

## 激光与光电子学进展

## 水下无线光信道并行仿真方法研究

张建磊<sup>1\*</sup>, 寇琳琳<sup>1</sup>, 王洁<sup>1</sup>, 刘雪辰<sup>2</sup>, 杨祎<sup>1</sup>, 贺锋涛<sup>1</sup><sup>1</sup>西安邮电大学电子工程学院, 陕西 西安 710121;<sup>2</sup>海军装备部, 陕西 西安 710077

**摘要** 针对基于蒙特卡洛的水下无线光信道仿真计算量大、计算效率低的问题,提出了利用 Open MP (open multi-processing) 和 CUDA (compute unified device architecture) 的水下无线光信道并行仿真方法。通过将计算密集型部分移植到多线程并行计算的方式提高仿真计算效率。在此基础上引入 3 种优化方案,通过剔除无效光子和限制高散射事件的方式,加速数据合并,减少主存显存的数据交换量,进一步提高仿真效率。对比分析了在不同水质、不同计算环境以及不同光子数和距离等条件下的加速效果。结果表明,相比于传统串行仿真,图像处理器 (GPU) 水下光散射并行计算方法的加速最高可达 300 倍;中央处理器 (CPU) 水下光散射计算方法的加速最高可达 90 倍。

**关键词** 海洋光学; 并行架构; 水下无线光通信; 蒙特卡洛仿真

中图分类号 TN911.74

文献标志码 A

DOI: 10.3788/LOP202259.1901001

## Research on Parallel Simulation Method of Underwater Wireless Optical Channel

Zhang Jianlei<sup>1\*</sup>, Kou Linlin<sup>1</sup>, Wang Jie<sup>1</sup>, Liu Xuechen<sup>2</sup>, Yang Yi<sup>1</sup>, He Fengtao<sup>1</sup><sup>1</sup>College of Electronic Engineering, Xi'an University of Post & Telecommunications, Xi'an 710121, Shaanxi, China;<sup>2</sup>Naval Equipment Department, Xi'an 710077, Shaanxi, China

**Abstract** A parallel simulation method for underwater wireless optical channel simulation is proposed based on open multi-processing (Open MP) and compute unified device architecture (CUDA) to address the computational complexity and low computational efficiency of the Monte Carlo-based model. The simulation calculation efficiency is improved by transplanting computationally intensive parts to threaded parallel computing. Three optimization schemes are introduced to accelerate merging by eliminating invalid photons, limiting high-scattering events, and reducing the amount of data exchange between main and video memories, further improving the simulation efficiency. The acceleration effects in various water types, computing environments, and photon numbers and distances are evaluated. The parallel calculation efficiency of graphics processing units has an acceleration effect of up to 300 times, whereas that of central processing units has an acceleration effect of 90 times compared with traditional serial simulation.

**Key words** oceanic optics; parallel architecture; underwater optical wireless communication; Monte-Carlo simulation

## 1 引言

目前水下无线通信技术主要分为以下 3 类: 声学通信、射频通信以及无线光通信 (Underwater optical wireless communication, UOWC)。相比水下声学及射频通信, 水下无线光通信具有传输速率高、通信容量大以及实现成本低等优势<sup>[1-2]</sup>。然而, 由于海水信道的固有光学属性以及复杂的海洋环境, 光信号在海水信道中传输时会受到吸收、散射等影响产生多径效应, 进

一步导致波形失真和码间串扰<sup>[3]</sup>。

Beer-Lambert 定理<sup>[4]</sup>可作为水下信道衰减模型的理论基础。然而, 该定理却有 2 个条件: 1) 接收发射两端完全对齐; 2) 所有经过散射的光子能量完全丢失。实际上, 光子经历多次散射后仍会有部分被接收端捕获, 该方法低估了接收端接收到的光功率。为了改进上述问题, Mobley 等<sup>[5]</sup>将辐射传输方程 (Radiative transfer equation, RTE) 引入 UOWC 信道建模, 但是 RTE 方程包含 5 个自变量, 方程开放, 无法确定准确解。RTE 描

收稿日期: 2021-08-24; 修回日期: 2021-09-19; 录用日期: 2021-10-19

基金项目: 装备预研教育部联合基金 (8091B032130)、国家自然科学基金 (61805199)、陕西省技术创新引导专项基金 (2020TG-001)、水下信息与控制重点实验室基金

通信作者: \*zhangjianlei@xupt.edu.cn

述了光在海水信道中传播时由于海水介质的吸收散射等影响发生衰减的过程,这一过程的核心就是 RTE 方程的求解,常用的求解 RTE 方程数值解的方法有蒙特卡洛(Monte Carlo)光线追踪法<sup>[6]</sup>、不变嵌入法<sup>[7]</sup>、离散坐标法<sup>[8]</sup>和矩阵法等。针对这一问题,Illi等<sup>[9]</sup>对 RTE 方程的求解方法进行改进,利用 Newton-Cotes 公式以及改进的辛普森公式对方程进行快速收敛,从而计算出 RTE 的准确解,并且描述了利用三种散射相函数(TTHG (two-term Henyey-Greenstein)\HG (Henyey-Greenstein)\FF(Fournier-Forand))的信道建模方法。相比上述方法,蒙特卡洛仿真(Monte-Carlo simulation, MCS)法能够在兼顾水体固有光学属性、散射相函数以及各水域条件的情况下三维建模,方法简单可操作性高,并且可以清晰表现每个光子的吸收散射过程。但是,MCS法是通过模拟大量的仿真样本得到更真实的仿真结果<sup>[10]</sup>,所以仿真时会出现计算时间长、效率低等问题,即使有如方差缩减技术<sup>[11]</sup>、贝叶斯蒙特卡洛(Bayesian Monte Carlo, BMC)法<sup>[12]</sup>等,也仅仅是在提高光子的利用率方面改进。为了解决上述问题,Chen等<sup>[13]</sup>采用了“半解析”的MCS法进行仿真模拟,该方法通过“期望值估计”以及“对发射接收范围内的概率进行估计”的方法将MCS法中的随机取样改为积分的数值运算,该方法能够有效降低因选取样本的数据规模较大而预分配内存空间不合理的问题,然而该方法使用的随机取样点为“数学取样点”而非“物理取样点”,因此在分析时仅仅只能对结果进行数值分析,其中的物理含义并不明晰。为了解决此问题,Yuan等<sup>[14]</sup>通过将接收器处的光功率表示为概率密度函数(PDF)的多次积分,并且使用改进的蒙特卡洛积分(Monte Carlo Integral, MCI)法求解。与广泛使用的MCS模型相比,MCI模型采用了更简单的算法结构,计算速度更快。改进后的MCI法相比于MCI而言,其计算效率提高了12倍,相比于MCS法计算效率提高了5.6倍。但是,该结果仍然低于并行后的MCS法,并且利用MCS法仿真时各光子散射的路径清晰,可以通过统计不同的数据得到相应的散射特征,例如散射次数、时间、权重、空间分布等。

随着中央处理器(CPU)核心数日趋增加以及统一计算设备架构(Compute unified device architecture, CUDA)的逐渐完善,并行算法为MCS法提供了良好的架构条件。基于MCS法的UOWC信道模型具有串行数据处理较少、各光子间散射相对独立等特点,很适合引入并行算法。因此,本文基于蒙特卡洛算法并利用CUDA和Open MP(Open multi-processing)进行并行处理,引入3种优化方案,分析了不同计算环境和参数条件下的加速效果。

## 2 理论推导

### 2.1 辐射传输方程

水下无线光信道可以通过求解 RTE 进行解析化

建模。此时,单位时间内单位立体角发散的能量<sup>[15]</sup>为

$$dP_{div} + dP_{ext} = dP_{sca} + dP_{src} \quad (1)$$

基于能量守恒定律的 RTE 方程,其中影响能量交换的因素如下。

1) 单位时间内,体元辐射能量

$$dP_{div} = \frac{\partial \Phi(\mathbf{r}, \mathbf{s}, t)}{\partial \mathbf{s}} ds dA d\Omega = \frac{\partial \Phi(\mathbf{r}, \mathbf{s}, t)}{\partial \mathbf{s}} d\Omega dV, \quad (2)$$

式中: $\mathbf{r}$ 为散射发生的位置; $\mathbf{s}$ 为单位方向矢量; $t$ 为时间; $ds$ 是沿着光传播方向 $\mathbf{s}$ 的长度元; $dA$ 是垂直于 $\mathbf{s}$ 方向的面元; $d\Omega$ 为单位立体角; $dV$ 为单位体积元; $\Phi(\mathbf{r}, \mathbf{s}, t)$ 为单位时间内流经单位面积、单位立体角的能量。

2) 单位时间内,立体角单元内体元由于吸收和散射所引起的能量损失

$$dP_{ext} = (\mu_t ds) [\Phi(\mathbf{r}, \mathbf{s}, t) dA d\Omega], \quad (3)$$

式中: $\mu_t$ 为衰减系数; $\mu_t ds$ 为在 $ds$ 中的消光概率(由于吸收散射导致的)。

3) 任意方向 $\mathbf{s}'$ 入射到体元上时,单位时间内散射至方向 $\mathbf{s}$ 的单位立体角内的能量

$$dP_{sca} = (\mu_s dV) \left[ \int_{4\pi} \Phi(\mathbf{r}, \mathbf{s}', t) P(\mathbf{s}' \cdot \mathbf{s}) d\Omega' \right] d\Omega, \quad (4)$$

式中: $\mu_s$ 为散射系数, $\mu_s = N_s \sigma_s$ , $N_s$ 为散射体的数量密度, $\sigma_s$ 为散射体的散射截面; $\Phi(\mathbf{r}, \mathbf{s}', t) d\Omega'$ 为单位时间 $t$ 、单位立体角 $d\Omega'$ 内由单个散射体获得的能量; $P(\mathbf{s}' \cdot \mathbf{s}) d\Omega$ 为沿 $\mathbf{s}'$ 方向传播的光散射进入方向 $\mathbf{s}$ 所在立体角 $d\Omega$ 内的概率。

4) 单位时间内,光源产生的能量

$$dP_{src} = S(\mathbf{r}, \mathbf{s}, t) dV d\Omega. \quad (5)$$

将式(2)~式(5)代入式(1)可得

$$\mathbf{s} \cdot \nabla \Phi(\mathbf{r}, \mathbf{s}, t) + \mu_t \Phi(\mathbf{r}, \mathbf{s}, t) = \mu_s \int_{4\pi} \Phi(\mathbf{r}, \mathbf{s}', t) P(\mathbf{s}' \cdot \mathbf{s}) d\Omega' + S(\mathbf{r}, \mathbf{s}, t), \quad (6)$$

式中: $\mathbf{s}' \cdot \mathbf{s}$ 等于2个向量之间夹角的余弦值,光子在信道中不断碰撞时,原运动方向会改变。

### 2.2 基于 MCS 的水下无线光信道模型

水下光子传输的散射过程如图 1 所示,发射端

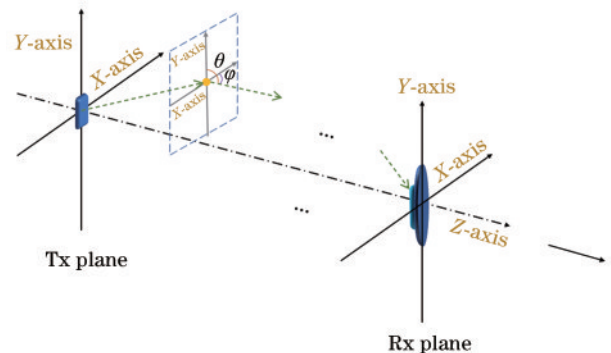


图 1 水下光子传输的散射过程

Fig. 1 Scattering process of underwater photon transmission

(Tx plane)为光子的初始位置,接收端(Rx plane)为光子经历散射后的终止位置。光子在信道中传输时,通过对方位角  $\varphi$ 、散射角  $\theta$  和步长  $d$  抽样确定光子的下一次散射位置,多次位置迭代后确定光子的散射轨迹。散射过程<sup>[16]</sup>如下。

1) 方位角  $\varphi$ 、散射角  $\theta$  和步长  $d$  可表示为

$$\begin{cases} \varphi = 2\pi\xi \\ \theta = \arccos \left\{ \frac{1}{2g} \left[ 1 + g^2 - \left( \frac{1 - g^2}{1 - g + 2g\xi} \right)^2 \right] \right\}, \end{cases} \quad (8)$$

$$d = -\frac{\ln(1 - \xi)}{\mu_t}, \quad (9)$$

式中: $\xi$ 为随机数,取值范围为 $[0, 1]$ ;  $g$ 为各向异性因子,当 $g \rightarrow 0$ 时,表示前向散射与后向散射概率相等。当 $g \rightarrow 1$ 时,表示前向散射强于后向散射,本次仿真中 $g = 0.924$ 。

2) 通过抽样决定光子的散射角和方位角,进而确定下次散射的方向矢量:

$$\begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \begin{bmatrix} \frac{\mu'_x \cdot \mu'_z}{\sqrt{1 - \mu_z'^2}} & -\frac{\mu'_y}{\sqrt{1 - \mu_z'^2}} & \mu'_x \\ \frac{\mu'_y \cdot \mu'_z}{\sqrt{1 - \mu_z'^2}} & \frac{\mu'_x}{\sqrt{1 - \mu_z'^2}} & \mu'_y \\ -\sqrt{1 - \mu_z'^2} & 0 & \mu'_z \end{bmatrix} \cdot \begin{bmatrix} \sin \theta' \cos \varphi' \\ \sin \theta' \sin \varphi' \\ \cos \theta' \end{bmatrix}, \quad \text{if } \mu_z'^2 < 1, \quad (10)$$

为了避免除数接近于 0,光子传输方向的方向余弦矢

量可以修正为

$$\begin{bmatrix} \mu_x \\ \mu_y \\ \mu_z \end{bmatrix} = \text{sgn}(\mu'_z) \cdot \begin{bmatrix} \sin \theta' \cos \varphi' \\ \sin \theta' \sin \varphi' \\ \cos \theta' \end{bmatrix}, \quad \text{if } \mu_z'^2 \approx 1, \quad (11)$$

式中: $\text{sgn}()$ 为符号函数,更新后的方向余弦应满足

$$\mu_x^2 + \mu_y^2 + \mu_z^2 = 1. \quad (12)$$

3) 光子的下一次位置  $(x, y, z)$  为

$$\begin{cases} x = x' + d \cdot \mu'_x \\ y = y' + d \cdot \mu'_y \\ z = z' + d \cdot \mu'_z \end{cases} \quad (13)$$

4) 光子的部分权重会在每一步结束时被吸收,权重将变为

$$\omega_n = \omega_{n-1} \times \frac{\mu_s}{\mu_t}. \quad (14)$$

式中: $\omega_n$ 为散射  $n$  次后光子的权重,具有新权重的光子将会在信道中继续散射和吸收,直至触发截止条件。

MCS为模拟光子在海水信道中的散射提供了一种高效准确的仿真方案。为了提高运算效率,减少计算耗时,采用并行架构以增加仿真样本数量和提高仿真效率。

### 3 算法实现

MCS通过模拟更多的光子产生更稳定的结果,当模拟样本量越大时,结果越接近实际值,仿真中可以通过输出光子散射参数得到相应的信道光学特性,如:时间参数  $t$ ,光子权重  $\omega$ ,光子位置参数  $(x, y)$  等。UOWC 散射理论已在第 2 小节中给出,水下无线光信道仿真框架如图 2 所示,不同水域的参数如表 1 所示( $\mu_a$ 为吸收系数, $\mu_s$ 为散射系数, $\mu_t$ 为衰减系数, $\omega_0$ 为单次散射率)。

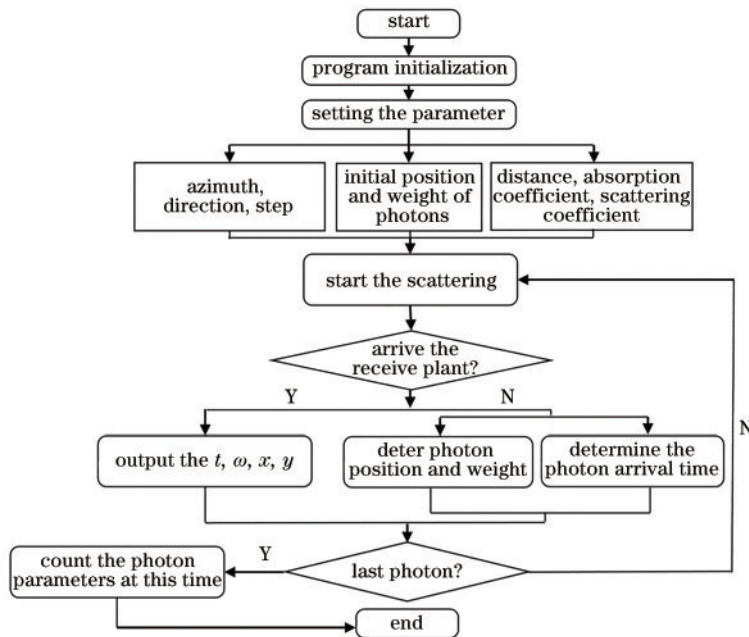


图 2 水无线光信道仿真框架

Fig. 2 UOWC channel simulation of framework

表 1 不同水域的光学特性

Table 1 Optical properties of different water types

Water type	$\mu_a / m^{-1}$	$\mu_s / m^{-1}$	$\mu_t / m^{-1}$	$\omega_0$
Clear ocean ( I )	0.114	0.037	0.151	0.245
Coastal ( II )	0.179	0.291	0.398	0.731
Harbor ( III )	0.366	1.824	2.190	0.833

### 3.1 基于 Open MP 的并行加速仿真

Open MP 是一种共享内存系统的多处理器程序设计的指导性编译方案,具有架构简单、运行效率高、可移植性好的优点<sup>[17]</sup>。该方法定义了一系列可移植可扩展的开发接口,在单台多处理器计算机上就能发挥优秀的并行效果。基于 Open MP 的 UOWC 信道的并行架构模式如图 3 所示。基于 Open MP 的 UOWC 的并行算法模型如图 4 所示。

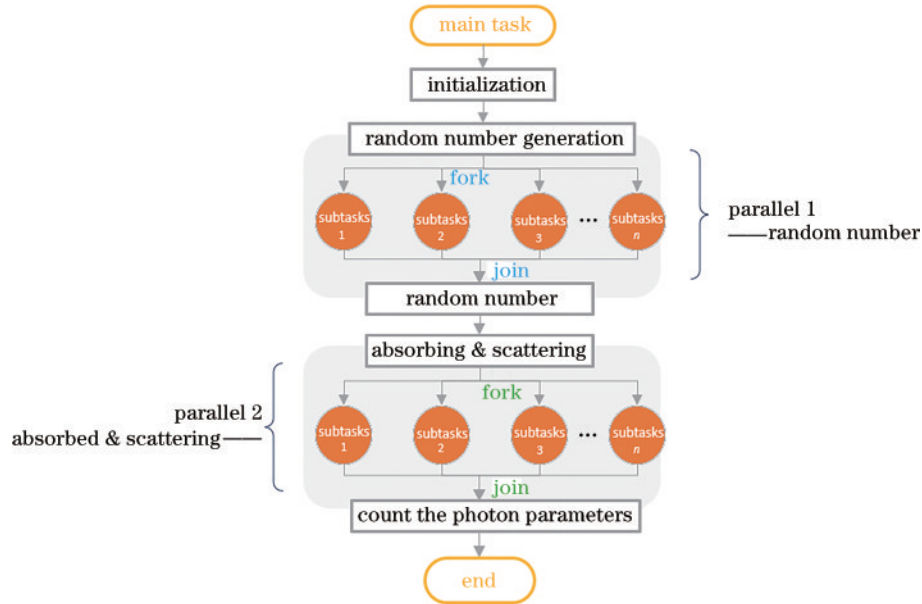


图 3 基于 Open MP 的并行架构模型

Fig. 3 Parallel architecture model based on Open MP

```

Algorithm 2: Algorithm of Open MP parallelized UOWC code
1. input // Parametric values
2. initialization // Initializing the variables
3. quicksort // Define function
4. random number // Generate random array

#pragma omp parallel for private(4)

5. for (n < thePhoton_N; ++n) // Start to absorption and scattering
6. absorption && scattering

#pragma omp parallel for schedule(6)

7. end for // End parallel
8. index // Addressing merge
9. FILE* fp // File storage
10. print x, y, ω, and t values // End
    
```

图 4 基于 Open MP 的 UOWC 并行算法模型

Fig. 4 Parallel algorithm simulation based on Open MP

### 3.2 基于 CUDA 的并行加速仿真

CUDA 是 NVIDIA 推出的一种并行编程模型。通过图像处理器(GPU)(设备端,执行并行部分)和 CPU(主机端,执行串行部分)“协同合作”的方式统一到异构计算系统中<sup>[18-20]</sup>。主机端对设备端进行存储、分配,设备端对计算密集型部分进行处理<sup>[20]</sup>。CUDA 编程模型和 CUDA 数据管理模型如图 5 所示。

优化部分如下:

1) GPU 在工作时只存在一个 CUDA 流,并且每个 CUDA 流中并行核函数和数据交换函数依次执行,如果采用多个 CUDA 流,会顺序执行,在本次仿真中,并行部分主要为随机数生成和光子的吸收散射过程。为了保证各线程的计算负载能力相当,所有光子的散射过程将均匀分配给各线程并行计算,降低线程空闲。

2) 在光子散射碰撞的过程中,每一个线程负责计算一个光子的散射过程。光子散射后会产生 4 种状态:反向散射;权重过小而“死亡”的光子;未到达接收范围内的无效光子;完成散射的光子(如图 6 所示)。因此,需要对传回主存的光子进行分类、排序、合并和压缩,剔除其中的无效光子,从而减少显存与主存数据交换的数据量,减少 GPU 中的条件分支,更高效地完成提取和分析。

3) 光子在经历过多次散射后,权重会下降到较小的水平。具有小权重的光子在进行传输模拟时会耗费大量的计算时间,而其对整体的辐亮度贡献很小。在仿真过程中,可以在误差允许的范围内减少小权重光子的抽样次数,以达到提高计算效率的效果。目前,最常用的一种算法是轮盘赌<sup>[21]</sup>,当光子权重过小时,通过

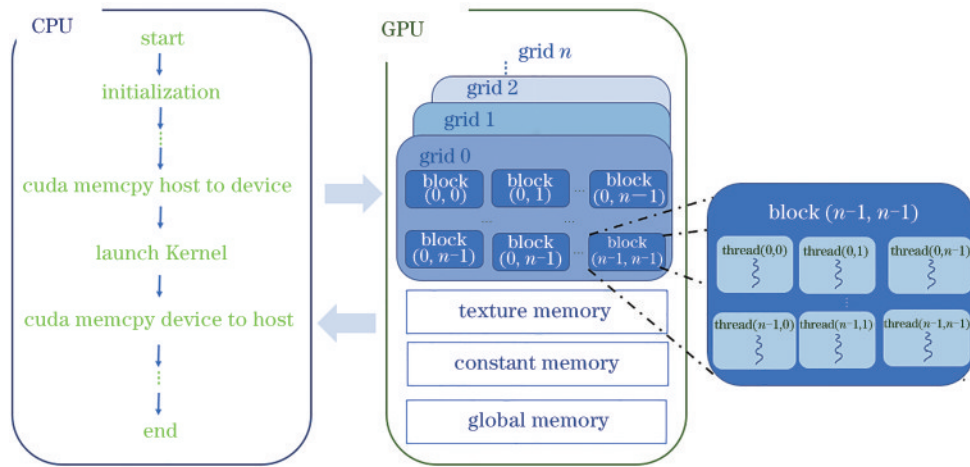


图 5 CUDA 编程模型和 CUDA 数据管理模型

Fig. 5 CUDA programming model and CUDA data management model

<b>case 1</b>	program initialization – scattering parameters – scattering – propagation – arrived receive plate – end
<b>case 2</b>	program initialization – scattering parameters – scattering – propagation – photon weigh too small – photon dead – end
<b>case 3</b>	program initialization – scattering parameters – scattering – propagation – outside the received plate – photon dead – end
<b>case 4</b>	program initialization – scattering parameters – scattering – propagation – $\mu_a < 0$ – dead – end

图 6 光子散射的截断方案

Fig. 6 Truncation scheme for photon scattering

随机抽样确定光子的存活状态。在仿真过程中,会预先设定光子的消亡阈值( $\tau$ ),当此时光子权重( $\omega_{n-1}$ )小于消亡阈值时,光子的权重由随机数( $\chi$ )选择,该光子的权重有一定的概率更新为  $\omega_n$ ,具有  $\omega_n$  权重的光子将继续下一次散射,此过程表示为

$$\omega_n = \begin{cases} \omega_{n-1} \cdot \tau, & \text{if } \chi \leq \frac{1}{\tau} \\ 0, & \text{others} \end{cases} \quad (15)$$

程序中所使用的轮盘赌算法如图 7 所示。

基于 CUDA 的 UOWC 信道仿真的算法模型如图 8 所示。

CUDA 程序的框架如图 9 所示。

基于 CUDA 的并行加速模型是“CPU+GPU”协同合作方式进行的,具体实施步骤如下:

1) 设定水下无线光通信中各光学特征参数,如吸收系数、散射系数、各向异性因子等。

2) 将程序中运算独立、复杂度高、运算时间长的部分(随机数生成部分和各光子的吸收散射过程)分配于 GPU 端(Kernel 1 和 Kernel 2)并行计算。其中,线程分配部分由 CPU 生成。利用 CUDA 平台实现主存与显存中数据的相互传输。

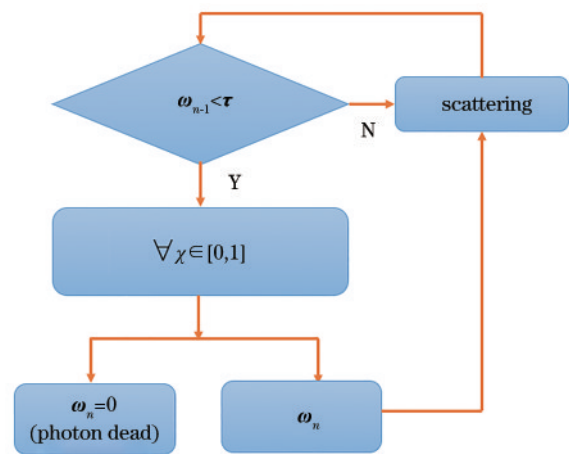


图 7 程序中所使用的轮盘赌算法

Fig. 7 Roulette in the program

```

Algorithm 3 : Algorithm of CUDA UOWC code
1. input // Parametric values
2. initialization // Initializing the variables
3. quicksort // Define function
4. random number // generate random array
5. __global__ set_random
6. __global__ void kernel_random // Launch the first kernel function
7. __global__ void solveKernel // Launch the second kernel function
8. for (n < thePhoton_N; ++n) // Start to absorb and scatter
9. absorb && scattering
10. cudaMalloc() // Memory allocation on device
11. cudaMemcpy() // Copy data from host to device
12. computer the parametric values
13. cudaMemcpy() // Copy data from device to host
14. end for
15. index // Addressing && merging
16. FILE* fp // File storage
17. print x, y, \omega and t values // End
    
```

图 8 基于 CUDA 的 UOWC 信道仿真并行算法模型

Fig. 8 Parallel algorithm for UOWC channel simulation based on CUDA

3) 步骤 2) 中 Kernel 1 为随机数并行生成,并利用数组存储;Kernel 2 为各光子的吸收散射过程,方位角、散射角和步长的抽样通过对随机数组调用生成。

4) 最后一个光子散射结束后,各光子的数据由

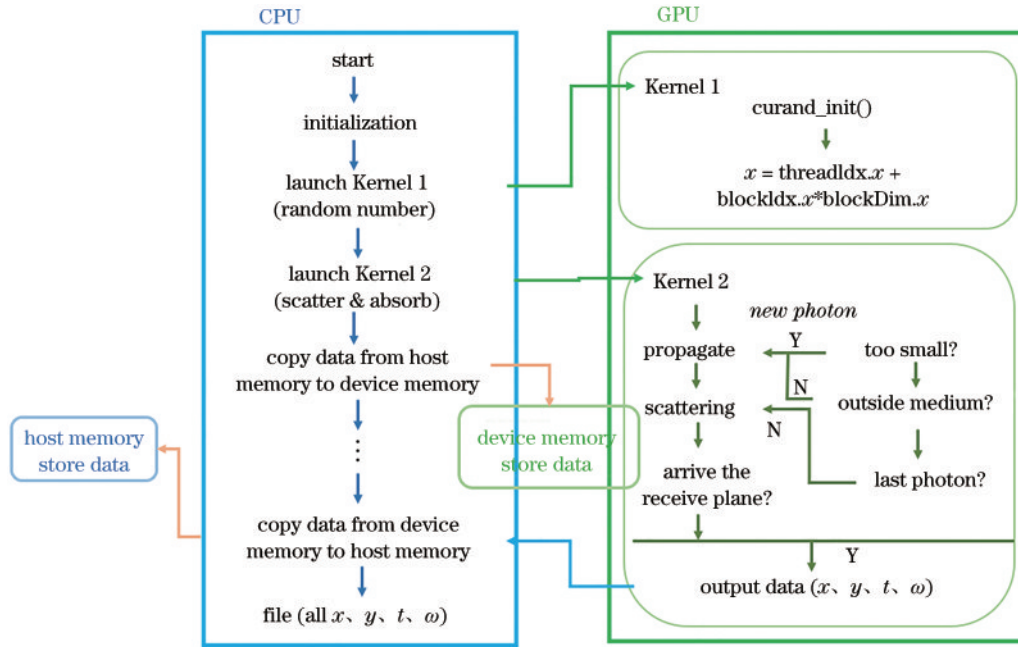


图9 CUDA程序的框架

Fig. 9 Framework of CUDA program

GPU传回CPU,将传回数据进行筛选、排序、存储。

5) 释放内存,结束运算。

## 4 讨论与分析

测试比较了3种平台在不同计算环境、不同输入参数条件下的并行加速效果。为了消除测试的随机性,计算结果为3次测试数据的平均值。2种计算环境硬件参数如表2所示。

### 4.1 不同计算平台下的仿真结果

图10和图11仿真了距离为30 m时,  $3 \times 10^6$  个光子在3种计算平台和3种海水类型下的信道脉冲响应。利用了2种测试方法均方误差(Mean squared error, MSE)和决定系数(Nash-Sutcliffe model efficiency

表2 两种计算环境

Table 2 Two computer environments

	Computer environment 1	Computer environment 2
	GeForce MX 150	GeForce GTX 1050 TI
CUDA core	384	768
Clock rate /MHz	936	1366
Global memory /Mb	6098	12252
CPU	i7-8550U	E5-2670
CPU core	4	8

coefficient, 常称为  $R^2$ ) 验证仿真的准确性,表3展示了其拟合优度。

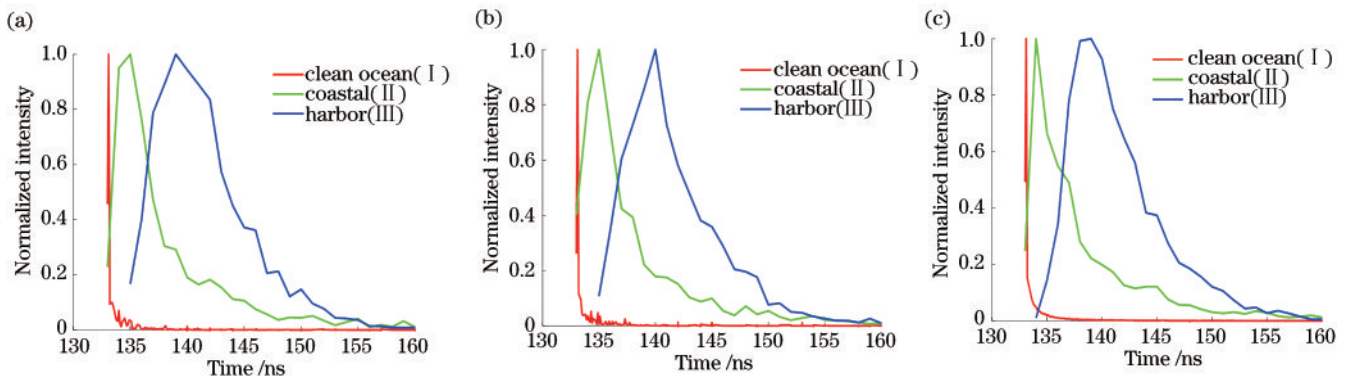


图10 计算环境1中不同水域的信道脉冲响应(a)计算平台为Matlab;(b)计算平台为Open MP;(c)计算平台为CUDA

Fig. 10 Impulse response for different water types in computer environment 1. (a) Computing platform is Matlab;

(b) computing platform is Open MP; (c) computing platform is CUDA

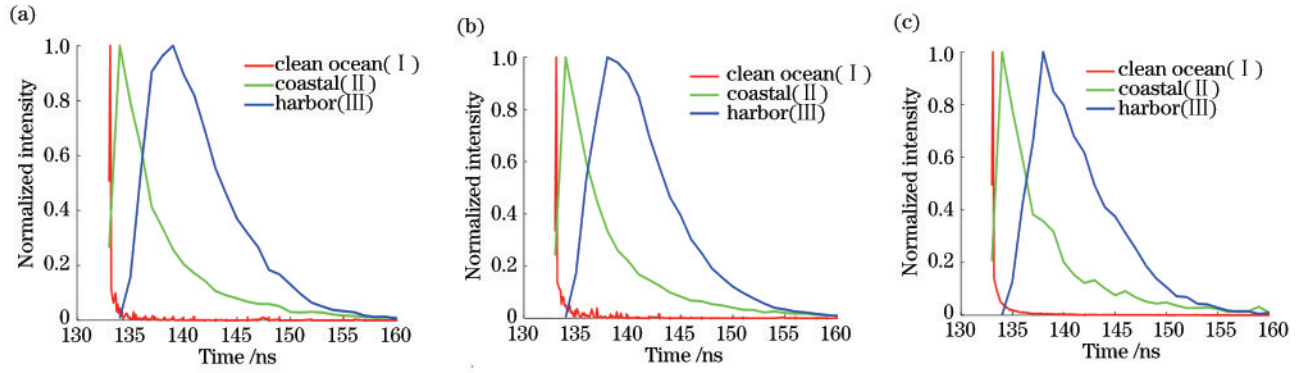


图 11 计算环境 2 中不同水域的信道脉冲响应。(a)计算平台为 Matlab;(b)计算平台为 Open MP;(c)计算平台为 CUDA

Fig. 11 Impulse response for different water types in computer environment 2. (a) Computing platform is Matlab; (b) computing platform is Open MP; (c) computing platform is CUDA

表 3 不同水域、平台、计算环境条件下的测试参数和测试结果

Table 3 Test parameters and test results in different waters, platforms and computing environmental conditions

Computer environment	Computing platform	Simulation parameter			Result		
		Photon number	Distance /m	Water type	MSE	R <sup>2</sup>	
1	Matlab	3 × 10 <sup>6</sup>	30	I	0.012760	0.9616	
		3 × 10 <sup>6</sup>	30	II	0.069580	0.9156	
		3 × 10 <sup>6</sup>	30	III	0.052060	0.9594	
		3 × 10 <sup>6</sup>	30	I	0.005969	0.9906	
		3 × 10 <sup>6</sup>	30	II	0.060130	0.9137	
		3 × 10 <sup>6</sup>	30	III	0.041380	0.9832	
	Open MP	CUDA	3 × 10 <sup>6</sup>	30	I	0.004200	0.9962
			3 × 10 <sup>6</sup>	30	II	0.061600	0.9295
			3 × 10 <sup>6</sup>	30	III	0.059200	0.9567
		Matlab	3 × 10 <sup>6</sup>	30	I	0.010900	0.9728
			3 × 10 <sup>6</sup>	30	II	0.059800	0.9280
			3 × 10 <sup>6</sup>	30	III	0.040100	0.9846
2	Open MP	3 × 10 <sup>6</sup>	30	I	0.005900	0.9914	
		3 × 10 <sup>6</sup>	30	II	0.062100	0.9163	
		3 × 10 <sup>6</sup>	30	III	0.053900	0.9563	
	CUDA	3 × 10 <sup>6</sup>	30	I	0.003300	0.9976	
		3 × 10 <sup>6</sup>	30	II	0.059200	0.9842	
		3 × 10 <sup>6</sup>	30	III	0.052500	0.9604	

$$E_{MS} = \frac{\sum_{i=1}^N [F_s(I_i) - F(I_i)]^2}{N}, \quad (16)$$

式中:  $F_s(I_i)$  为模拟数据  $I_i$  的累积分布函数;  $F(I_i)$  为确定点  $I_i$  处分布  $f_{hs}(x)$  的分布函数值。MSE 越接近 0 表示拟合越好。

$R^2$  的测试指标计算公式为

$$R^2 = 1 - \frac{S_e}{S_t}, \quad (17)$$

式中:  $S_e = \sum_{i=1}^M (f_{s,i} - f_{p,i})^2$ ;  $S_t = \sum_{i=1}^M (f_{s,i} - \bar{f})^2$ ,  $f_{s,i}$  和  $f_{p,i}$  是仿真数据的概率,  $\bar{f}$  为  $f_{s,i}$  在区间  $M$  内的均值。 $R^2$  越

接近 1 表示拟合越好<sup>[22]</sup>。

#### 4.2 不同计算环境下的加速效果

表 4 和表 5 分别为计算环境 1 和 2 中 10 万、20 万和 30 万光子在 20 m 与 35 m 距离下的加速效果。表中所涉及的 M 为 Matlab、O 为 Open MP、C 为 CUDA。Speedup of M to O、Speedup of M to C、Speedup of O to C 分别为; Matlab 与 Open MP 运行时间的比值; Matlab 与 CUDA 运行时间的比值; Open MP 与 CUDA 运行时间的比值。

经实验结果得出: 由于 Matlab 一般情况下为串行运算, 每条指令顺序完成, 因此在多任务处理方面运行效果不佳。相比于 Matlab 运算, 基于 CPU 的 Open MP 具有明显的加速效果: 在计算环境 1 中的加速效果

表 4 计算环境 1 中的加速效果

Table 4 Acceleration effect in computing environment 1

Number of photons	Distance / m	Speedup of M to O	Speedup of M to C	Speedup of O to C
$1 \times 10^5$	20	16.21	101.26	<b>6.25</b>
$1 \times 10^5$	35	16.23	75.33	4.64
$2 \times 10^5$	20	33.92	116.98	3.45
$2 \times 10^5$	35	<b>58.83</b>	<b>189.31</b>	3.22
$3 \times 10^5$	20	52.75	132.38	2.51
$3 \times 10^5$	35	40.53	121.85	3.01

表 5 计算环境 2 中的加速效果

Table 5 Acceleration effect in computing environment 2

Number of photons	Distance / m	Speedup of M to O	Speedup of M to C	Speedup of O to C
$1 \times 10^5$	20	32.04	243.20	7.59
$1 \times 10^5$	35	33.66	234.44	6.97
$2 \times 10^5$	20	54.59	<b>351.56</b>	6.44
$2 \times 10^5$	35	44.24	351.17	<b>7.94</b>
$3 \times 10^5$	20	<b>93.05</b>	210.86	2.27
$3 \times 10^5$	35	76.93	207.76	2.70

为 16~52 倍,在计算环境 2 中为 32~93 倍。计算环境 1 中,CPU 为四核,而计算环境 2 中为八核,当并行指令开始后,计算环境 1 中将会有 4 个执行核心对并行语句同步计算,而计算环境 2 中将会有 8 个核心并行计

算,因此,计算环境 2 中运行效率更高。通过将大量并行处理任务拆解到线程与网格运算,达到加速效果。相比于 Matlab 串行运算,CUDA 最高可加速约 300 倍。

#### 4.3 不同仿真参数条件下的加速效果

针对计算环境 2 中 Open MP 以及 CUDA 进行仿真结果进行分析,结果如表 6~8 所示。

通过上述数据可知,在相同水域、相同传输距离、不同光子数的条件下 CUDA 对 Open MP 的最大加速比达到了 10.64;相同水域、相同光子数、不同传输距离的条件下最大加速比达到 3.43;相同光子数、相同传输距离、不同水域的情况下最高加速比为 2.5。

最高加速效果与最低加速效果,表 6 相差约为 8,表 7 相差约为 0.75,表 9 相差约为 0.53。由此可知,光子个数对仿真运行速度的影响较大,距离次之,水域最小。

结合本文结果以及 MCS 法分析如下:由于蒙特卡洛法是利用概率现象作为模拟对象的数值模拟方法,通过对大量具有相同概率分布或数学期望的数值进行随机抽样,再以之作为参数参与程序运算的,因此,在仿真中,即使使用多线程或多核心进行并行运算,仍不可避免地生成大量的随机数,大大降低了程序的运行速度。表 6 改变的参数为光子数,随着光子数的不断增加,随机数生成部分需要更多的时间生成调用以及更大的内存存储,造成程序运行时间的延长。为了进

表 6 不同光子数的加速效果

Table 6 Acceleration effect of different photon number

Number of photons	Type of sea water	Distance /m	Running time of Open MP /s	Running time of CUDA /s	Speedup of C to O
$1 \times 10^5$	Ocean water (I)	50	5.074	0.477	<b>10.64</b>
$3 \times 10^5$		50	14.485	3.786	3.83
$5 \times 10^5$		50	21.491	7.844	2.74
$1 \times 10^6$		50	43.959	10.877	4.04

表 7 不同距离的加速效果

Table 7 Acceleration effect of different distances

Number of photons	Type of sea water	Distance /m	Running time of Open MP /s	Running time of CUDA /s	Speedup of C to O
$3 \times 10^5$	Ocean water (I)	10	12.231	3.776	3.24
$3 \times 10^5$		30	12.755	3.781	3.37
$3 \times 10^5$		50	13.683	3.988	<b>3.43</b>
$3 \times 10^5$		100	15.764	5.872	2.68

表 8 不同海水类型的加速效果

Table 8 Acceleration effect of different water types

Number of photons	Type of sea water	Distance /m	Running time of open MP /s	Running time of CUDA /s	Speedup of C to O
$3 \times 10^5$	Ocean water ( I )	30	7.287	3.705	1.97
$3 \times 10^5$	Coastal ( II )	30	8.005	3.903	2.05
$3 \times 10^5$	Harbor ( III )	30	10.003	4.002	<b>2.50</b>



一步提高程序的运算效率,本文所阐述的并行仿真中对于随机数生成部分进行了改进,将原有的在循环中需要迭代而产生的随机数改为数组调用形式,通过对比表 6 中的结果不难发现,光子数对仿真运行速度的影响较大。表 7 以及表 8 为相同光子数,不同距离以及不同水域条件下进行模拟的结果。通过数据仍然可以看出 CUDA 相比于 Open MP 运行速度更快,但是最高

与最低的加速效果相差不大,因此可以得出结论:在生成随机数数量相差不大时(即光子数相同时)CUDA 的运算速度高于 Open MP。

#### 4.4 不同优化条件下的加速效果

测试了(II)、(III)类海水条件下 300 万光子在优化前后的加速效果,如表 9 所示。表中所述 Non-optimized 为未进行优化,Optimized 为进行优化。

表 9 不同优化方法下的加速效果

Table 9 Acceleration effect of different optimization methods

	Computer environment 2				Computer environment 1		
	Photon number	Distance / m	Water type	Running time of CUDA / s	Running time of Open MP / s	Running time of CUDA / s	Running time of Open MP / s
Non-optimized	$3 \times 10^6$	30	II	831	1051	1059	1347
Optimized	$3 \times 10^6$	30	II	422	659	562	723
Non-optimized	$3 \times 10^6$	30	III	917	1194	1084	1471
Optimized	$3 \times 10^6$	30	III	562	714	633	830

从表 9 数据可知,优化后程序有明显的加速效果。测试中光子权重阈值为  $1 \times 10^{-5}$ , 此时有 50%~71% 的光子进入轮盘赌。其中,提速效果最明显的为计算环境 2 中 CUDA 在(II)类海水中的加速效果,加速比约为 1.9,Open MP 的最大加速比也约为 1.7。相比于优化前,优化后的程序能够在硬件资源不变的条件得到更优越的加速效果。

## 5 结 论

利用 CUDA 和 Open MP 对基于 MCS 的 UOWC 信道仿真提出了 2 种并行方案。通过将计算密集型部分移植到各线程并行计算的方式提高程序的运算效率,通过剔除无效光子、限制高散射事件的方式使得加速比大幅提高。相比于传统的串行计算方式,CUDA 和 Open MP 具有更高的运行速度。不同于 Matlab 的运行方式,并行计算的程序执行结束后不再保留无用的光子数据和处理程序,能够很大程度上提高程序的运算时间、节省存储空间,解决在不同仿真条件下光子的散射模型运行时间长的问题。

### 参 考 文 献

- [1] Spagnolo G S, Cozzella L, Leccese F. Underwater optical wireless communications: overview[J]. Sensors, 2020, 20(8): 2261.
- [2] Saeed N, Celik A, Al-Naffouri T Y, et al. Underwater optical wireless communications, networking, and localization: a survey[J]. Ad Hoc Networks, 2019, 94: 101935.
- [3] Zhu S J, Chen X W, Liu X Y, et al. Recent progress in and perspectives of underwater wireless optical communication[J]. Progress in Quantum Electronics, 2020, 73: 100274.
- [4] Zeng Z Q, Fu S, Zhang H H, et al. A survey of underwater optical wireless communications[J]. IEEE Communications Surveys & Tutorials, 2017, 19(1): 204-238.
- [5] Mobley C D, Stramski D. Effects of microbial particles on oceanic optics: methodology for radiative transfer modeling and example simulations[J]. Limnology and Oceanography, 1997, 42(3): 550-560.
- [6] Mobley C D, Sundman L K. Effects of optically shallow bottoms on upwelling radiances: inhomogeneous and sloping bottoms[J]. Limnology and Oceanography, 2003, 48(1): 329-336.
- [7] Mobley C D. Light and water: radiative transfer in natural waters[M]. San Diego: Academic Press, 1994.
- [8] Jin Z, Stamnes K. Radiative transfer in nonuniformly refracting layered media: atmosphere-ocean system[J]. Applied Optics, 1994, 33(3): 431-442.
- [9] Illi E, Bouanani F E, Park K H, et al. An improved accurate solver for the time-dependent RTE in underwater optical wireless communications[J]. IEEE Access, 2019, 7: 96478-96494.
- [10] Zhang S, Zhang K, Dong Y H. A uniform spatial channel model for underwater wireless optical communication links [C]//GLOBECOM 2020-2020 IEEE Global Communications Conference, December 7-11, 2020, Taipei, Taiwan, China. New York: IEEE Press, 2020.
- [11] Plass G N, Kattawar G W, Guinn J A, Jr. Radiative transfer in the earth's atmosphere and ocean: influence of ocean waves[J]. Applied Optics, 1975, 14(8): 1924-1936.
- [12] Koning A J. Bayesian Monte Carlo method for nuclear data evaluation[J]. The European Physical Journal A, 2015, 51(12): 207-213.
- [13] Chen P, Pan D L, Mao Z H, et al. Semi-analytic Monte Carlo model for oceanographic lidar systems: lookup table method used for randomly choosing scattering angles [J]. Applied Sciences, 2018, 9(1): 48-61.
- [14] Yuan R Z, Ma J S, Su P, et al. Monte-Carlo integration

- models for multiple scattering based optical wireless communication[J]. IEEE Transactions on Communications, 2020, 68(1): 334-348.
- [15] Boas D. Diffuse photon probes of structural and dynamical properties of turbid media: theory and biomedical applications[EB/OL]. [2021-04-06]. <https://repository.upenn.edu/dissertations/AAI9636132/>.
- [16] Qadar R, Kasi M K, Kakar F A. Monte Carlo based estimation and performance evaluation of temporal channel behavior of UWOC under multiple scattering [C]//OCEANS 2017-Anchorage, September 18-21, 2017, Anchorage, AK, USA. New York: IEEE Press, 2017.
- [17] Zhong Z, Rong M A, Lei H E, et al. An efficient large-scale mesh deformation method based on MPI/Open MP hybrid parallel radial basis function interpolation[J]. Chinese Journal of Aeronautics, 2020, 33(5): 1392-1404.
- [18] NVIDIA CUDA C programming guide[EB/OL]. [2021-05-04]. <http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA C Programming Guide.pdf>.
- [19] NVIDIA, NVIDIA CUDA computer unified device architecture programming Gui-de Version 2.0, NVIDIA [EB/OL]. [2021-04-24]. <http://developer.nvidia.com/object/cuda.html>.
- [20] Block B J. Platform independent, efficient implementation of the Ising Model on parallel acceleration devices[J]. The European Physical Journal Special Topics, 2012, 210(1): 147-157.
- [21] Stpiczynski P. Algorithmic and language-based optimization of Marsa-LFIB<sub>4</sub> pseudorandom number generator using Open MP, Open ACC and CUDA[J]. Journal of Parallel and Distributed Computing, 2020, 137: 238-245.
- [22] Shin M, Park K H, Alouini M S. Statistical modeling of the impact of underwater bubbles on an optical wireless channel[J]. IEEE Open Journal of the Communications Society, 2020, 1: 808-818.