

基于 GPU 的遥感图像融合并行算法研究

赵进¹, 刘昌明¹, 宋峰², 张丽萍³

(1. 海军计算技术研究所, 北京 100086; 2. 68310 部队, 陕西 西安 710600;
3. 91731 部队, 北京 102200)

摘要: 基于通用 GPU 并行计算技术, 结合遥感图像数据融合处理特点, 利用 NVIDIA 公司的 CUDA 编程框架, 在其 GPU 平台上对 BROVEY 变换和 YIQ 变换融合算法进行了并行研究与实现。实验结果表明, 随着遥感图像融合算法的计算复杂度、融合处理的问题规模逐渐增加, GPU 并行处理的加速性能优势也逐渐增大, GPU 通用计算技术在遥感信息处理领域具有广阔的应用前景。

关键词: 遥感图像融合; GPU; 并行; CUDA; 优化

中图分类号: TP391

文献标识码: A

文章编号: 1674-7720(2013)06-0035-03

Study of remote sensing image fusion parallel algorithms based on GPU

Zhao Jin¹, Liu Changming¹, Song Feng², Zhang Liping³

(1. Navy Institute of Computing Technology, Beijing 100086, China;

2. PLA of 68310, Xian 710600, China;

3. PLA of 91731, Beijing 102200, China)

Abstract: Based on the GPU parallel computing technology, this article combined remote sensing image data fusion processing characteristics using NVIDIA's CUDA (Compute Unified Device Architecture) programming framework with BROVEY and YIQ fusion algorithm. Experimental results show that the computational complexity of the remote sensing image fusion algorithm and the size of the problem increase, the accelerated performance advantage of GPU increases. This study shows that the computing technology of GPU has broad application prospect in the field of remote sensing image processing.

Key words: remote sensing image fusion; GPU; parallel; CUDA; optimization

随着遥感传感器技术的发展, 获取遥感数据的方式已经由单一可见光模式发展成多种传感器模式。每一种传感方式获取的数据信息相对单一, 融合不同传感方式获取的数据信息可以有效提高数据信息量, 提供更为精确的遥感图像信息。

POHL C 和 GENDEREN J L^[1]对图像融合给出了如下定义: 通过特定算法将两幅或多幅图像合成一幅新的图像。图像融合可以分为像素级、特征级和决策级 3 个层次, 其中像素级融合需要处理的数据量最大, 计算过程也最复杂。随着遥感图像的时间、空间和光谱分辨率逐步提高, 融合算法处理结果的精度要求同时在逐步提高, 导致图像融合的处理速度需求逐步增大, 计算量大、计算过程复杂的遥感图像融合在处理速度上面临着新的问题与挑战。近几年来, 基于 GPU 的异构平台在通用计算领域得到快速发展, 已经在许多方面得到了有效应用^[2-4], 为遥感图像融合快速处理技术研究提供了新的思路。

《微型机与应用》2013 年 第 32 卷 第 6 期

结合 CPU-GPU 异构平台面向通用计算领域的性能优势, 针对遥感图像融合面临的处理速度问题, 通过分析图像融合处理过程的特点, 将像素级融合的 BROVEY 变换和 YIQ 变换融合算法在基于 GPU 平台上进行了并行研究与实验, 获得了突出的性能。

1 研究背景

1.1 CUDA 编程模型

NVIDIA 公司的 CUDA^[5]语言是基于 C 语言的扩展, 主要使用 API 调用底层功能进行处理与计算, 使熟悉 C 语言的编程人员能够快速运用 CUDA 开发通用计算程序。在 CUDA 结构中, CPU 端被称为 Host, GPU 端被称为 Device, 采用 SIMT(Single Instruction Multiple Thread)模式执行程序^[5]。用户把可以放在 GPU 上并行执行的程序组织称为 Kernel 内核程序。在 CPU 端执行的程序称为 Host 宿主程序, 控制 Kernel 的启动、加载或保存与 GPU 的通信数据, 以及执行部分的串行计算。Device 端在执

欢迎网上投稿 www.pcachina.com 37

行时创建很多的并行线程 Thread, 线程组织成线程块 Block, 而 Block 再组成网格 Grid。每个 Thread 执行自己的程序 Kernel, Block 内的线程通过共享存储器(Shared Memory)分享数据。并行线程通过 GPU 上的众多计算内核实现并行处理, 实现整个程序的性能加速^[5]。

1.2 两种典型的像素级融合算法

遥感图像融合在遥感图像配准基础上进行, 待融合的两幅遥感图像具有相同大小, 每个像素点的融合或是独立执行, 或者只需要少量周围像素点数据协助执行。

1.2.1 BROVEY 变换融合算法

BROVEY 变换融合算法^[6]是一种基于色度的变换, 属于彩色空间的方法。它是将多光谱的色彩空间进行分解得到色彩与亮度, 再用其全色图像进行计算。这种算法像素点彼此单独执行, 具有良好的并行性, 算法执行简单, 在保留了较多光谱信息的同时, 也具有较快的处理速度。

1.2.2 YIQ 变换融合算法

YIQ 颜色系统是一种用于电视信号传输 NTSC 制式的彩色编码系统。其中, Y 对应亮度信息, I、Q 分量反映图像与硬件相关的彩色信息。YIQ 变换融合算法^[7]属于彩色空间的方法, 计算较为复杂, 像素点在变换到 YIQ 分量后, 需要使用直方图匹配, 具有一定的并行性。

2 并行设计及优化实现

2.1 优化策略

2.1.1 数据异步传输

遥感图像融合 CUDA 并行计算程序在 CPU-GPU 异构平台上执行, 该平台上 CPU 与 GPU 工作任务不同。CPU 为 GPU 准备执行数据, 同时执行逻辑控制等复杂事务。CPU 加载数据时, 采用同步方式启动传输指令; 传递数据时, CPU 处于空闲状态, 计算资源不能得到充分利用。优化时 CPU 采用异步方式传输数据, 启动传输指令后直接进行后续事务处理, 使 CPU 执行与加载数据重叠, 提高资源利用率。

2.1.2 访存优化

GPU 线程对全局存储器(Global Memory)进行一次访存需要 400~600 个时钟周期, 对共享存储器(Shared Memory)、寄存器(Register)等快速存储部件访存一次只需要 4 个左右的时钟周期。优化时充分利用 GPU 显存中的多层次存储部件, 发挥快速存储部件读取数据优势, 最大化提高执行性能。

2.2 融合算法的 CUDA 并行实现

综合融合算法 CUDA 优化策略, 进行融合算法 CUDA 程序并行设计与实现时, 重点研究以下问题。

2.2.1 线程块参数与网格参数的设置

CPU-GPU 异构计算模式中, 每个线程块只能拥有有限线程数, 每个流处理单元 SP 最多可同时执行 256 整数倍线程, 根据遥感图像数据以二维方式组织的特点, 将块 Block 设置为 16×16 二维形式, 每个块有 256 个

线程, 使每个 SP 单元能够满负载执行, 最大限度地提高资源利用率。设计实现时, 线程网格大小的设置与图像大小有关, 网格的宽度设置成 $(\text{imagewidth} + \text{dimBlock}.x - 1) / \text{dimBlock}.x$, 高度设置成 $(\text{imageheight} + \text{dimBlock}.y - 1) / \text{dimBlock}.y$, 保证线程块是一个整数。线程块规模与遥感图像规模有关, 与 GPU 处理核心数量无关。这样同一问题能够在不同型号的 GPU 上执行, 形成 CUDA 并行程序良好的可移植性。

2.2.2 线程网格规模大于图像规模

令 i 为线程网格宽度索引号, j 为线程网格高度索引号, imagewidth 为图像宽度, imageheight 为图像高度。通过加入 $\text{if}(i < \text{imagewidth} \&\& j < \text{imageheight})$ 语句控制线程执行对应像素点。空执行线程最多可以为 $255 \times I + 255 \times J - 255 \times 255$ 个, 其中, I 是线程网格的宽度, J 是线程网格的高度。与具有的大量执行线程相比, 空执行线程占很小比例, 几乎不需要执行时间。与并行加速获取的提升时间相比, 空执行少量线程浪费的时间可以忽略。

3 实验与结果分析

实验硬件为: Inter (CR) Core (TM) i5 四核 2.67 GHz CPU, 2 GB 内存, NVIDIA GeForce GTX 460 GPU, 1 GB 显存, 336 个 SP 计算单元。操作系统为 Linux Ubuntu 10.10, 串行程序为 C 语言实现的标准程序。融合的图像是 IKONOS 扫描北京故宫的遥感卫星图像。

对两种融合算法的串行程序和实现的 CUDA 并行程序分别使用不同尺寸遥感图像进行了测试。令 CPU 执行融合算法标准串行程序时间为 T_{CPU} , CPU-GPU 标准加速平台上总执行时间分为两个部分, 一部分是 GPU 设备初始化和数据传输等额外总时间 T_{allocate} , 另一部分是 GPU 计算执行时间 T_{execute} 。对同一规模的图像, 定义总时间 total 加速比 $\text{speedup}_{\text{total}} = \frac{T_{\text{CPU}}}{T_{\text{allocate}} + T_{\text{execute}}}$, 定义执行时间 execution 加速比 $\text{speedup}_{\text{execute}} = \frac{T_{\text{CPU}}}{T_{\text{execute}}}$ 。图 1 用柱状图展示了实现的两种算法加速比具体情况。

从实验结果可以看出, 这两种算法的 $\text{speedup}_{\text{total}}$ 都随着图像规模的增加而增加, 计算复杂度高的算法表现趋势更突出。在小规模图像中出现了 CPU-GPU 并行执行总时间大于 CPU 串行执行时间 $\text{speedup}_{\text{execute}}$, 除 YIQ 算法在 4096×4096 图像有所回落外, 其余都随着图像规模的增加而增大。导致这样的现象有以下几点原因。

(1) CPU-GPU 异构计算平台拥有更多的计算资源, 使 $T_{\text{execute}} < T_{\text{CPU}}$ 。

(2) 图像规模与算法计算复杂度都较小时, $T_{\text{allocate}} > (T_{\text{CPU}} - T_{\text{execute}})$, 出现 CPU-GPU 异构平台并行执行时间大于 CPU 串行执行, 性能下降。随着图像规模与算法计算复杂度某一项或同时增加出现 $T_{\text{allocate}} < (T_{\text{CPU}} - T_{\text{execute}})$, 总执行时间在 CPU-GPU 异构平台上得到加速。

(3) 在相同的计算复杂度下, 图像规模使 GPU 计算

《微型机与应用》2013 年 第 32 卷 第 6 期

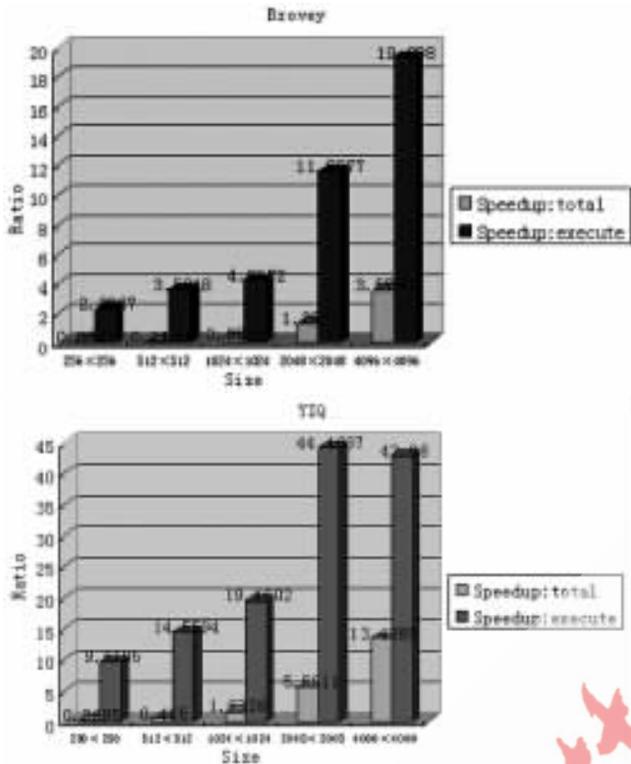


图 1 性能加速比

核心执行达到饱和时, speedup_{execute} 趋于稳定甚至下降。但因为计算资源的优势, 总加速性能仍有提升。

对于融合算法 CUDA 的并行执行, 定义为:

$$P_{execute} = T_{execute} / (T_{execute} + T_{allocate})$$

$$P_{allocate} = T_{allocate} / (T_{allocate} + T_{execute})$$

两种融合 CUDA 程序中 $T_{allocate}$ 与 $T_{execute}$ 的关系如图 2 所示。

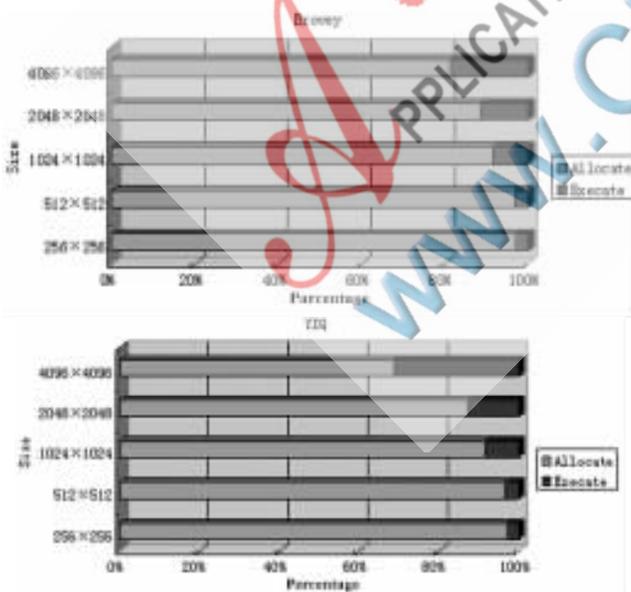


图 2 分配显存与执行时间所占比例

从图中可以看出, 图像规模(如 256×256 和 512×512)小时, $P_{execute}$ 小, 算法计算复杂度的不同导致加速比出现

不同情况。因此, 为进一步提升加速性能, 融合算法 CUDA 并行执行需要最大化地提高 $P_{execute}$ 。

从实验结果可以得到, 随着算法计算复杂度与问题规模的增大, 加速性能逐渐增加。实验表明, GPU 并行处理能够很好地应用于遥感图像融合的算法, 将 GPU 应用到遥感加速处理领域具有很好的应用前景。

本文从遥感图像融合处理背景出发, 阐述了研究融合处理加速的意义, 然后针对近些年兴起的 CPU-GPU 异构加速平台, 对 BROVEY 变换和 YIQ 变换遥感图像融合算法进行了 CUDA 并行优化实现。研究表明, GPU 通用计算技术在遥感图像融合领域具有广阔的应用前景。

参考文献

- [1] POHL C, Van Genderen J L. Multisensor image fusion in remote sensing: concepts, methods and applications[J]. International Journal of Remote Sensing, 1998, 19(5): 823-854.
- [2] OWENS J, LUEBKE D, GOVINDARAJU N, et al. A survey of general-purpose computation on graphics hardware[J]. Computer Graphics Forum, 2007, 26(1): 80-113.
- [3] GARLAND M, GRAND S, NICKOLLS J, et al. Parallel computing experiences with CUDA[J]. IEEE Micro, 2008, 28(4): 13-27.
- [4] LAHABAR S, AGRAWAL P, NARAYANAN P J. High performance pattern recognition on GPU[J]. National Conference on Computer Vision Pattern Recognition Image Processing and Graphics, 2008: 154-159.
- [5] NVIDIA. NVIDIA CUDA compute unified device architecture, Programming Guide, Version 2.0. NVIDIA, 2008. [EB/OL]. http://www.nvidia.com/object/cude_home_new.html.
- [6] GILLESPIE A R, KAHLE A B, WALKER R E. Color enhancement of highly correlated images-II. channel ratio and chromaticity transformation techniques[J]. Remote Sensing of Environment, 1987, 22: 343-365.
- [7] Dong Guangjun, Huang Xiaobo, Dai Chenguang. Comparison and analysis of fusion algorithms of high resolution Imagery[C]. International Symposium on Photoelectronic Detection and Imaging: Related Technologies and Applications, Proceedings of SPIE, 2008, 6625(66250H).

(收稿日期: 2013-01-21)

作者简介:

赵进, 男, 1986 年生, 硕士研究生, 助理工程师, 主要研究方向: 信息安全、遥感图像处理。

刘昌明, 男, 1971 年生, 本科, 工程师, 主要研究方向: 信息安全。

宋峰, 男, 1974 年生, 本科, 工程师, 主要研究方向: 信息安全。