

引用格式: Li Weiliang, Gao Wei, Ye Sanxing. Multi-scale Segmentation of Remote Sensing Image based on Multi-core Computer[J]. Remote Sensing Technology and Application, 2012, 27(4): 560-565. [李维良, 高伟, 叶三星. 基于多核计算机的遥感影像多尺度分割[J]. 遥感技术与应用, 2012, 27(4): 560-565.]

基于多核计算机的遥感影像多尺度分割

李维良, 高 伟, 叶三星

(中国地质大学(武汉)信息工程学院, 湖北 武汉 430074)

摘要: 多尺度分割作为一种成熟的影像分割方法, 在遥感影像信息提取中得到广泛应用, 但算法整体效率较低。利用多核计算机实现了基于数据并行的遥感影像多尺度分割。传统的影像 IO(Input and Output)方法在影像数据量较大的情况下无法满足多核计算机并行处理的需要, 设计了一种新的影像 IO 策略消除了这种缺陷; 此外, 在遥感影像多尺度并行分割的过程中, 普遍存在分割结果无法直接进行合并的问题, 利用对特定区域重分割的方法在保证效率的前提下解决了这个问题。结果表明: 针对各种数据量与尺寸的遥感影像, 并行分割效率有了较大提升, 并且分割算法具备了处理大数据量影像的能力, 极大地增强了通用性。利用多核计算机提升影像分割效率取得了显著成效。

关 键 词: 多尺度分割; 多核计算机; 影像 IO 策略; 数据分区合并

中图分类号: TP 391 **文献标志码:** A **文章编号:** 1004-0323(2012)04-0560-06

1 引 言

多尺度分割作为一种成熟的影像分割方法, 在面向对象的高分辨率遥感影像信息提取中得到广泛应用。但算法内存消耗大、计算时间较长、整体效率较低。针对算法改进可以从计算机实现角度来进行, 如将影像数据划分为若干数据分区进行数据并行化处理。

目前, 许多学者已经针对这方面进行了研究: 吴炜等^[1]利用集群对均值漂移影像分割算法进行了数据并行处理, 结果表明使用 n 个节点集群所消耗的时间为处理能力最差节点所需时间的 $1/n$, 能够显著地提高算法的效率。沈占峰等^[2]同样利用集群对均值漂移分割算法进行了并行化处理, 并提出了一种新的数据“缓冲区”式分区方法, 实验取得良好成效。Happ 等^[3]研究与探讨了在多核架构上进行影

像分割数据并行的有效性, 实验结果表明最终的分割效率有了较大提升。

现有的研究大部分是基于集群进行的, 存在部署困难、成本较高等缺点, 不适合桌面遥感软件的应用推广, 少数基于多核计算机的研究也主要是从流程的角度论证多核架构对提高分割效率的有效性, 并没有过多地考虑某些关键因素尤其是影像数据量的影响, 因此, 算法往往只针对中小型数据, 通用性受到制约。此外, 现有的研究在解决合并各数据分区分割结果时出现接边无法很好对应, 产生“分隔线”问题时, 提出了一系列算法, 这些算法在一定条件下能够取得理想效果, 但在复杂性、实施难度以及特定条件下的可信度等方面存在一些不足。这些问题本文将在后面做详细探讨。

在总结前人研究的基础上, 本文利用多核计算机, 研究和探讨基于数据并行的多尺度分割算法的

收稿日期: 2011-08-18; 修订日期: 2012-04-18

基金项目: 中央高校基本科研业务费专项资金资助项目(CUGL120267)。

作者简介: 李维良(1987—), 男, 湖北十堰人, 硕士研究生, 主要从事遥感影像处理方面的研究。E-mail: cuglwl@163.com。

通讯作者: 高 伟(1979—), 男, 湖北襄阳人, 讲师, 博士, 主要从事遥感影像处理方面的研究。E-mail: cuggaowei@hotmail.com。

关键技术,重点解决目前分割过程中存在的无法处理大数据量影像以及“分隔线”消除的问题。

2 多尺度分割算法的并行处理

2.1 多尺度影像分割算法

多尺度分割算法是一种面向对象的高分辨率遥感影像分割方法,它的关键在于两个影像对象异质度的描述。对于一个 d 维的特征空间,设两相邻对象的特征值分别为 f_{1d} 和 f_{2d} ,则异质度定义为^[4]:

$$h = \sqrt{\sum_d (f_{1d} - f_{2d})^2} \quad (1)$$

影像对象的光谱特征以及形状特征都可以作为特征空间的一维。通过式(2)对每一维特征求标准差来进一步标准化特征空间距离,其中 σ_{fd} 为对象特征的标准差^[4]:

$$h = \sqrt{\sum_d ((f_{1d} - f_{2d}) / \sigma_{fd})^2} \quad (2)$$

分割过程同时对影像对象的光谱特征和形状特征进行操作,具体步骤为^[5]:首先设置分割参数,包括作为影像对象合并终止条件的尺度阈值以及各个特征的权值。随后开始分割过程,每次进行分割时,搜寻邻域影像对象,遵循异质度最小的原则,把相邻影像对象合并为一个较大的影像对象。第一次分割过程中,单个像元被看作最小的影像对象;第二次及以后的分割过程中,以前一次分割过程中生成的影像对象为基础计算异质度,判断异质度 h 与尺度阈值的关系,若 h 小于或等于尺度阈值,则继续进行第 $n(n > 2)$ 次的分割,反之分割结束。

2.2 多尺度影像分割算法的并行化

基于数据并行的多尺度分割,首先根据线程数对影像进行数据划分,采取如图 1 所示的“行划分法”对影像进行均匀划分,划分后每个子影像称为一个“数据分区”,每一个线程对应处理一个数据分区。



图 1 影像数据分区

Fig. 1 Data partition of image

相互独立。分割结束后,数据分区边缘会出现“分隔线”,影响分割结果的精度,需要采取分区合并算法进行处理,基于多核计算机的并行分割处理流程如图 2 所示。

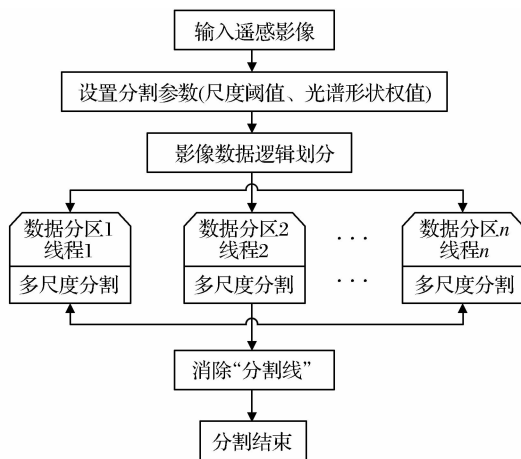


图 2 基于多核计算机的并行分割流程

Fig. 2 The flow chart of paralleled segmentation based on multi-core computer

3 影像分割并行化关键技术

采用多核计算机进行基于数据并行的遥感影像多尺度分割主要面临两个问题:

(1) 影像数据量的限制影响并行分割算法的通用性。目前,大部分桌面遥感软件在利用多核处理器加速时只在算法层做了并行化,影像读写层仍是基于传统的单线程读写模式,即先将全部或部分影像数据通过传统读写方式载入内存,再对这部分数据进行算法并行化处理。多尺度分割需要随时获取相邻影像像元(或像元集合)的基本信息,在实现过程中,大多采用将影像数据全部调入内存,再分配给各计算单元进行处理^[7]。这对计算机集群来说问题不大,因为在算法执行之前,数据就已经被划分为若干数据分区并发送给对应计算节点,每台计算机的数据负载较轻。但在多核计算机上,这样的处理方式并不适用,数据量过大会引起系统内存崩溃。

(2) “分隔线”影响分割结果的准确性。基于数据并行的影像分割结束后合并各数据分区结果时,存在接边无法很好地衔接,数据分区之间存在明显的“分隔线”,使得分割结果不可靠。

针对第一个问题,目前国内外的研究并不多见,本文设计了一种新的影像数据分区 IO 策略,保证分割效率的同时消除了数据量对算法实现的限制。针对第二个问题,国内外已有一些研究,胡晓东等^[6]

然后各数据分区开始并行进行分割,分割过程

在并行分割结束后将“分隔线”与其两侧分割块轮廓的交点分为在“分隔线”交汇的 a 类交点与不在“分隔线”交汇的 b 类交点。两个相邻的 a 类交点间的线段称为 A 类线,此外为 B 类线段,对 A 类线两侧的分割块进行合并,对 B 类线两侧的分割块进行“次分割”,以此消除“分隔线”对结果的影响,但该算法比较复杂,且直接合并 A 类线两侧的分割块,在某些情况下也显得主观因素过重,值得商榷,因为“分隔线”两侧接边能够很好衔接的分割对象在现实中属于同一影像对象的情况是可能存在的。Happ 等^[3]提出一种在并行分割过程中就考虑各数据分区相邻分割块相互影响的方法,但会导致分割过程中数据分区逐渐变得不规则,算法实施难度较大,且各计算节点会出现负载逐渐不均衡。本文对现有的研究成果进行综合取舍,在并行分割结束后,搜索“分隔线”两侧分割块所包含的像素并重新进行分割,在保证效率的前提下消除了“分隔线”对分割结果的影响。

3.1 多核架构下影像数据分区读写策略

为了保证并行分割算法的通用性不受影像数据量的限制,必须设计一个合理的影像数据分区 IO 层,使得分割过程能够对较大数据量影像文件进行读写,内存不会发生崩溃,同时,多线程对所属数据分区能够做到在逻辑上同时进行读写,彼此之间互不影响,便于算法的实现,如图 3 所示。

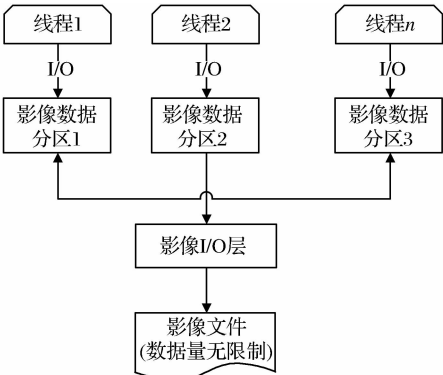


图 3 基于多核平台的影像数据 IO 层

Fig. 3 The image input and output level based on multi-core computer

为达到对影像数据进行有效组织和管理的目的,需对影像进行分块,以影像块为最小单位对影像文件进行加载与回写,可减少系统对影像文件的访问次数,提高系统性能^[7]。本文采用“行划分”法对影像进行均匀分块,每个影像块包含若干行数据,每一个分块称为一个“影像块”。影像的读写在时空上具有局部性,可以在内存中设置一个容量相对较小

的缓存区来对影像文件进行缓存,提高读写效率。缓存区以影像块为单位建立,本文通过影像块队列来维持上述缓存区。分割过程中像元读写趋势是从上到下,从左到右,因此缓存区中影像块的调度采用先进先出策略,如图 4 所示。

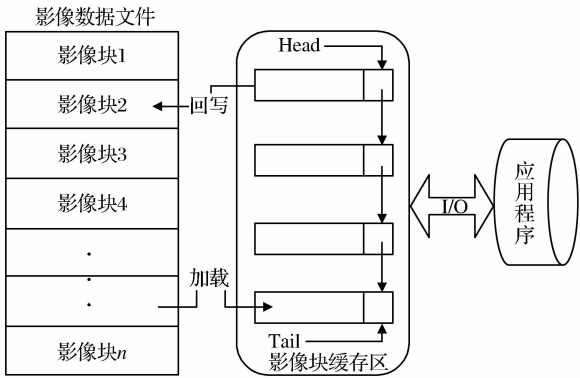


图 4 缓存块加载和回写

Fig. 4 The loading and flushing of cache block

在建立图 4 所示的缓存机制后,可以实现单线程对遥感影像文件的高效读写,且只要对缓存区大小进行合理设置,分割过程不会出现内存崩溃的情况。然而,在并行分割过程中,多线程对所属的数据分区同时进行读写,使得一个线程往往会直接影响缓存区中属于其他线程数据分区的影像块的加载与释放工作,导致影像块缓存区不一致。因此,需要对上述影像缓存机制进行完善。本文为每个线程所对应的影像数据分区分别维护一个较小的缓存区,缓存区之间相互独立,避免产生上述问题,如图 5 所示。

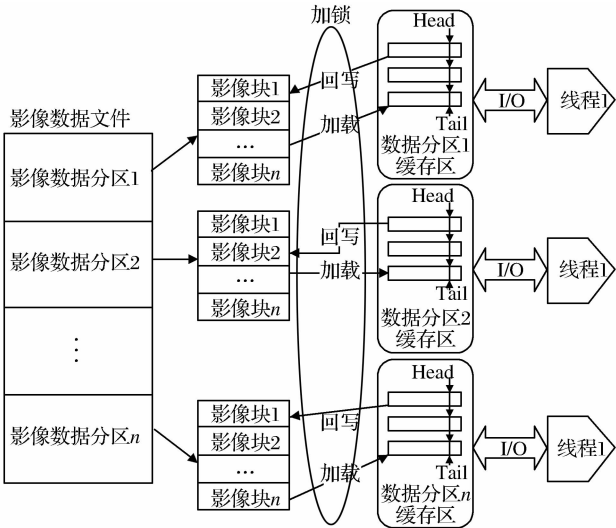


图 5 影像数据分区 IO 策略

Fig. 5 The strategy of image data partition input and output

本方法在逻辑上允许许多线程对所属数据分区同时进行读写,有利于分割算法的实现,且多线程在相应缓存区中存取数据时做到了一定程度的并行化,保证了读写效率。原始影像文件是所有线程的公共资源,因此线程进行影像块加载与回写操作时需要使用锁机制。

多线程对影像并行读写时,每个线程像元读写步骤为:

(1) 指定操作类型(读或写)、波段号、像元行列号,根据行列号确定像元所处的数据分区,以及数据分区中的影像块号。

(2) 判断第(1)步中影像块是否位于线程所属数据分区的缓存区中,若影像块尚未加载,跳至第(3)步;否则,直接在缓存区中对像元进行读或写操作,若为写操作,需同时设置影像块的脏标志。操作完毕后返回。

(3) 计算缓存区剩余容量,若能容纳第(1)步中指定的影像块,对影像块上锁,从影像文件中读取影像块数据并解锁,放入缓存区的尾部,返回第(2)步;否则,判断缓存区队列头部的影像块是否为脏,若为脏,跳至第(4)步,若不为脏,跳至第(5)步。

(4) 对缓存区队列头部的影像块上锁,回写影像块数据到影像文件并解锁,取消影像块脏标志,返回第(3)步。

(5) 直接释放缓存区队列头部影像块所占内存,返回第(3)步。

3.2 消除“分隔线”

影像并行分割过程中,各数据分区是独立进行的,没有考虑彼此之间的影响,因此,并行分割结束之后,各数据分区之间会产生一条明显的“分隔线”。“分隔线”往往会将本属于同一分割对象的像元集合分成两个对象,有时这两个分割对象的接边不能很好地衔接^[6],更有甚者,即使是“分隔线”两侧接边能够很好衔接的分割对象在现实中也可能不应属于同一影像对象。总之,“分隔线”使得影像分割结果不可信。

“分隔线”对分割结果影响的消除既可以在并行分割之后进行,也可以在并行分割过程中就加以考虑,目前针对两种方式都取得了一定的研究成果^[3,6],但现有的“分隔线”消除算法在算法复杂性、实施难度以及算法可信度等方面存在一定不足或有待验证,因此,需要平衡各种因素,找到最适合本文研究的“分隔线”消除算法。本文通过搜索与“分隔线”相邻的分割对象内的所有像素,将它们设为无效

并进行重分割,算法较为简单可行,重分割过程没有将“分隔线”两侧的分割块区别对待,在一定程度上也保证了重分割过程的客观性与可信性,算法详细步骤如下:

(1) 定义一个像素点的集合 PointSet,初始化为空。

(2) 遍历“缝合线”所经过的所有像素点,依次判断它的上下两像素是否被设为无效(可以是一个特定的标号),若像素无效,不进行处理;否则,以该像素为种子像素入栈,对该像素所属分割对象所包含的所有像素进行扫描线种子填充,填充值为无效值,同时记录被填充像素的坐标,并加入集合 PointSet 中。

(3) 以集合 PointSet 中所有像素为初始对象,对这些像素重新进行分割,重分割过程中可根据实际情况重新设置尺度参数。

4 结果与分析

根据以上研究思路,对并行分割算法进行了实现。实验环境为: Intel i5 双核 CPU, 2 GB 内存及 OpenMP 并行计算环境。实验中并行分割时,线程数量采用 2,与 CPU 核数一致。

首先,为了验证“分隔线”消除算法的最终效果,对 QuickBird 全色影像与相应多光谱影像进行融合,取融合结果前 3 波段的 RGB 合成影像作为实验影像进行分割,影像尺寸为 500×500 像素,实验中尺度阈值设为 2 000,光谱特征权值与形状特征权值均设为 0.5。

图 6(a)~(d)分别为实验影像原图、串行分割结果、“分隔线”未消除情况下的并行分割结果以及“分隔线”消除后的并行分割结果。单从图 6(c)中可以看出,并行分割结束后,数据分区之间的“分隔线”十分明显,受其影响,“分隔线”两侧分割结果并不理想。将图 6(c)、6(d)对比后可见,“分隔线”经过重分割处理后被完全消除,分割结果可信度也得到提高。

此外,为了验证本文所设计的影像分区 IO 策略的可行性以及利用多核计算机进行多尺度分割的有效性,接下来的实验采用了多幅不同数据量与尺寸的遥感影像进行分割,表 1 显示了这些影像的基本信息。为达到实验目的,本文设置缓存区总容量为原始影像数据量的 $1/3$,且不超过 200 Mb,并平均分配给每个影像数据分区所对应的缓存区;影像分块大小采用固定的每 16 行作为一个影像块。

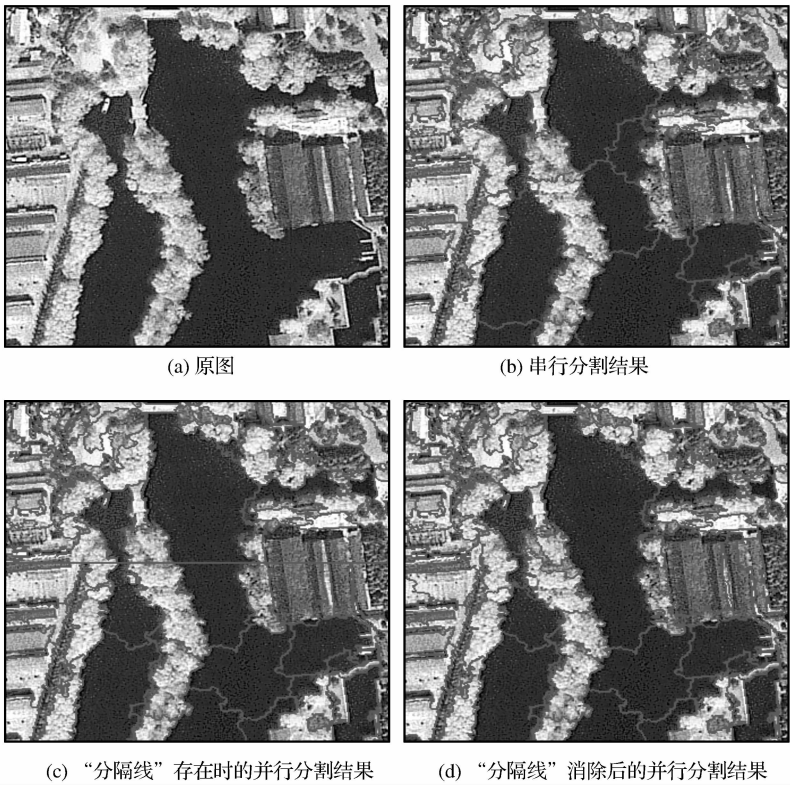


图 6 数据分区合并结果

Fig. 6 The result of merging data partition

表 1 影像数据基本信息

Table 1 Basic information of the images		
影像名	数据量/Mb	尺寸/像素
影像 1	500	6 500×6 500
影像 2	780	8 300×8 300
影像 3	1 100	10 000×10 000

从表 2 可见,基于传统单线程影像 IO 层的并行分割只有在影像数据量较小时才适用,当影像数据量过大时,系统内存会崩溃,分割无法正常运行。而采用本文所设计的方法之后,只要合理地设置影像数据分区缓存区大小,分割算法具备了处理大数据量影像的能力,极大地增强了通用性。

表 2 基于两种不同 IO 策略的分割对照

Table 2 The contrast of segmentation base on two different input and output strategies		
影像名	基于传统 IO 方法的并行分割能否成功	基于本文方法的并行分割能否成功
影像 1	能	能
影像 2	能	能
影像 3	否	能

表 3 显示了串行分割与并行分割之间效率的对比。从中可见,随着影像数据量的增加,“分隔线”消

除用时所占比例在整个分割过程中越来越低,对整体分割效率的影响不大,针对各种数据量与尺寸的遥感影像,并行分割效率都有了较大提升,利用多核计算机提升影像分割效率取得显著成效。

表 3 影像分割效率对照

Table 3 The contrast of segmentation efficiency			
影像名	串行分割用时/S	并行分割用时/S	“分隔线”消除占并行分割用时比例/%
影像 1	3 292	2 031	3.0
影像 2	5 258	3 212	1.7
影像 3	8 359	5 129	1.5

5 结 语

在多核计算机上进行基于数据并行的遥感影像多尺度分割,如果采用传统单线程影像 IO 模式,必须首先将影像数据全部调入内存后再做处理,无法处理数据量过大的影像。并且,并行分割完毕后,数据分区无法很好地衔接也是采用数据并行方式所无法回避的问题。为解决这些问题,设计了一种新的影像数据分区 IO 策略,实验结果证明本方法具有良好的可行性。同时,在总结前人研究的基础之上,利用对相应影像对象重分割的方法消除了“分隔

线”,使分割结果更为可靠。

本文仅从方法的角度论证解决并行分割时所面临的主要问题的可行性,并没有对全部关键技术的所有特殊情况均作考虑,例如针对不同尺寸与数据量的影像如何设置影像块的大小及缓存区的容量,以及重分割过程中尺度参数是否需要重新设置以及如何更好地设置等,这些问题都有待进一步的研究与探讨。

参考文献(References):

- [1] Wu Wei, Shen Zhanfeng, Luo Jiancheng, *et al.* Implementation of Multi-scales Segmentation for High Resolution RS Images based on Cluster[J]. Computer Engineering and Applications, 2009, 45(34): 7-9. [吴炜, 沈占锋, 骆剑承, 等. 均值漂移分辨率遥感影像多尺度分割的集群实现[J]. 计算机工程与应用, 2009, 45(34): 7-9, 24.]
- [2] Shen Zhanfeng, Luo Jiancheng, Wu Wei, *et al.* Implementation of Parallelization of Mean-Shift Algorithm for Multi-scale Segmentation of Remote Sensing Images[J]. Journal of Harbin Institute of Technology, 2010, 42(5): 811-815. [沈占锋, 骆剑承, 吴炜, 等. 遥感影像均值漂移分割算法的并行化实现[J]. 哈尔滨工业大学学报, 2010, 42(5): 811-815.]
- [3] Happ P N, Ferreira R S, Bentes C. Multiresolution Segmentation: A Parallel Approach for High Resolution Image Segmentation in Multicore Architectures[C]//The International Archives of The Photogrammetry & Remote Sensing and Spatial Information Sciences, Vol. XXXVIII-4/C7.
- [4] Baatz M, Schape A. Multi-resolution Segmentation: An Optimization Approach for High Quality Multi-scale Image Segmentation[J/OL]. <http://www.agit.at/papers/2000/baatz-FP-12.pdf>, 2000.
- [5] Chen Qihao. Research and Realization of Multi-sources Remote Sensing Data on Object Oriented[D]. Wuhan: China University of Geosciences, 2007. [陈启浩, 面向对象的多源遥感数据分类技术研究与实现[D]. 武汉: 中国地质大学, 2007.]
- [6] Hu Xiaodong, Luo Jiancheng, Shen Zhanfeng, *et al.* Data Sewing Algorithm for Parallel Segmentation of High-Resolution Remotely Sensed Image[J]. Journal of Remote Sensing, 2010, 14(5): 922-927. [胡晓东, 骆剑承, 沈占锋, 等. 高分辨率遥感影像并行分割结果缝合算法[J]. 遥感学报, 2010, 14(5): 922-927.]
- [7] Zhang Chang, Liu Qiang, Dai Changli, *et al.* The Design and Implementation of Large-scale Imagery Database[J]. Science of Surveying and Mapping, 2010, 35(3): 208-209, 164. [张畅, 刘强, 戴昌礼, 等. 大规模影像数据库的设计与实现[J]. 测绘科学, 2010, 35(3): 208-209, 164.]

Multi-scale Segmentation of Remote Sensing Image based on Multi-core Computer

Li Weiliang, Gao Wei, Ye Sanxing

(Department of Information Engineering, China University of Geosciences, Wuhan 430074, China)

Abstract: As a mature image segmentation method, Multi-scale segmentation is used widely in remote sensing image information extraction, but the whole algorithm efficiency is low. In this paper, data paralleled segmentation of remote sensing image is achieved based on multi-core computer. The traditional image IO method can't satisfy the needs of parallel computing based on multi-core computer. A new image IO strategy is devised to eliminate this drawback; In addition, in the process of parallel segmentation, it universally exists a problem that the segmentation results can not be merged directly. The method of re-segmentation is used to solve the problem to the special areas. The experimental results show that parallel segmentation efficiency has improved significantly on images of various quantity and size. Besides, the segmentation algorithm has been provided with the ability of processing image of large data quantity and the applicability is greatly enhanced. Image segmentation based on multi-core computer acquires remarkable effect.

Key words: Multi-scale segmentation; Multi-core computer; Image input and output strategy; Data partition merging