

DEM 生成算法并行化研究

吕建峰 刘定生 焦伟利 李国庆

(中国科学院中国遥感卫星地面站, 北京 100086)

摘要 数字高程模型 DEM (Digital Elevation Model), 是一种表示三维空间连续起伏状态的数学模型, 如今在各行业应用十分广泛. 针对 DEM 生成过程中计算复杂、数据量大的特点, 在分析几种常用的 DEM 生成算法的基础上, 以线性内插算法为样本, 对 DEM 生成算法的并行化处理问题进行了深入研究. 研究中, 分别从数据并行和算法并行的角度, 对 DEM 生成算法并行化进行了分析, 并在网络分布式机群下进行了数据处理实验, 取得了较好的并行处理效果. 最后, 进一步根据实验结果, 讨论了两种分解方法的并行效率, 提出了 DEM 生成算法并行化的有效途径.

关键词 遥感 数字高程模型(DEM) 并行处理 算法并行 数据并行

中图法分类号: P237-39 P209 文献标识码: A 文章编号: 1006-8961(2002)05-0506-07

Parallel Algorithms Research on the Interpolation of DEM

LV Jian-feng, LIU Ding-sheng, JIAO Wei-li, LI Guo-qing

(China Remote Sensing Satellite Ground Station, Chinese Academy of Sciences, Beijing 100086)

Abstract Digital elevation model (DEM) are collections of elevation points and could be used to describe the geographical features and to show the terrain surface in three dimensions over a certain area. This features are more important for various applications. DEMs are created by collecting elevations and referencing them to corresponding points in the mapped area. The elevations add a Z value to the ground's X and Y horizontal coordinates. However, it is always more difficult to get DEM and normally needs more computing time to generate DEM since its complicated algorithm and huge data. In order to speed up the generating of DEM, this paper makes some sensitive study on the parallel processing of the interpolation of DEM. Based on the analyzing of several common interpolation methods of DEM, the linear interpolation (sometimes called Proportional Distance Method) was chosen as our basic algorithm. Meanwhile, data-parallel and algorithm-parallel approaches were applied to this basic algorithm. Through the experiences at networked parallel processing system some good results were obtained and proposal for further research on this topic was provided.

Keywords Remote sensing, Digital elevation model (DEM), Parallel processing, Algorithms-parallel, Data-parallel

0 引言

数字高程模型 DEM (Digital Elevation Model) 是一种表示三维空间连续起伏状态的数学模型, 如今应用十分广泛, 如铁路、公路勘探设计的选线, 地学分析中的坡度、坡向计算和地貌分析, 精确制导武器的地形匹配, 飞机导航, 地物的三维立体显示以及

数字正射影像校正等工作都将用到 DEM. 由此可见, DEM 在测绘、遥感、地质、地理和军事等领域都具有广泛的应用前景, 它已成为空间技术应用的重要研究课题之一.

如今, DEM 表示方法已有多种, 如规则网格模型、等高线模型、不规则三角网和数学分块法等, 其中规则网格模型是普遍采用的 DEM 模型. 在该模型中, 地形图可由一个带有高程值的二维矩阵所表

示,其不仅数据排列规则,且结构简单,用其计算分析每一点的高程值具有较好的实时性,也便于计算坡度、坡向、阴影和轮廓等;等高线模型则是首先提供地形图中每条等高线以及相对应的高程值,然后沿着等高线按一定距离采样,并储存其值,由于这样得到的只是分析区域的部分高程值,因此需要用内插方法确定其他点的高程值,故该模型通常需转化为规格网格模型或不规则三角网;不规则三角网(TIN)适用于采样点不规则分布的情况,因为它是直接利用原始离散采样点来表示地形,并将所有的采样点连成连续三角形,而三角形形状和大小则取决于采样点的位置和密度.由于这种 TIN 模型能根据地形的复杂程度改变采样点的密度和位置,故能避免数据冗余,而且能突出山脊、山谷、地形变化区等精度要求较高处的特征,TIN 还能较好地表示复杂地形,且绘制三维立体地形图效果较好.

由于规则网格模型能比较方便地存入计算机,且适合图象的计算、分析和处理,因此是目前应用最普遍的 DEM 模型.一般 DEM 生成是指由等高线模型或不规则三角网通过内插来生成规则网格模型的过程.另外,由于现有的 DEM 生成算法一般都是串行算法,因此在处理稍大图象时,速度显得较慢,可见,如何提高处理速度已是人们急待解决的问题.

由于 DEM 计算复杂、数据量大,比较适合利用并行计算来提高其处理速度,因此,并行计算已经成为当前国内外计算机科学研究与应用的前沿课题.国际上已有很多人研究 DEM 算法的并行实现途径^[1],如英国 Glamorgan 大学的 David Kidner 等采用 20 台 Pentium II 233 来做并行计算,其中的主要任务就是生成 130 000 000 点的 DEM,整个计算耗时约 20min^[2].

1 DEM 生成算法简介

DEM 的数据源可以通过对航测图象进行解析绘制等高线而得,也可以是通过实测离散点数据或通过数字化等高线而得.由于这些源数据是以不规则离散高程点的形式来表示一个规则平面的空间高度,因此需对这些源数据进行空间内插,以获得按规则网格排列的数字高程模型.

DEM 内插方法主要有线性内插、多项式内插、样条函数内插、傅立叶级数内插、最小二乘趋势内插及平均移动法内插^[2,3]等.目前大多数 DEM 软件均

采用多项式内插和线性内插,如,GRASS 采用线性内插及平均移动法;ARC/INFO 采用线性内插和五次多项式内插(TIN)及双线性方程内插、三次卷积积分内插(Lattice)等^[4].下面简介其中几种:

1.1 窗口拟合法

这种方法是通过计算以待插点为圆心,以 R 为半径的圆内诸点的加权平均值来确定高程值,其内插的精度由已知点到内插点的距离来确定,一般以距离的平方作为权倒数来进行加权平均.设 n 为圆内插值点的个数,则插值公式为

$$Z(x, y) = \frac{\sum_{i=1}^n Z(x_i, y_i) d_i^2}{\sum_{i=1}^n d_i^2} \quad (1)$$



图1 窗口拟合法

1.2 移动拟合法

移动拟合法是典型的逐点内插法,该方法是通过对待插点取用一个多项式曲面来拟合该点附近的地表面.这里设取二次多项式来拟合,则一定区域内特定点的高程值可写为如下一般式:

$$Z_p = AX^2 + BXY + CY^2 + DX + EY + F \quad (2)$$

若将坐标原点平移到待插点,即 $X' = X - X_p$, $Y' = Y - Y_p$,则代入上式,即得移动拟合法插值公式

$$Z'_p = AX'^2 + BX'Y' + CY'^2 + DX' + EY' + F' \quad (3)$$

而且由于上式中有 6 个待求参数,因此需最少取用 6 个数据点(距离待插点最近)的 (X, Y, Z) 代入式(2)才能求得.对于待插点 $X' = Y' = 0$,式中 F' 值即为待插点的高程值.由此可见,在确定 6 个参数时,最好选用多于 6 个数据点,再利用最小二乘法来求解.这样计算就非常复杂,同时因为移动拟合法是 DEM 生成中很常用的方法,所以其并行化很具有实际意义.并行化时,只需采用数据并行方法,即先把要处理的原始等高线图划分为若干小块,然后分给多个处理机同时并行计算.由于处理每一点时,需要周围一定范围内的数据,所以分发数据时,各个小块应加上阴影数据,以便每个处理机不需要数据交

换就能独立处理该块内的每一点。

1.3 线性内插法

在由等高线数据内插 DEM 时,文献[5]中,Shuji Murai 提出了剖面内插法和比例内插法(如图 2、图 3)两种方法,这两种方法均属线性内插方法。

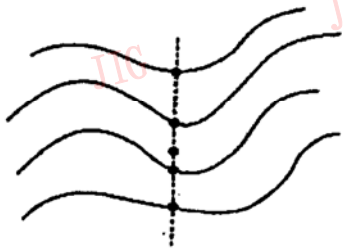


图 2 剖面内插法

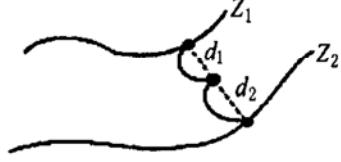


图 3 比例内插法

其中,剖面内插法就是每一点的值通过剖面上各等高线值内插得到.这样虽结果图效果柔和,但是会损失很多细节,而且在一些变化比较大的区域还会出现错误;比例内插法中,待插点的值是由离该点最近的两条等高线来决定,其插值公式为

$$Z(X, Y) = \frac{Z(X_1, Y_1)d_2 + Z(X_2, Y_2)d_1}{d_1 + d_2} \quad (4)$$

其中, $Z(X_1, Y_1)$, $Z(X_2, Y_2)$ 为距离最近的两条不相同等高线的高程值. d_1 , d_2 为该点距两条等高线的距离值.这种方法在下列情况下运行效果较好:等高线延伸到所处理区域的边界、等高线比较密集、等高线连续、等高线之间没有孤立点等.其程序的运行速度直接取决于等高线的疏密程度。

本文所用的内插方法是以上两种方法的综合,并在搜索方案上提出逐象限搜索法,即把待插值点作为原点,首先,在 4 个象限各搜索两条不同的等高线,然后判断搜索得到的等高线之间的关系,再选择两条合适的等高线进行插值.最初选择的算法是在每个象限只搜索一条等高线,但是对于图 4 中出现的情形,这种算法则会出现错误的内插结果或无法进行内插:

对于图 4(a) 中 A 区域,由于它位于整幅图的左上角,因此如果采用 4 个象限各搜索一条等高线的

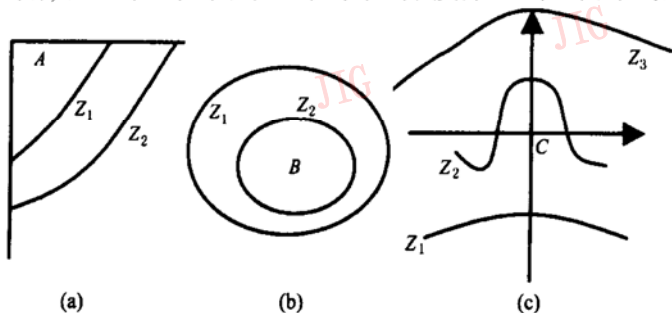


图 4 等高线的特殊情况

方法,则只能找到等高线 Z_1 ,这样整个 A 区域只能赋予同一个值,都等于等高线 Z_1 的高程值;图 4(b) 中, B 区域周围有封闭等高线存在,即位于山峰或山谷地区,如果只搜索一条等高线,则 B 区域也只能赋等高线 Z_2 的值,这样将出现平顶现象.由此,对这两种情况, A 区域及 B 区域应该根据周围其他等高线的趋势给予赋值,而不应该为一个值.对于图 4(c) 中的原点 C 及周围区域,同样只能搜索到等高线 Z_2 ,致使 C 点及周围均为 Z_2 的高程值,从而出现错误.其实, C 点高程值理应为等高线 Z_1 与 Z_2 的内插结果。

如果采用每象限搜索两条等高线的方法,则可以解决上述问题.处理前,首先参照图 5 对原始等高线数据结构化,然后在处理某一点时,先读取该点的结构,如果中间项不为零,则不需进行内插,可直接赋以高程值,否则需要进行内插处理.根据结构的第 1 和第 3 项即能知道该点同行左右方向第 1 条等高线的位置,然后向上、下搜索.假如向上搜索,先读取该点上一行象元的数据,如果象元高程值为零,则读取左右等高线的信息,再计算等高线与待处理点的距离,并与前面的等高线比较,从而找出当前距离最近的等高线位置;如果搜索过程中,象元的高程值不为零,则表示该象元为等高线上的点,也说明搜索过程要穿越等高线,此时需对后面搜索到的等高线作一标记.搜索过程以到达边界或已找到两条不同等高线,且不能找到更近的等高点时作为停止的标准.4 个象限搜索结束后,再把得到的等高线按一定规则排列.这些规则主要有:①去除高程值相同的等高线;②被加了穿越标记的等高线均应排在没有标记的等高线后面;③同一类等高线按距离远近排序,近的排在前面,排好以后,再利用排在前两位的等高线进行内插.如果最后得到的等高线只有一条,那么就将其赋为等高线的高程值。

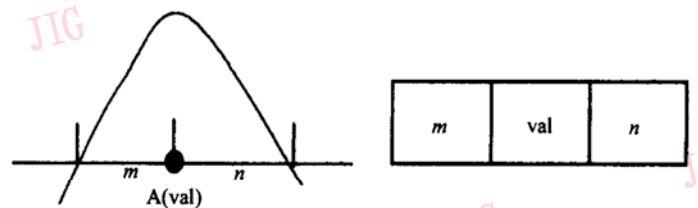


图 5 等高线文件数据结构化示意图

图 4(a) 和图 4(b) 的情况类似,当搜索两条等高线时,均可以通过穿越最近的等高线来找到第 2 条等高线,再根据趋势内插出高程值,如图 6 所示。

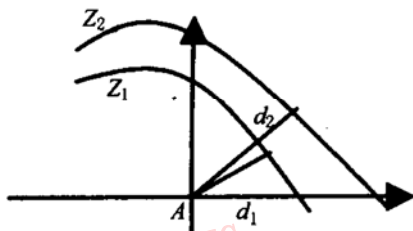


图 6 趋势内插法

图中 Z_2 为穿过 Z_1 找到的第 2 条等高线, d_1 为 A 和 Z_1 的距离, d_2 为 A 和 Z_2 的距离, 由于 Z_2 有穿越标记, 所以用下式进行内插:

$$Z(x, y) = \frac{Z_1(x_1, y_1)d_2 - Z_2(x_2, y_2)d_1}{d_2 - d_1} \quad (5)$$

图 4 中 A 和 B 区域均可进行趋势内插, 而不再是同一个值. 对于图 4(c) 情况, 其搜索结束后, 等高线的排序结果为 Z_2, Z_1, Z_3 , 此时取 Z_2 和 Z_1 进行内插, 即可以得出 C 区域正确的高程值.

2 DEM 生成算法的并行化分析

2.1 算法的选取及串行算法分析

为使 DEM 生成算法并行化的研究集中在 DEM 生成过程中数据特点与算法结合的并行处理上, 本文选取具有典型代表意义的线性内插算法进行了并行化研究. 由于该算法的特殊性, 因此可以分别从数据并行和算法并行的角度进行并行处理, 以便进行算法并行和数据并行的对比分析^[6].

在这里, 线性内插算法的源数据是栅格化的等高线文件, 目标是通过内插处理形成规则网格模型数据, 以便使网格中每一点都有相应的高程值. 参照 GRASS 软件中的线性内插算法^[7], 对其进行部分改进后的算法框图如图 7 所示. 改进后的线性内插算法流程如下:

(1) 程序对源数据文件进行预处理, 并在读取栅格化等高线图象时, 把每一点结构化(如图 5 所示).

由图 5 可见, 数据结构包括如下 3 项: 其中, 中间一项为读取的原数据, 若为零, 即为待插点, 若非零, 则表示有等高线通过; 第 1 项为该点距同一行左边第 1 条等高线的距离 m , 如果左边无等高线, 则该值为零; 第 3 项为该点距右边第 1 条等高线的距离 n . 在对后续点进行结构化的同时, 需把前面结构化的结果存盘, 以便最终在硬盘上产生一个结构文件.

(2) 根据系统内存大小和所处理的图象列数,

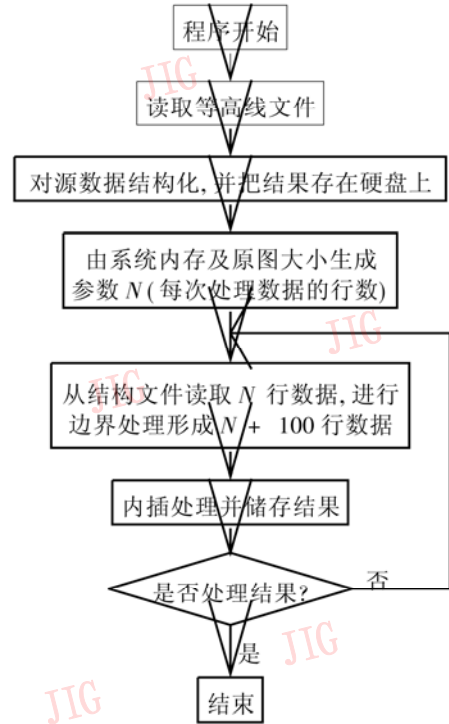


图 7 线性内插串程序框图

动态地产生处理参数 N (每次处理的行数). 内插时, 每次从结构文件中读取 N 行数据, 同时进行边界处理生成 $N+100$ 行数据, 即在 N 行数据上下各多读 50 行数据, 当遇到上下边界时, 不足 50 行, 则补零. 这样, 在开始确定 N 的大小时, 实际上得到了 $N+100$ 行图象数据.

(3) 系统将 $N+100$ 行结构化数据读入内存, 对 N 行中的每一点进行内插处理. 处理某点时, 以该点为原点, 再按从第 1 到第 4 象限的顺序, 在 4 个象限内各找到两条离这点最近的等高线, 通过集中比较后, 找出其中两条距离此点最近, 且代表不同高程值的等高线来进行线性内插; 在处理的同时, 储存结果, 即每处理 N 行就存储结果, 并把内存及时释放, 以供下面的计算利用.

上述处理流程中, 其中最后一步从读取结构文件到内插处理结束这部分计算复杂度高, 重复计算量大, 比较适合于作分布式并行计算.

2.2 算法的并行化分析

近 10 年来, 人们在探索高效的并行处理途径时, 提出了许多并行计算模型, 如 PRAM、VLSI、BSP 以及分布式计算模型 $\log P$ 和 C^3 模型等^[8], 同样, 对并行算法也有多种. 典型的常用算法有: ① SIMD (单指令流多数据流), 即各处理机同一时刻执行相同的指令, 仅数据不同; ② MIMD (多指令流多数据流)^[9], 即各处理机对于不同数据可以独立进行操作. 这两种方式, 从更广泛的意义上来看, 亦可

称之为数据并行和算法并行,下面就从这两种方法讨论 DEM 线性内插算法的并行化。

(1) 数据并行实现

本文采取主从方式进行数据并行处理,即一台主机控制 M 台从机进行并行计算,由主机进行原始数据的分发、处理结果的接收、以及控制各从机的同步等,而从机则按照主机的命令来对不同的数据进行相同的并行处理。

串行算法中,从读取结构文件进行插值处理到储存结果这一部分,需对原图象进行逐行重复性处理,因数据计算量很大,可以考虑用数据并行的方法来实现并行处理。而对原图进行结构化处理部分实现并行则比较困难,故仍采用串行方式处理。由于串行处理部分占整个程序的比例较大(约7%~8%),这在一定程度上制约了整个处理效率的提高。

由于结构化后的数据量是原数据量的3倍,如果把结构化后的数据由主机向从机分发,则显得通讯量太大,而且在主机结构化原数据的同时,所有从机均处于等待状态,这就降低了并行效率,所以实现结构化时,采用主机只向从机发送原始数据,并在逐行发送的同时,所有从机均进行结构化处理,并把结果存在各自的硬盘上。这样原始数据发送结束时,从机就各自有了结构化后的数据。

随后,各从机根据内存和所分配给的图象大小,确定参数 N (每次处理的行数),并将图象分割为一系列 N 行数据,分别进行并行处理。处理时,各从机

从本地硬盘读取 N 行待处理数据,并作必要的边沿处理后,按照内插公式进行处理,处理后把结果直接送回主机。主机则循环接收 M 台从机的结果数据。从机处理完 N 行数据后,再读取下一个 N 行数据,直到结束。

在这种并行实现方法中,主机在从机进行内插处理时,只收集数据和存储数据。当有 M 台从机计算时,主机则采用循环逐行接收法,这样在每台从机完成一行数据的处理时间内,主机要接收 M 行数据并完成存储。实测结果表明,程序运行时,主机和从机的 CPU 利用率均达到 99% 以上,基本上实现了各处理机间的负载平衡。

这种并行方法易于掌握,且编程简单,但并行效率也易受到限制,根据加速比定律,由于算法中所有运算均需进行,同时由于通讯带宽的限制和通讯延迟及同步延迟,当并行规模为 p 台处理机时,其加速比小于 p ,因此不可能出现超加速比,后面的实验结果也证实了这一点。

实验测试时,当处理机数 $p = 2$ (即一台主机和一台从机时),因为运算过程和串行处理程序基本相同,没有实测意义,所以从 $p = 3$ 开始测试。测试结果见表 1、图 8、图 9。其中,原始等高线图象数据大小分别为系列 1: 1024×1024 、系列 2: 2048×2048 、系列 3: 4096×4096 ,3 幅图象来自同一幅大图象的不同部分,等高线的疏密程度基本类似,比较便于讨论并行处理效率。

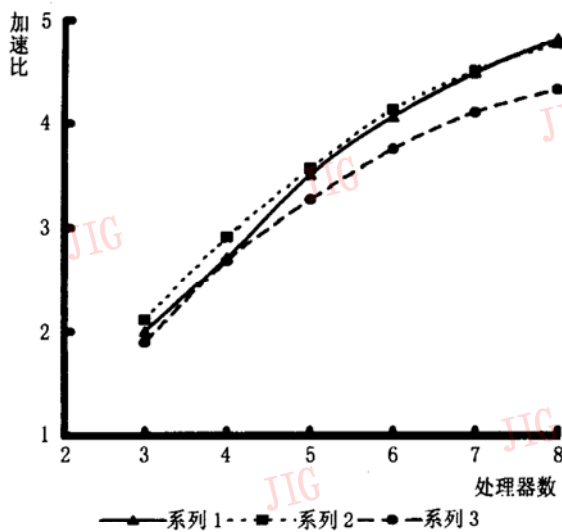


图 8 数据并行处理加速比曲线

(2) 算法并行实现

采用数据并行处理,由于受到多种物理因素(如处理机的运算速度、通讯带宽等)的限制,因此处理机数量增加到一定程度时,其加速比将趋于常数,且

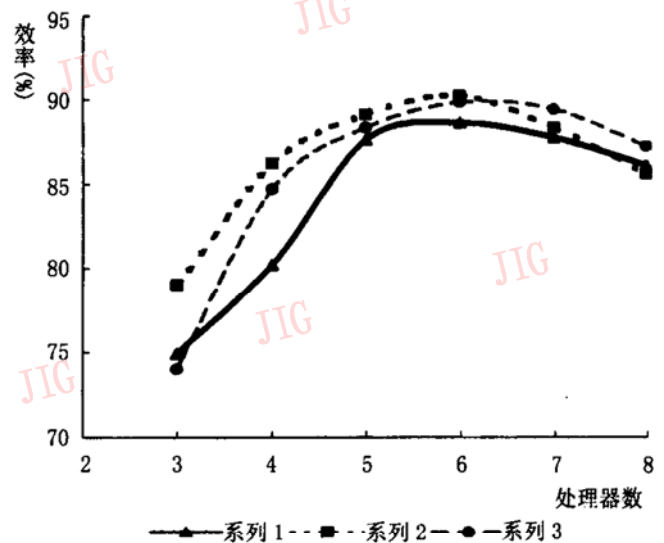


图 9 数据并行处理效率曲线

并行处理速度不再提高,于是人们开始从算法并行的角度来研究并行处理,希望能得到更高的加速比。简单的说,算法并行泛指在同一个并行处理系统中,不同的处理机分别完成同一个处理程序中不同的处

理算法部分,即不同的处理机其处理的算法不同,但所处理的数据可以相同或不同。

对于不同的算法,为实现高速并行处理,其所采用的算法并行化策略也许不同.特别对于某些搜索算法,在并行化时,如果把搜索的中间结果反馈给其他进程,则有可能中途就已经得出结果,这样就可能提前取消串行算法中无谓的搜索,从而可以提高加速比,甚至出现超加速比.因为 DEM 内插生成算法就比较类似于这种情况,所以可考虑用算法并行的方法来对其并行化,具体实现方式如下:

选取处理机数 $p = 5$,内插前,主机把全部原始数据发给从机,而各从机则在接收数据的同时进行结构化处理,并在本地硬盘中存储结果.在内插时,每个从机都参与每一点的处理.由于串行算法中,每象限要找出两条不同的等高线(如果存在),因此只有当 4 个象限都搜索结束,并已将得到的 8 条等高线(一部分是相同的)进行过比较处理后,才能确定待插点的高程值.算法并行处理时,由于考虑 4 台从机是分别在以待插值点为原点的 4 个象限内进行搜索,且找到一条最近的等高线就停止搜索,并把结果返回主机,为此,在主机处理收集的 4 条等高线中,如果 4 条等高线是同一等高线或者不适宜插值,那么就通知各从机进行第 2 轮搜索,进一步搜索与第 1 轮不同的等高线;如果第 1 轮搜索结果已经满足插值需要,那么主机就进行内插运算,并通知各从机进行下一点的处理,这样就节省了

处理该点所需的搜索时间,从而使并行效率大为提高。

由于算法中把插值计算的部分计算量(判断比较和内插)移到了主机,而判断的时候,从机处于等待状态,因此整体看来,主机的负荷稍重.实测中,主机和从机选用了相同性能处理机,这样在程序运行时,主机 CPU 利用率大于 99%,从机 CPU 利用率为 97~98%.如果选用主机的性能稍强于从机,那么就能得到更理想的并行效果。

另外,算法中主机与从机需进行频繁的数据交换,由于其通讯本身的建立就需要一定时间,因此过多的中间结果交换将使并行效率有所降低,这是本算法有待改进的地方之一。

由于处理机数与算法的设计直接相关,不能随意增减,所以只测了 $p = 5$ (4 台从机)和 $p = 3$ (两台从机)的情况,将其和数据并行相同条件进行比较,其加速比和效率都有明显提高。

3 DEM 生成算法并行处理测试结果

在由 8 台 Pentium Pro 微机构成的网络分布式并行处理环境下,对上述算法进行了测试实验.该系统使用标准 100M 网络交换机来实现各微机的互联,系统配置 Linux 操作系统和 MPI 并行计算平台.各微机内存为 128M,主频 200MHz.实验结果见表 1、表 2。

表 1 数据并行测试结果

图象系列	计算规模		处理机数					
			3	4	5	6	7	8
系列 1	图象大小: 1 024× 1 024 串行处理时间: 130s 其中: $t_1 = 8s, t_2 = 122s$	并行处理时间(s)	65	48	37	32	29	27
		加速比 S	2.00	2.71	3.51	4.06	4.48	4.81
		效率 S/S_i	0.749	0.802	0.876	0.886	0.877	0.861
系列 2	图象大小: 2 048× 2 048 串行处理时间: 500s 其中: $t_1 = 31s, t_2 = 469s$	并行处理时间(s)	237	172	140	121	111	105
		加速比 S	2.11	2.91	3.57	4.13	4.50	4.76
		效率 S/S_i	0.790	0.862	0.891	0.902	0.883	0.856
系列 3	图象大小: 4 096× 4 096 串行处理时间: 1 998s 其中: $t_1 = 175s, t_2 = 1 823s$	并行处理时间(s)	1057	745	611	533	487	462
		加速比 S	1.89	2.68	3.27	3.75	4.10	4.32
		效率 S/S_i	0.740	0.847	0.883	0.898	0.894	0.872

表 2 算法并行测试结果

	图象大小					
	1 024× 1 024		2 048× 2 048		4 096× 4 096	
处理机数量 p	5	3	5	3	5	3
并行处理时间(s)	34	50	127	179	558	756
加速比 S	3.89	2.44	3.94	2.79	3.58	2.64
效率 S/S_i	0.953	0.973	0.983	1.046	0.967	1.034

这里,串行处理时间 T_s .即单机按串行算法进行

处理时所需的实际时间.从算法分析的角度来看,可将串行处理时间分解为 $T_s = t_1 + t_2$,其中, t_1 为串行程序中不可并行处理部分的运行时间, t_2 为串行程序中可以并行处理部分的运行时间.并行处理时间 T_p 为在指定的处理机数 p 下,实际的并行处理时间。

理论并行处理时间为: $T_i = t_1 + t_2/p$; 实际加速比为 $S = T_s/T_p$; 理论加速比为 $S_i = T_s/T_i$; 并行实现

效率为 $S/S_i = T_i/T_p$.

4 结果分析与讨论

由表1、表2、图8、图9可见:

(1) 对于某一幅图象,由表1和图8可知,随着处理机的增加,其并行处理时间越来越短,且加速比变大.当处理机数目继续增加时,加速比增长变慢,最后趋于某个固定值 S_0 .在这里, S_0 可以称为该系统下此算法的最大加速比.该数值大小将取决于系统的处理机性能、通讯带宽和算法设计中的通讯次数以及每次通讯的数据量.欲进一步提高加速比,须从改进算法,减少处理机间通讯延迟,增加数据交换带宽和改进处理机计算能力等入手.

(2) 由图9可见,当处理机数目超过一定数量时,系统的并行处理效率开始降低.这可解释为当处理机数量增加时,系统中各处理机之间的通讯与数据交换时间将大为增加.当处理机数量超过某一临界点后,这种附加的非数据处理时间将逐步占据主要位置,从而使并行处理效率降低.

(3) 由图8和图9可见,本算法中当处理的图象变大,也即通讯次数和通讯量增加时,则并行处理的加速比和效率均开始下降.若要维持高的加速比和效率,则要重新改进算法.

(4) 比较数据并行和算法并行的测试结果,发现算法并行处理的加速比和效率均高于数据并行处理.而且在有大规模数据交换和通讯延迟的情况下效率可超过1,这直接证明了超加速比的存在,同时也说明,对于DEM生成,采用良好设计的算法来进行并行处理,可得到较好的综合性能.

(5) 本项研究采用了较为简单的线性内插算法,其所形成的并行处理方式同样可适用于其他的内插算法.由于线性内插算法计算的复杂性较低,若对其他更为复杂的算法,在直接或适当修正并行分解方式后,则有可能获得更好的并行处理效果.

(6) 本实验中所采用串行算法是在GRASS软件原算法的基础上经过改进形成的.由于原算法是直接对等高线文件进行处理,而没有结构化的步骤,因此其串行运行时间是现算法的好几倍.但原算法经并行分解后的并行处理效率很高.由此看来,算法的计算复杂度越高,算法并行效果越好,而算法越优化,则算法并行效率就会降低.

参考文献

- 1 Kidner D B, Rallings P J, Ware J A. Parallel processing for terrain analysis in GIS: Visibility as a Case Study [J]. *GeoInformatica*, 1997, 1(2): 183~ 207.
- 2 David Kidner, Ian Fitzell, Phil Rallings *et al.* A distributed approach for planning radio communication[EB/OL]. <http://www.geovista.psu.edu/sites/geocomp99/Gc99/083/gc-083.htm>, GeoComputation99, Mary Washington College in Fredericksburg, VA, USA, on 25-28 July 1999.
- 3 黄力,邱涤珊,王峰. GIS中DEM的表示方法[J]. *电脑与信息技术*, 1999, (2): 10~ 13.
- 4 陈军,赵仁亮. GIS空间关系的基本问题与研究进展[J]. *测绘学报*, 1999, 28(2): 95~ 102.
- 5 郭清. DEM内插方法比较及ARC/INFO内插方法分析[J]. *铁路航测*, 1999, (1): 7~ 11.
- 6 Shuji Murai. GIS Work Book [M]. Japan Association of Surveors, 1996.
- 7 吕建峰. 遥感数据处理算法并行研究及其在正射影像中的实现[D]. 北京:中科院中国遥感卫星地面站, 2000.
- 8 Steven F. Clamons, GRASS4.2 Programmer's Manual, 1997
- 9 李晓梅,莫则尧,文尚猛等. 面向结构的并行算法——设计与分析[M]. 长沙:国防科技大学出版社, 1996.
- 10 孙家昶,张林波,曲学斌等. 网络并行计算与分布式编程环境[M]. 北京:科学出版社, 1996.



吕建峰 1997年毕业于北京大学电子学系,中国科学院遥感卫星地面站硕士研究生.主要从事遥感图象并行处理方面的研究.



刘定生 中国科学院中国遥感卫星地面站副站长,研究员.主要研究方向为数字图象处理、遥感信息提取、遥感数据并行处理及大型遥感图象快速处理系统等.



焦伟利 1988年毕业于武汉测绘科技大学,现为中国科学院中国遥感卫星地面站副研究员.主要研究方向为遥感图象处理、地理信息系统及其应用等.



李国庆 1990年毕业于北京大学地球物理系,现为中国科学院中国遥感卫星地面站副研究员,中国科学院地理科学与资源环境研究所和中国科学院中国遥感卫星地面站联合培养在职博士研究生.主要研究方向为遥感图象并行处理、大型遥感图象处理系统等.