高分辨率遥感图像配准并行加速方法

郝昀超 王显珉

(北京航空航天大学计算机学院数字媒体北京市重点实验室 北京 100191)

摘 要 基于SIFT算法的遥感图像配准精度高、稳定性强,但图像幅宽大、提取特征点数量多使得配准过程耗时长。提出了一种高分辨率遥感图像配准的并行加速方法。该方法在特征点提取时利用GPU实现了高斯金字塔建立过程中的并行加速,并对提取出的大量特征点使用共享内存来进行局部极值高速缓存,降低了特征点提取所需的运算时间;同时通过分块处理以及OpenMP多线程技术实现了特征点匹配及仿射模型计算过程的CPU并行处理。实验表明:本方法相对于传统的SIFT算法平均加速3倍,并且对于固定大小的图像,本方法的特征点提取时间和特征点个数具有线性关系,加速比随着提取出特征点数量的增加而增大。

关键词 GPU,遥感图像,SIFT,配准

中图法分类号 TP274.2 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.9.006

Parallel Acceleration Method for Very High Resolution Remote Sensing Image Registration

HAO Yun-chao WANG Xian-min

(Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

Abstract The method for remote sensing image registration based on scale-invariant feature transform(SIFT) has the advantage of hig haccuracy and good stability. However, the method is very time-consuming because of the large size of the image and the huge quantity of feature points. This paper presented a parallel acceleration method for very high resolution remote sensing image registration which builds the Gaussian pyramid by hardware implementation on GPU. We used the shared memory to cache the temporary extremum at high speed when identifying the keypoint, which effectively decreases the time for the keypoint extraction. Meanwhile, we divided the whole image into blocks and used OpenMP to match the feature-points and build parallel acceleration of the affine model. Compared with the traditional registration method——SIFT, this method is 3 times faster. We concluded that the runtime of the keypoint extraction has linear relationship with the quantity of the keypoints, and the acceleration ratio raise with the density of the keypoints going up.

Keywords GPU, Remote sensing image, SIFT, Registration

1 引言

遥感图像的配准是遥感图像匹配、变化监测、多光谱融合等应用的关键步骤[1]。随着技术的进步,遥感图像分辨率越来越高,我国高分辨率遥感卫星分辨率已经达到分米级。由于图像数据量的增加,如何快速进行高分辨率遥感图像的配准成为关键问题。

SIFT算子的提出,使得遥感图像自动配准成为可能[2]。但是由于SIFT算法的计算复杂度高,高分辨率遥感图像的图像数据量较大,普通的SIFT算法需要的时间过长,尤其是高分辨率遥感影像的特征点往往密集,因此普通SIFT算法不适合快速配准。

目前,许多学者对基于SIFT的配准算法进行了改进,文献[3]提出先利用GPU加速的SIFT算法进行粗配准,再利用ORB算法进行精配准[3]。文献[4]通过比较对应点对的坐标,剔除掉有异常的坐标差的图像,提高了配对点精度[4]。这些方法均考虑到了特征点稳定性和运算时间的问题,但没有考虑高分辨率图像的巨大时耗问题,因此只可以运行较小的图片,对于较大的图片,运算时间呈几何倍数增加,难以满足快速配准[5]。文献[10]利用GPU加速的小波分析进行遥感图像配准,该方法由于较好的存储优化策略,加速效果十分明显;但没有针对宽幅遥感图像进行讨论。

针对上述问题,本文提出了一种高分辨率遥感图像配准的并行加速方法,该方法可以保证在遥感图像中提取出大量的SIFT特征点,并有较高的配准速度;此外,在特征点密集的情况下仍可以保持较快的运算速度。

2 并行加速方法及SIFT配准算法概述

2.1 统一计算设备架构

统一计算设备架构(Compute Unified Device Architec-

到稿日期:2014-06-20 返修日期:2014-08-12

郝昀超(1990—),男,硕士生,主要研究方向为并行计算、遥感图像处理,E-mail:luckiehao@gmail.com;王显珉(1984—),男,博士生,主要研究方向为人工智能、遥感图像处理。

ture,CUDA)是 NVIDIA 公司生产的用于显卡开发的通用计算体系。NVIDIA 将其设备架构描述为单指令多线程(Single Instruction Multiple Thread,SIMT),这种设备拥有复杂的内存结构体系和多种内存链接机制,为程序提供最大化的带宽支持[7]。CUDA 的特殊架构使得它十分有利于处理图像数据。由于带宽是影响 CUDA 处理数据的瓶颈,如何优化使其影响最小成为关键。

在 CUDA 计算体系结构中,执行资源被组织成多核流处理器(Streaming Multiprocessor,SM)。warp 是 SM 中的线程调度单元。分配给某个 SM 的线程块将进一步划分成由 32 个线程组成的 warp 单元。warp 的调度是影响处理性能的关键,GPU 全局储存器访问的时延较长,所以 CUDA 利用执行延迟较短的线程来覆盖全局储存器访问延迟较长的线程,这样的调度机制被称为"延迟隐藏"。这种机制使得 GPU 没有像 CPU 那样引入大量的缓存和分支预测机制,使得 GPU 不适合较多的分支运算。SIFT 特征点提取时基本为逐点运算,有利于 GPU 运算,而 KD 树算法分支运算较多,本文提出利用 CUDA 进行 SIFT 特征提取;而 KD 树和 RANSAC 等步骤利用 OpenMP 分块并行化,充分利用硬件资源。

2.2 OpenMP

OpenMP 是由 OpenMP Architecture Review Board 提出的用于共享内存并行系统的多线程程序设计的一套指导性的编译处理方案,其因易用性而被广泛接受。OpenMP 通过高层抽象描述对程序进行并行化,编程人员只需要通过 pragma 指明自己的意图,由此编译器可以自动将程序进行并行化,并在必要之处加入同步互斥以及通信。通过将图像分块,并对每一块都分配一个线程进行处理,以达到块间并行[6]。本文将耗时占 80%的 SIFT 算法部分进行 CUDA 并行化处理,之后的特征点匹配和仿射变换采用 OpenMP 多线程处理,以充分利用硬件资源。

2.3 SIFT 算法概述

SIFT(Scale-Invariant Feature Transform)方法的特点在于对特征点的旋转和尺度具有旋转不变性。这种不变性用于遥感图像时具有很高的稳点性、准确性以及辨识能力,使一个特征可以准确找到同名控制点,为仿射校正提供了基础。SIFT 算法的主要步骤为[7]:1)尺度空间极值探测;2)局部特征点定位;3)方向赋值;4)特征描述子生成;5)特征点匹配;6)仿射变换模型计算。

3 并行加速方法实现

3.1 方法概述

高分辨率遥感图像配准的并行加速方法的流程如图 1 所示。首先将参考影像和待配准影像读入数据集,并且做好分块处理;然后,利用流处理技术将分块数据集依次读入 GPU 内存,在核函数运行时同步进行传输,针对遥感图像特点及 SIFT 算法原理,利用 CUDA 的纹理特性对尺度空间构建进行并行加速,并对特征点计算的中间结果进行缓存;最后将特征点数据输出至主机端内存,利用 OpenMP 强大的多线程处理,将分块的数据传入,进行特征点匹配、异常匹配点排除和仿射变换。将矫正后的图像输出,完成配准任务。

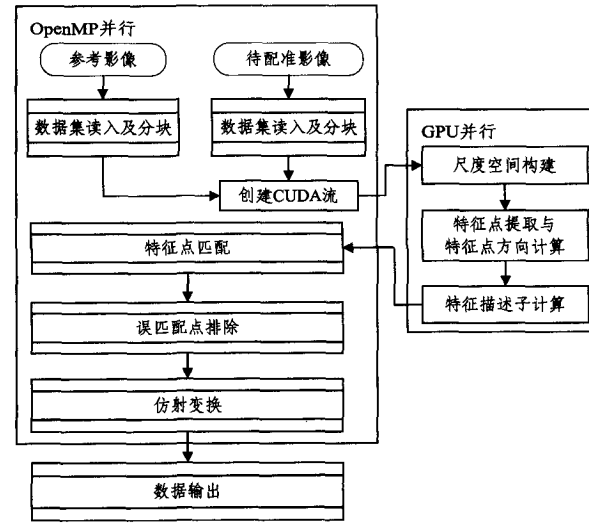

图 1 配准算法基本流程

3.2 数据集读入及分块

本方法根据原图像大小以及设备性能,将图像等分为相同大小的矩形,保证处理过程中有足够的主机内存和 GPU 显存。为保证在 GPU 上处理的线程控制以及保证从全局内存到共享内存传输的速度,本文将图像分成 1024×1024 大小的正方形块,存入固定内存中(Pinned Memory)。为保证算法一致性,在无法整除的边缘用 0 补齐。

3.3 创建 CUDA 流

CUDA 流是一系列按顺序执行的指令集合[9]。高分辨率遥感图像的数据量极大,GPU 的设备内存较小,数据如何高效传入和储存是制约 GPU 运算速度的关键。本文采用分块的处理方式,将宽幅遥感图像分成 GPU 易于处理的小块,再通过 CUDA 流处理,充分利用并行特性,减少 GPU 数据流的传输时间。为了不使用分支语句,消除幽灵元素。

由于设备差异较大,CUDA 流的数量是保证流处理高效运行的基础,本文定义了冗余时间变量 RT,表示在一个流处理结束到另一个新的流处理开始的空余时间,将 RT 表示为

$$RT=n\times\frac{S}{v_{H2D}}-t_{pro}+\frac{S_p}{v_{D2H}}$$

其中,n 表示块的个数,S 表示数据总量,v_{H2D} 表示主机端到设备端的数据传输速度,v_{D2H} 表示设备端到主机端的数据传输速度,t_{pro} 表示数据数据处理时间,S_p 是处理过的数据量。RT 由 n 倍输入图像的总量 S 除以从主机端到设备端的传输速度减去处理特征点的时间决定,这部分时间越短,效率越高,但是如果为负就会出现错误。在保证 RT 为正的前提下,可以保证流在比较理想的状态下进行工作,但是这部分时间与特征点处理的时间 P 的关系密切,而 P 由特征点的个数所决定,所以 RT 值在程序运行阶段是动态变化的。本文使用固定权值的方法保证 RT 为正。

3.4 尺度空间的构建

图像数据从主机内存传入设备内存,存放于全局内存中。由于纹理内存具有快速随机访问的特性,并且可以享受硬件加速的双线性差值,而构造尺度空间需要将原图像进行下采样,因此将这部分内存与纹理内存绑定,采取接合式访问。

由于高斯函数在 x 方向和 y 方向上的相关性为 1,因此

可以将二维空间卷积分解成两次一维卷积。通常用 $m\times m$ 的高斯核对大小为 $n\times n$ 的图像做卷积，需要 $n^2\times m^2$ 次运算；而如果将这次卷积分成行卷积和列卷积，则只需要 $2\times m\times n^2$ 次运算，可以大大减少时间复杂度，而且更加利于 warp 控制。为最大限度实现 warp 控制，并且高效处理所有像素，采用 128 线程循环处理的分配模式。

3.5 特征点提取与特征点方向计算

特征点提取的主要步骤是与高斯差分金字塔的同一层的 8 个相邻像素对比，再与上下两层的 18 个相邻像素对比。本文方法充分利用共享内存的高速访问机制，将比较大小的中间结果进行高速缓存，一方面可以提高数据的重利用率，另一方面减少了运算次数，具体过程如图 2 所示。

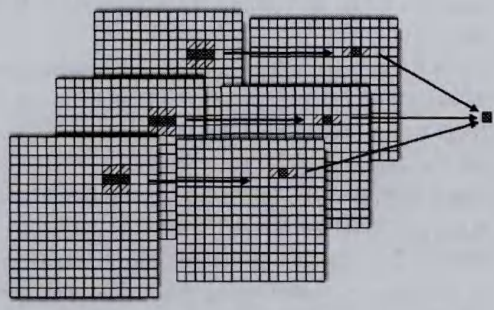

图 2 共享内存数据处理示意图

首先为高斯差分金字塔的每一个层建立一块共享内存空间，每个元素都与其上下两个元素进行比较，将结果放入另一块内存中，例如图 2 中，深色内存元素和上下两个带斜线元素进行比较，然后将结果存入另一块共享内存中，深色内存再与左右两个内存进行比较，将结果取出，相邻两层进行比较，得到最终的最大值或者最小值。串行算法中比较 27 个像素点的步阶复杂度为 27，而 CUDA 并行算法的步阶复杂度仅为 6，逻辑提速 78%。由于这种设计使得本文算法十分有利于特征点密集型图像的计算，因此本算法十分有利于遥感图像特征点提取，在特征点数量众多的图像上效果明显。

为提高编程效率，降低由于数据传输带来的不必要时间损耗，本文获得候选特征点后直接进行特征点排除，为每一个剩余的特征点分配一个线程，当点数多于线程数时做循环处理。本文针对每一个点均采用一个线程运算，线程之间无交流，最终使用线程同步进行统一对齐。将位置的尺度差值精确到亚像素精度，每一个线程进行 5 次迭代，与阈值进行比较，如果小于阈值，此线程进入下一次循环或者退出函数。下一步，每一个线程计算 Hessian 矩阵时，如果主曲率大于给定的阈值，则此线程进入下一次循环或者退出函数，直至所有线程都退出函数。

3.6 特征描述子计算

该步骤是所有 SIFT 特征计算最耗时的部分，所用时间与特征点数量严格正相关，本文采用为每一个特征点分配一个线程块的计算模式。根据 SIFT 方法，每个特征点由 128 维特征描述子描述，根据其论述[7]，SIFT 特征的 128 维的每一维都具有独立性，于是每一个线程块中有 128 个线程并行协同计算。每一维都独立进行直方图统计操作，最大限度地减小原子操作带来的性能损失。在 16×16 的空间上分配 128 个线程，同样需要迭代两次来运算。计算完成后，将带匹配的特征点传输至主机内存中，进行下一步处理。

3.7 特征点匹配

本文采用了构建 KD 树的匹配方式，由于希望 GPU 和 CPU 的协同工作能力达到最大，该步骤将放在 CPU 上处理。此步骤消耗时间与特征点的个数正相关。

3.8 误匹配点排除及仿射变换

由于对特征点匹配结果的密切依赖，此步骤也要在 CPU 上进行处理。为保证鲁棒性，本方法采用 RANSAC 方法计算仿射变化模型，将生成的结果串行写入文件数据中。分块处理流程只有在 GPU 计算平台上是逐块进行，其他步骤均为各个块间同时进行，如图 3 所示。

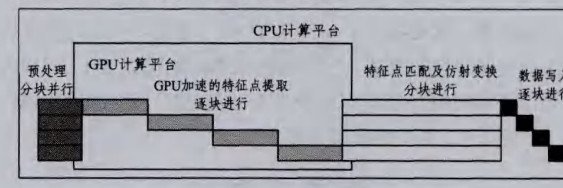

图 3 不同计算平台间并行分工

4 实验分析

本文选取 4 张高分辨率遥感影像作为测试图集，这 4 张分别为不同地物，分辨率均为 1m。表 1 分别记录了使用传统 SIFT 方法与本方法在相同的条件下运行测试图集的总体运行时间、特征点提取时间以及特征点提取数量 3 项指标，加速比定义为传统方法与本方法的比值。对比实验平台均为：Intel Core i5 处理器(主频 3.3GHz)，NVIDIA GTX630 显卡(流处理器频率1620MHz，流处理器单元(SP) 96 个)，4GB 内存。

表 1 实验结果

编号		1	2	3	4
地貌		荒地	港口	山区	城市
尺寸		17184×15893	12280×12340	12248×12340	12244×12340
总体运行时间	传统 SIFT 方法(s)	438.4	175.9	315.7	442.0
	本方法(s)	156.5	63.4	110.9	115.1
	加速比	2.80	2.77	2.85	3.84
特征点提取时间	传统 SIFT 方法(s)	323.2	130.8	231.0	360.4
	本方法(s)	41.3	18.3	26.2	33.5
	加速比	7.83	7.15	8.81	10.75
特征点提取个数	传统 SIFT 方法	245322	89217	176664	253084
	本方法	245340	89216	176681	253092
	误差比(%)	0.003	0.001	0.009	0.003

总体运行时间的实验结果如表 1 所列。结果表明：

(1)在保证特征点提取数基本一致的情况下，本方法与传统 SIFT 方法的加速比大概是 3 左右，在提取特征点部分，加速可达到 7～10。误差主要由浮点数的计算特性引起。

(2)加速效果与特征点密集程度有关，比如港口图大部分是海水，特征点较少，加速比为 4 幅图中最小的，而城市图像拥有特征点的密度是最高的，所以具有最高的加速比。

本文进行了另一组实验，将每个图像分成 1000×1000 的数据块，记录每块提取出的特征点个数与运行时间，并绘制成散点图，如图 4 所示。

从图 4 中可以看出，此步骤消耗的时间与特征点个数正相关，主要是由于最后一步将每一个特征点分配给一个线程

块进行计算，相同特征点个数会反映出不同的计算时间，时间较低的为特征点密集型的图像。

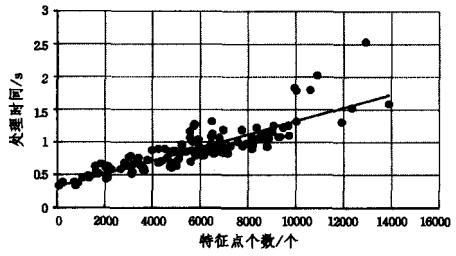

图 4　计算时间与特征点数目散点图

结束语　本文提出了一种基于 SIFT 的高分辨率遥感图像的配准并行加速算法，分别利用 CUDA 架构以及 OpenMP 多线程技术对特征点提取以及匹配过程进行了加速优化，提高了传统 SIFT 算法的配准速度。

参 考 文 献

[1] Zitova B, Flusser J. Image registration methods: a survey[J]. Image and Vision Computing, 2003, 21: 997-1000

[2] Li Qiao-liang, Wang Guo-you, Liu Jian-guo. Robust scale-invariant feature matching for remote sensing image registration[J]. IEEE Geoscience and Remote Sensing Letters, 2009, 6(2): 187-291

[3] Zhang Yun-sheng, Zhou Pei-long, Ren Yue, et al. GPU-accelerated large-size VHR images registration via coarse-to-fine matching[J]. Computers and Geosciences, 2014, 66: 54-65

[4] 雷小群，李芳芳，肖本林. 一种基于改进 SIFT 算法的遥感影像配准方法[J]. 测绘科学，2010，35(3)：143-145
Lei Xiao-qun, Li Fang-fang, Xiao Ben-lin. A registration method of RS image based on improved SIFT algorithm[J]. Science of Surveying and Mapping, 2010, 35(3): 143-145

[5] Kirk D B, Wen-mei W. Programming massively parallel processors: a hands-on approach[M]. Morgan Kaufmann, 2010

[6] Dagum L, Menon R. OpenMP: an industry standard API for shared-memory programming[J]. Computational Science & Engineering, 1998, 5(1): 46-55

[7] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110

[8] CUDA C Programming Guide[OL]. http: docs. nvidia. com/cuda/cuda-C-programming-guide/#axzz3iTPutLEx

[9] Nvidia CUDA Computer Unified Device Architecture[S]. Programing Guide, Version 2. 0 beta 2, 2008

[10] 周海芳，赵进. 基于 GPU 的遥感图像配准并行程序设计与存储优化[J]. 计算机研究与发展，2012，49(S1)：281-286
Zhou Hai-fang, Zhao Jin. Parallel programming design and storage optimization of remote sensing image registration based on GPU[J]. Journal of Computer Research and Development, 2012, 49(S1): 281-286

(上接第 23 页)

参 考 文 献

[1] Lane N D, Miluzzo E, Lu H, et al. A survey of mobile phone sensing[J]. Communications Magazine, IEEE, 2010, 48(9): 140-150

[2] Weiser M. The computer for the 2 1 st century [J]. Scientific american, 1991, 265(3): 94-104

[3] Campbell A, Choudhury T, Hu S, et al. NeuroPhone: brain-mobile phone interface using a wireless EEG headset[C]// Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds. ACM, 2010: 3-8

[4] Poh M Z, Kim K, Goessling A D, et al. Heartphones: Sensor earphones and mobile application for non-obtrusive health monitoring[C]// International Symposium on Wearable Computers, 2009(ISWC'09). IEEE, 2009: 153-154

[5] Miluzzo E, Wang T, Campbell A T. Eyephone: activating mobile phones with your eyes[C]// Proceedings of the Second ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds. ACM, 2010: 15-20

[6] Nanayakkara S, Shilkrot R, Yeo K P, et al. EyeRing: a finger-worn input device for seamless interactions with our surroundings[C]// Proceedings of the 4th Augmented Human International Conference. ACM, 2013: 13-20

[7] Probst K, Perteneder F, Leitner J, et al. Active office: towards an activity-promoting office workplace design[C]// Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts. ACM, 2012: 2165-2170

[8] Berkhoff C, Ochoa S F, Pino J A, et al. Clairvoyance: A framework to integrate shared displays and mobile computing devices [J]. Future Generation Computer Systems, 2014, 34: 190-200

[9] Polli A M, Korn M, Nylandsted Klokmose C. Local area artworks: collaborative art interpretation on-site[C]// Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication. ACM, 2013: 79-82

[10] Xively[OL]. https://xively. com/

[11] Open. Sen. Se[OL]. http://open. sen. se/

[12] 孙效华，冯泽西. 可穿戴设备交互设计研究[J]. 装饰，2014(2)：28-33
Sun Xiao-hua, Feng Ze-xi. Interaction Design for Wearable Devices[J]. Zhuang Shi, 2014(2): 28-33

[13] 郑能干，吴朝晖，林曼，等. 电子织物研究进展[J]. 计算机学报，2011，34(7)：1172-1187
Zheng Neng-gan, Wu Zhao-hui, Lin Man, et al. A Survey on electronic textiles[J]. Chinese Journal of Computer, 2011, 34(7): 1172-1187

[14] 徐强，孙建华. 可穿戴多人协同支撑软件系统研究[J]. 电子科技大学学报，2010(S1)：57-60
Xu Qiang, Sun Jian-hua. Research on Wearable Computing in the Multi-collaborative of System[J]. Journal of University of Electronic Science and Technology of China, 2010(S1): 57-60

[15] Brunette W, Sodt R, Chaudhri R, et al. Open data kit sensors: a sensor integration framework for android at the application-level [C]// Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services. ACM, 2012: 351-364

[16] Fielding R T. Architectural styles and the design of network-based software architectures[D]. University of California, 2000

[17] OAuth[OL]. http://oauth. net/

[18] UPnP Device Architecture V1. 1[OL]. http://upnp. org/specs/arch/UPnP-arch-DeviceArchitecture-v1. 1. pdf

[19] Cling[OL]. http://4thline. org/projects/cling