

基于 CUDA 的三维数据场航路规划方法

笄良龙, 臧 涛, 杨廷武, 刘 贝

(海军潜艇学院航海观通系, 青岛 260071)

摘 要: 针对数据量庞大、复杂的三维数据场环境下航路规划速度偏低的问题, 提出一种基于统一计算设备架构(CUDA)的三维数据场航路规划方法。该方法以三维水下声场为威胁模型, 水下航行的潜艇为背景, 运用 CUDA 对大规模数据场环境下对航路进行规划, 对可并行计算部分与 CUDA 进行计算, 仿真结果证明该方法可以提高规划速率、优化初始航路。

关键词: 航路规划; 统一计算设备架构(CUDA); 航路优化

3D Data Field Route Planning Method Based on CUDA

DA Liang-long, ZANG Tao, YANG Ting-wu, LIU Bei

(Department of Navigation and Communication, Navy Submarine Academy, Qingdao 260071)

【Abstract】 Aiming at the low rate of route planning due to huge, complex 3D data, this paper proposes a 3D data field route planning based on Compute Unified Device Architecture(CUDA). The method takes the 3D underwater acoustic field as the threaten model, the navigational submarine as the background, uses CUDA to do the route planning under massive data field, computes the part which can parallel computing with CUDA. Simulation result proves that the method can raise planning speed, and optimize initial route.

【Key words】 route planning; Compute Unified Device Architecture(CUDA); route optimization

航路规划是指在一定的战术环境背景下, 满足一定条件的从起始点至目标点的最优路径^[1-2]。其条件往往为躲避危险、绕开障碍和缩短航程等。面对三维空间数据场庞大的数据量, 要保证在不损失数据情况下较快地规划出一条优化路径, 传统的串行方法往往使 PC 机的负荷过重, 而并行的路径规划方法已逐步被人们认同。

1 威胁场的建模与仿真

对于水下航行的潜艇, 敌声纳对潜艇探测能力直接关系潜艇的暴露与否, 水下声场是声纳探测能力的具体表现。以水下声场为威胁场在一定程度上客观地表达了潜艇所处位置的威胁情况。

采用 WKBJ 简正波方法^[3]计算声场数据如下:

$$I(r) = \frac{8\pi}{r} \left\{ \left| \sum_l \phi_l(z_1) \phi_l(z_2) \sqrt{v_l} \cos(v_l r) \right|^2 + \left| \sum_l \phi_l(z_1) \phi_l(z_2) \sqrt{v_l} \sin(v_l r) \right|^2 \right\} \\ TL = 10 \lg \frac{I(1)}{I(r)} \quad (1)$$

其中, $I(r)$ 为声强; l 是波导简正波的号数; z_1 和 z_2 分别是声源和接收深度; $\phi_l(z)$ 和 v_l 是简正波的本征函数和本征值; TL 为传播损失。对于计算出的传播损失 TL 可以表达威胁场, TL 越小表示威胁程度越大, 反之, TL 越大表示威胁程度越小。对某位置计算出的各方位扇面声场进行插值等计算后可以得到三维声场数据, 运用 Monte Carlo 方法对某一传播损失处进行可视化表达。

2 基于统一计算设备架构的航路规划

2.1 统一计算设备架构简介

GPU 自开发以来, 由于具有高内存带宽驱动的多个核

心, 其浮点运算速度远高于 CPU, 为图像和非图像处理提供丰富资源^[4-5]。统一计算设备架构(Compute Unified Device Architecture, CUDA)作为一种在 GPU 上的新型硬件和软件架构, 无须映射到图像应用程序编程接口(Application Programming Interface, API)便可在 GPU 上管理和进行一系列并行数据计算, 且可以返回计算结果, 增强了 GPU 的功能。

使用 CUDA 编程时, 可以将应用程序中数据并行、计算密集的函数编译到设备指令集中, 将得到的执行内核下载到设备上, 主机通过执行一系列执行内核完成相应操作。主机和设备保留自己的动态随即存取存储器(Dynamic Random Access Memory, DRAM)。用户可以通过优化的 API 调用设备的高性能直接内存访问引擎将数据从一个主机 DRAM 复制到设备 DRAM 中或从设备 DRAM 复制到主机 DRAM, 如此可将 GPU 看作 CPU 的一个协处理器。

为了使用方便, CUDA 将执行内核的线程批组织为线程块的网格, 每个内核对应分配一个的线程块网格, 每个网格中对应一定数量的线程块, CUDA 线程框架如图 1 所示。

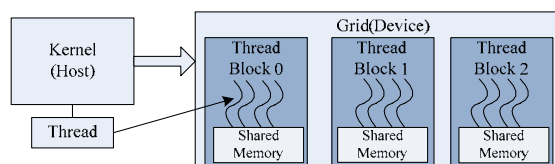


图 1 CUDA 线程框架

基金项目: 国家部委基金资助项目

作者简介: 笄良龙(1967 -), 男, 教授、博士生导师, 主研方向: 军队作战环境学; 臧 涛, 硕士研究生; 杨廷武, 博士研究生; 刘 贝, 硕士研究生

收稿日期: 2009-03-18 **E-mail:** zt_qtxy@126.com

2.2 基于栅格的路径搜索

基于栅格的路径搜索是对规则离散数据模型的一种最优搜索，在复杂环境下往往比以图论和仿生学算法为基本理论的路径搜索更有效、便捷。本文采用一种距离值传递法进行路径搜索，该方法将路径搜索考虑为寻找一条从终点退回到起点的路径。路径规划器在搜索者所在环境中从终点开始以波的形式向外传播，传播经过自由区域，遇到障碍时距离值波的前峰绕过障碍物，最终到达起始点，栅格搜索如图 2 所示。对于代表搜索者起始位置的环境中的每一点，到达终点的最短路径可以通过从起点开始沿其周围距离值下降最陡路径点连接得到。如果没有找到下降到终点路径，那么表示起点和终点间没有路径存在，即终点不可到达。

	30	27	S	23	22	23	24	27	28
	29	26	23	20	19	20	21	24	27
				16	17	20	23	26	
4	3	4	7	10	13	16	19	22	25
3	7	3	6	9	12		20	23	26
4	3	4	7	10	13		23	24	27
4	3	4	7	11	14		23	24	27
3	7	3					20	S	26
4	3	4	7	10	13	16	19	22	25
7	6	7	8	11	14	17	20	23	26

图 2 栅格搜索

与其他方法相比，距离值传递法还支持多目标情况下寻找路径。在此情况下，路径将自动趋向较近的目标。

2.3 航路目标函数

设从起始点 A 到目标点 B 存在一条最优航路，航路规划的目标综合考虑航行距离和威胁因素使航线的总代价为最小，采用如下代价方程描述相邻航路点之间的代价：

$$F(i, j) = \lambda_1 D(i, j) + \lambda_2 T(j) + \lambda_3 C(j) \quad (2)$$

其中， $F(i, j)$ 表示从第 i 点到第 j 点的路径代价； $D(i, j)$ 表示 2 点之间路径的距离代价由实际距离表示； $T(j)$ 表示 j 点的声场威胁代价。

$$T(j) = (FOM - TL) \quad (3)$$

其中， FOM 为声纳优质因子； $C(j)$ 为 j 点深度代价。

$$C(j) = \begin{cases} c \exp(1 - \frac{H}{H_0}) H & H \leq H_0 \\ 0 & H > H_0 \end{cases} \quad (4)$$

其中， λ_1 、 λ_2 、 λ_3 为加权系数。

2.4 CUDA 的并行航路规划

基于 CUDA 的航路规划原理以基于栅格的路径规划为基础，在三维环境下，运用 CUDA 并行计算能力，对多个位置点同时进行代价计算及赋值。实际应用主要由以下 3 个部分组成：

(1) 设备内存和参数设定

初始化路径数据场与环境威胁数据场。路径数据场存储计算得到的各位置点代价，起点初始值为 0，其他位置为一个较大的值(如 10 000)。环境威胁数据场存储环境的威胁数据，其值与所在位置的传播损失相关。路径数据场与环境威胁数据场运用内存拷贝函数 *cudaMemcpy* 由主机内存拷贝至设备内存。

设定 CUDA 的相关执行配置和参数 *Kernel <<< gridDim, blockDim, Ns >>> (parameter)*，其中，*gridDim* 表示栅格维数；*blockDim* 表示块的维数；*Ns* 表示指定在共享内存的字节数量；*parameter* 表示参数。为了使每个线程都能对

应一个独立的位置点，令 *gridDim*blockDim* 维数与数据场大小相等。

(2) CUDA 搜索航路

进行上述设置后，对每个索引为 (x, y, z) 的线程进行操作，即可以对应到每个位置点 (x, y, z) 的操作。每个线程由线程 ID(*thread ID*) 标识，是块的线程号。对于三维线程，每个索引为 (x, y, z) 的线程可以表示为

$$x = blockDim.x * blockIdx.x + threadIdx.x$$

$$y = blockDim.y * blockIdx.y + threadIdx.y$$

$$z = blockDim.z * blockIdx.z + threadIdx.z$$

其中，*blockIdx* 表示块索引；*threadIdx* 表示块内的线程索引。

搜索方法为数次调用执行内核对多个位置点同时进行代价的计算，通过不断计算并覆盖相应位置的路径数据场数据达到最终搜索目的。由于距离值传递法是利用波传递原理向外传播，许多位置点在开始阶段并不需要进行计算，因此对其进行相应限制，即在位置点 (x, y, z) 与起始点距离小于波的传递范围时才进行相应操作，CUDA 搜索如图 3 所示。在每次调用执行内核后，将传递范围值相应增加。

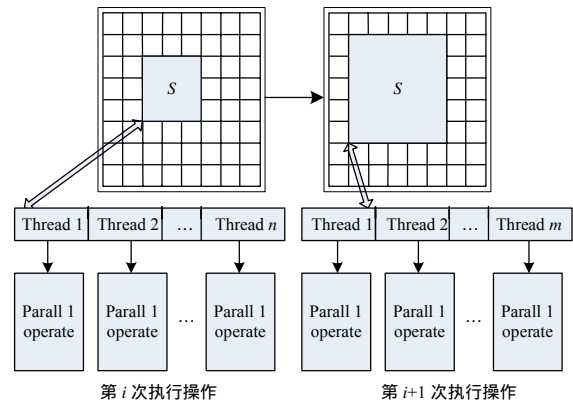


图 3 CUDA 搜索

对应每个线程操作为判断线程索引 (x, y, z) 与起始点距离是否小于波的传递范围，若小于传递范围则取相应路径数据场数据作计算及赋值操作，计算该位置往其相邻位置传递的代价函数，若小于原位置则替换原位置数据。

(3) 搜索路径获得

将路径数据场数据运用内存拷贝函数 *cudaMemcpy* 由设备内存拷贝至主机内存，获取路径具体步骤为建立路径链表 *Path* 将目标点放入 *Path*，寻找链表首个节点相邻位置点路径数据值最小的位置，将其压入链表，从链表首个节点寻找下一个首节点，直到找到起始点。

2.5 整体航路优化

对于已生成的初始航路，由于搜索自由度限制使得其航向间转角只能为 45° 或 45° 的倍数，从而限制了得出航路的航程和其转向方向，不能满足以实际为准的航路规划需求。在此对初始航路进行一定的优化处理。

基本原理为将航路 AB 沿 B 向航路点 C 扩展。在扩展航线 AE 中经过栅格的属性相同，直到碰到另一属性栅格，得到相邻的栅格 D，则 D、E 为新的航路点。再将航路 DE 沿 E 向航路点 C 扩展，直至到达目标点。

通过以上优化处理得出的航路更接近于实际需求的航路，减少了航向改变频率和变化的幅度，具有较好的效果。航路的航线优化如图 4 所示。

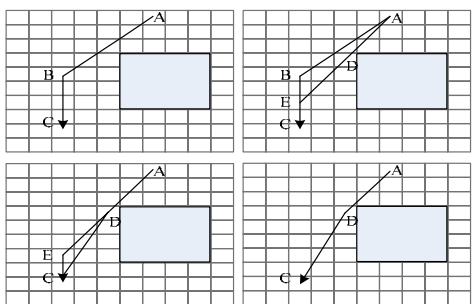


图4 航路的航线优化

3 仿真实验

本文以 VC.Net 为编程环境。所有程序在一台显卡为 GeForce 8600 GT 的 PC 机上运行。声场数据使用广义相积分 (WKBZ)方法经相应插值得到大小为 129 km×129 km×6.4 km 的数据,水平分辨率为 500 m,垂直分辨率为 50 m。实验选取了 5 个不同距离的起始点和结束点组。每组通过分别运用直接方法和 CUDA 并行方法计算消耗时间。计算时间消耗对比如表 1 所示。

	50 km	80 km	120 km	160 km	200 km
原方法	13.2	23.5	34.7	42.9	65.8
CUDA 方法	0.7	1.1	1.8	2.9	4.5

由表 1 可知,该方法与原方法相比效率上有很大提高。在一定范围内该方法效率与范围的增大相对应,在范围较大时效率有所降低。在判断位置点与起始点距离以决定是否进行计算赋值时,对一些已稳定的位置进行重复计算,在范围较大时得到体现。

规划得到的声场环境下的航路规划结果如图 5 所示。由

此可知,航路从起始点开始尽量避免威胁较大的区域,通过威胁较小的区域最终到达目标点。

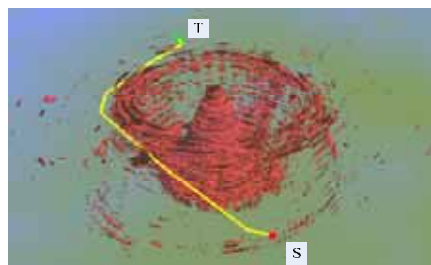


图5 航路规划结果

4 结束语

本文采用三维声场环境下基于 CUDA 的航路规划,在航路规划时使用一定的并行方法以加速航路搜索,取得了较好的效果。该方法同样适用于其他三维数据场中的路径规划问题。

参考文献

- [1] 冯琦. 军用无人机发展趋势[J]. 光电与控制, 2003, 10(2): 9-13.
- [2] 吕太之. 基于改进概率分解的路径规划方法[J]. 计算机工程, 2007, 33(21): 160-165.
- [3] 张仁和. 水平不变海洋声道中的 WKBZ 简正波方法[J]. 声学学报, 1994, (1): 1-13.
- [4] Clara S. NVIDIA CUDA Compute Unified Device Architecture: Programming Guide Version 1.1[EB/OL]. (2008-07-22). <http://developer.download.nvidia.com/compute/cuda>.
- [5] 徐志鹏. 基于图形硬件的批量式 SOM 学习过程[J]. 计算机工程, 2005, 31(17): 15-17.

编辑 陆燕菲

(上接第 229 页)

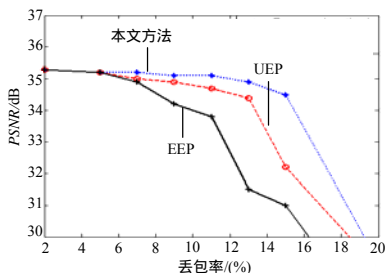


图8 GOP Length=8 的 Hall 序列仿真结果

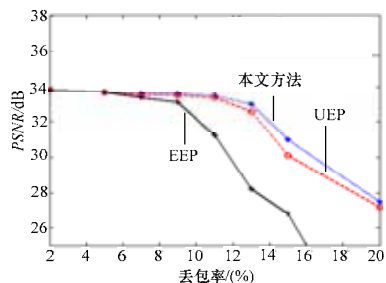


图9 GOP Length=8 的 Foreman 序列仿真结果

5 结束语

本文对 SVC 不同增强层比特流进行分析,按其对于重建图像的重要性将增强层比特流分为 3 类: A 类包括最低时域层的增强层数据; B 类包括信噪比第一增强层数据和第二时域

层增强层数据; C 类数据包括信噪比第二增强层的数据和最高时域层的增强层数据,并对这 3 类数据采用了不同冗余度的前向纠错保护。实验表明,本文提出的 H.264 SVC 编码视频数据的非均匀误码保护方案与传统 EEP 及现有 UEP 保护方法相比,可分别获得 2 dB 和 0.5 dB 的 PSNR 增益,具有较好的传输鲁棒性。

参考文献

- [1] Reichel J, Schwarz H, Wien M. Scalable Video Coding: Working Draft 1[C]//Proc. of JVT'05. Hong Kong, China: [s. n.], 2005.
- [2] 段青青, 宋学瑞. 一种 H.264/AVC 中的快速运动估计算法[J]. 计算机工程, 2008, 34(16): 244-246.
- [3] Liebl G, Schierl T, Wiegand T, et al. Advanced Wireless Multiuser Video Streaming Using the Scalable Video Coding Extension of H.264/MPEG4-AVC[C]//Proc. of ICEM'06. Toronto, Ontario, Canada: [s. n.], 2006.
- [4] Schierl T, Schwarz H, Marpe D, et al. Wireless Broadcasting Using the Scalable Extension of H.264/AVC[C]//Proc. of IEEE International Conference on Multimedia and Expo. Amsterdam, Netherlands: IEEE Press, 2005.
- [5] Cheng Lianji, Zhang Wenjun, Chen Li. Rate Distortion Optimized Unequal Loss Protection for FGS Compressed Video[J]. IEEE Trans. on Broadcasting, 2004, 50(3): 126-131.

编辑 张帆

