

基于集群和 GPU 的高光谱遥感影像并行处理

王茂芝, 郭 科, 徐文哲

(成都理工大学 数学地质四川省重点实验室, 四川 成都 610059)

摘 要: 以高光谱遥感影像数据处理中的主成分分析(PCA)和最小噪声分离(MNF)以及光谱相关系数填图(SCM)算法的并行化为目标,分别在集群环境下基于 MPI 设计并实现了协方差矩阵并行算法,以及基于 GPU 设计并实现了 SCM 并行算法,并在高光谱遥感影像数据处理中得到应用和验证。实验结果表明,高光谱遥感影像数据处理高性能计算对于提高和改善其时间性能具有显著效果,是高光谱遥感工程化应用快速处理重要的技术手段。

关键词: 高光谱遥感; 高性能计算; 集群; GPU

中图分类号: TP751.1 **文献标志码:** A **文章编号:** 1007-2276(2013)11-3070-06

Hyperspectral remote sensing image parallel processing based on cluster and GPU

Wang Maozhi, Guo Ke, Xu Wenxi

(Geomathematics Key Lab of Sichuan Province, Chengdu University of Technology, Chengdu 610059, China)

Abstract: The parallel algorithms design and implementation of covariance matrix, related to PCA and MNF, and SCM used in hyperspectral remote sensing image data process was discussed in this paper. The covariance matrix parallel algorithm was designed and implemented under cluster circumstance based on MPI. On the other hand, parallel algorithm of SCM was designed and implemented based on GPU. Both of these two parallel algorithms were verified during the application on hyperspectral remote sensing data processing. The experiment results prove that high performance computing is effective during the data process of hyperspectral remote sensing image, and it should be an important technique for generalization of hyperspectral remote sensing to engineering quick application. The results also prove the correctness of the parallel algorithms proposed in this paper.

Key words: hyperspectral remote sensing; high performance computing; cluster; GPU

收稿日期: 2013-03-19; 修订日期: 2013-04-24

基金项目: 863 计划“宽幅高光谱小卫星载荷关键技术研究”重点项目(2008AA121103);

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

中国地质调查局地调项目“航空高光谱成像系统集成与示范应用”(1212011120227);

四川省教育厅自然科学重点项目“高光谱遥感影像并行计算关键技术研究”(13ZA0065)

作者简介: 王茂芝(1974-), 男, 副教授, 博士, 主要从事高光谱遥感及其应用方面的研究。Email: wangmz@cdut.edu.cn

0 引言

随着光谱分辨率和空间分辨率的不断提升,光谱成像仪获取的高光谱遥感影像数据容量成海量级方式增长,导致其对处理平台和环境提出了较高的要求,同时也对高光谱遥感影像的处理算法在性能提出了挑战,计算机硬件技术和高性能计算技术的发展为缓解这种压力在平台和技术上提供了支撑。参考文献[1]从“基于核方法的高光谱数据分类”、“光谱和空间信息结合的影像处理”以及“并行实现”三个方面对目前高光谱遥感影像处理技术的进展进行了描述,指出了并行计算技术和方法在高光谱遥感影像处理中的重要性。参考文献[2]基于同构和异构并行环境对高光谱遥感影像处理中的分布主成分变换、分布迭代自组织数据分析技术、以及自动形态端元提取三个方面的并行算法设计与实现进行了探讨。国内学者在高(多)光谱遥感影像集群并行处理方面也开展了相关研究,如参考文献[3]在集群环境下设计实现了投影寻踪并行算法,参考文献[4]提出了一种基于尺度的影像数据划分策略及其 MPI(Message Passing Interface)并行处理。参考文献[5]基于 MPI 实现了灰度相关匹配算法和快速傅里叶变换算法。

近年来,随着 GPU (Graphic Processing Unit) 技术的发展,基于 CUDA (Compute Unified Device Architecture) 的高光谱遥感影像并行处理逐步得到应用^[6-7]。GPU 由于其强大的浮点计算能力在通用并行计算方面有巨大的潜力^[8],但是在 CUDA 架构出现之前,开发人员只能通过 OpenGL 或者 DirectX 等 API (Application Programming Interface) 来访问 GPU,这大大地限制了 GPU 在通用并行计算领域的推广^[9]。CUDA 架构的出现很好地解决了上述问题。

文中以集群环境下高光谱遥感数据处理 PCA (Principal Component Analysis) 和 MNF (Minimum Noise Fraction) 算法中的协方差矩阵并行算法设计和实现,以及基于 GPU 的 SCM (Spectral Correlation Coefficient Mapping) 并行填图为例,探讨高光谱遥感数据并行处理中涉及的并行算法设计、并行程序实现、并行性能分析等系列问题,为高光谱遥感影像工程化应用高性能计算提供一些参考。

1 高光谱遥感影像并行处理实例

1.1 基于集群的协方差矩阵并行计算及其在高光谱遥感数据处理中的应用

1.1.1 几个定义

n 维随机变量 (X_1, X_2, \dots, X_n) 的协方差矩阵定义为 $C = (c_{ij})_{n \times n}$ 。其中 $c_{ij} = \text{Cov}(X_i, X_j) = E\{[X_i - E(X_i)][X_j - E(X_j)]\}$, $i, j = 1, 2, \dots, n$, 显然 $c_{ij} = c_{ji}$ 。

在实际应用中,均值及协方差矩阵通常通过样本进行估算,假设随机变量 X 和 Y 的观察值分别为 x_k 和 $y_k (k=1, 2, \dots, m)$ 。均值估计公式定义如下:

$$\mu \equiv \bar{x} = \frac{1}{m} \sum_{k=1}^m x_k \quad (1)$$

随机变量 X 和 Y 的协方差估计定义为:

$$c_{xy} = \text{Cov}(X, Y) \equiv S_{xy} = \frac{1}{m-1} \sum_{k=1}^m (x_k - \bar{x})(y_k - \bar{y}) = \frac{1}{m-1} \sum_{k=1}^m (x_k - \bar{x})(y_k - \bar{y}) \quad (2)$$

其中, $\bar{x} = \frac{1}{m} \sum_{k=1}^m x_k$, $\bar{y} = \frac{1}{m} \sum_{k=1}^m y_k$ 。

在高光谱遥感影像数据立方体中,随机变量 X 或 Y 的含义为影像空间所有像元在某个波段上的取值, c_{xy} 的含义为 X 和 Y 对应两个波段的协方差, X 和 Y 的观察值 x_k 和 y_k 即为影像空间所有像元在对应波段的反射率值。

1.1.2 并行算法设计

显然,从公式(2)可以看出,要得到协方差矩阵元素值需要先计算样本均值 \bar{x} , \bar{y} , 所以问题转化为 \bar{x} , \bar{y} 以及公式(2)的并行计算。限于篇幅,基于上述思路的并行算法在此略去,文中提供一种转化后的并行算法,先把公式(2)转化为公式(3)。

$$c_{ij} = c_{xy} \equiv S_{ij} = S_{xy} = \frac{1}{m-1} \sum_{k=1}^m (x_k - \bar{x})(y_k - \bar{y}) = \frac{1}{m-1} \sum_{k=1}^m (x_k y_k - x_k \bar{y} - \bar{x} y_k + \bar{x} \bar{y}) = \frac{1}{m-1} \left(\sum_{k=1}^m x_k y_k - \sum_{k=1}^m x_k \bar{y} - \sum_{k=1}^m \bar{x} y_k + \sum_{k=1}^m \bar{x} \bar{y} \right) = \frac{1}{m-1} \left(\sum_{k=1}^m x_k y_k - \bar{y} \sum_{k=1}^m x_k - \bar{x} \sum_{k=1}^m y_k + m \bar{x} \bar{y} \right) = \frac{1}{m-1} \left[\sum_{k=1}^m x_k y_k - \bar{y} m \left(\frac{1}{m} \sum_{k=1}^m x_k \right) - \bar{x} m \left(\frac{1}{m} \sum_{k=1}^m y_k \right) + m \bar{x} \bar{y} \right] =$$

$$\begin{aligned} \frac{1}{m-1} \left(\sum_{k=1}^m x_k y_k - \bar{y} m \bar{x} - \bar{x} m \bar{y} + m \bar{x} \bar{y} \right) &= \frac{1}{m-1} \left(\sum_{k=1}^m x_k y_k - m \bar{x} \bar{y} \right) = \\ \frac{1}{m-1} \left(\sum_{k=1}^m x_k y_k - m \times \frac{1}{m} \sum_{k=1}^m x_k \times \frac{1}{m} \sum_{k=1}^m y_k \right) &= \\ \frac{1}{m-1} \left(\sum_{k=1}^m x_k y_k - \frac{1}{m} \sum_{k=1}^m x_k \sum_{k=1}^m y_k \right) & \quad (3) \end{aligned}$$

设样本容量为 m , 并行计算节点数为 p ($p \geq 2$), 记 $q = \lfloor \frac{m}{p} \rfloor$, $rem = m \bmod p$. 则上式可进一步展开成如下形式。

$$\begin{aligned} c_{ij} = C_{XY} &= \frac{1}{m-1} \sum_{k=1}^m (x_k - \bar{x})(y_k - \bar{y}) = \\ \frac{1}{m-1} \left(\sum_{k=1}^m x_k y_k - \frac{1}{m} \sum_{k=1}^m x_k \sum_{k=1}^m y_k \right) &= \\ \frac{1}{m-1} \left[\left(\sum_{k=1}^q x_k y_k + \sum_{k=q+1}^{2q} x_k y_k + \dots + \sum_{k=(p-1) \times q+1}^{p \times q+rem} x_k y_k \right) \right. & \\ \left. \frac{1}{m} \left(\sum_{k=1}^q x_k \sum_{k=1}^q y_k + \sum_{k=q+1}^{2q} x_k \sum_{k=q+1}^{2q} y_k + \dots + \sum_{k=(p-1) \times q+1}^{p \times q+rem} x_k \sum_{k=(p-1) \times q+1}^{p \times q+rem} y_k \right) \right] & \quad (4) \end{aligned}$$

由此得到协方差矩阵并行计算算法, 描述如下:

(1) 假定协方差矩阵维数为 n , 其第 i 行和第 j 列对应的随机变量为 X 和 Y , 协方差为 $c_{ij} = C_{XY}$. 设样本容量为 m , 随机变量 X 和 Y 的样本观察值为 x_k, y_k , $k=1, 2, \dots, m$, 并行计算节点数为 p ($p \geq 2$), 记 $q = \lfloor \frac{m}{p} \rfloor$, $rem = m \bmod p$. 假定第 p 个节点为根进程所在节点。

(2) 根据公式(4), 给各计算节点分配局部样本数据并计算局部样本乘积和以及局部样本和之积。节点 r ($r=1, \dots, p-1$) 的局部样本数据为 x_k, y_k ($k=(r-1) \times q+1, \dots, r \times q$), 其局部样本乘积和以及局部样本和之积分别为 $\sum_{k=(r-1) \times q+1}^{r \times q} x_k y_k$ 与 $\sum_{k=(r-1) \times q+1}^{r \times q} x_k \sum_{k=(r-1) \times q+1}^{r \times q} y_k$; 节点 p 的局部样本数据为 x_k, y_k ($k=(p-1) \times q+1, \dots, p \times q+rem$), 其局部样本乘积和以及局部样本和之积分别为

$$\sum_{k=(p-1) \times q+1}^{p \times q+rem} x_k y_k \text{ 与 } \sum_{k=(p-1) \times q+1}^{p \times q+rem} x_k \sum_{k=(p-1) \times q+1}^{p \times q+rem} y_k.$$

(3) 除根进程节点外, 各计算节点 r ($r=1, \dots, p-1$) 把由第(2)步计算结果(局部样本乘积和以及局部样本和之积)传送给根进程节点 p , 并由根进程节点根据

$$\begin{aligned} c_{ij} = C_{XY} &= \frac{1}{m-1} \left[\left(\sum_{k=1}^q x_k y_k + \sum_{k=q+1}^{2q} x_k y_k + \dots + \sum_{k=(p-1) \times q+1}^{p \times q+rem} x_k y_k \right) - \right. \\ \left. \frac{1}{m} \left(\sum_{k=1}^q x_k \sum_{k=1}^q y_k + \sum_{k=q+1}^{2q} x_k \sum_{k=q+1}^{2q} y_k + \dots + \sum_{k=(p-1) \times q+1}^{p \times q+rem} x_k \sum_{k=(p-1) \times q+1}^{p \times q+rem} y_k \right) \right] & \text{ 计} \end{aligned}$$

算样本协方差 $c_{ij} = C_{XY}$ 。

(4) 遍历 i, j ($i, j=1, 2, \dots, n$), 重复算法(2)-(3)步, 计算 $C_{ij} = C_{XY}$. 由于协方差矩阵为对称阵, 所以只需遍历矩阵的上三角或下三角就完成了协方差矩阵的计算。

1.1.3 高光谱遥感影像协方差计算数据映射

在高光谱遥感影像处理中, 经常需要对影像数据进行 PCA 和 MNF 变换以实现对其降维、特征提取等目的^[10], PCA 和 MNF 算法的重要内容之一就是协方差矩阵的计算。下面以高光谱遥感影像数据处理中协方差矩阵计算为例实现上述并行算法。其数据映射与协方差矩阵构建描述如下。

高光谱遥感影像数据在形式上表现为数据立方体, 假定数据立方体的 Z 轴表示高光谱遥感影像的波段, X, Y 轴表示影像数据的空间位置信息。PCA 和 MNF 算法中的协方差矩阵反映的是高光谱遥感影像不同波段之间的关系, 也就是说, 此时协方差矩阵的阶数为高光谱遥感影像的波段数, 每一个波段的数据元素个数为 X, Y 平面的像元数。从而对于高光谱遥感影像的协方差矩阵计算数据做如下映射。

假定第 i 和第 j 波段对应的随机变量记为 X, Y , 则其协方差为:

$$\begin{aligned} c_{ij} = C_{XY} = \text{Cov}(X, Y) &= \frac{1}{m-1} \sum_{k=1}^m (X_k - \bar{X})(Y_k - \bar{Y}) = \\ \frac{1}{m-1} \sum_{k=1}^m (x_k - \bar{x})(y_k - \bar{y}) & \end{aligned}$$

式中: m 指第 i 或第 j 波段的像元数, 也称为高光谱遥感影像的大小。从而高光谱遥感影像的协方差矩阵记为 $C = (c_{ij})_{n \times n}$, 其中 n 为高光谱遥感影像的波段数。 c_{ij} 的计算按公式(4)进行。上述数据映射并行算法应用实验结果及性能分析见 2.1 小节。

1.2 基于 GPU 的 SCM 并行填图

1.2.1 SCM 算法简介

SCM 是一种常见的高光谱图像监督分类算法, 它通过计算像元光谱矢量与波谱库光谱矢量之间的相关系数来确定二者之间的相似度, 相关系数越大, 说明两者越相似。假设 n 个波段的两种光谱矢量分别为 $T = (t_1, t_2, \dots, t_n)$, $R = (r_1, r_2, \dots, r_n)$, 则相关系数 r 的求解公式为:

$$r = \frac{\sum (T - \bar{T})(R - \bar{R})}{\sqrt{[\sum (T - \bar{T})]^2 [\sum (R - \bar{R})]^2}}$$

1.2.2 SCM 并行算法设计

文中利用 CPU 与 GPU 协同编程模型实现 SCM 算法的并行化。其中 CPU 用于处理串行任务:包括高光谱影像数据的读取,GPU 显存的分配,填图等工作;GPU 则处理并行任务:基于波谱库的并行插值以及 SCM 光谱相关系数并行计算。由于原始影像上每个像元与训练样本集的计算过程是完全独立的,所以 SCM 算法具有非常高的数据并行性。SCM 算法的并行化分两步实施:第一步由波谱库并行插值产生参考光谱,第二步以像元点为基准并行计算相关系数。下面分别描述。

由于用于匹配或分类识别的波谱库波段数据范围和实际的高光谱遥感影像波段数据范围不一致,所以需要利用波谱库数据进行插值得到和实际高光谱遥感影像波段数据相匹配的数据。文中采用线性插值进行处理。假设原始高光谱影像有 N 个波段,波谱库中的待识别地物有 M 种,则将进行 $N * M$ 次插值运算,注意到针对每种地物而言其插值运算是独立的,从而采用在 GPU 中产生 M 个线程,每个线程负责一种地物参考光谱插值运算的并行策略来实现光谱插值算法的并行化。

对于 SCM 相关系数的并行计算,由 SCM 计算公式可知各像元点与波谱库中待识别地物的相关系数计算是相互独立的,文中采用的并行策略是以像元点数目为基准在 GPU 中产生相应的线程数,每个线程负责一个像元点相关系数的计算,并根据计算结果对该像元点分类。需要说明的是,基于波普库的并行插值算法描述与 SCM 的并行计算相似。主要区别有两点,一是在开始的 CPU 端代码做数据准备时所执行的操作为:拷贝遥感影像波段长度值数据以及波普库待插值地物数据到显存;二是 GPU 端代码中线程数由待插值的波普库地物数决定,为 M 。

2 实验及结果

2.1 协方差矩阵并行计算结果

文中基于 MPI 在集群环境下实现了上述协方差并行算法。集群包含 16 个计算结点,每个结点由微机组成(配置为超越 E705/2.53 GHz 双核,2 GB 内存),计算结点通过局域网互联形成集群并行计算硬件环境,软件编程环境为 MPICH2。实验测试数据

来源于 863 项目,由中国国土资源航空物探遥感中心提供。数据波段数为 158,X,Y 轴的空间分辨率都为 30 m,其中 X 方向像元数为 191,Y 方向像元数为 273。所以,协方差矩阵计算中的参数 $n=158,m=191 \times 273=52143$ 。实验结果见表 1。从表 1 可以分析同一数据量在不同节点下的时间性能,见图 1,以及相同计算节点不同数据规模的时间性能,见图 2。

表 1 不同数据量和节点数的并行时间对比

Tab.1 Parallel computing time of different data size and different computing node

Time/s	Data sizes(bands*rows*columns)			
	158*191*273	158*382*273	158*382*546	158*764*546
1	12.719	26.404 1	52.063	105.594
2	6.4351	12.606 4	37.135 9	74.761 4
4	7.047 6	14.801 9	26.479 2	51.441 6
8	8.456 7	16.612 6	34.365 6	55.003 7
16	9.990 6	20.691 4	38.680 4	64.905 9

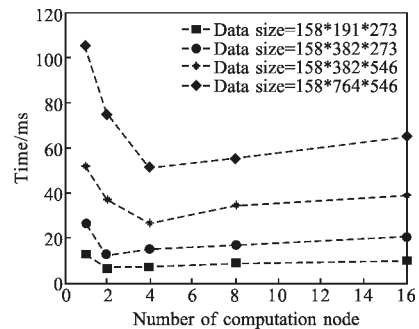


图 1 固定负载不同计算节点数的性能对比

Fig.1 Performance comparison of fixed load and different computing node

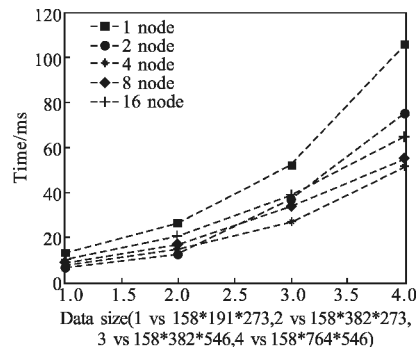


图 2 相同计算节点数不同负载的性能对比

Fig.2 Performance comparison of fixed computing node and different load

基于集群的并行实验性能分析,可以从固定加速比模型(Gustafsun 定律)和固定负载模型(Amdahl 定律)两个方面展开。对于 Gustafsun 定律,对比表 1 的第 1 列和第 3 列可以发现:当计算量为 $158 \times 191 \times 273$, 计算节点为 2 时,加速比为 1.98,如第 1 列所示;当计算量和计算节点都倍增时,即计算量为 $158 \times 382 \times 546$, 计算节点为 4 时,加速比为 1.97,如第 3 列所示。对比第 2 列和第 4 列可以得到同样结论,此时加速比分别为 2.09(第 2 列)和 2.05(第 4 列)。实验结果不仅验证了 Gustafsun 定律,同时从另外一个角度说明文中并行算法设计和并程序实现的正确性。对于 Amdahl 定律,从图 1 可以看出,当固定计算负载时,计算时间的总体变化趋势随着计算节点的增加是下降的。但由于串行因素的影响,加速比不会随着计算节点的增加而一直递增,如何确定最优计算节点数是一个值得深入研究的问题。

2.2 SCM 并行填图结果

基于 GPU 的 SCM 并行填图实验环境配置为 Intel core(TM)i7 CPU, 2.80 GHz 处理器, 4.0 GB 内存, NVIDIA GeForce 310M 显卡, Win7 32 位操作系统。CPU 编程环境为 Microsoft Visual C++ 2008, GPU 编程语言为 CUDA, 绘图采用 OpenGL, 实验数据是江西某地的高光谱遥感数据, 实验结果如表 2 所示。

表 2 SCM 并行算法时间性能对比

Tab.2 Performance comparison parallel algorithm of SCM

Data sizes	$158 \times 191 \times 273$	$158 \times 382 \times 273$	$158 \times 382 \times 546$	$158 \times 382 \times 1092$
Serial time /ms	16.57	29.83	48.14	98.39
Parallel time/ms	359	717	1139	2527
Speed-up ratio	21.6	24.04	24.28	25.68

从表 2 可以看出,基于 GPU 的 SCM 串并行填图算法时间消耗随着数据量的倍增大致呈线性递增方式变化,其加速比虽有改善,但递增幅度不大。如何根据原始数据规模和实验软硬件环境确定并行策略、数据分发方式、通信处理模式等问题是 GPU 并行值得进一步深入分析和探讨的问题。

3 结论

虽然测试数据在某种程度上不是海量级别,从实例并行计算结果仍然可以得到如下几个结论。

(1) 并行算法能够有效地改善和提高算法时间性能。对于高光谱遥感海量数据的处理引入高性能计算技术手段是可行的。在高光谱遥感海量数据的工程实际应用中,介入高性能计算技术手段是必要的。文中实验加速比结果表明引入并行计算后算法性能都得到提升。

(2) 不同应用和算法所采用的并行策略以及并行实现平台导致的并行性能有较大的差异。如在文中测试数据下,基于集群的 PCA 并行算法其加速比为 2 左右,而基于 GPU 的 SCM 并行填图其加速比可达 25 左右。如何根据算法特性(如计算型、通信型、混合型等),适应应用差异,设计并行算法,选择并行计算平台和编程环境,取得最佳并行效果是值得深入探讨的问题。特别地,基于 CPU 和 GPU 混合构架的并行算法设计和实现是值得重点关注的技术。

(3) 实验结果及性能分析表明文中提出的并行算法在设计和实现以及应用上是正确的。

值得进一步探讨的问题包括:

(1) 在并行算法设计过程中,如何适应不同应用确定最优并行计算节点以及搭建相应并行计算平台和环境值得进一步深入研究,参考文献[11]提供了一些思路。

(2) 在并行算法实现和并行编程过程中,数据分发和处理方式以及编程模式的选择对于并行性能具有较大影响。参考文献[12]提供了一些参考。

(3) 不同并行环境下,如基于 GPU^[13]和集群^[3]的性能对比分析,以及基于 CPU+GPU 的混合构架并行算法设计、实现及性能分析,都是值得进一步研究的课题。

参考文献:

- [1] Antonio Plaza, Jon Atli Benediktsson, Boardman Joseph W, et al. recent advances in techniques for hyperspectral image processing [J]. Remote Sensing of Environment, 2009, 113: S110-S122.
- [2] Antonio Plaza, David Valencia, Javier Plaza. An experimental comparison of parallel algorithms for hyperspectral analysis

- using heterogeneous and homogeneous networks of workstations[J]. *Parallel Computing*, 2008, 34: 92-114.
- [3] Wang Wei, Zhao Huijie, Dong Chao. Parallel algorithm of anomalies detection in hyperspectral image with projection pursuit[J]. *Journal of Beijing University of Aeronautics and Astronautics*, 2009, 35(3): 342-346. (in Chinese)
王维, 赵慧洁, 董超. 基于投影寻踪的高光谱图像异常检测并行算法 [J]. *北京航空航天大学学报*, 2009, 35(3): 342-346.
- [4] Shen Zhanfeng, Luo Jiancheng, Chen Qiuxiao, et al. High-efficiency remotely sensed image parallel processing method study based on MPI [J]. *Journal of Image and Graphics*, 2007, 12(12): 2132-2136. (in Chinese)
沈占锋, 骆剑承, 陈秋晓, 等. 基于 MPI 的遥感影像高效能并行处理方法研究 [J]. *中国图象图形学报*, 2007, 12(12): 2132-2136.
- [5] Lv Jie, Zhang Tianxu, Zhang Biyin. Applications of MPI parallel computing on image processing [J]. *Infrared and Laser Engineering*, 2004, 33(5): 496-499. (in Chinese)
吕捷, 张天序, 张必银. MPI 并行计算在图像处理方面的应用[J]. *红外与激光工程*. 2004, 33(5): 496-499.
- [6] Yang Renzhong, Chen Minhao, Shi Lu. Interferential hyperspectral real time spectrum reconstruction technology based on CUDA[J]. *Remote Sensing Technology and Application*, 2011, 26(4): 420-425. (in Chinese)
杨仁忠, 陈敏浩, 石璐. 基于 CUDA 的干涉型高光谱实时光谱复原处理技术 [J]. *遥感技术与应用*, 2011, 26(4): 420-425.
- [7] He Guojing, Liu Delian, Zhang Jianqi. High speed spectral matching approach for hyperspectral image based on CUDA [J]. *Aero Weaponry*, 2011, 4: 3-6. (in Chinese)
何国经, 刘德连, 张建奇. CUDA 架构下高光谱图像光谱匹配的快速实现[J]. *航空兵器*, 2011, 4: 3-6.
- [8] Shi Kun, Hao Yingming, Wang Mingming, et al. Real-time simulation method of infrared sea background [J]. *Infrared and Laser Engineering*, 2012, 41(1): 25-29. (in Chinese)
石坤, 郝颖明, 王明明, 等. 海面背景红外实时仿真[J]. *红外与激光工程*, 2012, 41(1): 25-29.
- [9] Jason Sanders, Edward Kandrot. *CUDA by Example: An Introduction to General-Purpose GPU Programming* [M]. Boston: Addison-Wesley Professional, 2011: 126-150.
- [10] Wei Feng, He Mingyi, Mei Shaohui. Hyperspectral data feature extraction using spatial coherence based neighborhood preserving embedding [J]. *Infrared and Laser Engineering*, 2012, 41(5): 1249-1254. (in Chinese)
魏峰, 何明一, 梅少辉. 空间一致性邻域保留嵌入的高光谱数据特征提取[J]. *红外与激光工程*, 2012, 41(5): 1249-1254.
- [11] Tiranee Achalakula, Stephen Taylor. A distributed spectral-screening PCT algorithm [J]. *J Parallel Distrib Comput*, 2003, 63: 373-384.
- [12] Kleantlis Psarris. Program analysis techniques for transforming programs for parallel execution [J]. *Parallel Computing*, 2002, 28: 455-469.
- [13] Shane Ryoo, Christopher I Rodrigues, Sam S Stone, et al. Program optimization carving for GPU computing [J]. *J Parallel Distrib Comput*, 2008, 68: 1389-1401.