

均值漂移高分辨率遥感影像多尺度分割的集群实现

吴 炜^{1,2}, 沈占锋¹, 骆剑承¹, 陈秋晓³, 胡晓东^{1,2}

WU Wei^{1,2}, SHEN Zhan-feng¹, LUO Jian-cheng¹, CHEN Qiu-xiao³, HU Xiao-dong^{1,2}

1.中国科学院 遥感应用研究所, 北京 100101

2.中国科学院 研究生院, 北京 100049

3.浙江大学 城市规划系, 杭州 310058

1.Institute of Remote Sensing Application, Chinese Academy of Sciences, Beijing 100101, China

2.Graduate University of Chinese Academy of Sciences, Beijing 100049, China

3.Department of City Planning, Zhejiang University, Hangzhou 310058, China

E-mail: thuway@163.com

WU Wei, SHEN Zhan-feng, LUO Jian-cheng, et al. Implementation of multi-scales segmentation for high resolution RS images based on cluster. *Computer Engineering and Applications*, 2009, 45(34): 7-9.

Abstract: Multi_scales segmentation is important basis for high resolution RS information computation and key technologies for graphics information extraction. The existed multi_scales segmentation algorithms are usually memory cost, computation-intensive. What's more, these problems will become serious as the data accumulating and algorithms improving. To solve these problems, a parallel algorithm for mean shift multi_scales segmentation based on cluster is proposed and implemented, statistics the processing time, then analyzes and proves the effectiveness of the algorithm.

Key words: multi_scales segmentation; high resolution RS image; efficiency analysis; mean shift; cluster

摘 要: 多尺度分割是高分辨率遥感信息计算的重要基础, 是高分辨率遥感影像图谱认知中“图”提取的关键技术。当前提出的多尺度分割方法普遍存在着占用内存大, 耗费计算资源、计算时间长的缺点, 并且这些问题随着遥感数据量的增大、算法的改进等进一步加剧。针对这种情况, 根据当前集群计算技术的发展, 以均值漂移的多尺度分割方法为例, 实现了一种基于集群计算环境的多尺度分割算法, 集中解决任务分配和结果回收以及数据并行的方式, 统计了算法所消耗的时间, 对其的效率进行了分析, 通过实验说明了集群化对提高多尺度分割效率的有效性。

关键词: 多尺度分割; 高分辨率影像; 效率分析; 均值漂移; 集群

DOI: 10.3778/j.issn.1002-8331.2009.34.003 **文章编号:** 1002-8331(2009)34-0007-03 **文献标识码:** A **中图分类号:** TP14

以“像素”作为基本单元的像素级遥感信息处理和分析方式应用于高空间分辨率遥感存在着很多问题^[1]: 如单个像素所代表的地表面积较小, 表达一个地物需要多个像元共同表达, 更无法表达地物之间的关系。针对这种情况, 有研究者提出了面向对象的遥感影像处理与分析方法, 这种方法的核心在于通过对影像的分割, 使同质的像元组成大小不同的对象。因而, “对象”的提取成为面向对象分析方法的基础。

而多尺度分割是实现“像素”到“对象”转换的关键技术, 是高分辨率遥感影像处理和分析的基础环节, 是当前计算机视觉和遥感图像处理领域的热点问题。研究者也提出了多种分割方法, 如郭建聪等利用水分岭方法进行多尺度分割^[2]; 谭衢霖等利用异质性的区域增长进行对象提取^[3]; 当前比较成熟方法是

Definiens 公司开发的 eCognition 中使用多分辨率分割, 但是其分割算法属于商业秘密还没有完全公布^[4]。

这些算法都不同程度地提高了算法的精度和效率, 但是也普遍存在分割结果不稳定; 分割效率较低, 不适合处理海量的高分辨率遥感影像。Dorin Comaniciu 等提出了一种基于均值漂移的进行影像分割的方法不需要输入参数, 而根据图像本身特征进行分割, 取得很好的效果^[5]。针对遥感数据的处理效率较低的问题, 也存在耗用内存过多、计算复杂的问题, 不适合处理数据量较大的影像, 沈占锋等提出了基于 MPI 的遥感影像高效能并行处理^[6]。在以上研究的基础上, 以基于均值漂移的多尺度分割为例, 采用集群计算环境进行了实现, 提高了算法的效率。

基金项目: 国家自然科学基金(the National Natural Science Foundation of China under Grant No.40871203); 国家高技术研究发展计划(863)(the National High-Tech Research and Development Plan of China under Grant No.2009AA12Z123)。

作者简介: 吴炜(1985-), 男, 博士研究生, 主要研究领域为集群计算、遥感信息处理; 沈占锋(1977-), 男, 博士, 副研究员, 主要研究领域为分布式遥感影像处理; 骆剑承(1970-), 男, 博士, 研究员, 博士生导师, 主要研究领域为遥感信息处理与分析; 陈秋晓(1972-), 男, 博士, 副教授, 硕士生导师, 主要研究领域为城市遥感、城镇规划; 胡晓东(1982-), 男, 博士生, 主要研究领域为遥感信息计算和目标识别。

收稿日期: 2009-09-03 **修回日期:** 2009-10-12

1 均值漂移算法

1.1 均值漂移算法原理

均值漂移算法的基本思想是:通过反复迭代搜索特征空间中样本点最密集的区域,搜索点沿着样本点密度增加的方向“漂移”到局部密度极大点。其本质是自适应的梯度上升搜索峰的过程,是一种非参数的密度估计方法。广泛地用于变化检测、目标跟踪、影像分割等图像处理和计算机视觉领域^[7]。

其算法实现的一般步骤为:在特征空间中任意选择初始搜索半径为带宽为 h 的圆 O ,并根据核函数计算圆 O 范围内的采样点的密度估计函数 $mh(x)$,均值漂移向量可以记为 $Mh(x)=m(x)-mh(x)$,若其模 $\|Mh(x)\| < \varepsilon$,则迭代过程结束。否则,则以 $mh(x)$ 所在的点为圆心,重复上述过程,直到 $Mh(x)$ 的模小于容差 ε 。

从以上的介绍可以看出:均值漂移算法的关键在于采样点密度的估计 $mh(x)$ 计算的和带宽 h 的选择。一种经常采用的 $mh(x)$ 的计算方式如公式(1)所示:

$$mh(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (1)$$

其中,公式 $K(x)$ 为核函数, h 为带宽。对于核函数和带宽的选取已经有了较多的研究,这里不再赘述。

1.2 基于均值漂移的多尺度分割

从以上的分析可以看出:均值漂移的过程是使用带宽为 h 的窗口,根据公式(1)进行滤波的过程,该滤波很好地保留了边缘的信息,而消除了同一物体内部的差异性。将滤波后的图像进行聚类,即根据物体的光谱、空间等特征划分为不同的类别,在此基础上可以探测出物体的边界,从而实现多尺度分割。该方法一种非参数的密度估计方法,不需要对样本的分布进行任何假设,而只需选择核函数和带宽。从而多次分割结果具有相当的稳定性。

基于均值漂移的多尺度分割是在特征空间上进行的,首先需要将影像从 RGB 空间转换到 luv 的彩色空间,并将图像的行坐标 (x, y) 作为另外的两维,这样就构成一个五维的特征空间 (l, u, v, x, y) ,在此基础上进行基于均值漂移的滤波和聚类,可以探测出物体的边界,这就是基于均值漂移的影像分割过程。

1.3 基于均值漂移的多尺度分割存在的问题

基于均值漂移的多尺度分割需要将影像数据读入内存,并转换为 (l, u, v, x, y) 的五维空间上进行处理,并需要保存滤波、聚类等中间结果,消耗大量的内存资源。这对于处理高分辨率影像来说,就显得无能为力。例如:一幅 SPOT5 卫星获取的全色影像,每个像素采用 Unsigned int 存储,覆盖范围为 $60 \text{ km} \times 60 \text{ km}$,单景的融合影像就超过 1.5 G 。当前的计算机内存根本无法进行处理。

同时,由于需要不断地迭代寻找样本点中的局部密度极大值点,计算量巨大,往往无法满足其实用性要求。如采用四核,主频为 2.6 G 的计算机处理一景 3000×2000 的影像就需要 6 分钟以上,这无法满足遥感影像处理实时性的需要。

因此,内存和计算效率成为制约均值漂移影像分割应用的瓶颈,迫切需要对基于均值漂移的多尺度分割算法进行改进。一方面可以改进算法本身,提高算法的效率,如减少算法迭代的次数。但是,这种方法往往是以牺牲精度为代价的,并且提高效率也是有限的;另一方面,可以采用集群、大型机等并行计

算方式进行实现。而集群具有较高的性价比,较为成熟的编程环境,设计和实现一种基于集群的多尺度分割算法,解决分割算法的效率问题。

2 均值漂移多尺度分割的集群实现

2.1 集群计算环境

集群是一组独立计算机(节点)的集合体,其节点除了可以作为一个单一的计算资源供用户交互式使用外,还可以协同工作,并表现为一个单一的、集中的计算资源供并行用户使用^[8]。与传统的大型机相比,集群具有很高的性价比,利用现有的计算设备,通过网络互联,可以构建一个廉价的、具有较高性能的集群计算环境。

并且随着网络技术和节点之间的消息传递接口(Message Passing Interface, MPI)的发展,使得通信的成本逐渐减低。MPI 已经有了多种免费或者商业实现,如 MPICH、LAM_MPICH、MS_MPI,可以很好地满足节点之间通信的需要。网络技术和通信接口的发展,使构建起集群的编程环境变得相对容易,日益成为科研和生产单位构建并行计算环境的首选。

在集群计算环境中,每个节点都有独立的内存和处理器,整个集群的内存是各个节点内存之和,集群的内存突破了单机内存的极限,从而变得很大。利用数据处理之间的并行性,进行各个节点之间的并行处理,提高了处理的效率,可以解决计算复杂的问题。

2.2 基于集群的并行化实现的模型

在集群中,应该有一个节点负责和用户的交互(也可以参与运算),用户通过其指定需要处理影像和处理的参数,并将这些任务传递给中心节点;由中心节点接受任务,并分配给各个远程节点进行并行处理,处理结束后,并将处理结果进行收集。其软件结构如图 1 所示。

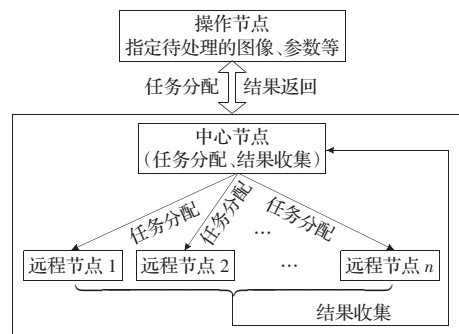


图 1 集群的软件结构图

2.3 并行分割实现的关键技术

2.3.1 任务分配与结果收集

并行算法实现的难点在于任务在不同节点之间的分配、图像和处理参数的传递和处理过程的同步。数据和参数在节点之间的传递主要保证数据的完整性和正确性。

一幅图像不仅有亮度值矩阵,还有图像的长度、宽度和波段数、存储格式(如数据类型、存储顺序等)等头信息。这些消息具有不同的结构,如亮度值矩阵是字符型,宽度和高度是无符号整型或整型,波段数是短整型。这些参数要是分散传输的话,在传输过程中很容易出现错误,如信息丢失,或者没有传输给相应的参数。而 MPICH 提供的构造类型 MPI_Type_struct ,较好

地解决了这个问题,一幅影像的头信息可以定义为:

```
MPI_Type_struct
{MPI_SHORT bandNum,MPI_UNSIGNED width,
MPI_UNSIGNED height,MPI_BYTE pixelType,
MPI_BYTE saveSequential}
```

这样就将图像的各种信息作为一种导出的数据类型,保证传输和接受的正确性。而图像本身的亮度值矩阵可以使用多维数组或者一维数组在不同节点之间的传输,需要注意的是数据的格式(如使用什么类型的数据进行存储以及存储的顺序)和数据的大小,这两种信息都可以保存在图像的头文件当中。

在传输处理数据的同时,需要传输进行处理的参数,图像具有相对固定的格式,而参数的随算法的变化很大,MPI 提供了数据打包的功能 *MPI_Type_indexed*,可以表示为:

```
<MPI_Type0,offset0>,<MPI_Type1,offset1>,...,<MPI_Typen,offsetn>
```

其中,前面是数据的类型,offset 是数据的偏移量,这样就可以实现异构的参数在不同的节点之间进行传输,并保证正确性。

2.3.2 多尺度的并行分割算法

基于集群的分割是通过多个节点协同执行同一操作来提高处理的效率,可以采用一种数据并行的方法,将数据分配到各个节点进行处理,数据分配的方式如图 2 所示。

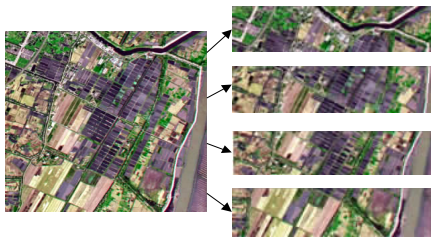


图 2 数据分块示意图(以四个节点为例)

在通过集群管理模块自动获取的一些集群的节点数 m 和节点的主机名、IP 地址等信息的基础上,其算法描述如下:

- (1)根据节点数,将待处理的图像分成相同大小的 m 个数据块,并将处理的参数和图像数据块进行打包,并传输给相应的节点;
- (2)各个节点进行并行的多尺度分割,由于各个节点的处理能力和图像本身的复杂程度不一致,需要在节点之间进行同步;
- (3)将各个节点的处理结果进行收集,并将分散图像处理的结果合并为一幅完整的图像。

分割结果在分块处会存在一条明显的分割线,影响了分割结果的精度,需要采取分块合并算法进行处理,这里就不再讨论。

3 实验与效率分析

所在研究团队长期从事遥感信息提取和高性能地学计算方面研究,形成了较为完善的底层模型,发展和实现多种遥感信息处理和分析算法,开发遥感信息计算软件 Since2008。在此基础上,利用 Visual C++结合 MPI 对上述分割算法进行了集群实现,并采用不同的大小的影像进行了分割实验,统计了分割时间,并据此分析算法的效率。

3.1 实验简介

3.1.1 集群平台

利用实验室的 4 台计算机,通过网络互联构成了一个集群计算的硬件平台,在此基础上安装了 Window XP 操作系统实

现对计算资源的管理,并安装了 MPI 的开源实现包 MPICH2,构建了一个异构的集群计算环境。各个节点之间的处理性能相差较大,其主要配置如表 1 所示。

表 1 集群中节点的配置情况

节点标识	品牌	机器名	处理器信息			内存/GHz
			处理器型号	核数	主频/GHz	
节点 1	Lenovo	WUWEI	Intel® Q600	4	2.00	2.0
节点 2	DELL	CHENGWUJIN	Intel® PWS490	2	1.86	2.0
节点 3	IBM	IRSA_UWAY	Pentium® M	1	2.00	1.0
节点 4	Lenovo	ZHOUYN	Intel® D	2	3.00	1.0

3.1.2 多尺度分割实验

分别选用 4 幅图像(为了表述的方便,以下由小到大依次称为图像 1~4),并采用不同的节点个数进行了分割实验。节点之间数据传输速度达到 54 m/s,并且是节点的内存之间进行传输,不需要写入硬盘,图像传输的时间相对于图像处理的时间不到 1/100,可以忽略不计。对 4 幅图像采用上述集群的不同的节点个数和相同节点个数不同节点进行处理所用的时间进行了统计,其结果如表 2 所示。

表 2 多尺度分割所花费的时间表

图像信息	1 个节点				2 个节点		3 个节点		4 个节点
	1	2	3	4	1-2	2-3	1-2-4	2-3-4	1-2-3-4
1 000×750	8.20	10.60	11.70	14.90	5.33	6.40	5.24	5.47	3.99
2 000×1 500	36.93	47.73	53.97	68.87	24.23	28.18	23.40	26.67	18.56
3 000×2 250	88.37	115.10	142.87	178.20	58.23	66.90	23.00	58.07	43.43
4 000×3 000	120.33	154.28	186.83	237.16	78.07	98.33	23.40	82.23	64.93

3.2 效率分析

将 4 个节点、4 幅影像的处理时间做成一幅折线图,如图 3 所示:可以清楚地看出:多个节点的进行处理所消耗的时间大致为最慢的单个节点所消耗时间的节点数分之一,用公式可以表示为:

$$Time(cluster) \approx \frac{\max(Time(0), Time(i), \dots, Time(m))}{m}$$

其中,Time(i)表示第 i 个节点处理所需要的时间,Time(cluster)表示集群所需要的时间,m 为集群中节点的个数。上述的节点公式,也能够从表 2 中的两个节点和三个节点的节点数中得到证明。采用数据的平均分配方式,需要等待最慢的节点处理完后才能进行返回,处理能力最慢的节点决定了集群处理的效率。

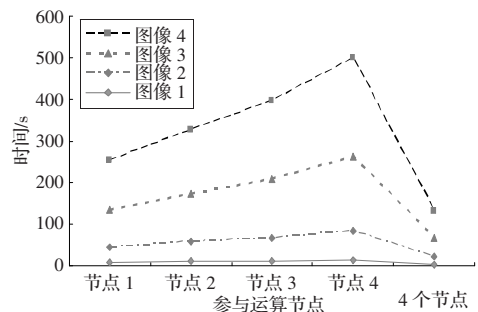


图 3 集群处理时间与单个节点处理时间的关系

4 结论

提出了一种基于集群计算环境的均值漂移多尺度分割算