. Future extensions will focus on adding diffraction corrections (e.g., GTD/UTD/PTD) [13, 14] to improve fidelity. These extensions introduce additional interaction

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30

pathways (e.g., ray spawning) and increase variability in per-ray work, requiring dynamic scheduling and careful memory management. Moreover, because hit-misses constitute a major portion of the workload, future work will explore early filtering heuristics (e.g., rasterizing a bounding silhouette before launch) and directional BVH optimizations, such as angle-specific subtrees, to reduce unnecessary intersection queries.

Acknowledgments. This work has received funding from the Swedish Research Council's Research Environment grant (SEE-6GIA 2024-06482).

References

[1] M. M. Taygur, I. O. Sukharevsky, and T. F. Eibert. "Computation of Antenna Transfer Functions with a Bidirectional Ray-Tracing Algorithm Utilizing Antenna Reciprocity". 2018. DOI: 10.23919/URSI-AT-RASC.2018.8471651.

[2] Z. Yun and M. F. Iskander. "Ray Tracing for Radio Propagation Modeling: Principles and Applications" 2015. DOI: 10.1109/ACCESS.2015.2453991.

[3] P. Li, J. Fan, and J. Wu. "Exploring the key technologies and applications of 6G wireless communication network" 2025. DOI: 10.1016/j.isci.2025.112281.

[4] R. Mittra, ed. *Computational Electromagnetics: Recent Advances and Engineering Applications*. 2014. DOI: 10.1007/978-1-4614-4382-7.

[5] H. Ling, R. Chou, and S. Lee. "Shooting and bouncing rays: Calculating RCS of an arbitrary cavity". 1986. DOI: 10.1109/APS.1986.1149823.

[6] S. Sefi. "Computational Electromagnetics: Software Development and High Frequency Modelling of Surface Currents on Perfect Conductors". DiVA portal. PhD thesis. KTH Royal Institute of Technology, 2005.

[7] M. Schiller et al. "GPU accelerated ray launching for high-fidelity virtual test drives of VANET applications". 2015. DOI: 10.1109/HPCSim.2015.7237048.

[8] J. Gómez et al. "Accelerated Ray Launching Method for Efficient Field Coverage Studies in Wide Urban Areas" 2023. DOI: 10.3390/s23146412.

[9] M. Pasquale. *SagittaSBR*. https://github.com/marco-pas/SagittaSBR. 2026.

[10] C. A. Balanis. *Balanis' Advanced Engineering Electromagnetics*. 1st ed. 2023. DOI: 10.1002/9781394180042.

[11] J. A. Stratton. *Electromagnetic Theory*. 1941. DOI: 10.1002/9781119134640.

[12] S. J. Kwon, J. Im, and H. G. Joo. "Resolution of Self-Intersection Issue in Monte Carlo Simulations Employing Graphics Ray Tracing Technology" 2023. Korean Nuclear Society.

[13] R. Kouyoumjian and P. Pathak. "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface" 1974. DOI: 10.1109/PROC.1974.9651.

[14] P. Y. Ufimtsev. *Fundamentals of the Physical Theory of Diffraction*. 2007. DOI: 10.1002/0470109017.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_30