

文章编号: 1006-6616 (2015) 02-0190-09

# GPU 用于高光谱数据高性能计算的应用实践与分析

许宁<sup>1,2,3</sup>, 肖新耀<sup>1,2,3</sup>, 胡玉新<sup>1,2</sup>, 温静<sup>4</sup>, 汪大明<sup>5</sup>

(1. 中国科学院空间信息处理与应用系统技术重点实验室, 北京 100190;

2. 中国科学院电子学研究所, 北京 100190;

3. 中国科学院大学, 北京 100049;

4. 中国国土资源航空物探遥感中心, 北京 100083;

5. 中国地质调查局油气资源调查中心, 北京 100029)

**摘要:** 高光谱遥感数据具有波段多、数据量大、处理复杂等特点, 基于 GPU 的高性能计算在遥感领域得到了快速发展, 为高光谱数据的快速处理提供了硬件和技术条件。采用 GPU 对高光谱遥感数据常用的 SAM、PPI 等处理算法进行应用实验, 验证基于 GPU 的高光谱遥感数据快速处理技术。实验采用新疆东天山地区的一景星载 Hyperion 数据, 利用支持 IDL 开发语言的 GPULib、CUDA 运行时 API 库进行算法效率的验证, 结果表明, 基于 GPU 的高光谱数据处理效率比常规的多核 CPU 主机处理效率有较大提升, 具有一定的应用推广价值。

**关键词:** 高光谱数据; GPU; 高性能计算; SAM; PPI

**中图分类号:** TP79

**文献标识码:** A

## 0 引言

高光谱遥感数据由于其丰富的光谱信息, 在地质勘探、环境监测、伪装判别等领域发挥了巨大作用, 是定量遥感发展的一个重要方向。在高光谱遥感数据的地质资源勘探应用领域, 端元提取、光谱角制图 (Spectral Angle Mapping, SAM)、匹配滤波等技术得到了广泛应用<sup>[1~3]</sup>。由于高光谱数据波段多、数据量大, 这些处理算法通常需要在特征空间或光谱空间进行大量的向量计算, 造成较大的计算资源开销, 因此高光谱数据处理算法的效率提升显得尤为重要。在国外, Gillis D 等<sup>[4]</sup>基于不同的多核处理器, 利用 MPI 多线程技术对 ORASIS 系统中的 N-Findr 等端元提取算法的处理效率进行了优化和对比; Tilton J C<sup>[5]</sup>利用 MPI 对 ETM 数据非监督分层分割算法进行了实验、分析和验证; 针对高光谱数据的目标检测, Wang Jianwei 等<sup>[6]</sup>利用 FPGA 实现了约束能量最小算法的实时处理。近年来, 基于图形处理器 (Graphic Processor Unit, GPU) 的高性能计算技术蓬勃发展, 在图像处理、地震资料分

收稿日期: 2014-11-13

基金项目: 中国地质调查局地质调查项目“地质勘查遥感系统集成与综合应用示范”(1212011120226)

作者简介: 许宁 (1982-), 男, 四川邛崃人, 博士生, 研究方向为光学遥感影像配准、融合处理及信息提取。E-mail:

x\_ning@aliyun.com

析、经融计算与分析等领域得到了推广和应用;在高光谱遥感数据处理领域,针对GPU并行计算技术也开展了一些研究和验证。Setoain J等<sup>[7]</sup>利用GPU对高光谱数据的端元提取算法进行了效率对比,发现对于SAM光谱分析算法,GPU相较CPU可提升效率10倍左右;Agathosd A等<sup>[8]</sup>利用多颗GPU协同开展了高光谱数据最小体积单形体分析解混方法研究,验证了GPU高性能计算的有效性。在国内,也有学者开展了基于GPU的遥感图像快速处理方法探索。杨靖宇等<sup>[9]</sup>针对SAM算法进行了GPU并行化方法研究;罗耀华等<sup>[10]</sup>利用GPU开展了MNF变换算法的并行处理和比较;宋义刚等<sup>[11]</sup>开展了基于GPU的PPI端元提取算法的优化和分析。

考虑到ENVI/IDL在遥感地质领域的广泛应用,本文着重探讨基于IDL开发语言的GPULib库在配置GPU的工作站环境中实现高性能计算效率的提升。实验利用一景新疆东天山地区的星载Hyperion高光谱遥感数据,针对一些常用的高光谱数据处理算法开展应用和验证,最后通过这些算法处理效率的对比来分析GPU在高光谱数据地质应用领域的应用前景。

## 1 并行计算环境

### 1.1 GPU及GPULib

目前常用的并行计算环境有OpenMP、MPI及支持GPU运行的CUDA等,本文只对GPU的组成和环境做简单介绍。GPU的造价低,且具有强大的浮点数据计算能力。与CPU类似,GPU由GPU内存、缓存和多个流处理器(Stream Multiprocessor)组成,其中流处理器是真正的处理单元,由多个处理核心(Core)及一级缓存、寄存器和指令发射单元等部分组成。

尽管GPU的运算能力非常强大,但其只是作为协处理器在CPU的协调下完成数据处理。图1为单一总线架构下CPU和GPU形成的非对称处理架构,GPU与CPU不是通过内部总线互连,而是通过PCI-E接口作为外围设备进行工作;同时,GPU和GPU不能访问对方的存储器,多个GPU之间也不能互相进行内存访问。这样,在数据处理的过程中,数据必须首先存储在CPU内存中,当GPU需要进行数据处理时,必须先将数据从CPU内存传输到GPU内存,处理完成后再将数据传回CPU内存。因此,基于GPU实现高速数据处理的效率不只取决于GPU的性能,还要取决于计算机系统架构与GPU的配合程度。

GPULib是Tech-X公司开发的一套支持ENVI/IDL、Matlab等开发语言的GPU运行算法库,它的运行基于CUDA平台,为单机IDL开发语言使用GPU提供了良好的环境。

### 1.2 基本性能测试

本次实验采用的工作站硬件环境:CPU为4核Intel Quad(Q6600)处理器,主频2.40 GHz,内存3.5 G,硬盘容量500 G,另外通过PCI-E插槽连接一块C2050型号的GPU。工作站的软件环境:Windows XP Pro 32bit操作系统,另外安装配置了CUDA Developer Drivers驱动和CUDA Toolkit工具包软件,采用的编程软件为IDL7.1,并配置了专门的GPULib算法库。

除高光谱遥感数据处理中常用的向量点积、矩阵乘法、矩阵转置等,本次还增加了快速傅立叶变换(FFT)的测试。基本性能测试分别对 $5000 \times 5000$ 和 $10000 \times 10000$ 大小的单精度浮点数据进行实验,结果见表1。从表1中可以看出,利用GPU进行矩阵乘法和FFT计算时,GPU可以获得很大的效率提升;但对于矩阵转置这种运算量很小的数据处理,GPU反而没有运行优势,因为GPU处理时消耗了更多的从CPU向GPU传输数据的时间。

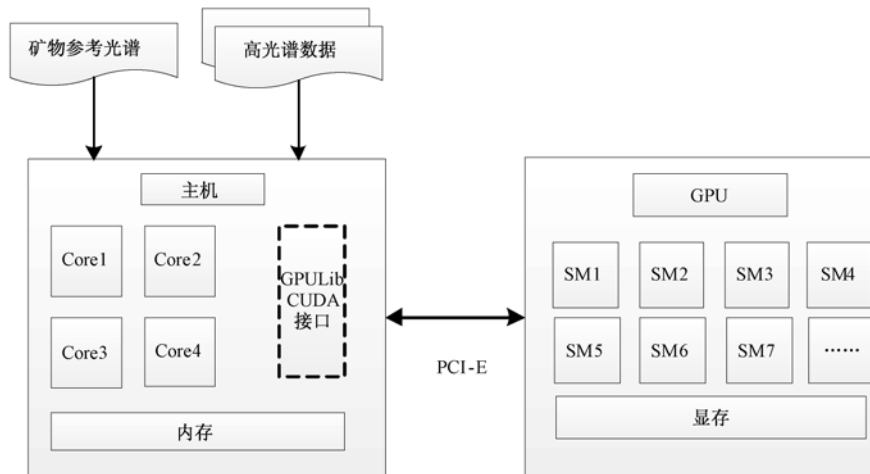


图1 工作站主机和 GPU 组成的工作环境

Fig. 1 Sketch of work environment constructed by Host and GPU

表1 CPU-GPU 基本性能测试

Table 1 Basic performance comparison of CPU and GPU

测试项目	数据大小	测试结果		
		矩阵乘法	矩阵转置	FFT 计算
Q6600 时间/s	5000 × 5000	126.859	0.000	4.047
	10000 × 10000	1626.453	0.001	23.121
C2050 时间/s	5000 × 5000	0.157	0.016	0.125
	10000 × 10000	0.452	0.032	0.405
处理时间对比	5000 × 5000	820.758	0.000	32.376
	10000 × 10000	3598.347	0.031	57.089

## 2 数据和算法设计

高光谱遥感数据具有波段多、数据量大、处理耗时等特点，而 GPU 具有造价低、体积小、速度快、并行性高的优势，因此 GPU 可以为高光谱数据的快速处理提供一种新的解决途径。基于上述思想，本文主要验证应用 GPU 的高光谱遥感数据快速处理关键技术，基于常用的 IDL 语言开发和实现典型的 SAM、PPI 算法的并行处理。采用星载 Hyperion 高光谱数据进行验证实验和分析，该数据在实验中已经完成了坏线去除、水汽吸收波段剔除以及大气校正处理，数据大小为 1700 行 × 250 列 × 176 波段，数据量为 285 M。在 SAM 算法验证过程中，采用不同数量（包括 5 条光谱和 200 条光谱库中的参考光谱）的光谱库数据进行了运算测试。

### 2.1 SAM 算法实验设计

SAM 算法在高/多光谱遥感领域应用较广，多用于遥感数据的分类处理，即像元光谱与参考光谱（包括在图像中选择的像元向量和光谱库向量）的相似度计算。其算法原理是将像元向量和参考向量看作是  $N$  维空间（ $N$  为波段数量）的 2 个点，通过计算 2 个向量在  $N$  维空间的向量夹角对比 2 个光谱向量的相似性，夹角越小，表明 2 个光谱的相似性越高。SAM 算法通过下式来计算测试光谱  $t_i$  与实验室光谱  $r_i$  之间的相似性<sup>[12]</sup>：

$$\alpha = \cos^{-1} \left[ \frac{\sum_{i=1}^{n_b} t_i r_i}{\sqrt{\sum_{i=1}^{n_b} t_i^2} \sqrt{\sum_{i=1}^{n_b} r_i^2}} \right] \quad (1)$$

式中:  $n_b$  为波段数。两个光谱之间的相似性不受向量长度及增益的影响, 因而可以减低地形对照度的影响,  $\alpha$  取值区间为  $0^\circ$ — $90^\circ$ 。

本次选用 SAM 算法进行基于 IDL + GPU 的处理实验。根据 GPU 和 CPU 的关系, 单颗 GPU 实验中二者之间的逻辑关系如图 2 所示。其中主机为多核单机工作站, 高光谱数据和光谱库数据首先被导入到主机内存中, 进行测试的数据根据需要通过接口传输到 GPU 显存中, GPU 再读取显存中的数据进行处理。

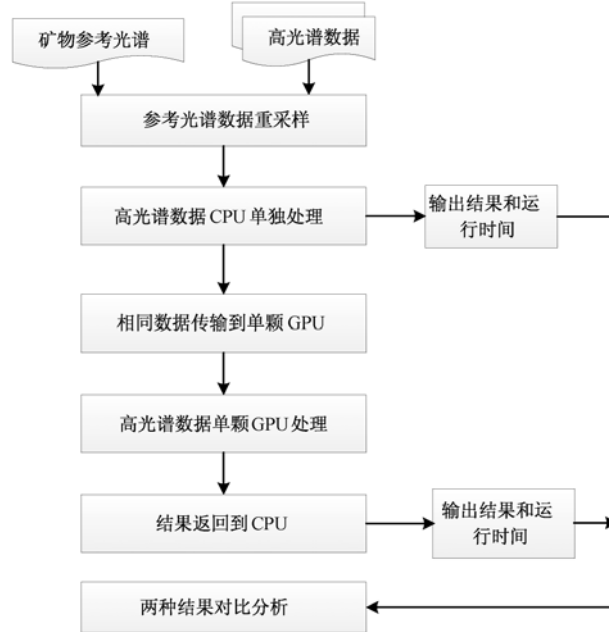


图2 基于单颗 GPU 实现的 SAM 处理流程图示

Fig. 2 Flowchart of implementation for SAM algorithm on single GPU

在 GPULib 库和 CUDA 运行时 API 等的支持下, 为了进行 CPU 和 GPU 处理效率的对比, 本文采用以下处理流程进行 SAM 算法的处理实验:

- ①首先获取待处理的高光谱遥感数据和参考光谱数据, 本文选择我国东天山地区的一景 Hyperion 高光谱数据, 选择美国 USGS 矿物光谱库中的云母、方解石等作为参考光谱数据;
- ②参考光谱的光谱分辨率和 Hyperion 数据不同, 需要对参考光谱进行重采样;
- ③将数据全部读入到工作站内存, 首先利用 CPU 进行处理, 输出整景数据的处理时间;
- ④初始化 GPU, 将相同的数据从 CPU 传送到 GPU 中;
- ⑤在 GPU 中进行 Hyperion 数据的 SAM 处理, 主要涉及到数据的矩阵乘法运算;
- ⑥将处理完成后的产品数据从 GPU 返回到 CPU, 输出处理后的结果和处理时间;
- ⑦对两种方式处理的结果进行对比, 包括产品结果的一致性和时间 (性能) 的变化。

## 2.2 PPI 算法实验设计

像元纯净指数 (Pixel Purity Index, PPI) 认为在  $N$ -维像元特征空间中 ( $N$  为高光谱数据

波段数), 所有像元成散点分布, 而那些比较纯的像元(端元)将分布在  $N$  维空间中散点分布形成的  $N + 1$  个顶点的凸面体的顶点处<sup>[13-14]</sup>。PPI 的处理采用由最小噪声分离(Minimum Noise Fractions, MNF)<sup>[15]</sup>变换得到的高光谱降维数据, 将降维后的像元光谱向量逐像元投影到随机生成的不同方向单元向量上, 对每个随机单元向量统计像元投影到各随机单元向量端点(或接近于端点)处的次数, 记录每次投影到端点的像元, 每个像元被标记为极值的总次数称为“纯像元指数”, 指数图像的像元越亮, 表明该像元越纯净, 更可能为图像中存在的光谱端元(Endmember), 其基本原理如图 3 所示。

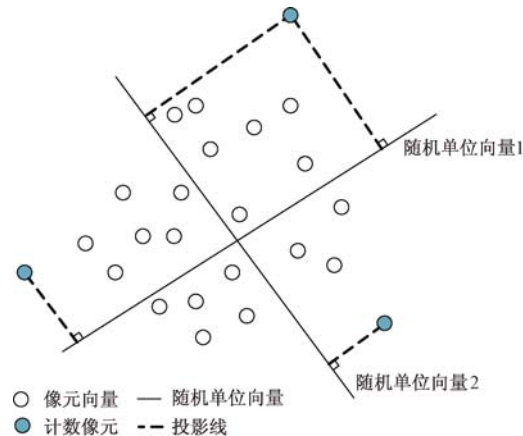


图 3 PPI 算法中像元投影极值计数原理

Fig 3 Principle of PPI algorithm for projection of pixel vectors on skewers

本次基于 IDL + GPU 的 PPI 并行算法实验处理流程如下:

①利用 IDL 实现的 PPI 算法与 ENVI 集成的 PPI 模块进行处理实验, 并进行效率对比。处理时对 ENVI 和程序 IDL 算法都采用了逐个像元迭代和分块迭代的方式, 其中分块迭代表示分块中的所有像元均参与与随机单位向量(skewer)的矩阵乘法运算;

②基于 GPUlib 库函数对原来的 IDL PPI 算法程序进行更改和优化;

③利用 ENVI 完成其 PPI 算法的处理测试, 包括逐像元处理和分块处理, 记录时间;

④在 CPU 主机上完成原来 PPI 算法的测试, 也包括逐像元的处理和分块处理, 记录时间;

⑤通过循环, 将高光谱数据逐像元传送到 GPU 进行处理, 记录时间;

⑥采用分块方式将数据传送到 GPU 进行处理, 并保存返回的处理结果, 记录时间;

⑦对 ENVI 处理结果和 IDL 处理的结果进行对比。

### 3 结果与分析

针对 SAM 算法, 按照 SAM 算法实验设计的处理流程, 利用 Hyperion 星载高光谱数据进行实验, 其中参考光谱从 USGS 光谱库中挑选; 为比较不同数量参考光谱对处理效率的影响, 分别选择 5 条和 200 条参考光谱进行 SAM 处理。本次采用单颗 GPU 进行实验, 实验输出的统计时间见表 2。在处理过程中需要对原始三维数据进行重列, 因此增加了矩阵重列时间的对比(矩阵重列时间是指矩阵完成行列重组形成新矩阵的 Reform 的时间)。

表2 CPU与GPU的SAM性能测试

Table 2 Performance comparison of SAM algorithm between CPU and GPU

测试项目	5条参考光谱		200条参考光谱	
	SAM计算时间	矩阵重列时间	SAM计算时间	矩阵重列时间
CPU/s	0.469	1.000	14.093	5.516
GPU/s	0.188	0.094	2.297	1.032
效率对比	2.500	10.638	6.135	5.345
平均效率	5.209		5.890	

可以看出,在只采用4核CPU进行处理和利用GPU进行处理的时间效率对比上,当参考光谱较少时,SAM处理效率提升并不明显,只有2.5倍;但采用200条光谱进行SAM计算时,二者的时间效率差了6倍。而在矩阵重列方面,参考光谱较少时,GPU对数据重列所花的时间更少,效率更高。

对于PPI算法,按照PPI算法实验设计中的处理流程进行处理,程序输出的测试时间见表3;对高光谱数据进行PPI处理的结果见图4,从左到右为选择参加处理实验的Hyperion高光谱三波段彩色数据、ENVI处理结果和程序处理结果。从结果图可以看出,CPU和GPU二者运行结果一致。

表3 测试程序PPI与ENVI性能测试(CPU)

Table 3 Performance comparison of PPI between test program and ENVI on CPU

测试项目	1000单位随机向量		10000单位随机向量	
	逐像元	分块优化	逐像元	分块优化
ENVI/s	245	13	2163	125
程序/s	501.641	183.200	5054.172	1320.618
效率对比	0.489	0.071	0.428	0.095

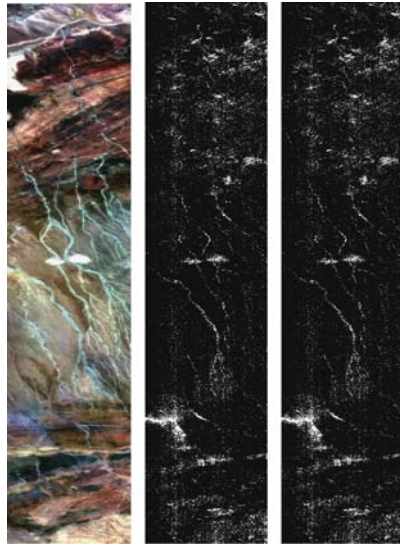


图4 进行PPI实验的处理数据和处理结果

Fig. 4 Hyperspectral data and results of PPI algorithm on CPU and GPU

PPI 算法运行效率的比较采用两种方式, 一种为测试程序与 ENVI 商业软件的比较, 一种为基于 CPU 和 GPU 的测试程序运行效率比较。从表 3 测试程序和 ENVI 自带的 PPI 算法的运行结果可以看出, 实验中编写的 IDL 程序采用逐像元处理 (采用循环方式, 生成一个随机单位向量 Skewer, 将图像所有像元光谱逐个与其计算内积) 时, 消耗的时间较长, 为 ENVI 逐像元处理的 2 倍; 进行分块处理 (利用 IDL 的矩阵运算优势) 后, ENVI 在效率上得到了很大的提升, 但测试程序采用分块处理获得的优势并不明显 (仅使用 CPU 处理)。

使用 GPU 参与处理后 (见表 4), 对于逐像元的处理, GPU 在处理效率上并没有什么优势, 并且由于数据传输时间的消耗, 其处理时间实际比 CPU 更长; 但对于分块处理, GPU 则体现了它的优势, 相较 CPU 效率提升达到了 10 倍以上。当然, 由于测试程序没有进行优化处理, 与 ENVI 相比, GPU 在处理效率上没有体现出较大优势, 在 1000 个随机单位向量参加运算时, CPU 处理时间为 13 s (见表 3), 而 GPU 为 10.3 s (见表 4); 在 10000 个单位随机向量参加运算时, CPU 处理时间为 125 s, GPU 为 95.682 s。

表 4 CPU 与 GPU 测试程序 PPI 性能对比

Table 4 Performance comparison of PPI using test program between CPU and GPU

测试项目	1000 单位随机向量		10000 单位随机向量	
	逐像元	分块优化	逐像元	分块优化
CPU/s	501.641	183.20	5054.172	1320.618
GPU/s	523.102	10.324	5128.320	95.682
效率对比	0.958974	17.745	0.985541	13.802

## 4 结论

本文基于高光谱数据处理的特点和目前 GPU 高性能计算的发展和现状, 以单机工作站挂接 GPU 为并行计算环境, 验证 GPU 在高光谱数据处理中的应用效能。通过对高光谱数据地质应用处理中常用的矩阵计算、SAM 算法、PPI 算法的应用分析, 设计验证流程, 分别对这些算法开展 GPU 和 CPU 处理效能的对比实验, 并通过 AVIRIS 机载数据和 Hyperion 星载高光谱数据进行了实验验证。实验结果表明, 基于 IDL 和 GPU 的小型并行计算环境可以实现高光谱数据处理效能的提升, GPULib 库可以为广大的研究人员提供相对简单的研究和应用环境; 单颗 GPU 集成的工作站在高光谱数据常用的 SAM 处理和 PPI 端元提取算法方面均得到了效率的提升, 平均效率可以达到 10 倍左右; 测试数据采用 GPU 并行处理算法得到的处理结果与 CPU 的处理结果一致。

## 参 考 文 献

- [1] 王润生, 甘甫平, 闫柏琨, 等. 高光谱矿物填图技术与应用研究 [J]. 国土资源遥感, 2010, (1): 1~13.  
WANG Run-sheng, GAN Fu-ping, YAN Bai-kun, et al. Hyperspectral mineral mapping and its application [J]. Remote Sensing for Land & Resources, 2010, (1): 1~13.
- [2] 王润生, 熊盛青, 聂洪峰, 等. 遥感地质勘查技术与应用研究 [J]. 地质学报, 2011, 85 (11): 1699~1743.

- WANG Run-sheng, XIONG Sheng-qing, NIE Hong-feng, et al. Remote sensing technology and its application in geological exploration [J]. *Acta Geologica Sinica*, 2011, 85 (11): 1699 ~ 1743.
- [ 3 ] 程宾洋. 高光谱遥感蚀变矿物填图算法并行设计与实现 [D]. 成都: 成都理工大学, 2013.
- CHENG Bin-yang. The parallel design and implementation of hyperspectral remote sensing mineral mapping algorithm [D]. Chengdu: Chengdu University of Technology, 2013.
- [ 4 ] Gillis D, Bowles J H. Parallel implementation of the ORASIS algorithm for remote sensing data analysis [C] // Plaza A J, Chang C I. *High performance computing in remote sensing*. US: Taylor & Francis Group, 2008.
- [ 5 ] Tilton J C. Parallel implementation of the recursive approximation of an unsupervised hierarchical segmentation algorithm [C] // Plaza A J, Chang C I. *High performance computing in remote sensing*. US: Taylor & Francis Group, 2008.
- [ 6 ] Wang Jianwei, Chang Chein-I. FPGA design for real-time implementation of constrained energy minimization for hyperspectral target detection [C] // Plaza A J, Chang C I. *High performance computing in remote sensing*. US: Taylor & Francis Group, 2008.
- [ 7 ] Setoain J, Prieto M, Tenllado C, et al. Parallel morphological endmember extraction using commodity graphics hardware [J]. *IEEE Geoscience and Remote Sensing Letters*, 2007, 4 (3): 441 ~ 445.
- [ 8 ] Agathos A, Li J, Petcu D, et al. Multi-GPU implementation of the minimum volume simplex analysis algorithm for hyperspectral unmixing [J]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2014, 7 (6): 2281 ~ 2296.
- [ 9 ] 杨靖宇, 张永生, 董广军. 基于 GPU 的遥感影像 SAM 分类算法并行化研究 [J]. *测绘科学*, 2010, 35 (3): 9 ~ 11.
- YANG Jing-yu, ZHANG Yong-sheng, DONG Guang-jun. Investigation of parallel method of RS image SAM algorithmic based on GPU [J]. *Science of Surveying and Mapping*, 2010, 35 (3): 9 ~ 11.
- [ 10 ] 罗耀华, 郭科, 赵仕波. 基于 GPU 的高光谱遥感 MNF 并行方法研究 [J]. *四川师范大学学报: 自然科学版*, 2013, 36 (3): 476 ~ 479.
- LUO Yao-hua, GUO Ke, ZHAO Shi-bo. Minimum noise fraction of hyperspectral remote sensing in parallel computing based on GPU [J]. *Journal of Sichuan Normal University: Natural Science*, 2013, 36 (3): 476 ~ 479.
- [ 11 ] 宋义刚, 叶舜, 吴泽彬, 等. 基于 GPU 的高光谱遥感图像 PPI 并行优化 [J]. *航天返回与遥感*, 2014, 35 (4): 74 ~ 80.
- SONG Yi-gang, YE Shun, WU Ze-bin, et al. Parallel optimization of Pixel Purity Index algorithm based on GPU for hyperspectral remote sensing image [J]. *Spacecraft Recovery & Remote Sensing*, 2014, 35 (4): 74 ~ 80.
- [ 12 ] Kruse F A, Lefkoff A B, Boardman J W, et al. The Spectral Image Processing System (SIPS): Interactive visualization and analysis of imaging spectrometer data [J]. *Remote Sensing of Environment*, 1993, 44: 145 ~ 163.
- [ 13 ] Boardman J W. Geometric mixture analysis of imaging spectrometry data [J]. *Proc. Int. Geoscience and Remote Sensing*



Symp, 1994, 4: 2369 ~ 2371.

- [14] 许宁, 胡玉新, 雷斌, 等. 一种基于 PPI 的高光谱数据矿物信息自动提取方法 [J]. 测绘科学, 2013, 38 (4): 138 ~ 141.

XU Ning, HU Yu-xin, LEI Bin, et al. Automated mineral information extraction based on PPI algorithm for hyperspectral image [J]. Science of Surveying and Mapping, 2013, 38 (4): 138 ~ 141.

- [15] Green A A, Berman M, Switzer P, et al. A transformation for ordering multispectral data in terms of image quality with implications for noise removal [J]. IEEE Transactions on Geoscience and Remote Sensing, 1988, 26 (1): 65 ~ 74.

## VALIDATION AND ANALYSIS OF HIGH PERFORMANCE COMPUTATION ON HYPERSPECTRAL IMAGERY BASED ON GPU

XU Ning<sup>1,2,3</sup>, XIAO Xin-yao<sup>1,2,3</sup>, HU Yu-xin<sup>1,2</sup>, WEN Jing<sup>4</sup>, WANG Da-ming<sup>5</sup>

(1. Key Laboratory of Technology in Geo-spatial Information Processing and Application System, IECAS, Beijing 100190, China;

2. Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China;

3. University of Chinese Academy of Sciences, Beijing 100049, China;

4. China Aero Geophysical Survey and Remote Sensing Centre for Land and Resources, Beijing 100083, China;

5. Oil & Gas Survey, China Geological Survey, Beijing 100029, China)

**Abstract:** Hyperspectral imagery has many characteristics, such as plenty of bands, large volume of data, high computing complexity. In recent years, high performance computation has been making great progress in remote sensing based on GPU, providing the hardware and technical conditions for the rapid processing of hyperspectral data. We implemented the experiments on a hyperspectral image which was obtained by Hyperion of EO-1 satellite in East Tianshan area, Xinjiang, using SAM and PPI algorithms based on CPU and GPU, trying to study the fast processing technology on hyperspectral data. . Actually the GPULib and CUDA API were used through IDL language and the data was tested by different algorithms. The results show that the processing efficiency of hyperspectral data in GPU is greater than CPU and the technology can be used in remote sensing image processing.

**Key words:** hyperspectral data; GPU; high performance computation; SAM; PPI