

基于 CPU/GPU 异构模式的高光谱遥感影像 数据处理研究与实现

汤媛媛^{1,2} 周海芳¹ 方民权¹ 申小龙¹

(国防科学技术大学计算机学院 长沙 410073)¹ (中国人民武装警察部队黄金地质研究所 廊坊 065000)²

摘要 近年来,基于 GPU 的新型异构高性能计算模式的蓬勃发展为众多领域应用提供了良好的发展机遇,国内外遥感专家开始引入高性能异构计算来解决高光谱遥感影像高维空间特点所带来的数据计算量大、实时处理难等问题。在此简要介绍了高光谱遥感和 CPU/GPU 异构计算模式,总结了近几年国内外基于 CPU/GPU 异构模式的高光谱遥感数据处理研究现状和问题;并面向共享存储型小型桌面超级计算机,基于 CPU/GPU 异构模式实现了高光谱遥感影像 MNF 降维的并行化,通过与串行程序和共享存储的 OpenMP 同构模式对比,验证了异构模式在高光谱遥感处理领域的发展潜力。

关键词 高光谱遥感, CPU/GPU, OpenMP, MNF

中图分类号 TP391 文献标识码 A DOI 10.11896/j.issn.1002-137X.2016.2.010

Hyperspectral Remote Sensing Image Data Processing Research and Realization Based on CPU/GPU Heterogeneous Model

TANG Yuan-yuan^{1,2} ZHOU Hai-fang¹ FANG Min-quan¹ SHEN Xiao-long¹

(School of Computer Science, National University of Defense Technology, Changsha 410073, China)¹

(Gold Geological Institute of CAPF, Langfang 065000, China)²

Abstract In recent years, the development of new high-performance heterogeneous computing based on GPU provides good opportunities in many application areas. Domestic and foreign remote sensing experts have started to introduce it to solve the issues like computation intensive and difficult real-time processing caused by high-dimensional space features of hyperspectral image. In this brief introduction to hyperspectral remote sensing and CPU/GPU heterogeneous computing model, we summarized hyperspectral data processing status and problems based on CPU/GPU heterogeneous pattern in recent years, and for small desktop supercomputer with shared storage, realized parallelization of hyperspectral imaging MNF dimensionality reduction on CPU/GPU heterogeneous model, and verified the development potential of heterogeneous patterns in the field of hyperspectral remote sensing processing by contrasting with the sequential program and OpenMP.

Keywords Hyperspectral remote sensing, CPU/GPU, OpenMP, MNF

1 引言

作为新型综合对地观测技术的高光谱遥感(Hyperspectral Remote Sensing),以能够鉴别物质分子及原子结构的测谱学为基础,融合了多门学科。它多达成百上千的光谱通道和纳米级别的光谱分辨率,能够为每个像素端元提供几乎与地面实测相同的一条精细、完整且连续的光谱曲线,是遥感发展新的里程碑^[1]。高光谱遥感使在多光谱遥感中无法探及的物质无处遁形,同时让遥感应用从定性分析转向定量分析,可以说它的出现和发展是遥感学科的一场革命,自兴起以来已经在森林、大气、地质、农业、海洋和军事等多个应用领域得到了广泛的研究和应用,有着广阔的发展前景和深远意义。然而它在提供高质量、高维度观测数据的同时,也带来了数据的

冗余度高、数据存储量大、数据计算量大、处理时间长和 Hughes 现象(维数灾难)等难题。

并行计算是有效提高处理速度和解决计算密集型问题的途径之一。高光谱遥感的实时性要求和高维特征使得并行化成为遥感领域的研究热点。在异构计算模式出现之前,国内外学者关注的高光谱数据并行化研究都只能在一定程度上改善其发展瓶颈,并不能完全满足工程实际需求。相关研究如开发基于原有串行算法的 CPU 同构并行算法和模型、利用加速平台或软件(如 FPGA、MATLAB 或遥感软件提供的并行库)等^[2]。近年来崛起的异构高性能计算并行模式以其强劲性能等特点成为主流计算模式,为众多领域大规模计算的发展提供了良好的加速平台。而 CPU/GPU(Graphic Processing Unit)异构模式更是以高性价比和低能耗比引领了当今

到稿日期:2015-03-01 返修日期:2015-05-08 本文受国家自然科学基金项目(61272146)资助。

汤媛媛(1987—),女,硕士生,工程师,主要研究方向为高性能图像处理, E-mail:447583109@qq.com;周海芳(1975—),女,教授,主要研究方向为高性能图像处理;方民权(1989—),男,硕士生,主要研究方向为高性能计算;申小龙(1989—),男,主要研究方向为高性能计算。

绿色高性能计算时代的潮流。

基于上述背景,本文将简要介绍高光谱遥感和 CPU/GPU 异构模式的研究现状,总结基于 CPU/GPU 异构模式的高光谱遥感影像数据的国内外研究现状和问题,最后面向共享存储的桌面小型超级计算机,基于 CPU/GPU 异构模式实现高光谱遥感影像 MNF 降维的并行化。

2 相关背景

2.1 高光谱遥感研究现状

高光谱遥感的发展主要包括两方面:成像光谱仪和高光谱数据处理方法的发展。成像光谱仪作为获取高光谱数据的重要设备,经历了从航空遥感到航天遥感的发展阶段,光谱分辨率也得到了从低分辨率、中分辨率到高分辨率的提升。高光谱数据处理方法主要有:降维、分类、融合、目标识别与检测、混合端元提取等。经过几十年的长足发展,高光谱数据处理方法不断丰富和拓展,被广泛应用于诸多领域,如地球科学、精准农业、林业、海洋、大气、水体、考古、军事等,验证了其广阔的应用前景^[2]。

2.2 CPU/GPU 异构计算模式

并行计算模式经历了从最初的以 CPU 为主同构计算模式演变为集合了 CPU 和协处理器的异构计算模式。异构模式的出现有效解决了同构模式面临的功耗墙、存储墙和频率墙的问题,实现了计算性能的大幅提升^[3]。当今异构模式主要有 4 种架构:APU、Cell/B. E、CPU/GPU、CPU/MIC。

基于 GPU 的异构模式以其可编程性好、性价比高和功耗比低等诸多优点受到人们的青睐。它采用 CPU+GPU 架构,包含两类差异巨大的处理器——CPU 和 GPU, GPU 作为协处理器处理数据密集型的并行任务, CPU 则负责复杂的逻辑事务处理,从而充分发挥 CPU 和 GPU 的独特优势,充分挖掘异构系统的潜能,降低计算成本和能耗^[4]。

目前市场上主流的 GPU 有 NVIDIA 和 AMD 两家,基于 GPU 的扩展并行是当今国际上发展最快的高性能计算技术之一,其发展已大大超过了摩尔定律,应用范围从计算机图形学扩展到通用计算领域,并表现出卓越的性能。在很多领域,如能源开发、生命科学、医学图像、天气预测、数据挖掘与分析、地球科学、金融行业、数值计算、计算化学、图像和视觉处理等领域^[2],科学家已将其应用移植到 CPU/GPU 异构模式,并获得了良好的性能提升。

2.3 基于 CPU/GPU 异构模式的高光谱遥感影像数据国内外研究现状

2008 年,Setoain 等^[5]提出将 GPU 用于高光谱图像的并行处理,并验证了 CPU/GPU 异构模式能够适用于高光谱图像处理领域。目前基于 CPU/GPU 异构模式的国内外研究大致分为以下 4 类^[2]:(1)面向 CPU/GPU 异构模式,将原有串行算法移植优化为并行算法来获得一定的加速效果,以此验证 CPU/GPU 异构模式的性能优势和适用性。(2)将基于 CPU/GPU 的异构并行算法与其他遥感软件或同构加速平台如 ENVI、Matlab、MPI、OpenMP 等进行对比,以体现其性能。(3)针对高光谱数据,深入研究基于 CPU/GPU 异构模式的加速及存储优化。(4)除了利用单个 GPU 进行加速外,研究多级并行降维算法,如单 GPU 下的 OpenMP+CUDA、MPI+OpenMP+CUDA 的混合编程模式,以及多 GPU 下的多级编程模型。

2.4 存在的问题

基于 CPU/GPU 异构模式的高光谱图像处理技术的大致流程为:根据串行算法和加速热点设计并行优化策略,重新排列原始图像数据,设计内核函数并开辟大量线程,实现 GPU 的两层并发机制。国内外对提高图像处理速度的研究主要集中在提高数据传输速度、优化算法策略、合理访存与数据结构这几方面。目前,基于 CPU/GPU 异构模式的高光谱遥感影像并行处理的研究主要集中在将串行算法移植优化到 GPU 平台,以验证异构模式的性能。其主要存在的问题有^[2]:(1)高光谱遥感算法基本上是从多光谱遥感移植而来,并未考虑到数据的高维特征,具有一定的局限性和缺陷;(2)高光谱图像的数据传输时间大大超过 GPU 的处理时间,对需要反复进行数据通信、迭代数量过多的算法效果一般;(3)目前的并行方法多是逐个像元处理,像元间及波段间的数据关联性需要人为控制,这也影响某些高光谱数据处理步骤的并行加速效果;(4)大多数基于 GPU 的高光谱数据并行研究主要是验证其在该平台下卓越的加速性能,对算法和存储的优化策略的涉及不够深入;(5)数据越大,线程越多,并发度越高,加速效果越明显,但目前 GPU 的线程数量是有限的。

3 基于 CPU/GPU 异构系统的高光谱 MNF 并行降维

3.1 高光谱遥感 MNF 降维算法原理

基于特征提取的最小噪声分离变换(Minimum Noise Fraction Rotation, MNF)是典型的高光谱遥感影像线性降维方法,它由两个主成分变换 PCA 组成,能够识别并隔离噪声数据,从而达到降维的目的^[6]。其串行算法流程为:

(1)通过滤波计算出原始图像噪声协方差矩阵 C_N , N 为波段数。

(2)提取 C_N 的对角矩阵,即对 C_N 进行特征值分解。

$$D_N = U^T C_N U \quad (1)$$

其中, D_N 为降序排列的特征值, U 为 C_N 的正交矩阵,也就是所对应的特征向量,得到变换矩阵

$$P = U D_N^{-\frac{1}{2}} \quad (2)$$

(3)对原始图像的协方差矩阵 C_D 进行变换。

$$C_{D-adj} = P^T C_D P \quad (3)$$

(4)对 C_{D-adj} 进行特征值分解。

$$D_{D-adj} = V^T C_{D-adj} V \quad (4)$$

其中, D_{D-adj} 为降序排列的特征值, V 为所对应的特征向量。

(5)计算变换矩阵 $T = PV$, 进行 MNF 变换。

$$Z = T^T X \quad (5)$$

X 为原图像数据。

MNF 串行算法的流程如图 1 所示。

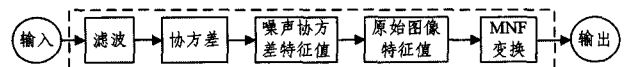


图 1 MNF 串行算法流程

3.2 高光谱遥感 MNF 降维算法加速热点分析

并行程序的性能优势只有与最优串行程序比较才能体现出来。基于编译器的优化开关选取是最主流、便捷的优化方法。为了找出最优串行程序,本文进行了编译器优化开关选取的测试,通过实验选定 Windows 环境下使用 release-O2 优化开关。

要分析 MNF 串行算法的加速热点, 必须通过执行该串行程序, 得出具体的运行时间, 进而找出花销较大的步骤。表 1 显示了在 Windows 环境下, 采用 4 组不同规模的高光谱遥感图像数据时, 执行 6 遍后取平均值的 MNF 降维串行算法程序(优化开关)的各步骤具体时间(不含 I/O)。

表 1 Windows 下 MNF 串行程序各段执行时间(s)

数据规模	滤波	协方差	噪声协方差特征值	原始变换矩阵特征值	MNF 变换	总时间
614 * 512	8.07	21.99	0.22	0.24	14.09	44.61
614 * 1087	19.65	49.17	0.30	0.30	31.88	101.30
753 * 1924	38.02	111.32	0.28	0.37	69.24	219.23
781 * 6955	134.08	402.32	0.27	0.33	257.95	794.95

MNF 算法原理表明, 执行过程中有大量的矩阵运算和迭代过程。串行程序实验证明, MNF 算法时间消耗最多的正是协方差矩阵运算和 MNF 变换过程, 其次是滤波。由表 2 可看出, 这 3 部分的时间和占总执行时间的 99% 左右, 是并行化的重点对象。噪声协方差特征值和原始矩阵变换矩阵特征值的时间开销所占比重随波段数增加而减少, 且耗时少, 根据 Amdahl 定律, 将这两部分并行化, 无法得到总性能的提升, 相反随着并行度的提高还可能因为核间通信导致加速比下降, 故忽略不计。

表 2 Windows 下 MNF 各步骤占总执行时间的百分比(%)

数据规模	滤波	协方差	噪声协方差特征值	原始变换矩阵特征值	MNF 变换
614 * 512	18.01	49.06	0.49	0.53	31.43
614 * 1087	19.34	48.39	0.29	0.29	31.37
753 * 1924	17.27	50.57	0.13	0.16	31.45
781 * 6955	16.80	50.41	0.03	0.04	32.32

3.3 MNF 算法加速热点并行设计及优化

输入数据后, MNF 算法的第一步就是采用滤波的方式对噪声协方差矩阵进行估计。本文使用了 3 * 3 大小的滤波模板, 从原始图像 X 取出 3 * 3 大小范围的数据(P0、P1、P2、P3、P4、P5、P6、P7、P8), 按降序重新排列并计算 P4 的值。滤波后图像的协方差矩阵 C_{i,j} = 原始图像 X_{i,j} - P4。MNF 算法滤波过程如图 2 所示。

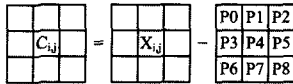


图 2 MNF 算法滤波过程

协方差矩阵的计算公式为:

$$\sigma_{x,y} = cov(X, Y) = \frac{\sum_{i=1}^s (X_i - \bar{X})(Y_i - \bar{Y})}{s-1}$$

传统展开变形后的公式为:

$$cov(X, Y) = \left[\sum_{i=1}^{s/N} (X_i - \bar{X})(Y_i - \bar{Y}) + \sum_{i=s/N+1}^{2s/N} (X_i - \bar{X})(Y_i - \bar{Y}) + \dots + \sum_{i=(N-1)s/N+1}^s (X_i - \bar{X})(Y_i - \bar{Y}) \right] / (s-1)$$

其中, X、Y 为不同的波段, s 是输入矩阵的行宽。可以看出协方差矩阵的计算重心在 (X_i - X̄)(Y_i - Ȳ) 上。传统的协方差矩阵需要计算局部数据的向量均值和全局向量均值, 最后在 Root 节点进行汇总计算, 进程间进行了 3 次通信。

为了降低进程间通信, 通过对协方差矩阵的不断变形:

$$\frac{1}{s-1} \left[\sum_{i=1}^s (X_i Y_i + \bar{X} \bar{Y} - X_i \bar{Y} - \bar{X} Y_i) \right]$$

$$\begin{aligned} &= \frac{1}{s-1} \left[\sum_{i=1}^s X_i Y_i + s \bar{X} \bar{Y} - \bar{Y} \sum_{i=1}^s X_i - \bar{X} \sum_{i=1}^s Y_i \right] \\ &= \frac{1}{s-1} \left[\sum_{i=1}^s X_i Y_i + s \bar{X} \bar{Y} - \bar{Y} s \bar{X} - \bar{X} s \bar{Y} \right] \\ &= \frac{1}{s-1} \left[\sum_{i=1}^s X_i Y_i - s \bar{X} \bar{Y} \right] \\ &= \frac{1}{s-1} \left[\sum_{i=1}^s X_i Y_i - \frac{1}{s} \sum_{i=1}^s X_i \sum_{i=1}^s Y_i \right] \\ &= \frac{1}{s-1} \left[\left(\sum_{i=1}^{s/N} X_i Y_i + \sum_{i=s/N+1}^{2s/N} X_i Y_i + \dots + \sum_{i=(N-1)s/N+1}^s X_i Y_i \right) - \frac{1}{s} \left(\sum_{i=1}^{s/N} X_i + \sum_{i=s/N+1}^{2s/N} X_i + \dots + \sum_{i=(N-1)s/N+1}^s X_i \right) * \left(\sum_{i=1}^{s/N} Y_i + \sum_{i=s/N+1}^{2s/N} Y_i + \dots + \sum_{i=(N-1)s/N+1}^s Y_i \right) \right] \end{aligned}$$

将任务模块分解成数据和计算更独立、细小的模块, 协方差矩阵的计算重心转变为 $\sum X$ 和 $\sum XY$ 。变换后协方差矩阵的进程间通信开销减少为 1 次, 计算量集中在 $\sum X$ 和 $\sum XY$ 上。

为了避免直接映射引起 GPU 的性能损失和 OpenMP 的 Fork-join 次数过多, 本文采用协方差矩阵的补偿均匀划分策略。由于协方差矩阵的对称性, 只需计算其下三角矩阵, 将需要计算的下三角矩阵视为矩形矩阵, 可完成负载均衡的计算任务。协方差矩阵的补偿划分流程如图 3 所示。

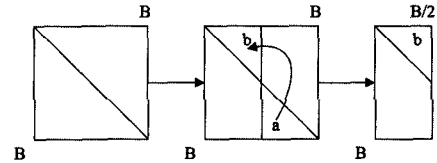


图 3 协方差矩阵的补偿划分流程

MNF 变换计算的结果依赖于前面步骤的结果, MNF 变换公式为 $T = PV$, V 为原始图像协方差矩阵的特征向量。 $T = PV = \{VP_1, VP_2, \dots, VP_N\}$, 通过对矩阵 P 的均匀分解, MNF 变换可以将 VP_i 映射到 OpenMP 线程/GPU 内核。 $T = PV$ 还可以对 V 矩阵分解, 将其拆成 $\{V_1 P, V_2 P, \dots, V_N P\}$ 。本文采用的是对矩阵 P 进行分割。

3.4 基于 OpenMP 的 MNF 并行降维算法

由于本文实验用的小型桌面超级计算机有 2 个 8 核的英特尔至强 CPU, 程序执行前设置 export OMP_NUM_THREADS=16, 即默认线程个数为 16。通过合理的任务分配和同步, 能够保证 OpenMP 计算任务的均衡性和正确性。在图 4 中, 使用 Fork-Join 执行模式, 这意味着读入高光谱数据后, 程序运行到 3 个加速热点并行域时会派生出多个从线程, 执行完毕后从线程合并为一个主线程, 各个步骤间都需要同步来保证该步骤的完成和下一个步骤的正确性。

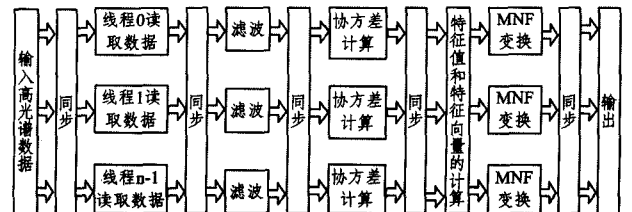


图 4 基于 OpenMP 的 MNF 并行降维算法流程

3.5 基于 CPU/GPU 异构模式的 MNF 并行降维算法

面向基于共享存储的小型桌面超级计算平台, 基于

CPU/GPU 异构模式,将 MNF 算法的加速热点(协方差矩阵运算、MNF 变换、滤波)映射到 GPU 执行来实现程序整体性能的提升。

基于 CPU/GPU 异构模式的 MNF 并行算法流程如图 5 所示。CPU 与 GPU 相互配合,其中,CPU 端负责启动 GPU、与 GPU 之间的数据交流、串程序的计算和逻辑控制等任务,包括读取高光谱数据到 CPU 主存、传送数据到 GPU 端的全局存储器、启动 Kernel 函数以控制 GPU 进行加速热点的并行计算,执行非加速热点步骤的计算和累加求和、接收 GPU 计算结果等。而 GPU 端则作为协处理器负责密集度高的并行计算部分,包括滤波、协方差矩阵、MNF 变换并行热点的并行计算,其中协方差矩阵的计算按照补偿划分策略,先计算划分后对角线上的协方差,再计算矩形协方差。最后 CPU 端输出最终计算结果,除了 I/O 通信外,在整个计算过程中,CPU 端和 GPU 端也有相互的数据传递。

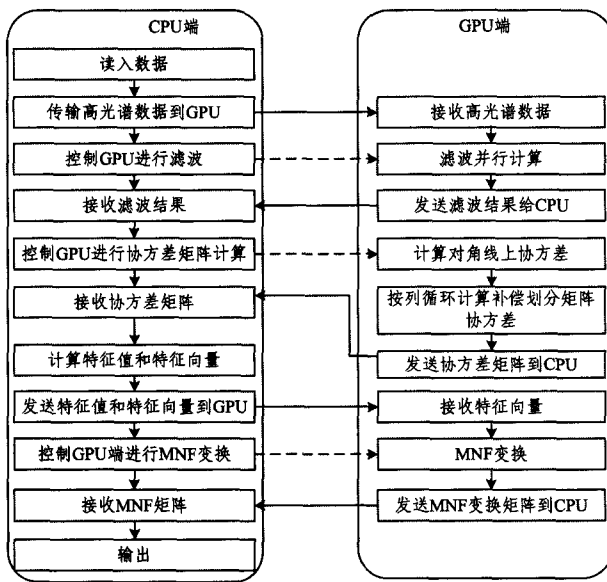


图 5 基于 CPU/GPU 异构模式的 MNF 算法流程

滤波的 GPU 端设计:本文为滤波并行计算设计一个 Kernel 函数,定义 GPUfilter <<<band, thread_num>>>(), <<< >>>运算符中的参数是 Kernel 函数的执行参数,用于说明 Kernel 中的线程块及线程数量,()里的参数则是滤波的参数,其中 band=224, thread_num=512。也就是说该 Kernel 函数中包含 224 个 block,每个 block 由 512 个 thread 组成,其中 block 的数量等于问题规模即波段数。每个 block 对应处理一个波段的数据。尽量减少全局存储器的访存并确保对齐访问来优化 GPU 全局存储器访存。利用共享存储替代全局存储,每个 block 声明一块 SM 存储空间,用于存储 block 内 thread 需要执行的对应数据块,并注意 Bank Conflict 以提高数据存取速度。

协方差矩阵的 GPU 端设计:基于矩阵的补偿均匀划分策略,采用 GPU 上的每个线程块计算协方差矩阵中的某个协方差。共启动 $B/2$ 列个 block, B 为波段数。每列有 B 个 block 去计算一列协方差矩阵,每个 block 对应单个协方差计算任务,每个 block 内有 512 个 thread。具体到映射实现,共开辟两个内核函数:kernel1 <<< $B/2$, thread_num>>>进行对角线上的协方差矩阵计算, kernel2 <<< $B/2$, thread_num>>>计

算下三角协方差矩阵,通过 $B/2$ 次循环启动共同完成协方差矩阵的运算。出于数据访存优化考虑,协方差计算使用了 GPU 读取速度较快的 SM 存储经常使用的数据。由于 block 内的 thread 在执行任务时是乱序执行的,需使用 _syncthreads ()函数来同步保证数据更新的一致性。这样,同一个 block 内的 thread 通过 shared memory 交换数据,并通过栅栏同步 (barrier)保证 thread 间能够正确地共享数据,实现了线程间的细粒度并行。

MNF 变换的 GPU 端设计:MNF 变换可视为矩阵乘运算。数据存放在 GPU 共享存储器上,假设 P 是 $B \times n$ 的矩阵, V 是 $S \times B$ 的矩阵,则结果矩阵 $T=PV$ 的尺寸为 $S \times n$,目标是求解计算长度为 B 的两个向量内积。求得变换矩阵 T 后,再对原始图像 X 进行 MNF 变换。考虑到将结果矩阵上的每个元素映射到 GPU 线程的方案受限于 GPU 能启动的线程个数,采用循环计算的方法,创建内核函数,启动 512 个线程块,每个线程块包含 512 个线程。线程块在分割好的数据区跳跃切换,线程块内的线程计算 block 内第 $blockIdx * blockDim + i * (gridDim * blockDim)$ 个内积运算(其中 $gridDim$ 表示线程格中的线程块数, $blockDim$ 表示线程块中线程数量, i 为循环变量),直到运算个数超过 $S \times n$ 。采用了基于共享存储器和栅栏同步的访存优化和数据对齐策略。通过最大化存储器吞吐量、利用率和指令吞吐量来充分挖掘 CPU/GPU 异构性能。

4 实验结果与分析

4.1 实验平台

实验平台为 AMAX 微型桌面超级计算机,详细配置为:1)两个 8 核的 Intel Xeon CPU E2650;2)一个 Kepler GK110 架构的 NVIDIA Tesla K20 GPU。在 Windows 7 环境下,采用 CUDA6.5 工具包及 Visual Studio2012 对程序进行编译及执行。

4.2 实验数据

实验数据来自美国 AVIRIS 提供的免费高光谱数据,详细信息如表 3 所列。

表 3 实验高光谱数据详细信息

数据规模	波段数	原始数据大小 (MB)	预处理后数据大小 (MB)
614 * 512	224	134.31	67.16
614 * 1087	224	285.15	142.58
753 * 1924	224	618.98	309.49
781 * 6955	224	2320.74	1160.37

使用 ENVI 软件将数据的各波段转换为 BMP 图像,利用 pre_img.cpp 编译命令行将生成的 BMP 图片合并输出为 img_(图像高度)_(图像宽度).dat 文件,即并行降维使用的高光谱数据。该步骤能够减少降维程序执行时的数据输入/输出(I/O)时间。

4.3 实验结果

按照设计的并行优化方案,实验测试得到的数据(执行时间和加速比)如表 4—表 7 所列。通过与串行程序和 OpenMP 算法各加速热点和总执行时间的比较,基于 CPU/GPU 异构模式的加速性能优势得到了体现,验证了 CPU/GPU 异构模式在大规模图像处理领域的应用前景。

users to items[C]//Proc of the 8th ACM Conf on Recommender systems. ACM,2014;145-152

- [13] Cremonesi P, Garzotto F, Negro S, et al. Looking for “good” recommendations: A comparative evaluation of recommender systems[M]// Human-Computer Interaction-INTERACT 2011. Springer Berlin Heidelberg,2011;152-168
- [14] Bennett J, Lanning S. The netflix prize[C]//Proc of KDD cup and workshop. 2007;35

- [15] Vargas S, Baltrunas L, Karatzoglou A, et al. Coverage, redundancy and size-awareness in genre diversity for recommender systems[C]//Proc of the 8th ACM Conf on Recommender systems. ACM,2014;209-216
- [16] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer,2009,42(8):30-37
- [17] Anderson C. The long tail: Why the future of business is selling less of more[M]. Hyperion,2006

(上接第 50 页)

表 4 基于 OpenMP 的 MNF 并行程序执行时间(s)

数据规模	滤波	协方差	噪声协方差特征值	原始变换矩阵特征值	MNF 变换	总时间
614 * 512	0.44	3.05	0.21	0.21	0.30	4.20
614 * 1087	0.90	6.55	0.23	0.22	0.59	8.48
753 * 1924	1.77	13.62	0.25	0.26	1.28	17.17
781 * 6955	6.39	49.53	0.23	0.23	4.77	61.13

表 5 基于 CPU/GPU 的 MNF 并行程序执行时间(s)

数据规模	滤波	协方差	噪声协方差特征值	原始变换矩阵特征值	MNF 变换	总时间
614 * 512	0.94	0.54	0.31	0.29	0.03	2.11
614 * 1087	0.99	1.11	0.28	0.24	0.06	2.65
753 * 1924	1.02	2.41	0.31	0.24	0.12	4.10
781 * 6955	1.33	9.40	0.28	0.25	0.44	11.71

表 6 基于 OpenMP 的 MNF 并行程序加速比

数据规模	滤波加速比	协方差加速比	MNF 变换加速比	总时间加速比
614 * 512	18.34	7.2	47.76	10.54
614 * 1087	21.91	7.51	54.38	11.84
753 * 1924	21.64	8.17	54.09	12.66
781 * 6955	20.99	8.12	54.22	12.82

表 7 基于 CPU/GPU 的 MNF 并行程序加速比

数据规模	滤波加速比	协方差加速比	MNF 变换加速比	总时间加速比
614 * 512	8.59	40.70	481.65	20.74
614 * 1087	20.36	44.35	569.30	36.63
753 * 1924	37.33	46.23	595.63	50.61
781 * 6955	100.87	42.80	591.28	61.80

从图 6 看出,两种并行方案均取得一定的加速比,其中基于 CPU/GPU 的方案加速比增长随着数据规模的增大尤为明显,这说明了本文提出的并行方案的有效性。OpenMP 加速比的增幅较平稳,主要是由于 GPU 的线程号和数据号一致,在取数据时,一次能取出 cacheline 的数据,可确保访存对齐,且 OpenMP 核数有限。相对于 OpenMP 并行方案,基于 CPU/GPU 异构模式取得的加速比更理想,而且数据规模越大,总加速比越大,其中最大总加速比为 61.8。这验证了 CPU/GPU 异构模式相对于传统同构模式在处理大规模数据上的并行性能优势。

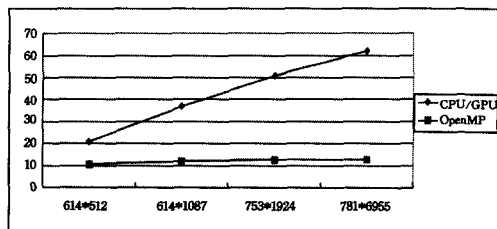


图 6 MNF 降维总加速比(CPU/GPU、OpenMP)

结束语 高光谱遥感影像处理属于计算量大、计算复杂、迭代数多的计算任务。采用异构并行手段能够在保证图像处理精度的同时,有效提高图像的处理速度。本文介绍了高光谱遥感和 CPU/GPU 异构计算模式,并阐述了基于 CPU/GPU 异构模式的高光谱遥感数据处理研究现状和问题,最后面向共享存储型小型桌面超级计算机,引入共享存储的 OpenMP 同构模式作为实验对比基础,设计并实现了基于 CPU/GPU 和 OpenMP 的高光谱遥感影像 MNF 降维的并行算法。实验取得了良好的加速比,其中 CPU/GPU 最大总加速比为 61.8, OpenMP 最大总加速比为 12.82,验证了异构模式在高光谱遥感处理领域的发展潜力。

参考文献

- [1] Su Hong-jun, Sheng Ye-hua, Yang He, et al. Orthogonal Projection Divergence-Based Hyperspectral Band Selection[J]. Spectroscopy and Spectral Analysis, 2011, 31(5): 1309-1313 (in Chinese)
- 苏红军, 盛业华, Yang He, 等. 基于正交投影散度的高光谱遥感波段选择算法[J]. 光谱与光谱学分析, 2011, 31(5): 1309-1313
- [2] Tang Yuan-yuan, Zhou Hai-fang, Fang Min-quan, et al. Hyperspectral Remote Sensing Image Data Processing on GPU[J]. Information Security and Technology, 2015, 6(4): 148-152 (in Chinese)
- 汤媛媛, 周海芳, 方民权, 等. 基于 GPU 的高光谱遥感影像数据处理[J]. 信息安全与技术, 2015, 6(4): 148-152
- [3] Ju Tao, Zhu Zheng-dong, Dong Xiao-she. The Feature, Programming Model and Performance Optimization Strategy of Heterogeneous Many-Core System: A Review[J]. Acta Electronica Sinica, 2015, 43(1): 111-119 (in Chinese)
- 巨涛, 朱正东, 董小社. 异构众核系统及其编程模型与性能优化技术研究综述[J]. 电子学报, 2015, 43(1): 111-119
- [4] 张舒, 褚艳利. GPU 高性能运算之 CUDA[M]. 北京: 中国水利水电出版社, 2009
- [5] Setoain J, Tenllado C, Prieto M, et al. Parallel hyperspectral image processing on commodity graphics hardware[C]//2006 International Conference on Parallel Processing Workshops. 2006: 465-472
- [6] Green A A, Switzer B H, et al. A transformioll for ordering multispectral data in terms of image quality with Implications for noise remaval[J]. IEEE Transactions on Geoscience and Remote Sensing, 1988, 26(1): 65-74