DOI:10.13203/j.whugis20190095 文章编号:1671-8860(2020)06-0941-08

一种景观指数的GPU并行算法设计

钟艾妮[1,2] 常栗筠[1] 马云龙[1,2] 亢孟军[1,2] 毛子源[1]

1 武汉大学资源与环境科学学院,湖北 武汉,430079
2 智慧感知与智能计算研究所,湖北 武汉,430079

摘 要:空间数据量的迅猛增长给传统单机模式的空间分析软件带来了巨大挑战,如景观格局分析软件FRAGSTATS已无法处理省级尺度的高分辨率土地覆盖数据。在两次遍历连通域标记算法的基础上,充分利用单机图形处理器的并行运算特性,提出一种改进的景观指数并行算法。该算法针对斑块尺度的斑块周长、斑块面积景观指数指标,实现了大规模区域景观指数的高效运算。应用该算法及串行算法,对不同分辨率下的土地利用分类栅格图像进行斑块尺度景观指数计算,结果表明,在大数据量的情况下,该算法能够大幅度提高景观指数的计算性能,相较串行算法效率提升了5倍,为海量数据的景观分析提供了更好的选择。

关键词:连通域标记算法;景观指数;单机图形处理器;并行计算

中图分类号:P208 **文献标志码**:A

景观指数是反映景观空间格局信息,定量描述地理现象形态特征的空间分析方法[1]。景观指数不仅在生态学领域应用广泛,而且也被用来描述、分析地学及城市规划中各种空间现象的特征[2-3]。在地理信息科学领域,景观指数常用于反映土地利用格局,在土地资源利用的分析评价研究中占有重要地位[4]。计算景观指数的常用软件为美国俄勒冈州立大学开发的FRAGSTATS软件,然而该软件针对大数据量的栅格计算耗时久,且可能由于内存使用限制导致无法生成计算结果[5]。实际数据生产和科学研究中常常需要对大数据量的影像进行分析,因此景观指数算法亟待优化。

斑块尺度的景观指数计算中,最关键的是斑块识别,即识别图像中的连通域。串行算法是传统的连通域标记算法,可分为广度优先算法和深度优先算法。广度优先算法中每个像元搜索同等深度的其他像元,确定局部连通域后,通过多次遍历实现整体连通域的标记,文献[6-10]提出了基于行程的标记、两次遍历(Two-Pass)、轮廓追踪标记等多种算法;文献[7]采用链表的结构存储等价标识值,但更新等价数组时扫描效率低;文献[9]在链式结构的基础上,利用数组结构来存储等价标识之间的关系,提升了等价数组定位查找的效率。深度优先算法的代表是种子填充算法,该算法从单个种子像元出发,不断向邻域搜索,实现单个连通域的标记后,再进行下一连通域的标识[11-12]。在这些连通域标记算法的基础上,文献[13-15]又不断提出优化效率或节省存储空间的新算法,并从二值图像向多值灰度图扩展。但串行的连通域标记算法计算时长受图像大小和分辨率的影响,大数据量环境下串行算法效率较低。

为提高大数据量情况下的计算效率,文献[16-17]利用计算机集群将连通域标记串行算法的计算任务分给多台计算机,从而达到提高计算效率的目的。该方法通过数据并行大幅优化了串行算法的计算效率,但搭建计算机集群要耗费大量财力和物力,资源利用率低,对实验环境要求较高。因此该方法适用于处理大批量复杂海量图像。

当前大多数普通计算机都配有中央处理器(central processing unit,CPU)和单机图形处理器(graphics processing unit, GPU)两种处理器,而GPU由于自身结构特性,不仅具有图形显示的功能,还能进行密集计算。英伟达(NVIDIA)公司

收稿日期:2019-04-07
项目资助:国家重点研发计划(2017YFB0503500)。
第一作者:钟艾妮,硕士,主要从事空间分析与可视化的研究。ainy_zhong@whu.edu.cn
通讯作者:亢孟军,博士,副教授。mengjunk@whu.edu.cn

在2007年提出统一计算设备架构(compute unified device architecture, CUDA),以实现结合CPU和GPU优点的异构并行计算。CUDA能通过简单的指令操作GPU,充分利用GPU的海量计算核心实现真正的并行工作,使得具有高并发性的算法的效率得以大幅提升[18-19]。文献[20-21]发现场模型数据的连续特征能有效利用GPU并行能力完成分析与计算,应用广泛,栅格图像也属于场模型。在此基础上,文献[22-26]实现了基于CUDA的连通域标记算法的并行化,每个线程处理一个或多个像元的邻域判断,通过多次遍历使像元标记值在邻域范围内最小化,直至不再更新。虽然较串行算法效率优化,但原串行算法本身效率较低,优化结果不够理想。此外,GPU单核利用率低,受线程数的限制而难以处理千万像素级的图像。

为了高效完成计算并降低对实验环境的要求,本文基于Two-Pass连通域标记算法,首先提出了基于斑块尺度景观指数的GPU并行算法(斑块尺度景观指数的指标是其他指标计算的基础,本文景观指数的计算内容为斑块周长、斑块面积);然后针对土地利用分类栅格图像,识别土地利用斑块及其土地利用类型,计算各斑块的周长和面积指标,并比较串行和并行两种斑块尺度的景观指数算法效率。该算法解决了GPU并行算法中连通域标记的不连续问题,减少了整体迭代次数,能够实现八邻域的斑块识别,相较串行算法提升了效率。

1 景观指数的串行、并行算法原理

1.1 改进的景观指数串行算法

传统八邻域的Two-Pass算法通过对图像栅格的两次遍历,实现连通域的标识。第一次遍历首先对每个像元赋以唯一标识值,然后通过如图1所示的邻域扫描模板进行邻域判断。邻域判断过程是将模板的P位置对准当前所扫描的像元,若模板范围内的1~4号位置存在相同属性值的像元,则将邻域内最小标识值赋给该像元,同时记录等价标识值,直至所有像元完成邻域判断;在第二次遍历时,根据记录的等价标识值,将该像元等价标识值中的最小值作为新的标识值赋给像元。该算法得到的连通域最终标识值不连续,而本文提出的改进算法能实现连通域的顺序标识,避免了再次遍历图像需要重新标识带来的时间消耗,有利于景观指数计算。

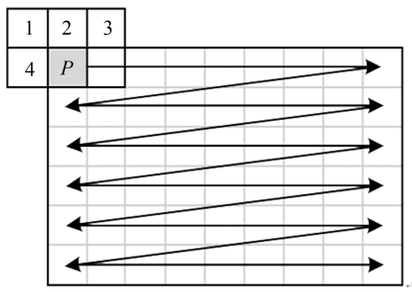

图1 第一次遍历中的邻域扫描模板

Fig.1 Scanning Mask in First Pass

本文改进的斑块尺度景观指数串行算法采用八邻域判断连通性,步骤如下:(1)用图1所示的邻域扫描模板,按照从左至右、从上至下的顺序扫描栅格像元,赋以像元唯一标识值,判断模板范围内像元连通情况。若1~4号位置存在一个或多个与P位置像元具有相同属性值的像元,则在等价数组中记录这些像元和P位置像元的标识值。(2)第一次遍历完成后,更新等价数组,使每个标识值的等价值取等价关系中的最小值。(3)再次遍历栅格像元,定义一个连通域顺序标识变量,初始值设为1。取每个栅格像元标识值的等价值,若标识值即等价值,则标识值及其等价值变为顺序标识变量,顺序标识变量值加1,否则标识值变为该等价值。(4)每一个像元的新标识值即为斑块编号,根据该编号在斑块景观指数数组中定位,然后根据像元的邻接关系更新该斑块的面积和周长。

1.2 改进的景观指数并行算法

§1.1中的串行算法是按照从左到右、从上至下的顺序扫描图像,并且在第一次和第二次遍历时对每个像元都执行相同的操作,因此为景观指数计算的并行化提供了可能。

本文提出的改进的景观指数并行算法流程如图2所示,首先通过按行划分数据块的方式,每个线程处理各自划分一定行数的栅格像元,然后经过边缘扫描和等价标识数组的更新,最终更新各自数据块内的等价标识,并用顺序的重标记值实现各斑块的景观指数计算。与串行算法相比,该算法需要3次线程同步来实现整体斑块标识,增加了数据块边缘的邻域判断和等价标识更新,可分为以下3个部分:(1)第一次遍历,各线程对分配的数据块进行斑块标识;(2)边缘处理,将边缘处被分割的斑块进行融合;(3)第二次遍历,不仅实现图像的最终斑块顺序标记,而且同时计算斑块尺度景观指数。

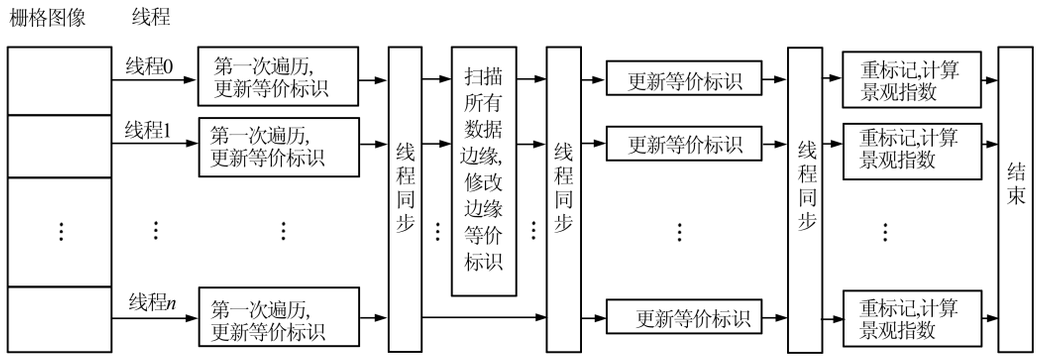

图2　本文算法流程图

Fig.2　Flowchart of Our Proposed Algorithm

1.2.1　第一次遍历

标记图像连通域需要两个重要数组:(1)记录连通域标记值的标识数组;(2)记录等价标识值的等价数组。对m行n列的栅格图像,标识数组的大小为$m\times n$,与栅格图像像元数相同,i行j列的像元标识初始值赋为$i\times m+j+1$,每个像元具有全局唯一标识值。通过查找等价数组中属于同一连通域的标识,能够实现最终连通域的标识。本文算法采用像元数加一个$(m\times n+1)$的数组记录等价标识值,等价数组的下标值对应标识值,从下标为1的数组开始,每个像元有一个等价标识值。等价数组中的元素可以分为根节点和非根节点两类,下标值等于元素值的数组元素为等价根节点,其他为非根节点,通过数组下标和元素值能够实现等价标识值的链式存储。根据非根节点指向的等价标识一直向下查找,最终可得到对应的等价根节点。

栅格图像在本文算法中的第一次遍历过程如下:(1) 每个线程用如图1所示的模板扫描各自分配的栅格像元进行邻接判断。(2)若无邻接像元,则该像元标识为等价根节点,等价数组的下标与元素值均为该标识值。取与像元邻接的最小标识的等价根节点值,赋给该像元标识的等价值。若邻域范围内存在邻接像元,其等价根节点和最小标识的等价根节点不同,则将较大的等价根节点指向较小的等价根节点。(3)通过等价根节点的更新,依赖该等价根节点的其余非根节点也能找到对应的新最小根节点。等价标识过程示意图如图3所示。由图3可以看出,扫描至P_1位置时,邻接像元标识为1,标识1的等价根节点为1本身,则在等价数组中记录标识8的等价值为1;扫描至P_2位置时,邻接像元的标识为4和8,较小的标识4的等价根节点为4,在数组中记录标识9的等价值为4;比较标识4和8的等价根节点值,发现标识8的根节点值1更小,则将等价数组中标识4的等价值改为1。

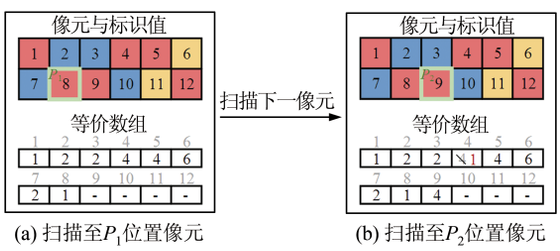

图3　等价标识记录示意图

Fig.3　Diagram of Equivalent Marking Recording

图4为本文算法单个线程中第一次遍历至第一次线程同步过程的示意图。对栅格像元完成第一次遍历并记录下等价标识值后,等价数组中还存在非根节点,因此需要更新等价数组,为非根节点查找等价根节点,以便后续计算。当第一次线程同步时,每个线程所分配像元标识的等价值已更新为等价根节点,此时已识别出每个数据块中的斑块,但此时斑块标识不连续,且位于数据块边缘(首行或末行)的斑块可能被分割。

1.2.2　边缘处理

为了实现边缘斑块的融合,需要对各个数据块的边缘数据进行扫描,使被分割的斑块具有相同的最小标识值。图5为边缘扫描模板示意图,边缘处理过程如下:(1)每个线程(除最后一个)将各自数据块的边缘数据取出(该数据块的最后一行和下一数据块的第一行),P位置对准上一数据块最后一行的像元,1、2、3位置对应下一数据块第一行的像元。(2)若存在与P位置像元相同属性值的邻接像元,取P位置像元的等价根节点和邻接像元的等价根节点进行比较,将较大的等价根节点指向较小的等价根节点,直到所有边缘数据都更新完。(3)边缘数据扫描完后,等价数组中所有等价信息都已记录完善,与图4的过程类似,

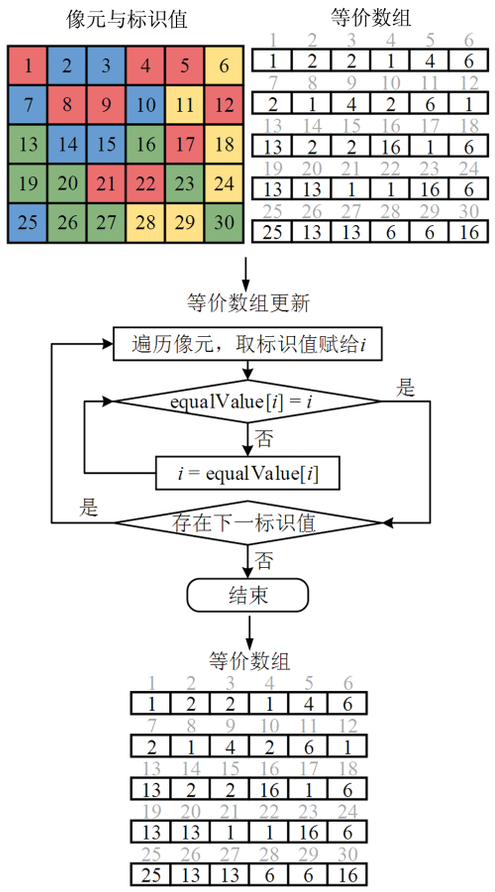

图4　第一次遍历过程

Fig.4　Process of the First Scanning

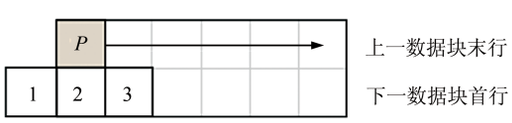

图5　边缘扫描模板

Fig.5　Scanning Mask in Margin Processing

再次进行等价数组的更新，每个数据块范围内的等价根节点个数即该数据块的斑块数，对非根节点查找其根节点，得到所有标识的等价根节点。

1.2.3　第二次遍历

等价数组中所有标识等价值均为等价根节点，此时标识数组进行等价值替换即能得到所有斑块，虽然每个斑块是被唯一值标识的，但标识不连续，在计算景观指数时不利于斑块的定位查找，会造成再次遍历图像的时间损耗。因此在第二次遍历时，利用上一步骤中更新等价数组时记录的各数据块内斑块数，对每个数据块内的斑块进行重标记，第二次遍历过程如图6所示。

为了记录斑块尺度景观指数的计算结果，定义包含斑块类型、斑块周长、斑块面积的结构体数组作为斑块数组，通过斑块编号定位每个像元所属斑块。斑块类型为像元属性值；斑块面积为属于该斑块的像元面积之和；斑块周长的计算需要考虑邻接关系，是所有边界像元长度之和。由于采用从左至右、从上至下的扫描方式，在扫描到单个像元时，首先对所属斑块周长增加一个像元的周长，判断该像元的上方和左方是否存在属于同一斑块的像元，若存在，则对所属斑块周长减去邻接边个数与像元边长的乘积，从而实现斑块周长的计算。

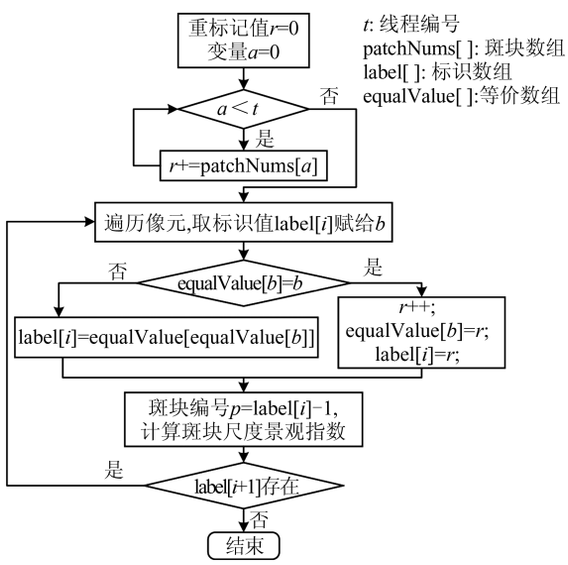

图6　第二次遍历过程

Fig.6　Process of the Second Scanning

2　实验与分析

2.1　实验数据与环境

本文实验采用的是Inter(R) Core i5-7500 3.40 GHz的4核CPU处理器，以及NVIDIA公司GeForce GTX 1050型号的GPU，显卡内存为2 GB，每个block的最大线程数为1 024个。操作系统为Windows 10，基于CUDA 10.0实现本文提出的景观指数并行算法。

实验数据为tif格式的土地利用分类栅格图像，来源于地理国情普查数据，像元值为土地利用分类编码值，连通域数量多，斑块形状复杂。实验中用到了4组不同的栅格图像，图像信息如表1所示，其中data_1为二值图像，data_1与data_4均为连通域数量多但连通域结构相对简单的实验数据，data_2、data_3为连通域复杂的多值图像。

2.2　实验内容

本文实验分析不同参数、不同算法及不同数据规模下的计算效率，包括以下内容：(1)本文算法在不同线程数下的斑块尺度景观指数计算效

表 1　实验数据信息

Tab.1　Experimental Data Information

实验数据	图像分辨率/像素	大小	备注
data_1	256×256	12 KB	建设用地的二值栅格图像
data_2	2 515×1 578	448 KB	多值土地利用分类栅格图像,包含8类用地类型
data_3	4 784×6 825	128 MB	多值土地利用分类栅格图像,包含8类用地类型
data_4	5 741×8 190	130 MB	多值土地利用分类栅格图像,包含8类用地类型

率;(2)本文算法的第一次遍历、边缘处理和第二次遍历这3部分计算耗时的对比与分析;(3)利用CPU串行算法、本文算法以及FRAGSTATS软件测试不同数据规模下土地利用栅格的斑块尺度景观指数计算效率,记录计算时间,并计算加速比R。R的计算公式为:

$$R=\frac{T_{\mathrm{CPU}}}{T_{\mathrm{GPU}}}$$

式中,T_{CPU}是CPU串行算法的计算耗时;T_{GPU}是本文算法的计算耗时。二者均不考虑图像数据读取部分的耗时,且T_{GPU}只考虑核函数执行时间,不考虑数据拷贝到GPU及计算结果拷贝回CPU的耗时。为避免计时误差,实验采取多次计时结果的平均值作为最终计算时间。

2.3　实验结果与分析

GPU具有大量计算核心,不同核心计算内容互不干扰时,效率能达到并行最优,理论上利用越多线程,并行计算效率越高。本文实验需要用到线程同步且数据块间具有一定的相互依赖性,因此只用到GPU的一个block,实验所用GPU的单个block支持的最大线程数是1 024个。§2.1中的实验数据在不同线程数下的计算时间如图7所示。

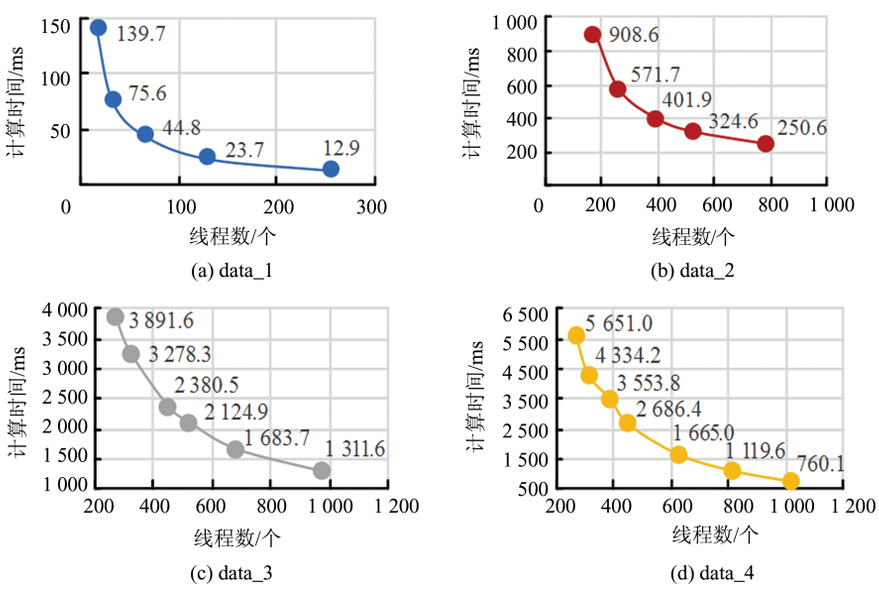

图7　本文算法在不同线程数下的计算时间

Fig.7　Computation Time of Our Proposed Algorithm Under Different Threads

并行算法中利用的线程数越多,单个线程需要计算的栅格行数就越少,并行环境下整体计算耗时便越少。当实验数据的规模较小时,本文算法能够利用的线程数量会受到限制,不能充分利用GPU的并行能力。并行算法的计算时间不仅受到数据规模的影响,还与图像中连通域的复杂程度有关。复杂连通域通常具有多分支结构,需要对处于不同分支的像元寻找根节点,该情况下的数据分割也会带来更多的边缘斑块分割,斑块融合则需要多次循环以查找根节点。

为进一步分析本文算法的计算效率,对该算法第一次遍历、边缘处理和第二次遍历这3部分的计算耗时进行了统计,如图8所示。针对4组实验数据分别选择最佳线程数,data_1采用256个线程,data_2采用789个线程,data_3采用975个线程,data_4采用1 024个线程。

本文算法相较于CPU串行算法多出了边缘处理的部分,由图8可知,当图像规模增大时,处理边缘数据的耗时在整体耗时中的占比会逐渐降低。根据图8所示的堆叠图可知,实验数据量

增大时，第一次遍历需要处理的数据行数也增多，耗时增长，而边缘处理只涉及数据块的首末两行。针对大数据量的栅格图像，边缘处理部分的耗时相对较低，能减少并行算法多出的步骤对计算效率的影响。

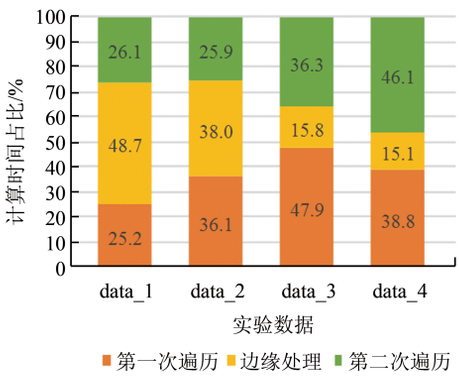

图8 本文算法各部分的耗时

Fig.8 Computation Time of Three Parts of Our Proposed Algorithm

表2为4组实验数据在FRAGSTATS软件、CPU串行算法以及本文算法下的耗时记录。由表2可知，本文算法计算效率相对较高，耗时更少，与串行算法的加速比随着图像数据规模的增大而增大，处理大数据量的栅格图像，计算斑块尺度景观指数的效率优化结果明显。

3 结 语

本文提出了一种基于景观指数的GPU并行算法，利用CUDA编程，以不同数据规模的土地利用栅格图像作为实验数据，讨论CPU串行算法和本文算法的计算效率。实验结果显示，本文算法能够高效计算大尺度、高分辨率土地利用栅格的斑块尺度指标，串行算法效率高，改造为适合GPU的并行算法能够对千万像素级的图像计算效率进一步优化，为处理大数据量栅格的景观指数计算提供新途径。在实际数据生产中，基于本文提出的并行算法，一台普通的配备CPU+GPU的机器，在计算过程中能充分利用GPU多核心架构的并行能力，同时还能利用CPU的逻辑控制能力来实现其他计算任务，以提升数据生产的效率。由于本文算法处理大数据量栅格图像需要大量内存，数据量超过GPU显存时不能进行计算，因此计算过程中的数据结构设计有待进一步优化，减少不必要的内存开销，以适应大数据量计算。

表2 FRAGSTATS、串行算法及并行算法耗时比较

Tab.2 Comparison of Computation Time of FRAGSTATS, Serial and Parallel Algorithms

实验数据	图像分辨率/像素	Fragstats软件耗时/ms	串行算法耗时/ms	本文算法耗时/ms	加速比R
data_1	256×256	3 100	16.0	12.9	1.24
data_2	2 515×1 578	9 500	406.1	250.6	1.62
data_3	4 784×6 825	5 330	3 143.5	1 311.6	2.39
data_4	5 741×8 190	13 440	4 473.5	760.1	5.88

参 考 文 献

[1] Peng Jian, Wang Yanglin, Zhang Yuan, et al. Research on the Influence of Land Use Classification on Landscape Metrics [J]. *Acta Geographica Sinica*, 2006, 61(2): 47-58(彭建, 王仰麟, 张源, 等. 土地利用分类对景观格局指数的影响[J]. 地理学报, 2006, 61(2): 47-58)

[2] Chen Wenbo, Xiao Duning, Li Xiuzhen. Classification, Application, and Creation of Landscape Indices [J]. *Chinese Journal of Applied Ecology*, 2002, 13(1): 121-125(陈文波, 肖笃宁, 李秀珍. 景观指数分类、应用及构建研究[J]. 应用生态学报, 2002, 13(1): 121-125)

[3] Liu Jiafu, Wang Ping, Li Jing, et al. An Algorithm for Land-Use Pattern Index and Its Application [J]. *Geography and Geo-information Science*, 2009, 25(1): 107-109(刘家福, 王平, 李京, 等. 土地利用格局景观指数算法与应用[J]. 地理与地理信息科学, 2009, 25(1): 107-109)

[4] Luo Minghai, Jiang Zilong, Chen Qi, et al. Carrying Capacity Evaluation of Land Resources in Wuhan Based on the Geographical Condition Monitoring [J]. *Geomatics and Information Science of Wuhan University*, 2018, 43(12): 2 317-2 324(罗名海, 蒋子龙, 程琦, 等. 地理国情在武汉市土地资源承载力评价中的应用[J]. 武汉大学学报·信息科学版, 2018, 43(12): 2 317-2 324)

[5] Mcgarigal K S, Cushman S, Neel M, et al. FRAGSTATS: Spatial Pattern Analysis Program for Categorical Maps [OL]. http://www.umass.edu/landeco/research/fragstats/fragstats.html, 2002

[6] Dillencourt M B, Samet H, Tamminen M. A General Approach to Connected-Component Labelling for Ar-

bitrary Image Representations [J]. *Journal of the ACM*, 1992, 39(2): 253-280

[7] Christophe F, Jens G. Two Linear Time Union-Find Strategies for Image Processing [J]. *Theoretical Computer Science*, 1996, 154(2): 165-181

[8] Chang Fu, Chen Chunjen, Lu Chijen. A Linear-Time Component-Labeling Algorithm Using Contour Tracing Technique [J]. *Computer Vision and Image Understanding*, 2004, 93(2): 206-220

[9] Wu K, Otoo E, Suzuki K. Optimizing Two-Pass Connected-Component Labeling Algorithms [J]. *Pattern Analysis & Applications*, 2009, 12(2): 117-135

[10] He L, Ren X, Gao Q, et al. The Connected-Component Labeling Problem: A Review of State-of-the-Art Algorithms[J]. *Pattern Recognition*, 2017, 70: 25-43

[11] Manohar M, Ramapriyan H K. Connected Component Labeling of Binary Images on a Mesh Connected Massively Parallel Processor [J]. *Computer Vision Graphics & Image Processing*, 1989, 45(2): 133-149

[12] Celebi M E. A Simple and Efficient Algorithm for Connected Component Labeling in Color Images [C]. SPIE Electronic Imaging Conference, San Francisco, California, USA ,2012

[13] Zhao Xiao, He Lifeng, Yao Bin, et al. A New Connected-Component Labeling Algorithm [J]. *IEICE Transactions on Information & Systems*, 2015, 98(11): 2 013-2 016

[14] Shen Xiajiong, Wang Jingjing, Fan Jiaming, et al. Labeling Algorithm of 8-Adjacent Connecting Area for Massive Gray Scale Images[J]. *Computer Engineering and Applications*, 2013, 49(20): 126-139(沈夏炯, 王晶晶, 范家铭,等. MGSI-8CA标记算法 [J]. 计算机工程与应用, 2013, 49(20): 126-139)

[15] Gupta S, Palsetia D, Agrawal A, et al. A New Parallel Algorithm for Two-Pass Connected Component Labeling[C]. 28th IEEE International Parallel & Distributed Processing Symposium, Phoenix , Arizona, USA, 2014

[16] Ma Yihang, Zhan Lijun, Xie Chuanjie, et al. Parallelization of Connected Component Labeling Algorithm [J]. *Geography and Geo-information Science*, 2013, 29(4): 67-71(马益杭, 占利军, 谢传节, 等. 连通域标记算法的并行化研究 [J]. 地理与地理信息科学, 2013, 29(4): 67-71)

[17] Liu Yang, Guan Qingfeng. A Parallel Algorithm for Landscape Metrics [J]. *Journal of Geo-information Science*, 2017, 19(4): 457-466(刘洋, 关庆锋. 景观指数的并行计算方法 [J]. 地球信息科学学报, 2017, 19(4): 457-466)

[18] Guan Xuefeng, Zeng Yumei. Research Progress and Trends of Parallel Processing, Analysis, and Mining of Big Spatiotemporal Data [J]. *Progress in Geography*, 2018, 37(10): 1 314-1 327 (关雪峰, 曾宇媚. 时空大数据背景下并行数据处理分析挖掘的进展及趋势 [J]. 地理科学进展, 2018, 37(10): 1 314-1 327)

[19] Yang Jinyu, Zhang Yongsheng, Li Zhengguo, et al. GPU-CPU Cooperate Processing of RS Image Ortho-Rectification [J]. *Geomatics and Information Science of Wuhan University*, 2011, 36(9): 1 043-1 046(杨靖宇, 张永生, 李正国,等. 遥感影像正射纠正的GPU-CPU协同处理研究 [J]. 武汉大学学报·信息科学版, 2011, 36(9): 1 043-1 046)

[20] Dong Luming, Zhang Bin, Zhao Xuesheng. A Seamless Terrain Rendering Algorithm Based on GPU Tessellation [J]. *Geomatics and Information Science of Wuhan University*, 2017, 42(3): 402-407(董路明, 张斌, 赵学胜. 一种基于GPU Tessellation的地形无缝绘制算法[J]. 武汉大学学报·信息科学版, 2017, 42(3): 402-407)

[21] Liu Jinshuo, Li Yangmei, Jiang Zhuangyi, et al. Fine-Grained Parallel Optimization of Large-Scale Data for PMVS Algorithm[J]. *Geomatics and Information Science of Wuhan University*, 2019, 44(4): 608-616(刘金硕, 李扬眉, 江庄毅, 等. 基于PMVS算法的大规模数据细粒度并行优化方法[J]. 武汉大学学报·信息科学版, 2019, 44(4): 608-616)

[22] Hawick K A, Leist A, Playne D P. Parallel Graph Component Labelling with GPUs and CUDA [J]. *Parallel Computing*, 2010, 36(12): 655-678

[23] Kalentev O, Rai A, Kemnitz S, et al. Connected Component Labeling on a 2D Grid Using CUDA [J]. *Journal of Parallel & Distributed Computing*, 2011, 71(4): 615-620

[24] Komura Y. A Generalized GPU-Based Connected Component Labeling Algorithm [J]. *Journal of the Association for Computing Machinery*, 2016, 39(2): 253-280

[25] Qin Fangtao, Fang Bin. GPU Accelerated Parallel Labeling Algorithm of Connected-Domains in Binary Images [J]. *Journal of Computer Applications*, 2010, 30(10): 2 774-2 786(覃方涛, 房斌. GPU加速的二值图连通域标记并行算法 [J]. 计算机应用, 2010, 30(10): 2 774-2 786)

[26] Wang Zehuan, Lai Junjie. An Improved Parallel Connected Component Labeling Algorithm and Its

GPU Implementation[C]. HPC China, Guangzhou, China, 2014(王泽寰, 赖俊杰. 一种改进的图像连通区域标记的并行算法及其在GPU上的实现[C]. 全国高性能计算学术年会,广州,2014)

A GPU-Based Parallel Algorithm for Landscape Metrics

ZHONG Aini[1,2] *CHANG Lijun*[1] *MA Yunlong*[1,2] *KANG Mengjun*[1,2] *MAO Ziyuan*[1]

1 School of Resource and Environmental Sciences, Wuhan University, Wuhan 430079, China

2 Institute of Smart Perception and Intelligent Computing, Wuhan University, Wuhan 430079, China

Abstract: Massive spatial data poses increasing challenges to traditional analysis software. For example, landscape pattern analysis software FRAGSTATS has been unable to process provincial-level high-resolution land cover data. Based on Two-Pass connected component labeling algorithm, this paper provides an improved parallel algorithm with GPU programming to solve the landscape metrics computation problem about massive land use data. This parallel algorithm for massive landscape metrics calculation takes full advantage of a general computer, and focuses on patch perimeter and area calculation. It can also accelerate computation speed by multithreading and iteration times reduction to decrease computation time than traditional serial algorithms. We apply the proposed algorithm and serial algorithm to calculate landscape metrics of the land use classification raster images at different resolutions under patch scale. The experiment result shows great improvement of calculation performance of landscape metrics, and the efficiency has been improved by 5 times comparing with the serial algorithm, which proves that our proposed algorithm is a better choice for landscape analysis of massive data.

Key words: connected component labeling algorithm; landscape metrics; graphics processing unit(GPU); parallel computing

First author: ZHONG Aini, master, majors in geographic information processing and visualization. E-mail: ainy_zhong@whu.edu.cn

Corresponding author: KANG Mengjun, PhD, associate professor. E-mail: mengjunk@whu.edu.cn

Foundation support: The National Key Research and Development Program of China(2017YFB0503500).

引文格式:ZHONG Aini, CHANG Lijun, MA Yunlong, et al. A GPU-Based Parallel Algorithm for Landscape Metrics[J]. Geomatics and Information Science of Wuhan University, 2020, 45(6):941-948. DOI:10.13203/j.whugis20190095(钟艾妮,常栗筠,马云龙,等.一种景观指数的GPU并行算法设计[J].武汉大学学报·信息科学版,2020,45(6):941-948.DOI:10.13203/j.whugis20190095)