doi: 10.6046/zrzyyg.2020253

引用格式:赵晓晨,吴皓楠,李林宜,等. 面向汛旱情监测的遥感影像 GPU 并行处理算法[J]. 自然资源遥感,2021,33(3):107-113. (Zhao X C, Wu H N, Li L Y. GPU – based parallel image processing algorithm for flood and drought monitoring[J]. Remote Sensing for Natural Resources,2021,33(3):107-113.)

面向汛旱情监测的遥感影像 GPU 并行处理算法

赵晓晨1,吴皓楠2,李林宜1,孟令奎1

(1. 武汉大学遥感信息工程学院,武汉 430079; 2. 珠江水利委员会珠江水利综合技术中心,广州 510611)

摘要:针对面向汛旱情监测应用中遥感影像处理耗时过长的问题,包括辐射校正、几何纠正、遥感指数计算等过程, 对其业务化工作流程进行了分解分析。结合统一计算架构(compute unified device architecture,CUDA)的存储结构 和程序设计模型,将数据处理过程划分为数据读取、直方图统计、栅格分割、波段计算、重采样和数据输出等模块, 对波段计算及重采样等模块设计了并行处理方案,并通过实验确定了栅格划分的最佳尺度,基于栅格数组图形处 理器(graphics processing unit,GPU)映射方法加速了数据传输效率,最终提出了基于 CUDA 架构 CPU – GPU 协同的 并行处理算法。实验结果表明,辐射校正及遥感指数计算的波段计算模块可节约 58.9% 的时间;几何纠正效果最 为显著,最邻近像元重采样和双线性内插重采样模块的最终加速比分别能够达到9 倍和7 倍以上。

关键词: 汛旱情监测; 几何纠正; 遥感指数计算; GPU; CUDA

中图法分类号: TP 751; P 237 文献标志码: A 文章编号: 2097-034X(2021)03-0107-07

0 引言

在汛旱情监测领域,遥感技术以其长期动态监测的优势,成为强有力的技术手段之一。遥感对地观测技术飞速发展的同时,带来了文件体积过大、处理耗时过长的问题,因此,提高数据处理效率、减少耗时成为亟待解决的问题之一。

汛情及洪涝监测应用的关键在于水体范围及水体变化的信息提取,目前常用归一化差异水体指数(normalized difference water index,NDWI)、改进的归一化差异水体指数(modified NDWI,MNDWI)、增强型水体指数(enhanced water index,WI)等方法提取水体信息,取得了一定成效。旱情监测的主要方法之一是利用植被指数研究不同时期作物长势,经典的指数包括归一化植被指数(normalized difference vegetation index,NDVI)、比值植被指数(ratio vegetation index,RVI)、植被状态指数(vegetation condition index,VCI)等。

统一计算架构(compute unified device architecture,CUDA)的基本思想是驱动程序将密集计算数 据划分成细小的区块,映射、加载到图形处理器 (graphics processing unit,GPU)中并行执行,尽量开 发线程级并行并将这些线程在硬件中动态调度和执行^[1-2]。CUDA采用主机程序嵌套内核函数的程序设计模式,主机/设备代码分离的编译模式,数据由主机端传输至设备端,并调用设备端的内核函数派生大量独立线程并行执行。GPU与中央处理器(central processing unit,CPU)协同处理技术是将GPU视为高性能并行计算设备,将CPU称为主机端,把GPU看作CPU的协处理器,避免了从通用算法到应用程序接口(application programming interface,API)的复杂映射,从而直接使用GPU的计算能力,实现了异构环境下对GPU资源的灵活调用。

本文在分析了遥感影像处理原理的基础上,将 处理过程划分为若干模块,使用 CUDA 技术对影像 处理方法做并行改进,提出了基于 CUDA 架构的 CPU-GPU 协同的遥感影像预处理及相关指数计算 并行算法,并选取了典型影像数据对该算法进行验 证,证明其有效性。

1 遥感影像预处理及指数算法分析

汛旱情监测应用中的遥感影像处理主要包括辐射校正、几何纠正、遥感指数计算等过程,这里对传统算法原理进行分解分析,并确定可优化模块。

收稿日期: 2020-08-17;修订日期: 2020-11-15

基金项目:国家重点研发计划课题"河湖岸线洲滩立体监测及河湖功能动态评估关键技术研究"(编号:2018YFC0407804)资助。

第一作者:赵晓晨(1995-),女,硕士研究生,主要从事水利遥感方面的研究。Email: zhaoxiaochen@ whu. edu. cn。

通信作者: 孟令奎(1967-),男,教授,主要从事网络 GIS、水利遥感技术及应用方面的研究。Email: Lkmeng@ whu. edu. cn。

1.1 辐射校正

辐射校正的目的是将影像像元量化的、无量纲 的像元灰度值定标为辐射亮度值或表观反射率,通 常定标为表观反射率的计算表达式为:

$$L = GainDN + Bias , \qquad (1)$$

式中:L为大气表观反射率;DN为像元灰度值; Gain为辐射增益参数;Bias为辐射偏置参数,Gain 和Bias统称为辐射定标参数。

实现辐射定标首先要读取影像头文件,获取辐射增益、辐射偏置、太阳高度角等元数据信息,其次 根据式(1)对各波段影像栅格进行波段运算,考虑 对波段运算处理过程进行并行优化。

1.2 几何纠正

影像几何纠正的目的是消除图像的几何畸变, 该过程是遥感数据预处理中耗时最长、计算量最大 的过程之一,也成为遥感影像快速完成处理的瓶颈。 这里采用有理多项式系数(rational polynomial coefficient, RPC)模型对影像纠正(图1),相关研究表 明^[3], RPC 模型相对于严格成像模型的拟合误差不 超过0.04 个像元,均方差小于0.01 个像元,因此可 以看作对严格成像模型的拟合。





Fig. 1 Geometric correction algorithm of RPC model

由图1知, RPC纠正算法的实际开发内容主要包括以下4个模块:①影像读取及批处理任务管理;②RPC正解函数构建;③RPC反解函数构建; ④纠正后影像栅格重采样。

前3个模块运行效率主要依赖于计算机存储设 备性能,重采样是 RPC 纠正中耗时最长的步骤。使 用传统的串行程序设计方式处理时,对所有像素依 次循环处理,造成了极大的流程阻塞,成为算法的最 大瓶颈。同时由于每个像素的重采样独立进行,算 法具有可并行性,因此考虑作为主要并行化模块进 行设计。

- 1.3 遥感指数计算
- 1.3.1 汛情遥感监测算法

水体范围变化是汛情及洪涝灾害的主要特征, 水体信息的有效提取也一直是水利遥感技术研究的 重点。常用的水体光谱指数有 NDWI^[4]和 MND-WI^[5]等,具体表达式分别为:

$$NDWI = (R_{\text{Green}} - R_{\text{NIR}}) / (R_{\text{Green}} + R_{\text{NIR}}) , \quad (2)$$

$$MNDWI = (R_{Green} - R_{MIR}) / (R_{Green} + R_{MIR}) , (3)$$

式中, R_{NIR}, R_{Green}和 R_{MIR}分别为近红外、绿光和中红 外波段反射率。

1.3.2 旱情遥感监测算法

NDVI, VCI和 EVI 等遥感指数可用于监测植被 长势,进而表征干旱程度。具体表达式分别为:

$$NDVI = (R_{\rm NIR} - R_{\rm Red})/(R_{\rm NIR} + R_{\rm Red})$$
 , (4)

 $VCI = (NDVI - NDVI_{min}) / (NDVI_{max} - NDVI_{min}) ,$ (5)

$$EVI = G \frac{R_{\rm NIR} - R_{\rm Red}}{R_{\rm NIR} + C_1 R_{\rm Red} - C_2 R_{\rm Blue} + L} , (6)$$

式中: R_{Red} 和 R_{Blue} 分别为红光和蓝光波段反射率; L 为土壤调节参数, L = 1; C_1 为红光大气改正参数, $C_1 = 6$; C_2 为蓝光大气改正参数, $C_2 = 7.5$; G为增益系数。

遥感指数计算在算法上无需直方图统计、栅格 索引变更等复杂过程,仅包含不同波段影像栅格数 据中同名像元的波段计算,这一过程内在并行性强, 因此将波段运算模块作为并行化改造的首要任务。

2 CUDA 并行算法设计

遥感影像处理算法并行模式如图 2 所示,遥感 影像预处理算法通常以影像数据的管理与解析作为 起始功能模块,将影像元数据和光谱数据栅格分别



应用于不同的算法模块。其中元数据主要用于相关 模型的构建及参数求解,而光谱数据作为影像信息 的主要承载,是包括波段运算、栅格重采样、统计学 分析、矢量化等多种算法模块的数据来源,其中波段 运算和栅格重采样具有较强的区域并行性,适合进 行并行化改造。同时不同功能模块间存在复杂的协 同关系,从而使算法理论能够准确地应用于光谱数 据本身最终完成产品的输出。

2.1 辐射校正并行算法

根据 1.1 节的原理分析,辐射校正算法的效率 瓶颈集中在波段运算环节。相比于遥感指数计算, 辐射定标涉及的波段更多,公式更复杂,分支结构更 多,计算密集度更高。从影像头文件中获取辐射偏 置、辐射增益、太阳高度角等定标参数,也是算法的 重要环节。结合 GPU 存储方式的分析以及定标参 数在算法中定位,实际开发过程中使用__constant__ 将参数映射至常量内存,提高算法效率。

首先采用 GDAL (geospatial data abstraction library) 库用于数据解析,并通过注册影像驱动的方式,实现 Arc/Info ASCII Grid(asc), GeoTiff (tiff), Erdas Imagine Images(img)和 ASCII DEM(dem)等不同格式的影像文件读取。

影像数据解析为栅格数组后,以数据集和波段 2级数据类型管理,针对异构环境下设备端显存空 间有限的问题,采用条带划分、分批次处理模式。这 种单一的波段运算不涉及空间运算,仅针对同一像 元位置不同波段灰度值,因此对栅格数组逻辑分块 可以避免多余的空间信息赋值工作,并且根据 GPU 的需求弹性调整分块大小。

使用核函数编写波段计算是实现并行处理的核 心步骤,实际编写中有以下关键点:①使用__global __函数限定符,主机端负责调用,设备端负责执行; ②在核函数中完成栅格数组索引与线程逻辑管理索 引的正确映射;③算法公式中涉及的变量选取合适 的变量类型确保分配足够的显存空间;③在正确表 述算法公式外,使用分支结构处理函数异常值。

其中线程管理索引决定了栅格数组元素在线程 逻辑结构中的分配方式,其公式为:

$$I = (b_{y} G_{x} + b_{x}) (B_{x} B_{y}) + t_{y} B_{x} + t_{x} , \quad (7)$$

式中: I 为栅格数组索引; $b_x \pi b_y \beta$ 别为线程块在网 格中 $x \pi y$ 方向上的索引, 对应内建变量 blockIdx. x和 blockIdx. y; G_x 为一个网格中 x 方向上线程块总 数, 对应内建变量 gridDim. x; $B_x \pi B_y \beta$ 别为一个线 程块中 $x \pi y$ 方向上线程总数, 对应内建变量 blockDim. x 和 blockDim. y; $t_x \pi t_y \beta$ 别为线程在线 程区块中 x 和 y 方向上的索引,对应内建变量 threadIdx. x 和 threadIdx. y。本文结合二维影像数据自身 特点,同样使用二维格网、线程块进行管理。

另外,由于核函数中采用了少量分支结构用于波段运算中异常值的处理,可能造成线程块中执行速率不一致。为此可使用 CUDA 中内建函数 cudaThread-Synchronize()保证同一线程块中各线程状态对齐。

综上,针对波段运算的并行算法设计中,主机端 负责批处理任务编排、影像文件读取和输出、影像栅格 数组划分等耗时较短的串行任务,设备端负责对影像 栅格数据并行执行核函数,完成指数算法及异常值处 理。这一设计也是 RPC 几何纠正并行算法的基础。

2.2 基于 RPC 模型的几何纠正并行算法

通过1.2 节中对基于 RPC 模型的几何纠正算 法的原理分析,结合 CUDA 程序设计模型设计了 CPU-GPU协同的 RPC 纠正算法,如图 3 所示。



该算法实施步骤如下:①影像解析及参数读 取,并存储于主机端内存; ②根据 RPC 模型构建正 反解函数,正解函数在主机端调用并执行,且仅用于 纠正后影像四至点的结算,反解函数将用于设备端 影像并行重采样;③矫正后影像栅格初始化,填充 重采样灰度数据,同时对该范围进行逻辑分块,以便 在有限的设备端存储空间下完成整张影像的重采 样,分块后影像条带范围被传入 GPU,在全局内存 中初始化地面点坐标数组,根据校正后影像条带范 围反解得到相对应的原始影像范围,将该范围内原 始影像栅格传入 GPU: ④ RPC 反解函数中各项系数 随第一个影像块传入 GPU,并使用 constant 限定 符存储为常量内存,然后将反解函数派生为多线程 任务,将分块影像条带范围内的地面点坐标并行解 算为对应的影像坐标,并根据此坐标对原始影像条 带进行重采样: ⑤将纠正后影像条带回传至主机 端,合并入纠正后影像栅格。最后使用 GDAL 库中的 Geo-TIFF 影像驱动封装为影像产品。其中用于并行坐标反解及重采样主要包括并行纠正核函数、 RPC 模型填充函数以及 RPC 反解函数等几类。

2.3 遥感指数并行算法设计

遥感指数并行算法与辐射校正并行算法相似, 对读取后的影像波段直接进行数学计算。对于多波 段影像,相较于辐射定标过程,指数计算涉及的波段 数更少,因此也更为简单。

2.4 访存优化

在 2.1 节波段运算并行算法设计中提到了对栅 格数组进行逻辑划分以避免显存溢出,以及栅格数 组在多级线程管理结构中的映射方法,这 2 项算法 设计实现了异构环境下高空间分辨率影像分批并行 处理,是提升并行算法效率的关键。因此这里结合 GPU 硬件参数和相关官方资料进行了实验,并对相 关算法参数进行优化。

线程束(warp)作为 GPU 执行任务时的调度单 位,由调度器分配依次进入流处理器簇(SM)执行。 SM 每次只执行线程块(block)中的一个 warp,称作 active warp,此时其余 warp 处于待命状态。当 active warp 在执行中需要等待时(如访问 GPU 全局内 存),调度器可以直接切换至其他 warp 执行。通过 这种方式 GPU 隐藏了多线程的延迟等待,实现了大 规模的并行。因此在程序设计时,可以在同一 block 设置尽量多的 warp,避免所有 warp 均处于等待访存 状态。但由于 SM 内寄存器数量有限,同一个 block 内 warp 数量过多可能造成单一线程可使用的寄存 器数量过少,影响数据访存效率。同时由于 warp 中 线程(thread)数为固定值,因此同一 block 内 thread 总数应设计为这一固定值的倍数,避免 warp 切换时 流处理器(SP)的闲置。通过查阅相关文档获得了 实验平台所使用 GPU 相关硬件参数如表1 所示。

表 1	GPU 硬件参数
Tah 1	GPU narameter

	1			
字段	含义	值		
Nama	刑旦	GeForce GT		
Ivame	望亏	640M LE		
Compute Capability	计算能力/ TFlops ^①	2.1		
Warp Size	线程束内线程数量/个	32		
Threads Per Block	线程块最大线程数/个	1 024		
Threads Per Multipro-	流处理器簇最大线程	1 526		
cessor	数/个	1 330		

①1 TFlops 代表处理器每秒钟可进行一万亿次(10¹²)操作。

3 并行优化实验及分析

3.1 实验平台与数据

GPU 硬件及 CUDA 详细参数如表 2。实验使用

便携计算机平台, CPU 版本为 Intel(R) Core(TM) i5-2520M CPU @ 2.50 GHz, GPU 为 NVIDIA Ge-Force GT 640 M LE, DDR3 内存为 16 GB。影像数据 Landsat8 Level -1 产品, OLI 和 TIRS 传感器, 共11 个 波段, 覆盖周期为 16 d, 扫描宽度为 170 km×180 km, 单波段影像数据量为 100 MB。高分一号(GF-1) L1A 产品, 共9 个波段, PMS 传感器幅宽为 60 km, 空间分辨率为 2~8 m, WFV 传感器幅宽为 800 km, 空间分辨率为 16m, 单景影像数据量为 1.2 GB。

表 2 GPU 硬件及 CUDA 详细参数 Tab. 2 GPU and CUDA parameters

	值
型号	GeForce GT 640M LE
计算能力/ TFlops	2.1
时钟频率/ MHz	1 505
总线带宽/ bits	128
流处理器簇	2
流处理器(CUDA 核心)	96
流处理器簇最大线程数/个	1 536
线程束尺寸	32
线程块尺寸	1 024
线程块寄存器数量/个	3 268
驱动版本	376.51
CUDA 版本	8.0
CUDA 运行库版本	6.5

3.2 实验参数设置

根据 3.1 节中的硬件参数,在控制实验平台其 他硬件参数不变的前提下,使用1景 GF-1 WFV 影 像(数据量 1.63 GB)测试了不同线程块尺寸(以 32 为基准)下 NDVI 并行处理算法中波段运算环节处 理速度,结果如图4所示。



可见在线程块内最大线程数范围内,线程块尺 寸对波段运算耗时的影响并不显著。原因可能在于 本算法涉及的数据类型较为单一,各线程对全局内 存访存延迟较短。

对于可能影响算法运行效率的另一环节——栅 格数组逻辑分块,同样控制了其他变量不变,以影像 宽度为基准,分别测试了不同行数的条带划分尺度 下波段运算耗时,实验结果如图5所示。



Fig. 5 Grid scale of performance

由图 5 可以看出, 栅格数组的划分尺度的确影 响了并行波段运算的效率, 更大的条带划分可以减少 CPU - GPU 间数据交换的频数, 缩短并行算法的时间。但条带划分仍需考虑 GPU 显存容量的上限, 结



$$T = \frac{H}{\left[nmHW/M \right]} , \qquad (8)$$

式中: *T* 为最佳条带宽度; *H* 和 *W* 分别为二维影像 栅格数组的高度和宽度; *n* 为算法使用的栅格数组 数量; *M* 为 GPU 全局内存; *m* 为影像波段数量。

3.3 实验过程与结果分析

3.3.1 汛旱情监测指数

分别使用基于 CUDA 的并行算法、传统串行算法处理一景 GF - 1 WFV 影像数据,进行 NDVI 和 NDWI 指数产品的生产,对各模块计算耗时统计后分别见图 6。以鄱阳湖为例计算光谱指数产品如图 7 所示。



图 6 指数并行算法实验结果





(a) NDVI

(b) NDWI

图 7 鄱阳湖多光谱指数

Fig. 7 Remote sensing index in Poyang Lake

对于简单波段运算类光谱指数算法,采用基于 CUDA的并行算法能够有效提高波段运算这一瓶颈 环节的效率,影像读取和产品输出环节的时间消耗 与串行算法基本一致。此外由于波段运算核函数 中具体内容与串行算法循环体中函数内容完全一 致,因此保证了产品结果的正确性。

3.3.2 辐射校正

分别使用基于 CUDA 的并行算法、传统串行 算法对1景 Landsat8 影像数据进行辐射定标。由 于 Landsat8 各波段影像使用单独的 TIFF 文件存 储,因此使用各波段处理的平均耗时作为对比。 结果如表3 所示。表3 中可见,由于各波段影像差 异和计算机存在性能因素,算法耗时存在一定范

•		
	表 3	传统串行和 CUDA 并行辐射处理耗时对比

Tab. 3	Processing	time of	CPU	and	CUDA	for	radiometric	correction
--------	------------	---------	-----	-----	------	-----	-------------	------------

(s)

か 戸 日	影像	象读取	直方图统计		波段运算		产品输出		合计	
	传统串行	CUDA 并行	传统串行	CUDA 并行	传统串行	CUDA 并行	传统串行	CUDA 并行	传统串行	CUDA 并行
B1	2.76	3.57	32.71	31.7	16.34	7.54	2.90	3.70	54.71	46.51
B2	3.42	3.29	28.76	40.29	22.90	8.30	3.16	2.21	58.24	54.09
B3	8.35	4.57	40.23	37.75	17.95	10.48	3.21	2.39	69.74	55.19
B4	4.57	7.63	22.95	39.32	14.21	6.54	1.98	3.25	43.71	56.74
B5	6.54	9.47	34.69	30.07	26.30	9.36	4.32	5.73	71.85	54.63
B6	2.32	3.12	32.81	31.60	18.68	10.23	6.31	4.16	60.12	49.11

15.49

19.86

B11

均值

										(续表)
油印旦	影像	影像读取 直方图统计 波段运算		取 直方图统计		产品输出		合计		
仮 授 写	传统串行	CUDA 并行	传统串行	CUDA 并行	传统串行	CUDA 并行	传统串行	CUDA 并行	传统串行	CUDA 并行
B7	3.21	2.67	39.65	28.40	19.73	6.98	3.29	5.27	65.88	43.32
B8	7.48	4.32	27.03	38.27	20.10	4.38	2.70	4.90	57.31	51.87
B9	3.24	6.32	26.54	33.91	19.43	7.49	2.40	3.57	51.61	51.29
B10	5.31	2.13	35.63	29.50	27.31	7.90	2.98	4.78	71.23	44.31

10.61

8.16

围的浮动。从整体来看,波段运算环节处理效率有 了大幅提升,相比于串行处理方法,并行算法可节约 58.9%的时间,但直方图统计环节仍然有一定的优 化空间,算法总耗时得到了一定程度的降低。

4.36

4.68

31.26

32.02

35.10

34.17

3.3.3 RPC 几何纠正

3.59

4.62

分别使用基于 CUDA 的并行 RPC 纠正算法、传统串行 RPC 纠正算法,对1景 GF-1 WFV 影像数据

进行几何纠正。在影像重采样环节使用了最邻近像 元法和双线性内插法,比较不同重采样算法对并行加 速比的影响,如图 8 和表 4 所示。根据图 8 和表 4 可 以发现,基于 CUDA 的并行重采样算法相比传统串行 算法可以实现 7~10 倍的加速比,且不同重采样方式 的加速效果较为接近。使用较为复杂重采样方式总处 理时间较长、并行处理占比较大,因此总加速比更高。

2.90

3.90

2.67

3.27



Fig. 8 Processing time contrast of CPU and GPU for RPC

表 4 RPC 并行校正算法重采样环节加速比

Tab. 4 Speedup radio of parallel RPC algorithm

重采样方式	步骤	串行/s	并行/s	加速比
最邻近像元	重采样	174.47	18.79	9.29
	总计	195.88	61.96	3.16
四份股由任	重采样	506.32	71.36	7.10
双线性内插	总计	529.60	118.75	4.46

4 结论

针对汛旱情监测数据处理包括辐射校正、几何 纠正及遥感指数计算等过程耗时过长的问题,面向 常见遥感数据如 Landsat 和 GF - 1 等,分析各类处 理过程的共性与性能瓶颈,基于 CUDA 架构提出了 基于功能模块划分的并行处理算法。

通过实验发现,调整核心参数进行算法访存优 化,可以有效提升并行运算的效率,实验结果表明该 算法在保证成果精度的同时,相比传统串行算法效 率有了显著提升。辐射校正与指数计算均涉及的波 段计算模块,经过并行优化后,较串行模式可减少 70%以上的时间;几何纠正不同重采样方式的加速 效果较为接近;通过调整核心参数进行算法访存优 化,可以有效提升并行波段运算的效率。本算法针 对便携化处理平台而设计,可为汛旱情监测提供切 实有效的高性能计算解决方案。

参考文献(References):

- 肖 汉,张祖勋. 基于 GPGPU 的并行影像匹配算法[J]. 测绘学报,2010,39(1):46-51.
 Xiao H, Zhang Z X. Parallel image matching algorithm based on GPGPU[J]. Acta Geodaetica et Cartographica Sinica, 2010, 39 (1):46-51.
 古 昭松 王 遼 本海仁 CPU 和 CPU 也同协理的类学工具逐
- [2] 方留杨,王密,李德仁. CPU和 GPU协同处理的光学卫星遥 感影像正射校正方法[J]. 测绘学报,2013,42(5):668-675.
 Fang LY, Wang M, Li D R. A CPU - GPU co - processing orthographic rectification approach for optical satellite imagery[J]. Acta Geodaetica et Cartographica Sinica,2013,42(5):668-675.
- [3] 张 过,潘红播,江万寿,等.基于 RPC 模型的线阵卫星影像核 线排列及其几何关系重建[J].国土资源遥感,2010,22(4):1 5. doi:10.6046/gtzyyg.2010.04.01.
 Zhang G, Pan H B, Jiang W S. Epipolar resampling and epipolar

geometry reconstruction of linear array scanner scenes based on

52.97

50.91

53.01

59.76

RPC model[J]. Remote Sensing for Land and Resources,2010,22 (4):1-5. doi:10.6046/gtzyyg.2010.04.01.

- [4] McFeeters S K. The use of the normalized difference water index (NDWI) in the delineation of open water features [J]. International Journal of Remote Sensing, 1996, 17(7):1425-1432.
- [5] 徐涵秋.利用改进的归一化差异水体指数(MNDWI)提取水体 信息的研究[J]. 遥感学报,2005,1(5):589-595.
 Xu H Q. A study on information extraction of water body with the modified normalized difference water index(MNDWI)[J]. Journal of Remote Sensing,2005,1(5):589-595.
- [6] 马秀丹,崔宾阁,钟 勇,等. 一种基于 GPU 和内存映射文件的 高分辨率遥感图像快速处理方法[J]. 海洋科学,2018,42(1): 139-146.

Ma X D,Cui B G,Zhong Y,et al. A fast processing method for high – resolution remote sensing image based on GPU and memory mapping file[J]. Marine Sciences, 2018, 42(1):139 – 146.

- [7] 沈夏炯,侯柏成,韩道军,等. 基于 GPU 和矩阵分块的增强植被指数计算[J]. 遥感信息,2018,33(3):63-69.
 Shen X J, Hou B C, Han D J, et al. Calculation of enhanced vegetation index based on GPU and matrix partition[J]. Remote Sensing Information,2018,33(3):63-69.
- [8] NVIDIA. CUDA 2.0 for windows CUDA 2.0 programming guide [EB/OL]. [2008 - 06 - 07]. http://developer.download. nvidia.com/compute/cuda/20/docs/NVIDIA CUDA Programming Guide 2.0. pdf. 20.
- [9] Rajpoot P S, Kumar A. Impact assessment of meteorological drought on rainfed agriculture using drought index and NDVI modeling: A case study of Tikamgarh district [J]. Applied Geomatics, 2019, 11 (1):15-23.

- [10] Huang F, Chen Y, Li L, et al. Implementation of the parallel mean shift – based image segmentation algorithm on a GPU cluster[J]. International Journal of Digital Earth, 2019, 12(3):328 – 353.
- [11] Heo J M, Jo G, Han H J, et al. Accelerated code generator for processing ocean color remote sensing data on GPU [C]//IEEE. IGARSS 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018:9218 - 9221.
- [12] Nakamura T, Murakami S, Sambo L, et al. Efficacy of a GPGPU acceleration to inundation flow simulation in tonle sap lake in cambodia[J]. Engineering Journal, 2019, 23(1):151-169.
- [13] Zhou H F,Zhao J. Parallel programming design and storage optimization of remote sensing image registration based on GPU[J]. Journal of Computer Research and Development, 2012, 49(1):281-286.
- [14] Guo X, Wu J, Wu Z, et al. Parallel computation of aerial target reflection of background infrared radiation: Performance comparison of OpenMP, OpenACC, and CUDA implementations [J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2016, 9(4):1653 - 1662.
- [15] Gegout P, Oberlé P, Desjardins C, et al. Ray tracing of GNSS signal through the atmosphere powered by CUDA, HMPP and GPUs technologies[J]. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 2014, 7(5):1592 – 1602.
- [16] Norman M, Larkin J, Vose A, et al. A case study of CUDA FOR-TRAN and OpenACC for an atmospheric climate kernel[J]. Journal of Computational Science, 2015,9(1):1-6.
- [17] Peña A J, Reaño C, Silla F, et al. A complete and efficient CUDA sharing solution for HPC clusters [J]. Parallel Computing, 2014, 40 (10):574 – 588.

GPU – based parallel image processing algorithm for flood and drought monitoring

ZHAO Xiaochen¹, WU Haonan², LI Linyi¹, MENG Lingkui¹

 School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China; 2. Pearl River Comprehensive Technology and Network Information Center, Guangzhou 510611, China)

Abstract: Aiming at the time – consuming problems in the remote sensing image processing for flood and drought monitoring, the authors analyzed related workflows and algorithms including radiometric correction, geometric correction, and the calculation of remote sensing indices. Based on the storage structure and program design model of compute unified device architecture (CUDA), the remote sensing image processing was divided into several modules, including data reading, histogram statistics, grid partition, band calculation, resampling, and data output. Among them, parallel processing schemes were designed for the modules of band calculation and resampling, and the optimal cell sizes were determined for the module of grid partition. Meanwhile, the data transfer efficiency was increased through the grid data mapping based on a graphics processing unit (GPU). Finally, a parallel processing algorithm based on CPU – GPU collaboration in CUDA was proposed. The experiment results are as follows. The modules of radiometric correction and band calculation of remote sensing indices showed a 58.9% saving in time. Meanwhile, the geometric correction module enjoyed the most significant time – saving effects, and the final speedup ratios of the resampling methods of nearest neighbor and bilinear interpolation reached up to nine and seven times, respectively.

Keywords: flood and drought monitoring; geometric correction; calculation of remote sensing ndex; GPU; CUDA (责任编辑:张 仙)