

文章编号: 1673-6338(2007)03-0207-03

# 基于 2 维行程的栅格数据快速动态压缩算法

吴正升<sup>1</sup>, 宋玮<sup>2</sup>, 王秀莲<sup>3</sup>

(1 信息工程大学 测绘学院, 河南 郑州 450052 2 华北水利水电学院, 河南 郑州 450011

3 78155 部队, 四川 成都 610000)

**摘要:** 分析了常规 2 维行程压缩算法存在的不足, 在此基础上提出了一种在遍历栅格数据过程中直接快速动态生成 2 维行程的栅格数据压缩算法。该算法以  $M_D$  码代替行列号来扫描栅格数据, 对于一个  $2 \times 2$  的窗口由  $M_D$  码反解行列号只需转换左上角格网单元, 同时以动态线性表代替静态线性表。在提取栅格单元的过程中, 直接检测对比格网单元的属性值, 动态分配内存来建立 2 维行程编码。最后, 大量实验表明, 算法在运行效率和内存占有量方面均优于常规的 2 维行程压缩算法。

**关键词:** 栅格数据; 线性四叉树; 2 维行程

**中图分类号:** P208 **文献标识码:** A

## Fast Dynamic Compressing of Raster Data Using 2D Run-Length

WU Zheng-sheng, SONG Wei, WANG Xiu-lian

(1. Institute of Surveying and Mapping, Information Engineering University, Zhengzhou 450052, China;

2. North China Institute of Water Conservancy and Hydroelectric Power, Zhengzhou 450011, China;

3. 78155 Troops, Chengdu 610000, China)

**Abstract:** Due to some disadvantages of the traditional 2D run-length compressing, a fast dynamic compressing of raster data using 2D run-length is put forward in this paper. The method can construct 2D run-length directly through only once tracing of the whole raster data. Firstly, it scans raster data by the decimal Morton code instead of the row and once uses a  $2 \times 2$  window to extract 4 sequential pixels from the original raster data. Secondly, it uses a dynamic linear table instead of a static table. Transforming the attribute value of the pixels, it directly compares among their attribute values, and dynamically makes 2D run-length. Finally, the some experiments show that this algorithm is superior to the traditional 2D run-length compressing in the running speed and memory requirement.

**Key words:** raster data; linear quadtree; two dimensional run-length

栅格数据结构是将地表面划分为均匀的紧密相邻的格网阵列, 每个格网单元代表了地表上的一个区域, 其地理位置由行列号确定, 格网单元的属性值为所表达的空间对象的属性值或代码, 其实质是一个能描述空间对象及其属性的 2 维矩阵<sup>[1]</sup>。与矢量数据结构相比较, 栅格数据结构表达地理要素比较直观, 容易实现多元数据的叠加操作。因此, 栅格数据是地理信息系统中一种常用的数据类型, 例如扫描数字化数据、矢量栅格转换数据、遥感数据以及数字高程模型等, 都属于这种数据类型。但是, 栅格数据的缺点也是显著的, 例如数据的精度取决于格网的边长, 当格网边长缩小时, 格网单元的数量将呈几何级数递增, 造成存储

空间的迅速增加; 同时, 由于相邻格网单元属性值的相关性, 造成栅格数据的冗余, 特别当表示不规则多边形时, 数据冗余度更大<sup>[2]</sup>。因此, 为了高效存储和快速传输海量栅格数据, 有必要研究快速有效的栅格数据压缩算法。

### 1 常规 2 维行程压缩算法分析

2 维行程压缩法是在对传统 1 维行程压缩法的 2 维扩展基础上, 融合了线性四叉树 (linear quadtree, LQT) 及其 Morton 码技术。它通过对已经形成的基于十进制 Morton 码 (简称  $M_D$  码) 的栅格图像的 LQT 记录进行扫描, 记录其入口的  $M_D$  地址和属性值; 然后依次扫描后续地址的属性值。

收稿日期: 2006-12-03 修回日期: 2007-03-27

基金项目: 河南省重大重点科技攻关项目 (0423034200)

作者简介: 吴正升 (1978-), 男, 四川广元人, 讲师, 硕士, 主要从事空间数据库、地理信息系统的教学与研究。

若后一地址的属性值发生变化(即不等于前一地址的属性值)则记录后一地址和属性值;若后一位置的属性值不发生变化,则不作记录,直到最后,并记录最后位置的地址和属性值即可<sup>[3]</sup>。

这种 2 维行程编码虽利用了线性四叉树的地址码,但已经没有结构规则的四叉树,甚至已失去了四叉树的概念。然而它比规则的四叉树具有更高的压缩效率,而且对于以后的插入、删除和修改等操作,因不必保持完整的四叉树结构而变得相当简单<sup>[4]</sup>。而且,线性四叉树和 2 维行程编码采用了相同的地址码,它们之间的相互转换非常容易和快速。因此,2 维行程编码在栅格数据压缩方面得到了广泛应用,Mark Lauzkn 和 Cebrioh<sup>[4]</sup> 等人就主张用 2 维行程来表达面域栅格数据。

但是常规 2 维行程编码压缩算法对如何形成  $M_b$  码过程中存在两个缺点:

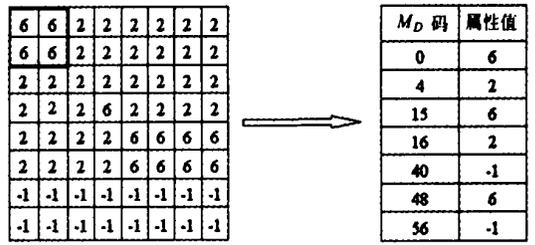
- 1) 对栅格数据按行列号顺序进行扫描,得把每个行列号都要转换成  $M_b$  码,需要运行时间;
- 2) 读取相应栅格数据的每个格网单元属性值赋值给以  $M_b$  码排序的 1 维数组,需要占用大量内存。特别是当栅格数据的格网单元数达数百万或更大时,在所需内存和运行时间上难以满足要求。

为了克服常规 2 维行程编码压缩算法存在的上述缺点,在文献 [1] 提出的线性四叉树快速动态编码基础上,提出一种只需对栅格数据遍历一次就能直接生成 2 维行程的快速动态压缩算法。

## 2 快速动态的 2 维行程压缩算法

### 2.1 算法的主要思想

算法的主要思想是以  $M_b$  码代替按行列号扫描栅格数据,以动态线性表代替静态线性表来存储数据。在以  $M_b$  码由小到大顺序提取栅格单元属性值的过程中,对于一个  $2 \times 2$  的窗口,由  $M_b$  码反解行列号时,只需转换左上角格网单元,减少了运行时间;同时利用动态线性表动态分配内存的特点,直接检测对比格网单元的属性值,动态压缩来建立 2 维行程编码,不需要专门开辟内存区来存放压缩结果,节省了大量内存。当完成对栅格数据的遍历后,动态线性表内记录就是所要求的 2 维行程编码压缩结果,最后把结果写成文件就完成了整个算法。如图 1 所示,图 1(a) 为原始栅格数据(6×8 矩阵,后两行为补充的位置,属性值为 -1);图 1(b) 给出了快速动态的 2 维行程压缩结果。



(a) 原始栅格数据 (b) 压缩结果

图 1 快速动态的 2 维行程压缩算法

### 2.2 算法的主要步骤

算法主要包括以下两个过程:快速提取格网单元属性值和比较属性值进行动态压缩和存盘。

#### 2.2.1 快速提取格网单元属性值

按  $M_b$  码的顺序循环,每次  $M_b$  码加 4 结束条件是  $M_b$  码大于或等于  $2^n \times 2^n$ 。依次从栅格数据中以  $2 \times 2$  的窗口(如图 1(a) 中栅格数据上的粗线框所示)提取 4 个格网单元的属性值。具体的做法是:由第 1 个  $M_b$  码反求栅格数据的行列号,该行列号刚好是 4 个单元中左上角单元对应的行列号,其他几个单元的行列号就很容易知道,再根据行列号就可以从栅格数据中提取相应单元的属性值。

$M_b$  码反求行列号方法:由于  $M_b$  码是由行号和列号的 2 进制数两两交叉组合的结果,因此可以由  $M_b$  码的二进制数采用位运算规则分解出行列号,其中奇位和偶位组合的十进制数分别对应行号 和列号  $J$ 。<sup>[1]</sup> 采用位操作运算来反解行列号比基于数学公式计算方法效率高很多。

#### 2.2.2 比较属性值进行动态压缩和存盘

把第 1 步得到的 4 个格网单元属性值依次与动态线性表(假设 PLisb)尾部数据结构作对比。若属性值发生变化,则记录现在的  $M_b$  码及其属性值,并把结构插入 PLis 的尾部;若属性值不发生变化,则不作记录,循环下一个单元。当循环条件结束时,就遍历完了所有栅格数据,动态线性表 PLis 中的数据就是压缩结果,把它写成文件,整个快速动态的 2 维行程压缩算法结束。动态线性表的存储  $s\_tree$  结构如下:

```

struct s_tree{
    int m_Mb; // Mb 码
    int m_Spr; // 属性值
}

```

### 2.3 算法实现

根据上述快速动态 2 维行程压缩算法的思想,用 VC++ 6.0 编程对该方法进行了具体实现,并在 P4 3.0 内存 1.0 GB 微机上进行调试和运行。算法采用的流程如图 2 所示。

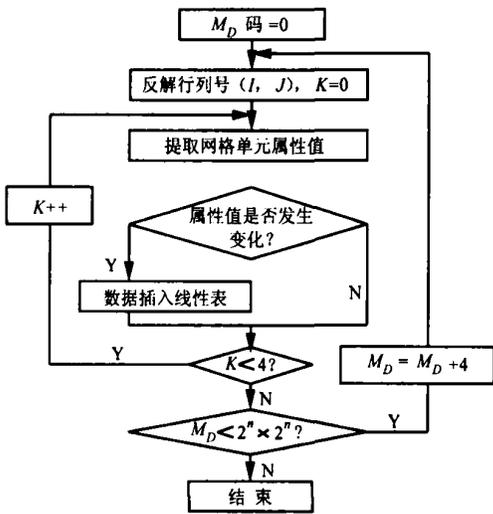


图 2 快速动态 2 维行程压缩算法流程图

### 3 实验与分析

为了验证算法的有效性, 特意选取不同大小和复杂程度的栅格数据 (以 BMP 图像实验), 先用算法进行压缩编码, 再反解码并在实验系统下还原显示。图 3 给出了一幅 BMP 图像采用算法压缩后反解码还原图, 把它和原始 BMP 图像进行对比, 两者属性值的分布完全一致, 这就验证了快速动态 2 维行程压缩算法的正确性。

为了比较快速动态法与常规 2 维行程压缩算法在运行速度及内存消耗上的差异, 分别以这两种方法对实验数据进行测试, 其测试结果如表 1 所示。

表 1 两种方法的运行时间和内存需求对比

压缩方法	128×128		512×512		1 024×1 024	
	时间 /ms	内存 /KB	时间 /ms	内存 /KB	时间 /ms	内存 /KB
快速动态	5	33.7	77	423.2	314	1 533.6
常规方法	19	80	262	1 280	1 064	5 120

### 4 结语

通过比较分析可以看出, 提出的快速动态 2 维行程压缩算法在运行效率和所需存储空间上均优于常规的 2 维行程压缩算法。加之 2 维行程编码比线性二叉树编码具有更高的压缩效率, 而且 2 维行程编码与线性二叉树编码进行转换也十分方便。因此, 快速动态 2 维行程压缩算法为解决栅格数据的快速压缩存储提供了一种可行的方法。

从表 1 可以看出: 快速动态方法相对于常规方法而言, 在运行效率上有很大的提高。

1) 运行时间少。整个编码过程只对栅格数据遍历一次;  $M_D$  码与行列号之间的转换次数少, 快速编码仅需对  $2 \times 2$  窗口中左上角的格网单元进行计算, 而不是对每个格网单元都进行  $M_D$  码到行列号的转换。

2) 内存需求小。快速动态方法利用动态线性表动态地分配内存, 在对栅格数据遍历的过程中, 不停地对属性值相同的相邻结点记录进行合并, 无需存储每个格网单元的  $M_D$  码及其属性值的对应关系表。而常规算法中, 由于原始栅格数据要调整成方阵, 就得事先开辟一个大小为  $2^n \times 2^n$  的 1 维数组来存储这种对应关系。同时, 快速动态方法也不需要专门开辟内存区来存放 2 维行程编码压缩结果。

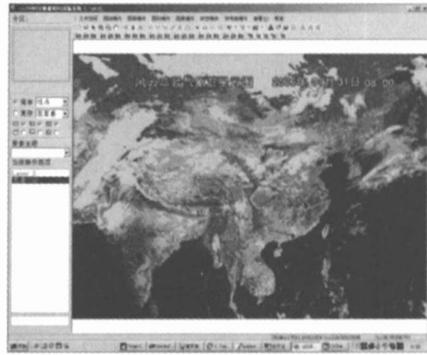


图 3 栅格数据压缩后反解码还原图

### 参考文献:

- [1] 盛业华, 唐宏, 杜培军. 线性二叉树快速动态编码及其实现 [J]. 武汉测绘科技大学学报, 2000, 25(4): 324-328.
- [2] 黄杏元, 马劲松, 汤勤. 地理信息系统概论 [M]. 北京: 高等教育出版社, 2001: 43-52.
- [3] 吴立新, 史文中. 地理信息系统原理与算法 [M]. 北京: 科学出版社, 2003: 174-183.
- [4] 龚健雅. 地理信息系统基础 [M]. 北京: 科学出版社, 2001: 102-115.
- [5] 杨敏, 汪云甲. 基于二叉树的栅格数据快速编码及其实现 [J]. 测绘工程, 2001, 10(4): 16-19.

责任编辑 安敏