

基于 GPU 的电磁计算与 SAR 回波仿真加速方法研究

吴雪, 夏伟杰, 蔡炜锋

(南京航空航天大学电子信息工程学院 南京 210016)

摘要: 电磁计算与 SAR (Synthetic aperture Radar, 合成孔径雷达) 回波仿真算法在雷达系统设计、雷达算法验证与目标自动识别算法研究等方面起着重要的作用, 针对传统的电磁计算与 SAR 回波仿真算法耗时过长的问题, 本文提出了一种基于 GPU (Graphic Processing Unit, 图形处理器) 的电磁计算与 SAR 回波仿真全流程加速方法。首先, 将传统算法中的三个核心计算部分——射线跟踪、电磁计算与 SAR 回波仿真进行多线程划分, 然后利用 GPU 通用计算平台, 并行处理所有线程单元, 加快仿真速度。最后, 实验结果表明: 该方法相对于单核 CPU 计算的加速比达到 450 左右, 相对于 10 核心 CPU 计算的加速比达到 50 左右。该方法在保证生成数据与传统算法一致的前提下, 实现了良好的加速效果。

关键词: 射线跟踪; 电磁计算; SAR 回波; 全流程加速; 并行计算

中图分类号: TN957.52 **文献标志码:** A **文章编号:** 2095-1000(2024)04-0108-07

DOI: 10.12347/j.ycyk.20240311001

引用格式: 吴雪, 夏伟杰, 蔡炜锋. 基于 GPU 的电磁计算与 SAR 回波仿真加速方法研究[J]. 遥测遥控, 2024, 45(4): 108-114.

Research on Acceleration Method of GPU-based Electromagnetic Calculation and SAR Echo Simulation

WU Xue, XIA Weijie, CAI Weifeng

(College of Electronic Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: Electromagnetic calculation and SAR echo simulation algorithm play an important role in radar system design, radar algorithm verification, and automatic target recognition algorithms research. To solve the problem of traditional electromagnetic calculation and SAR echo simulation algorithm being too time-consuming, an acceleration method for the entire process based on GPU is proposed. Firstly, the three core computing parts of the traditional algorithm-ray tracing, electromagnetic calculation, and SAR echo simulation are divided into multiple threads. Then, by using the GPU general computing platform, all thread units are processed in parallel to accelerate the simulation speed. Finally, the experimental results show that the acceleration ratio of this method is about 450 compared with single-core CPU and about 50 compared to a 10-core CPU. This method achieves a good acceleration effect on the premise that the generated data is consistent with the traditional algorithm.

Keywords: Ray tracing; Electromagnetic computing; SAR echo simulation; Full process acceleration; Parallel computing

Citation: WU Xue, XIA Weijie, CAI Weifeng. Research on Acceleration Method of GPU-based Electromagnetic Calculation and SAR Echo Simulation[J]. Journal of Telemetry, Tracking and Command, 2024, 45(4): 108-114.

0 引言

随着 GPU 性能的提升和可编程能力的不断发展, GPU 的应用范围也不再仅仅局限于图形绘制计算方面。其强大的并行运算能力使得许多计算密集型应用程序的执行速度得到明显提高^[1], 例如粒子物理模拟、气候模拟、3D 分子形状搜索以及

金融行业中的风险模型计算等。

合成孔径雷达自发明以来, 一直与军事紧密联系在一起, 并随着技术不断发展在军事领域得到了越来越多的应用。然而, 由于其成像机理复杂, 数据处理和图像处理设备价格高昂且体积庞大, 通过实测获取特定条件的 SAR 原始回波数据往往不太实际, 因此, 采用模拟方法获取目标的

SAR回波数据是一种常见的解决方法^[2]。

SAR仿真系统中,电磁计算与回波模拟是整个仿真过程中计算量最为复杂的部分,其计算速度的快慢直接影响到仿真系统的性能,计算所用的时间越长,整个模拟系统的性能就越低。已有的传统仿真方法尽管可以取得良好的仿真结果,但是由于计算量庞大,在常规的CPU或数字信号处理器等计算平台下,所消耗的计算时间较长,而基于GPU的高速浮点运算及并行处理技术可有效降低电磁场强与回波数据的计算时间,且能降低系统成本。

近年来,国内外研究人员在SAR相关计算方面已经取得了许多有实际应用价值的研究成果,例如,厉夫兵等人在GPU上完成了并行编程,实现了电磁计算中KdTree的快速遍历,与CPU单核计算相比,加速比^[3]达到了20左右;Mónica Denham等人提出了基于GPU的一种高效并行的SAR成像算法^[4],并测试对比了回波仿真中每个步骤的耗时;翟新刚等人利用CUDA(Compute Unified Device Architecture,计算统一设备架构)平台的并行执行能力,在GPU上实现了多模SAR成像与干涉条纹生成^[5]。

以上基于GPU的加速方法,只分别提出了射线跟踪、SAR回波仿真或成像的单独并行方案,但未针对SAR仿真系统进行整体分析,做到全流程中并行与串行结合的整体加速。因此,本文提出了一种基于GPU的电磁计算与SAR回波仿真全流程加速方法,区别于其他并行算法,该方法基于各核心运算部分的独立与并行性,对传统算法进行了改进,最大程度地体现了各计算部分的独立与并行关系,从而实现了总体系统的加速。

1 电磁计算与SAR回波仿真算法

传统的SAR回波仿真流程如图1所示,其中核心计算部分是根据设定的电磁波入射条件对目标进行射线跟踪与电磁计算,得到目标的RCS(Radar Cross Section,雷达散射截面)矩阵后,完成以线性调频信号为基础的SAR回波模拟。

在SAR仿真系统的核心计算部分中,首先,根据入射条件建立一个虚拟孔径平面,在该平面上,将一簇平行的入射射线投射到目标上,每一条射线之间的距离是其波长的十分之一,并确保该射线能完全覆盖被测物体。当光线位置被确定

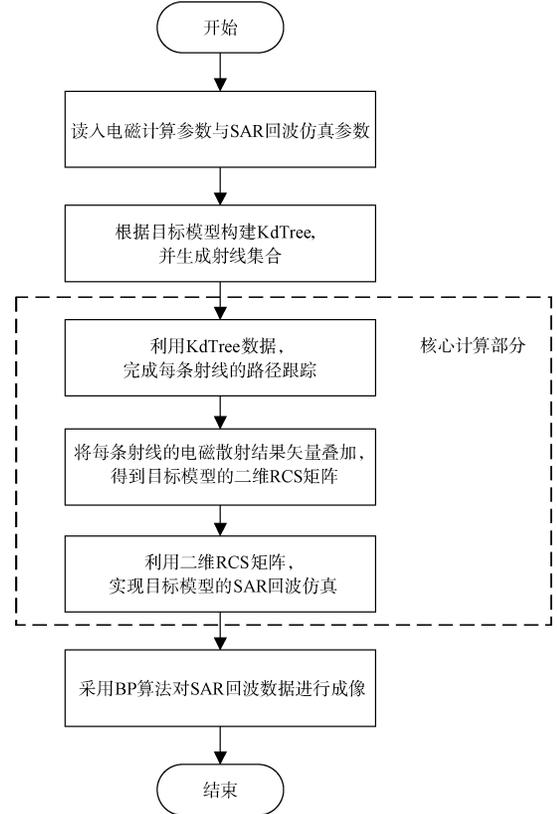


图1 传统的SAR回波仿真全流程

Fig. 1 The whole process of traditional SAR echo simulation

以后,根据反射规律实现每条射线的路径跟踪。

得到所有射线管的反射路径数据后,利用物理光学法和几何光学法进行场强跟踪^[6],求解每根射线管对应的电磁散射强度:

$$\mathbf{E}_{\text{scat}}(\mathbf{r}) = -jk_0\psi_0 \exp(jk_0\hat{\mathbf{i}} \cdot \mathbf{r}_m) \times \int_S \left\{ \hat{\mathbf{r}} \times \left[\hat{\mathbf{n}} \times \mathbf{E}^T - Z_0 \hat{\mathbf{r}} \times (\hat{\mathbf{n}} \times \mathbf{H}^T) \right] \right\} \times \exp[jk_0 \mathbf{r}' \cdot (\hat{\mathbf{r}} - \hat{\mathbf{i}})] ds \quad (1)$$

$$\mathbf{H}_{\text{scat}}(\mathbf{r}) = -jk_0\psi_0 \exp(jk_0\hat{\mathbf{i}} \cdot \mathbf{r}_m) \times \int_S \left\{ \hat{\mathbf{r}} \times \left[\hat{\mathbf{n}} \times \mathbf{H}^T + Y_0 \hat{\mathbf{r}} \times (\hat{\mathbf{n}} \times \mathbf{E}^T) \right] \right\} \times \exp[jk_0 \mathbf{r}' \cdot (\hat{\mathbf{r}} - \hat{\mathbf{i}})] ds \quad (2)$$

上式中,散射点 \mathbf{r}_m 处的总场分别用 \mathbf{E}^T , \mathbf{H}^T 来表示。

将每根射线管得到的电场投影在散射平面得到目标的二维电磁散射矩阵,通过散射电场 \mathbf{E}^T 可计算得到RCS值 σ :

$$\sigma = 4\pi R^2 \frac{|\mathbf{E}_{\text{scat}}|^2}{|\mathbf{E}_i|^2} \quad (3)$$

其中, \mathbf{E}_{scat} 为散射电场强度; \mathbf{E}_i 为入射电场强

度, SAR 时域回波仿真时, 按照方位向采样时间和距离向采样时间可将雷达发射的线性调频信号表示为:

$$s_1(t_a, t_{range}) = \text{rect}\left(\frac{t_{range}}{T_p}\right) \text{rect}\left(\frac{t_a}{L/v}\right) \exp\left(j2\pi f_c t_a + j\pi K_{range} t_{range}^2\right) \quad (4)$$

其中, t_a 与 t_{range} 分别为方位向跟距离向的采样时间, f_c 为雷达发射信号的载频, K_{range} 是距离向上的调频率, T_p 是信号的脉冲宽度, L 为 SAR 的孔径长度, v 为雷达速度。

最后, 根据电磁计算得到的 RCS 矩阵, 将目标场景 Q 中每一个电磁散射单元的回波信号进行矢量叠加, 可以得到整个场景的回波信号:

$$s(t_a, t_{range}) = \sum_{(x,y) \in Q} \sigma(x,y) \text{rect}\left(\frac{t_a - (T_c + y/v)}{L/v}\right) \cdot \text{rect}\left(\frac{D_{range}}{T_p}\right) \exp\left(j2\pi f_c D_a + j\pi K_{range} D_{range}^2\right) \quad (5)$$

其中, $\sigma(x,y)$ 是场景 Q 中坐标为 (x,y) 点的目标 RCS 值, D_{range} 为距离向时延, D_a 为方位向时延, T_c 为 (x,y) 点的波束中心时间。

2 GPU 并行计算

在传统的电磁计算与 SAR 回波仿真算法中, 遍历射线过多、电磁散射矩阵过大以及回波采样点的密集导致了 SAR 仿真系统的执行速度缓慢。本文对相应的核心计算部分进行独立性与并行性分析^[7], 将其中多次重复且互相独立的计算过程划分为多线程单元, 利用 GPU 上的 CUDA 平台, 并行处理所有线程单元, 达到提高算法性能的目的。

2.1 CUDA 并行编程模型

在 2006 年, NVIDIA 公司推出了一套基于 GPU 的通用并行计算平台 CUDA^[8]。CUDA 支持对更复杂的计算问题的快速求解, 它可以与不同的 CPU 架构协同工作, 在同一平台上完成计算, 同时也支持 C/C++、Python、Fortran 等多个语言。

在 CUDA 平台的架构下, 程序的执行分为了 (Host) 和设备端 (Device) 两大部分^[9]。其中在设备端上运行的程序一般都写在核函数 (Kernel) 中, 核函数内的多线程结构涉及到网格 (Grid)、线程块 (Block) 和线程 (Thread) 三个维度, 三个维度之间的关系如图 2 所示。在核函数中, 计算单元可划分为多个网格; 一个网格由多个线程块组成, 线程块

可以按一维、二维或三维的形式进行组合; 而多个线程组合构成一个线程块, 同一线程块内的线程可以进行同步或通过共享内存进行通信。将多次重复执行的计算程序写入核函数, GPU 以线程为单位同步调用多个线程执行核函数, 达到并行计算的效果。线程为程序执行流的最小执行单元, 每一个线程都有其唯一的线程 ID (Thread Id), 当所有线程并行处理时可以通过 Thread Id 进行索引, 找到对应的执行线程。

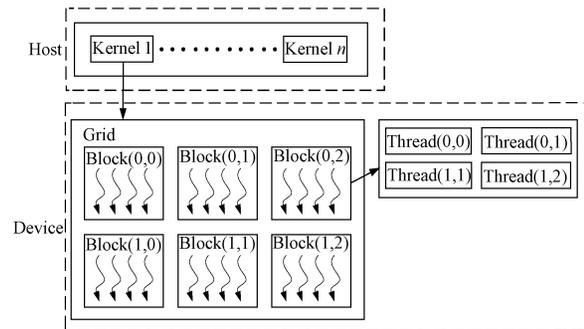


图 2 CUDA 线程层次结构

Fig. 2 Hierarchy of CUDA threads

2.2 算法并行化过程

基于 GPU 的电磁计算与 SAR 回波仿真全流程加速方法的流程图如图 3 所示^[10,11], 本文将 GPU 并行处理技术运用于射线跟踪、电磁计算和 SAR 回波仿真三个核心计算部分。首先, 针对已有的目标模型, 在 CPU 端完成 KdTree 的建立, 将射线路径的节点数据与相关参数传递到 GPU 后, 程序外部依次串行执行射线跟踪、电磁散射计算和 SAR 回波仿真三个核心计算部分, 内部对每个核心计算部分进行多线程划分, 在 GPU 上并行执行, 最后将所需的电磁散射数据和回波数据从 GPU 拷贝到 CPU, 完成保存并记录。

2.2.1 射线跟踪

将 KdTree 的数据从 CPU 传递到 GPU 后, 本文的射线路径跟踪计算采用的是 KdTree 的节点线索创建算法。节点线索的生成包括“继承”和“更新”两个部分, 其中“继承”指的是节点必须从父节点处获得全部的线索, 而“更新”是指更新节点分割时互相的指向关系^[3]。一根射线管在平面模型上的路径示意图如图 4 所示, 以 7 号节点和 8 号节点为例, 7 号节点必须继承 2 号节点与 1 号节点的全部线索信息; 当射线从 7 号节点射入 8 号节点时, 每个节点的更新也都要从它所有的父节点

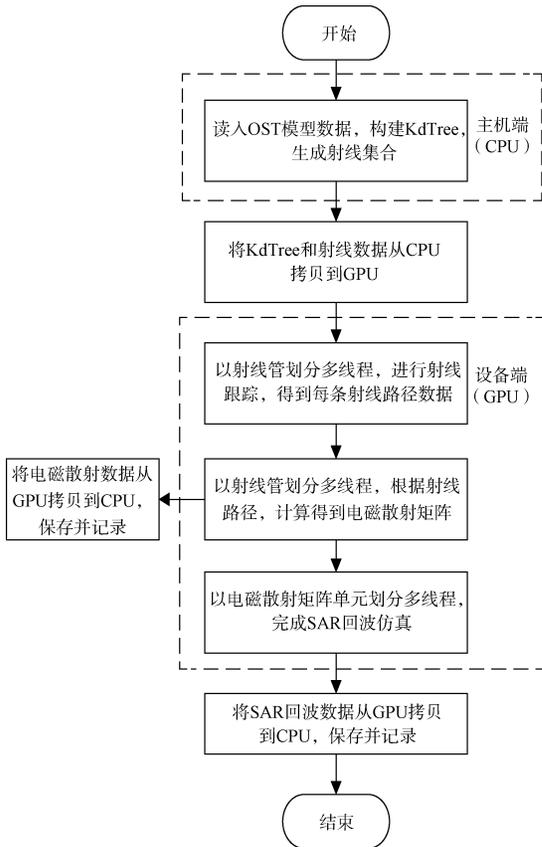


图3 算法整体流程图

Fig. 3 Flowchart of the overall algorithm

开始^[12]。所以当射线管过多时，KdTree的遍历会导致程序运行时间过长。

在射线跟踪的过程中，每根射线管的反射过程都与图4类似，且所有射线管之间互不干扰，因此本文进行射线跟踪计算时按照射线管划分多线程，把图3(b)中的目标节点数据传入GPU后，将每根射线管反射路径的计算过程作为GPU的单个线程，对所有射线管进行基于KdTree的射线追踪并行计算，得到射线的反射路径节点数据，有效提升了系统性能。

2.2.2 电磁散射计算

已知射线在目标上的反射路径后，利用物理光学法与几何光学法完成目标电磁散射矩阵的计算。电磁散射计算是以单个射线管为最小运算单元，通过追踪照射目标的射线组来实现的，其与射线跟踪计算类似，每根射线管的反射或散射过程也是互不干扰的。如图5所示，每根射线管产生的散射电场可以根据电磁波入射条件对应到二维电磁散射矩阵平面的任意位置中。因此，电磁散射计算中也按射线管划分多线程，每根射线管产

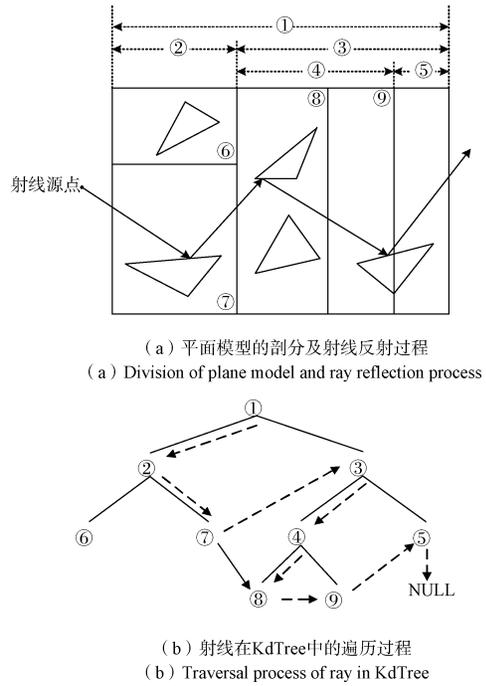


图4 平面模型中的射线追踪路径

Fig. 4 Ray tracing paths in a plane model

生的电磁场强计算作为GPU的单个线程。

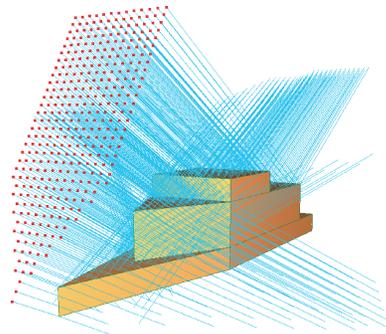


图5 射线跟踪示意图

Fig. 5 Ray tracing diagram

根据设定的分辨率，目标的散射电场在射线管平面的投影会形成一个二维矩阵，如图6所示。通常设定射线管尺寸 $\Delta = \lambda/10$ ，通过分辨率与入射条件得到每根射线管在矩阵中的位置，映射在同一分辨率下的射线管产生的电磁场可直接进行矢量叠加^[13]。分辨率越小，电磁散射矩阵越大，即电磁散射矩阵越能清晰地反映目标的结构特性。

2.2.3 SAR回波仿真

由式(1)至式(5)可知，SAR时域回波的计算是所有电磁散射单元产生的回波矩阵的矢量叠加，单位散射单元P点的回波仿真几何关系如图7所示， R_{near} 和 R_{pt} 分别为当前时刻下的场景近端斜距

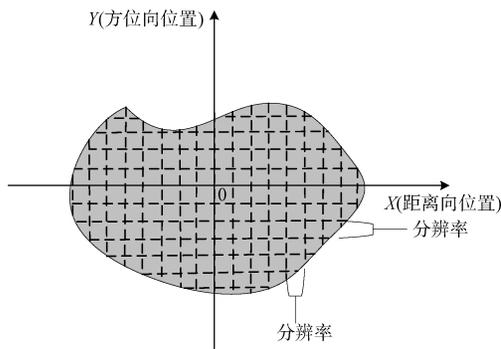


图 6 电磁散射矩阵示意图

Fig. 6 Schematic diagram of electromagnetic scattering matrix

和零多普勒斜距。在回波仿真过程中, 每个散射单元之间的计算互相独立。因此, 本文以电磁散射矩阵为基础, 按矩阵中的单元目标点划分多线程, 并行计算所有线程后, 将仿真结果叠加得到整个场景的回波数据^[6]。

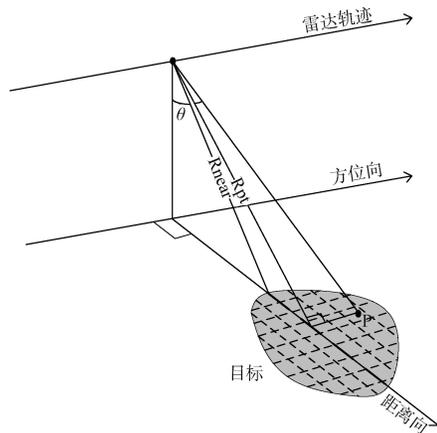


图 7 SAR 时域回波仿真几何图

Fig. 7 The geometry of SAR time domain echo simulation

同时, 对每个电磁散射单元进行回波矩阵计算时, 矩阵中每个方位向和距离向对应的回波数据计算也互不影响, 如图 8 所示, 散射的回波中每个采样脉冲是独立的。所以本文采用了嵌套调用核函数的方法, 在并行计算所有电磁散射单元的回波矩阵时, 再根据距离向采样点和方位向脉冲划分多线程, 以每个方位向脉冲上每个距离向采样点的回波数据计算作为单个线程, 嵌套并行计算回波矩阵中的所有采样点, 加速得到每个散射单元的回波矩阵数据。

3 实验结果

本文的仿真实验采用的是 Windows 10 操作系

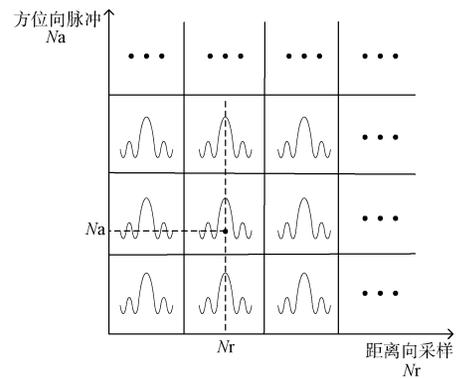


图 8 单个散射单元的时域回波示意图

Fig. 8 Time domain echo diagram of a single scattering element

统, CPU 型号为 Intel(R) Xeon(R) Silver 4210 CPU @ 2.20 GHz, GPU 型号为 NVIDIA GeForce RTX 2080 Ti, 编译环境为 Microsoft Visual Studio 2019 (CPU 端), CUDA toolkit 12.0 (GPU 端)。

为了对算法进行有效性验证, 本文对目标场景为 512 m×512 m 三级海况下的航母目标进行回波模拟, 设置模型分辨率为 1 m, 方位向脉冲为 256, 距离向采样点为 2 048, 即电磁散射矩阵大小为 512×512, 回波矩阵大小为 256×2 048。电磁波入射条件设定射线管尺寸为 1/10 倍入射波波长, 最大反射次数为 3 次, 射线管总数为 1 046 791 根。以上条件在 CPU 和 GPU 平台上得到的仿真结果一致, 海面的电磁散射矩阵图和 SAR 回波成像图分别如图 9 和图 10 所示。

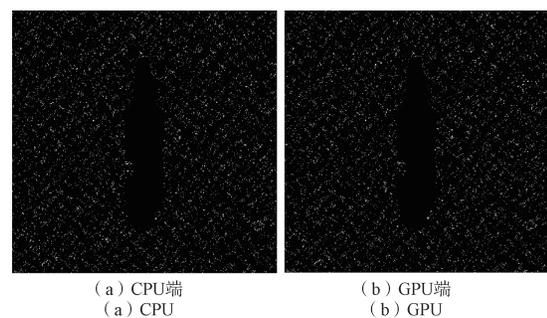


图 9 海面舰船电磁散射矩阵结果

Fig. 9 Results of electromagnetic scattering matrix of ships on sea surface

针对以上的目标场景与仿真参数, 本文使用单核 CPU、10 核心 CPU 和 GPU 测试各个核心计算部分的仿真时长^[14], 见表 1 和表 2, 电磁计算部分加速效果最显著, 全流程的单核 CPU/GPU 加速比为 476.794, 10 核 CPU/GPU 加速比达到了 52.501,

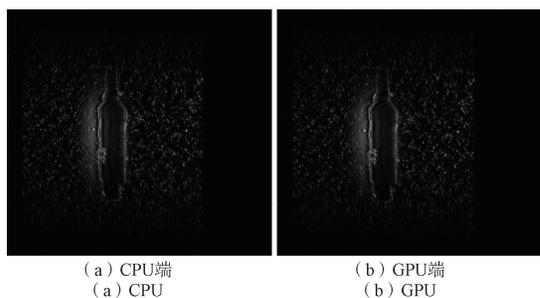


图10 海面舰船SAR时域回波成像图

Fig.10 SAR time domain echo imaging of sea surface ships

本文提出的方法有效提高了SAR仿真系统的速率。

表1 各计算部分耗时表格

Table 1 Time spent on each part of the calculation

计算部分	单核CPU/s	10核CPU/s	GPU/s
射线跟踪	37.578	5.195	0.674
电磁计算	180.311	24.713	0.241
回波仿真	1 706.928	182.04	3.122
全流程	1 924.817	211.948	4.037

表2 各计算部分加速比

Table 2 The acceleration ratio of each calculation part

计算部分	单核CPU/GPU	10核CPU/GPU
射线跟踪	55.754	7.7 077
电磁计算	748.178	102.543 5
回波仿真	546.741	58.309
全流程	476.794	52.501

4 结束语

本文提出了一种基于GPU的电磁计算与SAR回波仿真全流程加速方法，对SAR仿真系统中的核心计算部分完成多线程的划分，实现了基于CUDA平台下的异构架构编程，最后将单核CPU、10核CPU与GPU并行计算的时间进行对比，实验表明单核CPU/GPU加速比达到450左右，10核CPU/GPU加速比达到50左右，最终数据验证了本文提出的CUDA平台下SAR仿真全流程并行加速方法有效提升了SAR仿真系统的性能，大大降低了算法中核心计算部分的运行时间。在以后的研究中，还可以考虑以下问题，算法中数据精度float与double等导致的速度差异^[15]，在允许的误差范围内降低数据精度，达到加快算法运行速率的目的；CPU与GPU之间的数据传递与读取速度较

慢,可以将模型数据在不同GPU的内存区进行合理的计划和配置^[16]，从而进一步提升运算效率。

参考文献

- [1] 王康. 用于射线跟踪的KD-Tree并行构建算法研究[D]. 西安: 西安电子科技大学, 2020.
- [2] 沈晓峰, 吴文明, 田宏伟, 等. 基于GPU并行处理的SAR回波仿真器设计方法[J]. 测控技术, 2019, 38(5): 99-103.
SHEN Xiaofeng, WU Wenming, TIAN Hongwei, et al. Design method of SAR echo simulator based on parallel processing with GPU[J]. Measurement & Control Technology, 2019, 38(5): 99-103.
- [3] 厉夫兵, 苏永琪, 陈文剑. 基于线索KdTree的射线追踪并行计算[J]. 计算机工程与设计, 2023, 12(44): 3677-3682.
LI Fubing, SU Yongqi, CHEN Wenjian. Parallel computation of ray tracing based on rope KD-Tree[J]. Computer Engineering and Design, 2023, 12(44): 3677-3682.
- [4] DENHAM M, ARETA J, TINETTI F G. Synthetic aperture radar signal processing in parallel using GPGPU[J]. J Supercomput, 2016, 72: 451-467.
- [5] 翟新刚. 基于BP算法的SAR成像及干涉条纹生成的GPU实现[D]. 北京: 中国科学院大学, 2016.
- [6] 程先平. 海面背景典型目标SAR回波仿真及图像模拟软件设计与实现[D]. 南京: 南京航空航天大学, 2021.
- [7] 谭运馨, 黄海风, 赖涛, 等. 基于GPU的长轨SAR实时成像算法[J]. 数据采集与处理, 2023, 38(6): 1380-1391.
TAN Yunxin, HUANG Haifeng, LAI Tao, et al. GPU-based real-time imaging algorithm for long-track SAR[J]. Journal of Data Acquisition and Processing, 2023, 38(6): 1380-1391.
- [8] 翟新刚, 韦立登, 汪丙南, 等. 基于GPU的多模式SAR实时成像算法研究[J]. 电子测量技术, 2016, 39(10): 81-86.
ZHAI Xingang, WEI Lideng, WANG Bingnan, et al. Research on real-time imaging algorithm for multi-mode SAR with GPU[J]. Electronic Measurement Technology, 2016, 39(10): 81-86.
- [9] 任义. 基于GPU的SAR成像算法研究[D]. 重庆: 重庆邮电大学, 2017.
- [10] 白澜, 魏仁乐, 郭拯危, 等. 基于GPU的多模式SAR成像加速研究[J]. 太赫兹科学与电子信息学报, 2023, 21(8): 1037-1042.
BAI Lan, WEI Renle, GUO Zhengwei, et al. Multi-mode SAR imaging acceleration based on GPU[J]. Jour-

- nal of Terahertz Science and Electronic Information Technology, 2023, 21(8): 1037-1042.
- [11] 苟立婷, 李勇, 朱岱寅, 等. 基于 GPU 的圆迹视频 SAR 实时成像算法[J]. 雷达科学与技术, 2019, 17(5): 550-556, 563.
GOU Liting, LI Yong, ZHU Daiyin, et al. A Real-time algorithm for circular video SAR imaging based on GPU [J]. Radar Science and Technology, 2019, 17(5): 550-556, 563.
- [12] LI Fubing, SU Yongqi. Stackless KD-Tree traversal for ray tracing[C]//2021 2nd International Conference on Electronics, Communications and Information Technology(CECIT), Sanya, China, 2021, 12: 27-29.
- [13] 郑文军, 杨伟, 周礼来. 基于 GPU 并行技术的超大型海面舰船电磁散射仿真[J]. 电子科技大学学报, 2023, 52(4): 549-554.
ZHENG Wenjun, YANG Wei, ZHOU Lilai. Electromagnetic scattering simulation of extremely electrically large sea-ship scene based on GPU parallel technology [J]. Journal of University of Electronic Science and Technology of China, 2023, 52(4): 549-554.
- [14] 杨思军. 基于多核 CPU 的雷达信号并发处理架构设计[J]. 舰船电子对抗, 2021, 44(5): 62-66.
YANG Sijun. Design of radar signal concurrency processing architecture based on Multi-core CPU[J]. Ship-board Electronic Countermeasure, 2021, 44(5): 62-66.
- [15] 孟大地, 胡玉新, 石涛, 等. 基于 NVIDIA GPU 的机载 SAR 实时成像处理算法 CUDA 设计与实现[J]. 雷达学报, 2013, 2(4): 481-491.
MENG Dadi, HU Yuxin, SHI Tao, et al. Airborne SAR Real-time Imaging Algorithm Design and Implementation with CUDA on NVIDIA GPU[J]. Journal of Radars, 2013, 2(4): 481-491.
- [16] 宋佩涛. MOC 全堆芯中子输运 CPU/GPU 异构并行算法研究[D]. 哈尔滨: 哈尔滨工程大学, 2021.

[作者简介]

- 吴 雪 2000 年生, 硕士研究生。
夏伟杰 1979 年生, 博士, 教授, 博士生导师。
蔡炜锋 1999 年生, 硕士研究生。

(本文编辑: 潘三英)

(英文编辑: 赵尹默)