

基于 GPU 并行技术的超大型海面舰船电磁 散射仿真



郑文军^{1*}, 杨 伟², 周礼来²

(1. 中国西南电子技术研究所 成都 610036; 2. 电子科技大学电子科学与工程学院 成都 611731)

【摘要】为了解决超电大尺寸海面舰船场景中电磁散射计算的瓶颈问题, 研究基于多图像处理单元 (Multi-GPU) 并行加速技术的弹跳射线法 (SBR)。借助统一设备计算架构 (CUDA) 提供的多线程服务 (MPS), 构建 Multi-GPU 并行加速框架, 研究基于区域射线束划分 GPU 计算任务和实现方式; 研究基于矩阵网格的任务分割技术, 最大限度提高 GPU 全局内存利用率; 针对不同运算单元间的差异所带来的计算不同步问题, 设计基于动态负载均衡算法的调度系统, 进而提高计算资源利用率。仿真结果表明, 在双 GPU 硬件平台上, 该方案与现有并行技术算法相比, 在确保结果准确性的情况下加速比接近甚至超过 200%。因此, 该技术方案能够有效解决超电大海面舰船电磁散射问题。

关键词 超电大; Multi-GPU; 雷达散射截面; 海面舰船; 射线追踪法

中图分类号 TP399 文献标志码 A doi:10.12178/1001-0548.2022335

Electromagnetic Scattering Simulation of Extremely Electrically Large Sea-Ship Scene Based on GPU Parallel Technology

ZHENG Wenjun^{1*}, YANG Wei², and ZHOU Lilai²

(1. Southwest China Institute of Electronic Technology Chengdu 610036;

2. School of Electronic Science and Engineering, University of Electronic Science and Technology of China Chengdu 611731)

Abstract In order to solve the computation bottleneck of electromagnetic scattering in the extremely electrically large sea-ship scene, this paper studies the implementation technology of the shooting and bouncing rays (SBR) based on the parallel acceleration technology of multiple graphs processing units (Multi-GPU). This method utilizes the multi-process service (MPS) of compute unified device architecture (CUDA) to build the framework of Multi-GPU acceleration technique. The task division method is based on radiation beams in the projected grid region. In addition, the task division technology based on matrix grid is studied to improve maximally GPU global memory utilization. Finally, dynamic load balancing algorithm based scheduling system, which can enhance the usage of the computational resources and solve the asynchronization of computation caused by differences between GPUs has been designed. Some simulations show that compared with existing SBR method based on parallel computation technique, the speedup rate of proposed approach reach 200% with certain accuracy. Therefore, this method can effectively solve the electromagnetic scattering problem of extremely electrically large sea-ship scene.

Key words extremely electrically large size; Multi-GPU; radar cross section; sea-ship; shooting and bouncing rays

在实际雷达工程中, 雷达照射海面舰船目标通常为超电大尺寸规模, 超电大尺寸目标电磁散射计算始终是电磁学领域的重点及难点^[1-2]。基于射线追踪法的高频近似方法为解决此类问题最切实可行的手段。然而随着实际工程需求的日益严苛, 待求解

目标的尺寸也在逐步增大, 此时串行高频近似方法的计算效率已经难以满足要求。

因此, 研究人员在如何提高针对超电大目标散射的高频方法加速技术方面开展了相关工作^[3]。文献 [4] 等利用现场可编程门阵列 (FPGA) 加速射线

收稿日期: 2022-10-08; 修回日期: 2022-12-02

作者简介: 郑文军 (1981-), 男, 博士生, 主要从事雷达成像算法方面的研究。

*通信作者: 郑文军, E-mail: janal_zheng@163.com

相交运算,提高了射线追踪效率。文献 [5] 利用图形处理器 (graphics processing unit, GPU) 加速弹跳射线法 (shooting and bouncing rays, SBR), 对各类复杂电大尺寸目标的 RCS 进行分析。文献 [6-8] 基于 Spark、MPI 等并行技术研究计算集群在电磁算法中的应用。

目前,海面舰船雷达问题逐步向更大场景、更复杂环境方向发展,这势必需要更加强大的算力作为支撑。为了进一步提高并行计算效率,本文针对实际雷达场景遇到的超电大尺寸场景,研究了基于多图形处理器 (Multi-GPU) 并行加速的弹跳射线法框架,提出了基于区域射线划分 GPU 射线追踪任务的策略。

1 基于区域射线管的分配策略

对于海面舰船目标电磁散射特性而言,除了一次散射占较大比重外,舰船的舰面布置复杂,桅杆、开口结构等间相互作用,是改变舰船雷达波多次散射特征复杂的重要原因。因此,需要采用射线追踪法计算多次散射贡献。在射线追踪过程中,需要记录射线的轨迹,计算射线在分界面处产生的反射场信息。基于物理光学法 (physical optics, PO) 理论,分界面处等效电磁流可以由射线的入射场、反射场共同确定,进而再通过 Stratton-Chu 公式求得散射场。因此,该高频方法被标记为“PO-SBR”法^[9]。

由于海面舰船的超电大规模,其电磁射线管数量巨大,需要分批次处理。基于区域射线束的任务分配策略首先应确定计算区域。由于目标处于远场区,射线追踪范围即为目标在垂直于入射方向上的射线管平面投影。建立以 $\mathbf{o} = (r \sin \theta \cos \varphi, r \sin \theta \sin \varphi, r \cos \theta)$ 为原点,以方向矢量 $\mathbf{u} = (-\cos \theta \cos \varphi, -\cos \theta \sin \varphi, \sin \theta)$, $\mathbf{v} = (-\sin \varphi, \cos \varphi, 0)$, $\mathbf{w} = (-\sin \theta \cos \varphi, -\sin \theta \sin \varphi, \cos \theta)$ 为坐标轴的局部坐标系, θ , φ 分别为入射俯仰角与方位角。则空间中任意点 \mathbf{r} 在射线束平面 uov 的投影可确定为:

$$\mathbf{r}' = \mathbf{r} + (\mathbf{r} \cdot \mathbf{w}) \mathbf{w} + \mathbf{o} \quad (1)$$

式中, \mathbf{r}' 为 \mathbf{r} 投影于 uov 平面的位置矢量。为简化投影计算,构建以目标在各坐标轴上最大坐标及最小坐标为基准构建的轴对齐包围盒,由包围盒 8 个顶点在 uov 的投影可得到射线追踪区域。

基于区域射线束分割策略,对分割后的射线进行编号,则第 i ($i = 0, 1, 2, \dots$) 个射线中心的局部坐标为:

$$u_i = \left(\text{mod} \left(i, \left\lceil \frac{u_{\max} - u_{\min}}{\Delta} \right\rceil \right) + \frac{1}{2} \right) \Delta + u_{\min} \quad (2)$$

$$v_i = \left(\left\lceil \frac{i}{\left\lceil \frac{v_{\max} - v_{\min}}{\Delta} \right\rceil} \right\rceil + \frac{1}{2} \right) \Delta + v_{\min} \quad (3)$$

式中, u_{\max} 、 u_{\min} 、 v_{\max} 、 v_{\min} 分别表示投影在局部坐标轴 u 、 v 处的边界值; Δ 表示网格剖分密度; $\lceil \cdot \rceil$ 表示向上取整; $\lfloor \cdot \rfloor$ 表示向下取整; $\text{mod}(\cdot)$ 表示取模运算。

由于弹跳射线法以单一射线为最小计算单位,通过对照射目标射线束的追踪 (ray tracing, RT) 实现散射计算,以追踪区域中射线集合为单位进行划分并封装为子任务。需要保证散射体在射线管平面的投影全部落在射线管平面区域中,需要根据设定的射线管密度将射线管平面剖分为大小均匀的矩阵网络,通常设定网格剖分密度 $\Delta \leq \lambda/10$, 在射线管平面上构成 $M \times N$ 个射线管,且每个矩形网格的中心为射线管发射起点,射线追踪示意图如图 1 所示。

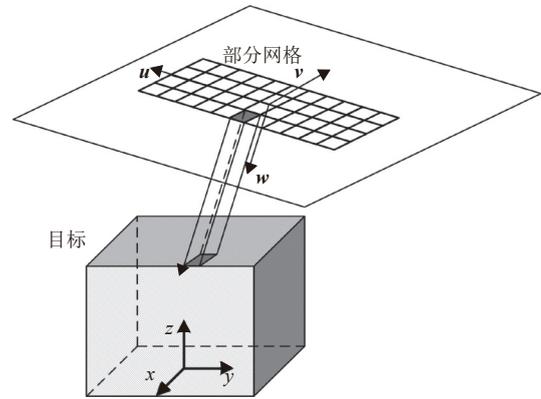


图 1 射线追踪示意图

2 基于矩形网格的任务分割

由于目标处于远场区,射线追踪范围即为目标在垂直于入射方向上的射线管平面投影。由于 GPU 中的全局内存空间有限,全局内存空间无法完整地存储所有射线管的每一级射线所包含的信息。在射线追踪的过程中,每一批次的核函数最多调用 T 个射线追踪任务。

在 $M \times N$ 个射线管构成的矩形网格中,对矩形网格进行按照如下分割规则进行划分:不失一般性,假设 $M \leq N$,交替选取未被划分的矩形网格中最长边 M_i 、最短边 N_i ,其中下标 i 表示其为第 i 次划

分的边长; 特别地, 第一次划分从长边或短边开始, 有 $M_0 = M$ 或 $N_0 = N$ 。将选取的边长作为矩形的一条边长, 在网格中截取一个极大的矩形网格分区, 其面积等于 T 。每次选取的矩形网格分区至少能够覆盖 $M_i \times 1$ 或 $N_i \times 1$ 个网格单元, 将尺寸为 $M \times N$ 的矩形网格分割成若干个矩形分区, 每一矩形分区中的所有射线管将作为一个批次的射线管任务, 如图 2 所示。第 i 次 GPU 全局内存利用率为:

$$S_i = \frac{\lfloor \frac{T}{M_i} \rfloor M_i}{T} \times 100\% \quad (4)$$

式中, $\lfloor \cdot \rfloor$ 表示向下取整。数学理论证明, 上述方法能够使得 S_i 最大, 从而使得整体内存利用率 $S = \prod S_i$ 最优, 提高算法的整体计算效率。在程序调用核函数时, 将矩形分区的其中一个包含射线管数目 T 的分区作为一个批次。每批次调用核函数之前, 将当前分区的射线管位置、长度、宽度等数据存储到全局内存中, 并在核函数调用时将此分区射线管的位置数据指针传入核函数中。在核函数中使用线程编号就可以定位每一个线程所对应的射线管的位置, 这样就完成了射线管映射到每一个线程的过程。

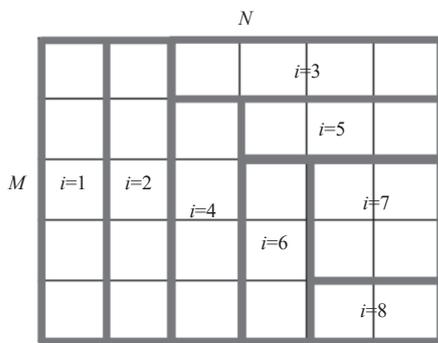


图 2 大规模射线管的任務分割方法

3 Multi-GPU 并行框架

考虑到中央处理器 (CPU) 与 GPU 并行计算能力存在极大的差距, 实际程序中繁重的射线追踪及电磁计算任务将全部交由 GPU 处理, CPU 端仅对程序进行整体控制, 无需执行高耗时操作, 因此采用 MPS 并发控制 GPU 计算^[10]。

基于 MPS 的 Multi-GPU 框架由任务调度线程及 GPU 控制线程组成。通过读取主机中挂载 GPU 设备信息进而为创建 thread 线程对象实现 Multi-GPU 控制, 各线程利用 `cudaSetDevice()` 指定对应

的设备加以控制, MPS 将为线程分配唯一的上下文 (Context), 该上下文始终指向绑定的 GPU 且包含此设备运行所需的内存、模块 (Modules)、数据流 (Streams)、本地内核 (Kernel) 缓存等信息, CUDA 根据上下文信息处理并控制对应 GPU 的运行。主程序完成线程分配后执行系统调度, 负责将划分后的任务集合中的追踪任务分配至不同设备进行处理, Multi-GPU 并行框架如图 3 所示。

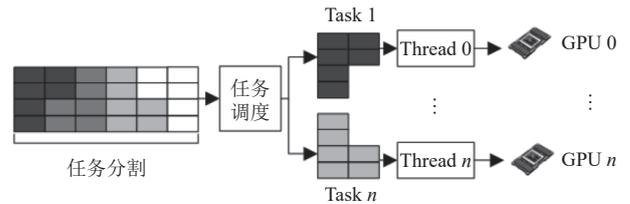


图 3 Multi-GPU 并行框架

4 基于动态负载均衡算法的调度

客观上不同 GPU 间性能存在一定差异, 如若单元间无法实现协同调配, 可能导致部分 GPU 空转闲置的现象发生, 程序性能急剧退化。现有的负载均衡算法大体可以分为动态及静态两类。静态负载均衡算法依照固化完成的任务分配策略对系统进行调度, 其运行时不依赖运算单元状态, 算法结构简单, 但无法适应环境变化所带来的负载失衡。

动态负载均衡算法随运算单元状态调整任务分配, 为保证代码健硕性, 避免线程冲突, 利用原子操作的不可分割性构建自旋锁对临界区代码进行保护。C++ 标准库中所带的原子量 (Atomic) 为算法实现提供了便利, 其中比较交换操作 (CAS) 算法函数中的 `compare_exchange_strong()` 与 `store()` 函数为多线程提供的原子操作, 保证线程安全。其中前者用于比较原子量数值是否符合预期, 后者用于为原子量赋值, 同时每个线程都将拥有 `atomic<bool> isComplete` 任务状态标志位。调度过程主要包括: 1) 调度器对各线程 `isComplete` 标志位进行轮询, 通过调用 `compare_exchange_strong()` 函数获取标志位数值, 当检测到 GPU 完成任务后, 调度器检查任务集合并重新推送新任务至线程, 并通过 `store()` 函数清除 `isComplete` 位; 2) 当线程接到标志位清除信号, 读入待处理的追踪任务并开始追踪计算, 计算完成后置位 `isComplete`; 3) 调度器检测到该标志位后重新推送新任务至线程。循环执行上述调度过程。若任务池中已无待处理数据, 调度器等待所

有 isComplete 置位后获取散射场计算结果并销毁线程。

5 海面舰船复合场景仿真分析

为了展示基于 Multi-GPU 并行技术的高频方法的计算能力, 首先对本方法的精确度进行验证, 随后比较不同目标尺寸电磁散射计算单 GPU 与多 GPU 的加速效果, 最后利用基于 Multi-GPU 的高频方法对 $2\text{ km}\times 2\text{ km}$ 大小的航母舰队电磁散射进行分析。其中, 运行平台基本硬件参数为: Intel Xeon E5-2620 V4 CPU, 双路 Nvidia Titan XP GPU。舰船模型主要包括: 1) 简化船模型, 尺寸 $27.0\text{ m}(L)\times 3.2\text{ m}(W)\times 5.0\text{ m}(H)$; 2) 货船, 尺寸 $140\text{ m}(L)\times 18\text{ m}(W)\times 17\text{ m}(H)$; 3) 监测船, 尺寸 $132.8\text{ m}(L)\times 20.0\text{ m}(W)\times 23.3\text{ m}(H)$; 4) 航母, 尺寸 $347.5\text{ m}(L)\times 87.8\text{ m}(W)\times 58.4\text{ m}(H)$, 如图 4 所示。

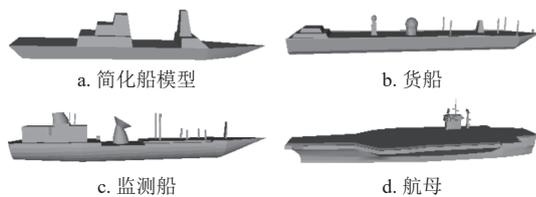


图 4 舰船模型示意图

5.1 精度验证对比

为了验证 Multi-GPU 加速的 PO-SBR 算法精度及性能, 利用本方法与商业软件 FEKO 的数值方法——多层快速多极子 (MLFMM) 对如图 4 所示的简化船模型及风驱海面进行电磁散射计算^[1], 如图 5 所示。其中, 雷达入射波频率为 400 MHz , 入射方位角 0° , 俯仰角 $-89^\circ\sim 89^\circ$ 。限于 FEKO 难以应对电大尺寸目标电磁散射计算, 海面大小为 $48\text{ m}(L)\times 16\text{ m}(W)$, 海面风速为 5 m/s , 相对介电常数 $\epsilon_r = 4 - j8$ 。

MLFMM 计算使用 12 线程, PO-SBR 分别采用单 GPU 与多 GPU 进行加速。其中, MLFMM 平均每角度计算耗时 395.2 s , 单 GPU 下 PO-SBR 平均每角度计算耗时 0.041 s , 双 GPU 下 PO-SBR 平均每角度计算耗时 0.028 s 。可以看到 GPU 加速下的 PO-SBR 与 MLFMM 相比存在明显的速度优势, 对于双 GPU 而言, 由于目标场景尺寸较小, 计算耗时在总耗时中占比低, 所以相较于单 GPU 加速而言效果无法有效体现。同时发现, 单 GPU 模式与双 GPU 模式的计算结果完全吻合, 说明了计算精度的一致性。

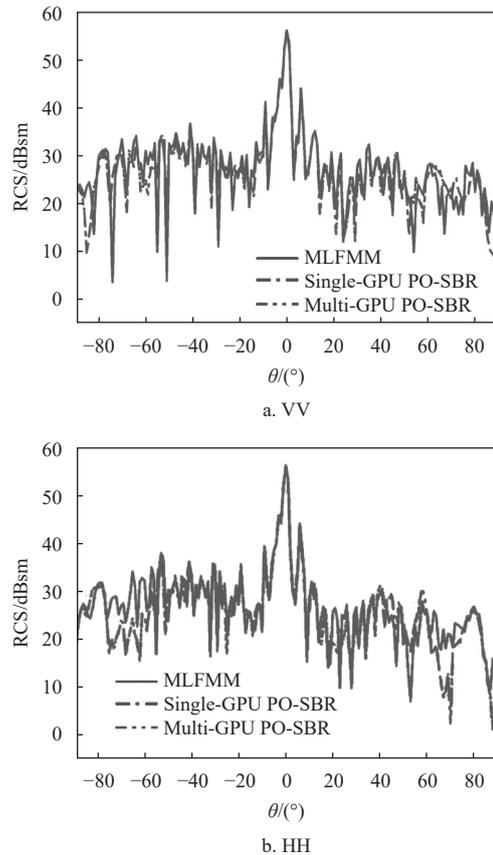


图 5 射线追踪法精度验证

5.2 大型场景效率分析

为了进一步确认本文算法与方案的计算效率, 采取如下两种场景进行分析说明。

场景 1: 货船编队, 其中舰船编队由 6 艘货船组成, 舰船前后间隔 200 m 、左右间隔 50 m 。海面大小分别为: $400\text{ m}(L)\times 400\text{ m}(W)$, $800\text{ m}(L)\times 800\text{ m}(W)$, $1\text{ 000 m}(L)\times 1\text{ 000 m}(W)$, 海面风速为 5 m/s 。船舰编队进行单站散射计算, 其中入射波频率为 3 GHz , 入射方位角为 90° , 俯仰角为 $0^\circ\sim 89^\circ$ 。

场景 2: 大型航母与监测船编队, 该战斗群由 2 艘航母及 6 艘监测船构成。电磁波工作频率为 10 GHz , 入射俯仰角为 45° , 方位角为 $0^\circ\sim 360^\circ$, 海面尺寸为 $2\text{ 000 m}(L)\times 2\text{ 000 m}(W)$, 海面风速为 5 m/s , 场景如图 6 所示。

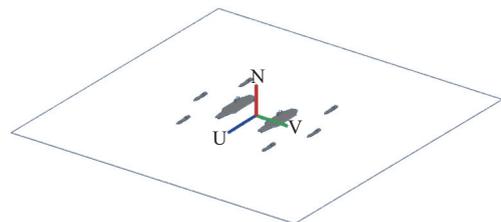
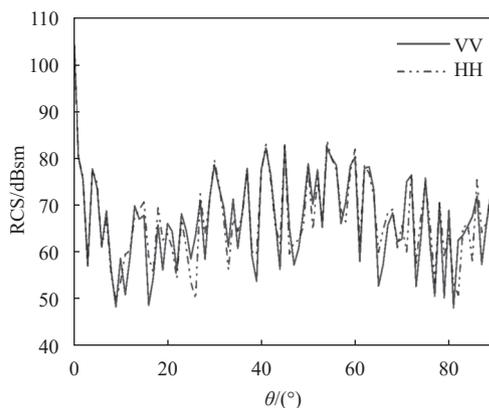


图 6 大型航母与监测船编队场景

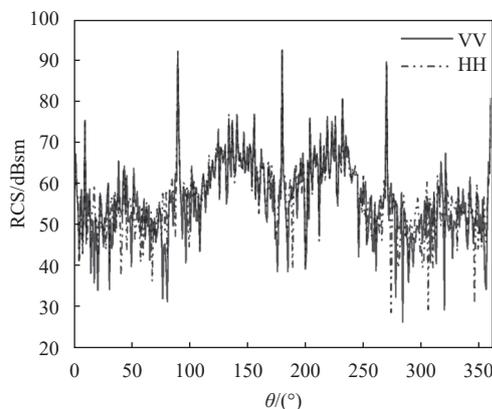
针对场景 1 的仿真, 不同海面对应不同的电尺寸规模, 表 1 为不同海面尺寸下两种方法的计算耗时比较。其中, 1 000 m×1 000 m 的海面对应 $10\,000\lambda \times 10\,000\lambda$ 规模大小, 其计算总耗时为 880 s, 平均单角度耗时 9.78 s, 两种电磁极化结果如图 7a 所示。双 GPU 加速下平均加速比均达到甚至超过 200%。双 GPU 模型下的加速比超过 200% 的原因: 上述 Multi-GPU 加速框架中独立的主线程使得任务分割为异步处理过程, 换言之在 Multi-GPU 中任务分割与射线追踪是同一时刻进行的, 任务分割耗时也由此节省。因此当计算量增大时, Multi-GPU 的加速比也随之增加。

表 1 场景 1 不同海面单站散射计算总耗时

海面尺寸	双GPU耗时/s	单GPU耗时/s	加速比/%
400 m×400 m	133.5	264.3	198
800 m×800 m	575.6	1 179.6	205
1 000 m×1 000 m	880.1	1 856.8	211



a. 场景 1 散射 (1 000 m×1 000 m 海面)



b. 场景 2 散射

图 7 两种场景不同极化散射结果

针对场景 2 的仿真, 其电尺寸规模远超过 $10\,000\lambda$ 。采用单 GPU 与双 GPU 两种不同方法计算

航母战斗群, 最终 RCS 结果保持一致, 如图 7b 所示。在计算耗时上, 单 GPU 平均每角度计算耗时 1 398.19 s, 双 GPU 平均每个观测角度计算耗时 653.59 s, 加速比为 213%。

在上述分析中可以发现, 单 GPU 计算难以解决上述场景耗时高的问题。Multi-GPU 由于其自生的可扩展性, 通过扩充 GPU 数量可以保证对任意真实场景的散射计算时间在可容忍范围内。相比于常见的 CPU 计算集群能够用更少的设备达到更大的加速比。因此相较于 CPU 集群, Multi-GPU 方案在处理并行问题上更具性价比。

6 结束语

实际雷达工程中, 受照射目标场景大小通常可达数万倍波长, 对该场景的散射计算属于超电大尺寸问题, 需要通过计算机并行技术加速算法实现。由于高频方法的局部性原则使其具有良好的并行性, 在处理此类问题时, GPU 相比于 CPU 展示出更高的性价比。现有的 GPU 加速通常为单设备加速, 在处理超电大尺寸问题耗时仍难以满足要求, 因此研究 Multi-GPU 加速的实现以动态提高算力是必要的。本文对海面舰船场景下的高频方法 Multi-GPU 并行加速技术的实现进行了研究。结果表明: 双 GPU 相比于单 GPU 算法平均加速 200% 左右, 可有效应用于超大场景下的电磁散射计算, 具有明显的应用潜力。

参考文献

- [1] “电磁计算”专刊编委会. 电磁计算方法研究进展综述[J]. *电波科学学报*, 2020, 35(1): 13-25.
The Editorial Board of Special Issue for "Computational Electromagnetics". Progress in computational electromagnetic methods[J]. *Chinese Journal of Radio Science*, 2020, 35(1): 13-25.
- [2] HUANG W F, ZHAO Z Q, ZHAO R, et al. GO/PO and PTD with virtual divergence factor for fast analysis of scattering from concave complex targets[J]. *IEEE Transactions on Antennas and Propagation*, 2015, 63(5): 2170-2179.
- [3] 孙立鹏, 方宁, 渠慎丰. 基于 OpenMP 的射线追踪并行计算方法[J]. *电子测量技术*, 2012, 35(1): 50-54.
SUN L P, FANG N, QU S F. Parallel computing of ray tracing by OpenMP[J]. *Electronic Measurement Technology*, 2012, 35(1): 50-54.
- [4] SUNDARARAJAN P, NIAMAT M Y. FPGA implementation of the ray tracing algorithm used in the XPATCH software[C]//Proceedings of the 44th IEEE 2001

- Midwest Symposium on Circuits and Systems. [S.l.]: IEEE, 2001: 446-449.
- [5] BREM R, EIBERT T F. A shooting and bouncing ray (SBR) modeling framework involving dielectrics and perfect conductors[J]. *IEEE Transactions on Antennas and Propagation*, 2015, 63(8): 3599-3609.
- [6] 刘祥. 基于并行 SBR 的电大尺寸目标近场电磁散射研究[D]. 西安: 西安电子科技大学, 2018.
LIU X. Near-Field electromagnetic scattering of large-size targets based on Parallel SBR[D]. Xi'an: Xidian University, 2018.
- [7] 徐万里. 基于 Spark 的近海与舰船目标复合散射特性研究[D]. 西安: 西安电子科技大学, 2019.
XU W L. Research on composite scattering characteristics of near sea and ship target based on spark[D]. Xi'an: Xidian University, 2019.
- [8] 郭立新, 麻军, 王安琪. PC 集群 MPI 并行矩量法研究一维大尺度粗糙面掠入射电磁散射特性[C]//2009 年全国微波毫米波会议. 北京: 电子工业出版社, 2009: 66-69.
GUO L X, MA J, WANG A Q. Study on 1-D large scale rough surface EM scattering at low grazing incidence by parallel MOM based on MPI of PC clusters[C]//Conference of Chinese Microwave/Millimeter-Wave. Beijing: Publishing House of Electronics Industry, 2009: 66-69.
- [9] YANG W, KEE C Y, WANG C F. Novel extension of SBR-PO method for solving electrically large and complex electromagnetic scattering problem in half-space[J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2017, 55(7): 3931-3940.
- [10] 周礼来. 海面舰船近场电磁散射建模及 GPU 并行技术研究[D]. 成都: 电子科技大学, 2022.
ZHOU L L. Research on near-field scattering modeling of ship on the sea and GPU parallel technology[D]. Chengdu: University of Electronic Science and Technology of China, 2022.
- [11] ALTAIR. Altair Feko[EB/OL]. [2022-10-03/2023-04-04]. <https://www.altair.com.cn/feko>.

编辑 叶芳