

# 面向离散点云并行插值数据划分方法研究

钱 辰<sup>1</sup>, 窦万峰<sup>1,2</sup>

(1. 南京师范大学计算机科学与技术学院, 江苏 南京 210023)  
(2. 江苏省信息安全保密技术工程研究中心, 江苏 南京 210097)

[摘要] 插值是数字高程模型的核心分析方法,也是构建数字高程模型的常用手段. 面对海量的离散点云数据,插值生成 DEM 的过程需要消耗大量的时间,将并行计算运用到插值计算中会显著缩短计算时间. 结合点云数据分布不均匀的特点,本文提出了一套针对离散点云并行插值生成格网 DEM 的数据划分方法,将读取、搜索邻域、插值计算、输出 4 个过程的处理时间量化,保证划分后形成的子块处理时间均衡,提高了并行计算效率.

[关键词] 数字高程模型,并行插值,离散点云,数据划分

[中图分类号] TP301.5 [文献标志码] A [文章编号] 1672-1292(2013)02-0063-05

## Research on Data Partition Method for Parallel Interpolation of Discrete Point Cloud

Qian Chen<sup>1</sup>, Dou Wanfeng<sup>1,2</sup>

(1. School of Computer Science and Technology, Nanjing Normal University, Nanjing 210023, China)  
(2. Jiangsu Research Center of Information Security and Privacy Technology, Nanjing 210097, China)

**Abstract:** Interpolation is the core analysis method of the digital elevation model, which is also a common means of the construction of the digital elevation model. It consumes a lot of time to generate an interpolated DEM when discrete point cloud data is magnitude. Applied parallel computing to interpolation calculation can significantly shorten the calculation time. Taking the point cloud data distribution characteristics into consideration, this paper raises a data partition method, which is used for parallel interpolation of discrete cloud points to generate grid DEM. This approach quantifies the processing time of four procedures, such as input, searching neighborhood, interpolation calculation and output. This method can keep the balance of the processing time of the sub-blocks thereby improving the efficiency of parallel interpolation.

**Key words:** digital elevation model, parallel interpolation, discrete point cloud, data partition

插值是连接地理信息系统中栅格和矢量两种数据格式的常用方法,是数字高程模型的核心技术和基本工具,也是数字高程模型表面建模的主要方法. 空间插值是用已知点的数值来估算其他点的数值的过程<sup>[1]</sup>. DEM 内插是指利用内插点空间位置相近的已知点的高程值求出未知点的高程值. 它是 DEM 的核心问题,贯穿于 DEM 的生产、质量控制、精度评定、分析应用的各个环节. DEM 数据在原始离散点较少时, DEM 插值速度较快. 但实际运用中,当使用单台计算机对大面积离散点进行插值时,由于插值算法的复杂性等原因,计算机的内存消耗加大,时间消耗将呈级数增长,如何加快插值速度是目前研究的热点问题.

高性能并行 GIS 是基于高性能计算的理论、方法和技术,在并行计算的软硬件架构支持下,通过对 GIS 中属于计算密集型或数据密集型的空间数据操作算法的并行化设计和开发,实现海量空间数据的高性能存取与处理操作<sup>[2]</sup>. 而并行计算是实现高性能计算最常用的方法.

并行计算中,数据划分是系统负载均衡的前提. 数据划分粒度的大小又涉及到数据和任务本身的属性以及相互之间的依赖关系. 因此合理的数据划分方法可以为具体并行环境下数据和任务的划分起到指导性的作用. 点云数据是构建 DEM 的主要数据源且数据量巨大,因此本文主要讨论离散点云并行插值生成

格网 DEM 的数据划分方法,从插值过程的执行时间角度考虑,借鉴传统的数据划分方法,提出了一套基于处理时间均衡的数据划分方法.

## 1 研究现状

传统的数据划分方法主要有:轮转法、散列划分、范围划分和混合范围划分法.在并行空间运算中,一维划分的主要问题是只从元组个数上考虑负载均衡,未考虑空间数据的空间属性特征和非均匀分布的特性,从而导致划分的空间数据可能数据数目均等,但数据量却分布不均.基于 ID 的划分根据空间对象的 ID 顺序将空间对象存入并行存储节点,当节点饱和时顺序存入下一节点,该方法考虑到并行存储节点上数据个数的均衡且易于实现,但方法非常粗略,是完全不考虑空间对象的任何属性或者关系特征的一种方法.Oracle Spatial<sup>[3]</sup>提供的空间数据划分算法是基于  $X$  或  $Y$  坐标值的范围划分,该方法仅考虑到空间对象的地理位置,可以保证划分后每个子网格内的空间数据本身有良好的空间相邻性,但无法保证节点间均衡的数据量.王永杰等<sup>[4]</sup>、周艳等<sup>[5]</sup>提出了一种基于 Hilbert 曲线层次分解的空间数据划分方法,该方法的优点是对于划分结果能够保持很好的空间邻近性且保持比较均衡的数据分布.

在针对 DEM 插值的数据划分方面也有不少研究成果.Shaowen Wang 等<sup>[6]</sup>使用 Globus Toolkit,针对密度不等的数据点通过基于四叉树的域分割算法,实现了在网格计算环境下并行空间内插算法的研究.吕建峰等<sup>[7]</sup>、Fang Huang 等<sup>[8]</sup>、Lluís Pesquer 等<sup>[9]</sup>采用按行划分的方法进行逐行内插,为保证划分后子数据块边缘点插值对邻域的需求,将按行划分的数据块上下各多分配一定的数据条宽度,通常选择 DEM 分辨率的一倍或两倍.Marc P Armstrong 等<sup>[10]</sup>采用了按列划分的方法,划分方法与按行划分基本相同.Shaowen Wang 等<sup>[6]</sup>、Xuefeng Guan 等<sup>[11]</sup>对数据进行正方形的块划分,正方形的公共边需要被相邻的两个正方形分别读取.胡金星等<sup>[12]</sup>采用了自适应格网分割方法,按照坐标轴交替分割直到每个数据块只有 2~3 个数据点.

结合离散点云插值生成 DEM 的计算特点,可以发现:由于大多数情况下,采样点的分布是不均匀的,等面积和等采样点的划分方法都会使划分后的子网格计算量不均匀,从而导致并行效率不高.等面积的划分方法下,插值算法执行的次数是相同的,但由于采样点分布不均匀,相同面积内的采样点个数不同,则计算机读取相同面积的数据所消耗的时间会产生细微的差异,同时密度的差异导致每次插值过程中寻求最邻近点的搜索时间也产生差异.等采样点的划分方法下,点分布不均可能会导致按照点数相等的原则划分后形成的子块的面积相差很大,面积的差异意味着插值计算执行的次数差异,即使数据的通信消耗的时间极短,大多数情况下可忽略不计,但若面积差异足够大,则应考虑数据的通信也需消耗一定的时间.同样,等点数划分下的面积不相等,会造成点的密度不均匀,从而影响邻近点的搜索时间.综合通信时间、计算次数和搜索时间的因素,等采样点的划分方法也有其局限性,制约了并行效率的提高.

## 2 基于处理时间均衡的数据划分方法

通过对现有划分方法的分析,本文提出一种基于计算时间均衡的数据划分方法,一定程度上可以克服上述方法的缺点,保证计算机对划分形成的子块的处理时间基本相同,提高了并行效率.

### 2.1 构建最小外接矩形

首先,构建原始数据的最小外接矩形,设矩形的长为  $m$ ,宽为  $n$ .设插值生成的目标 DEM 的分辨率是  $m_0 \times n_0$ .假设  $m$ 、 $n$  分别是  $m_0$ 、 $n_0$  的整数倍,不足整数倍可按边缘数据值补足.

### 2.2 基于四叉树的点云划分

划分的基本过程为:将原始图像(即最小外接矩形)分成 4 个相同大小的区域,计算每个区域内采样点的个数,并进行一致性检验,即判断每个区域点数是否接近于  $k$ ,若符合一致性要求,则停止划分,不符合则将该区再细分为 4 个相同大小的区域,并对每次细分后的区域进行一致性检测.这个迭代重复的过程直到点云中的所有区域都符合一致性标准才停止.一致性标准是指区域内采样点的个数不超过某个阈值  $k$ .图 1 所示为划分示意图.

### 2.3 基于线性四叉树的层次编码

如图 2 所示,线性四叉树的编码有如下规则:

(1)方向性:按照图 2 中的编码基准,以西北角为起点按顺时针方向递增.

(2)层次性:第  $n$  层子块的四进制编码应为: $q_1q_2\cdots q_n, q \in \{0,1,2,3\}, i=1,2,3,\cdots,n$ .

在这些编码中,处在第 1 位的是第 1 层,第 2 位是第 2 层,第  $n$  位是第  $n$  层( $n$  表示划分层次).例如图 2 中的 231 子块,按照图 1 的参考基准,表示该子块在第 1 层中的位置为 2,第 2 层的位置为 3,第 3 层的位置为 1.

## 2.4 处理时间分析

本文提出的基于处理时间均衡的数据划分方法,主要目

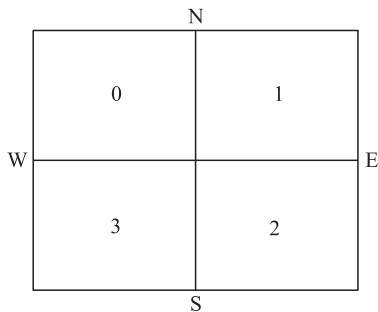


图 2 最小外接矩形的线性四叉树编码

Fig.2 Linear quadtree codes of minimum bounding rectangle

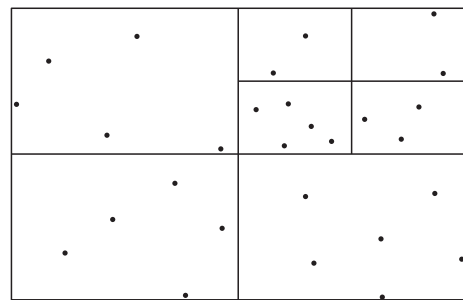


图 1 点数均衡的四叉树划分

Fig.1 Diagram of points balanced quad-tree partition

00	01	10	11
03	02	13	12
3		20	21
		230	231
		233	232
		220	221
		223	222

图 3 编码示例

Fig.3 Coding examples

的就是保证划分后的子块能尽可能地保证各并行节点间的负载均衡.一个完整的插值过程不仅包含主要的插值计算,还包括数据的读取和输出.因此,本文考虑的时间均衡不仅仅指插值计算的时间,还包括数据的读取和输出时间.邻域搜索是插值计算中相当耗时的一个环节.相对于邻域的搜索,插值结果的计算就比较简单,只需将最邻近点的值代入预先选定的插值公式,即可求解出未知点的高程.

设划分的级数为  $i$ ,划分后形成的面积最大的子块对应的划分层次为  $i_0$ ,该子块对应的点云密度下搜索出最邻近点的时间为  $t_0$ ,读取单位面积空数据的时间为  $t_1$ ,读取单个点云数据点的时间为  $t_2$ ,插值结果的计算时间为  $t_3$ (不包括寻找最邻近点的时间),输出单个数据的时间为  $t_4$ .

计算机处理点云数据插值生成 DEM 的过程主要分为:数据的读取、最邻近点的寻找、插值的计算、结果的输出.

### 2.4.1 数据的读取时间

划分形成的子块内点云的个数为  $k$ ,则读取数据点的时间为  $kt_2$ ,读取空数据的时间与子块的面积相关,经第  $i$  层划分后形成的子块面积为  $mn4^{-i}$ ,读取空数据的时间为  $t_1mn4^{-i}$ .则数据读取的总时间  $T_1$  为:

$$T_1 = kt_2 + t_1mn4^{-i}.$$

### 2.4.2 最邻近点的搜索时间

各子块需要执行插值计算的次数(即待插值点的个数)为  $\frac{mn4^{-i}}{m_0n_0}$ .经第  $i$  次划分后形成的子块与经第  $i_0$  次划分后形成的子块点云密度比为  $4^{i-i_0}$ .最邻近点的搜索时间与点云的密度相关,点云密度越大搜索耗时越短,单个点搜索出最邻近点的时间与它所在条带的密度成反比.设搜索时间与点云密度的相关系数为  $\alpha$ ,则单个点每次的搜索时间  $t_{si}$  为:

$$t_{si} = t_0 \frac{\alpha}{4^{i-i_0}}.$$

最邻近点搜索的总时间  $T_2$  为:

$$T_2 = \frac{mn4^{-i}}{m_0n_0} \cdot t_{si} = \frac{mn4^{-i}}{m_0n_0} \cdot t_0 \frac{\alpha}{4^{i-i_0}} = \frac{mnt_0\alpha 4^{i_0-2i}}{m_0n_0}.$$

2.4.3 计算插值结果

在选定插值方法的情况下,计算插值结果的过程相当于一个加权求和的过程,在已知最邻近点的情况下每次插值计算结果的计算量是相等的. 计算插值结果的总时间  $T_3$  为:

$$T_3 = \frac{mnt_3 4^{-i}}{m_0 n_0}.$$

2.4.4 结果的输出

在硬件条件相同的情况下,插值结果的输出只与需要输出的数据个数相关. 结果输出的总时间  $T_4$  为:

$$T_4 = \frac{mnt_4 4^{-i}}{m_0 n_0}.$$

综上所述,经第  $i$  层划分后形成的单一子块执行插值全过程的总时间  $T_i$  为:

$$T_i = T_1 + T_2 + T_3 + T_4 = kt_2 + t_1 mn 4^{-i} + \frac{mnt_0 \alpha 4^{i_0 - 2i}}{m_0 n_0} + \frac{mnt_3 4^{-i}}{m_0 n_0} + \frac{mnt_4 4^{-i}}{m_0 n_0}.$$

2.5 最优划分层次

第  $i+1$  层划分是将第  $i$  层划分后形成的一个子块又划分为面积相等的 4 个新子块,等面积块的时间差  $\Delta_i$  可表示为:

$$\Delta_i = T_i - 4T_{i+1} = \frac{3mnt_0 \alpha 4^{i_0 - 2i - 1}}{m_0 n_0} - 3kt_2.$$

其中,  $\Delta_i$  是关于自变量  $i$  的单调减函数. 若  $\Delta_i$  小于某给定阈值  $\varepsilon$  ( $\varepsilon$  为接近于 0 的某一个数),则可认为第  $i+1$  次及以后的划分效果不显著,应停止进一步划分,此时的  $i$  记为  $i^*$ ,是最优划分层次,该划分层次下形成的数据块称为标准子块,每一标准子块的处理时间为  $T_{i^*}$ .

2.6 基于时间均衡的再划分

点云数据经过  $i$  层划分后形成了若干面积不等的矩形块  $q$ ,每个矩形块对应的计算时间可通过函数  $T_i$  求得. 设时间系数  $\lambda$  是  $T_i$  与  $T_{i^*}$  的比值( $\lambda$  取整数),将矩形块  $q$  按面积等分为  $\lambda$  块,此时每块的处理时间接近于  $T_{i^*}$ . 按此方法将各层形成的子块按时间系数  $\lambda$  再次划分,即将整幅点云数据分割成若干面积不等但计算时间接近于  $T_{i^*}$  的矩形块.

2.7 编码调整

矩形块  $q_1 q_2 \cdots q_n$  经过第  $n+1$  次再划分之后形成了  $n+1$  层子块,编码也需要增加一位,变为  $q_1 q_2 \cdots q_n q_{n+1}$ ,第  $n+1$  位编码规则不变,即编码大小以西北角为起点顺时针递增,  $q_{n+1} \in \{0, 1, 2, \cdots, \lambda\}$ .

3 结语

内插算法是 DEM 数据初始化的重要手段,其计算效率也是影响数字地形分析高效计算的重要因素之一. 本文提出了一套针对离散点云并行插值生成格网 DEM 的数据划分方法. 该方法基于四叉树划分方法对离散点云进行初始划分和编码管理,通过对读取、搜索邻域、插值计算、输出 4 个过程的处理时间进行量化分析,最终划分成若干处理时间相等的子块,可有效地实现数据的均衡划分.

[参考文献](References)

[1] Kang-tsung Chang. 地理信息系统导论[M]. 北京:科学出版社,2003:245.  
Kang-tsung Chang. Introduction of Geographic Information System[M]. Beijing: Science Press, 2003:245. (in Chinese)

[2] 刘南,刘仁义. 地理信息系统[M]. 北京:高等教育出版社,2002.  
Liu Nan, Liu Renyi. Geographic Information System[M]. Beijing: Higher Education Press, 2002. (in Chinese)

[3] Ann Chervenak. The data arid: Towards an architecture for the distributed management and analysis of large scientific datasets [J]. Journal of Network and Computer Application, 2000, 23: 187-200.

[4] 王永杰,孟令奎,赵春宇. 基于 Hilbert 空间排列码的海量空间数据划分算法研究[J]. 武汉大学学报:信息科学版, 2007, 37(7): 650-653.

- Wang Yongjie, Meng Lingkui, Zhao Chunyu. Spatial partitioning of massive data based on Hilbert spatial ordering code[J]. Geomatics and Information Science of Wuhan University, 2007, 37(7): 650–653. (in Chinese)
- [5] 周艳, 朱庆, 张叶延. 基于 Hilbert 曲线层次分解的空间数据划分方法[J]. 地理与地理信息科学, 2007, 23(4): 13–17.  
Zhou Yan, Zhu Qing, Zhang Yeyan. A spatial data partitioning algorithm based on spatial hierarchical decomposition method of Hilbert space-filling curve[J]. Geography and Geo-information Science, 2007, 23(4): 13–17. (in Chinese)
- [6] Wang Shaowen, Marc P Armstrong. A quadtree approach to domain decomposition for spatial interpolation in Grid computing environments[J]. Parallel Computing, 2003, 29: 1481–1504.
- [7] 吕建峰, 刘定生, 焦伟利, 等. DEM 生成算法并行化研究[J]. 中国图像图形学报, 2002, 7(5): 506–516.  
Lü Jianfeng, Liu Dingsheng, Jiao Weili, et al. Parallel algorithms research on the interpolation of DEM[J]. Journal of Image and Graphics, 2002, 7(5): 506–516. (in Chinese)
- [8] Huang Fang, Liu Dingsheng, Tan Xicheng, et al. Explorations of the implementation of a parallel IDW interpolation algorithm in a Linux cluster based parallel GIS[J]. Computers and Geosciences, 2011, 37: 126–131.
- [9] Lluís Pesquer, Ana Cortes, Xavier Pons. Parallel ordinary kriging interpolation incorporating automatic variogram fitting[J]. Computers and Geosciences, 2011, 37: 464–473.
- [10] Armstrong M P, Marciano R. Parallel spatial interpolation[C]//Proceedings of the 11th International Symposium on Computer-Assisted Cartography(auto-Carto 11). Bethesda, MD: American Congress on Surveying and Mapping, 1993: 414–423.
- [11] Guan Xuefeng, Wu Huayi. Leveraging the power of multi-core platforms for large-scale geospatial data processing: Exemplified by generating DEM from massive LiDAR point clouds[J]. Computers and Geosciences, 2010, 36: 464–473.
- [12] 胡金星, 马照亭, 吴焕萍, 等. 基于格网划分的海量数据 Delaunay 三角剖分[J]. 测绘学报, 2004, 33(2): 163–167.  
Hu Jinxing, Ma Zhaoting, Wu Huanping, et al. Massive data delaunay triangulation based on grid partition method[J]. Acta Geodaetica et Cartographica Sinica, 2004, 33(2): 163–167. (in Chinese)

[责任编辑: 严海琳]

(上接第 54 页)

- [8] 牛泽平. 地图符号的设计——图形[J]. 大众标准化, 2010(S2): 76–77.  
Niu Zeping. Design of map symbol—graphics[J]. Popular Standardization, 2010(S2): 76–77. (in Chinese)
- [9] 齐琳, 沈婕, 张宏, 等. 面向警务 GIS 的地图优化表达方法研究——以南京市警务 GIS 为例[J]. 南京师大学报: 自然科学版, 2011, 34(1): 114–118.  
Qi Lin, Shen Jie, Zhang Hong, et al. Optimized map representing methods oriented police affairs GIS—Take police affairs GIS of Nanjing for example[J]. Journal of Nanjing Normal University: Natural Science Edition, 2011, 34(1): 114–118. (in Chinese)
- [10] 何丽华, 徐之俊. 地图注记设计若干问题的探讨[J]. 地理空间信息, 2011(6): 153–154.  
He Lihua, Xu Zhijun. Research on several issues of map annotation design[J]. Geospatial Information, 2011(6): 153–154. (in Chinese)
- [11] 王家耀, 孙群, 王光霞, 等. 地图学原理与方法[M]. 北京: 科学出版社, 2006.  
Wang Jiayao, Sun Qun, Wang Guangxia. Principles and Methods of Cartography[M]. Beijing: Science Press, 2006. (in Chinese)
- [12] 何宗宜, 张会霞. 《山西省历史地图集》的设计与计算机制图[J]. 测绘通报, 2003(12): 42–44.  
He Zongyi, Zhang Huixia. The computer map designing and compiling for the historical atlas of Shanxi Province[J]. Bulletin of Surveying and Mapping, 2003(12): 42–44. (in Chinese)
- [13] Bol P K. GIS, prosopography and history[J]. Annals of GIS, 2012, 18(1): 3–15.

[责任编辑: 严海琳]