一种由等高线模型生成规则格网模型的算法

陈国良 曹卫群 黄心渊
(北京林业大学信息学院, 北京 100083)

摘 要 针对地形可视化的要求,在分析由等高线模型生成规则格网模型的几种常用方法的基础上基于逐点内插和分块内插,提出了一种新的模型转换算法。该算法已经在 Microsoft Visual C++.Net 上实现,插值结果成功应用于大规模地形可视化平台。

关键词 DEM 内插 地形可视化
中图法分类号: TP391.9 **文献标识码**: A **文章编号**: 1006-8961(2007)06-1110-04

The Method of Generating the Regular Grid Model from the Contour Line Model

CHEN Guo-liang, CAO Wei-qun, HUANG Xin-yuan
(*College of Information, Beijing Forestry University, Beijing* 100083)

Abstract On the basis of analyzing existing commonly used interpolation methods for generating the regular grid model from the corresponding contour line model, this paper presents a new interpolation method based on block and point-by-point interpolation technology. This algorithm has already been accomplished with Microsoft Visual C++.net programming. And the result of interpolation has been successfully applied to the Terrain Visualization Platform.

Keywords digital elevation model, interpolation, terrain visualization

1 前 言

可视化技术能将数据转换成图形,给予人们深刻与意想不到的视觉效果,并给"数字林业"领域带来新的研究方式。随着计算机软硬件的加速发展和人们对模型精确度要求的不断提高,3 维地形数据变得越来越大,虽然由等高线数据生成规则格网数据的内插方法已经比较成熟,但在进行大规模地形可视化表现时,因为数据量较大,并且受到内存大小的制约,故会造成插值效率降低。本文以等高线模型和规则格网模型提供的地形数据为对象,对已有的插值方法做了相应的改进,改进的方法提高了插值效率。

2 常用的插值方法

按插值点的分布范围,可以将内插分为整体内插、分块内插和逐点内插 3 类[1]。

整体内插 它的插值函数由区域内所有采样点的属性值来确定,进而可利用这个插值函数进行全区域的特征拟合。常用的整体插值算法有多项式趋势面拟合、傅里叶分析、小波变换等,其优点是易于理解。简单地形特征因为参考点比较少,故可采用低次多项式来描述。但当地貌复杂时,参考点的个数将增加,则需要采用高次多项式来描述[2]。由于实际的地形非常复杂,整个地形很难用一个统一的多项式来拟合,因此对 DEM 数据进行内插一般不采

基金项目:教育部科学技术研究重点项目(重点 03029)
收稿日期:2006-03-09;**改回日期**:2006-03-28

第一作者简介:陈国良(1979 ~),男。2000 年获北京林业大学学士学位,现为北京林业大学信息学院在读硕士研究生。主要研究方向为计算机图形学、可视化。E-mail: guolchen@yahoo.com.cn

用整体内插的方法,而是采用局部内插。

分块内插　该内插是先把参考地形分成若干块,然后对各块使用不同的函数进行内插,而分块的大小则由地形的实际应用决定。一般来说,相邻分块间应有适当宽度的重叠,以保证相邻分块间能平滑、连续地拼接。常用的分块内插方法有线性内插算法、双线性多项式内插算法、二元样条函数内插、多面函数法、最小二乘配置法等。其中,基于 TIN 和正方形格网剖分法的双线性内插[3]是 DEM 分析与应用中较为常用的方法。

逐点内插法[4]　该内插由于是以待插点为中心,定义一个局部函数去查找周围的控制点,控制点的范围随待插点位置的变化而移动,因此又称移动曲面法。其中,邻近点的选择一般需考虑范围和点数两个因素。与分块内插法相比,逐点内插法由于十分灵活,且计算方法简单,同时对计算机的内存要求较低,因而其应用更为广泛。

3　适应地形可视化的等高线分块逐点内插算法及实现

3.1　地形可视化对所输入地形模型的要求

对于较小的地形区域,地形模型的三角面片数量不多,计算机硬件本身就能够满足其可视化的要求;但是对于大规模地形,即使事先采用一些简化方法[5],由于其整个地形模型的三角面片规模也可达数千万至上亿,超出了计算机硬件所能处理的能力,因此需要将地形分块,以降低数据规模。然而由于等高线模型中所有等高线都是连续保存的,每条等高线上的控制点也连续保存在外存中,因而不能直接应用于分块插值。

基于地形模型的上述特点,论文提出了采用分块插值的等高线插值算法,即首先将每条等高线拆分至各个地形块,然后在块内进行插值。算法还对数据进行了组织[6,7],即将地形模型分块逐点内插的结果以块为单位连续保存在外存中,这样可提高内外存数据交换的速度。

算法已经成功地应用在一个地形可视化平台上。

3.2　分块逐点插值

论文采用分块加权逐点内插的方法,即先在待插点周围的圆形区域(控制区域)内寻找控制点(已获得高程值的点),然后通过进行加权平均来获得待插点的高程值。各个控制点取相同权值,插值点高程计算的公式为

$$Z = \frac{1}{N}\sum_{i=0}^{N-1} Z_i \tag{1}$$

其中,N 为所找到的控制点数目,Z_i 为第 i 个控制点处的高程值。

控制区域由以下函数定义,

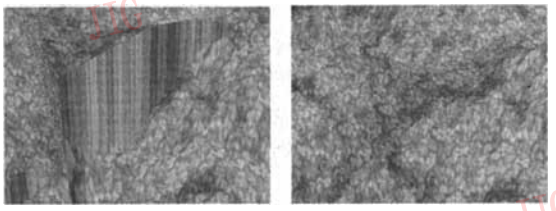
$$(x_1 - x_2)^2 + (y_1 - y_2)^2 \leqslant R^2 \tag{2}$$

其中,x_1, y_1 为控制点的 x, y 坐标值,x_2, y_2 为当前待插点的 x, y 坐标值,R 为圆形控制区域的半径。

与一般逐点内插方法不同,分块逐点内插会面临一些问题。

首先是块与块之间的拼接问题。相邻块的相邻边界处的点,分别在所属的块内计算高程值,割裂了相互之间的连贯性,因而会产生高度跳变(如图 1(a)所示)。

为解决这个问题,插值时应在块与块之间设置一定宽度的重叠,也就是对各块进行适当地扩充。在插值过程中先对扩充后的块进行插值,然后取当前插值块在扩充块中对应的区域作为插值结果(如图 2 所示,黑色区域为扩充块的范围),插值效果如图 1(b)所示。



(a) 直接插值可视化结果　　(b) 有重叠区域插值可视化结果

图 1　块边缘插值效果

Fig. 1　Interpolation at the edge of blocks

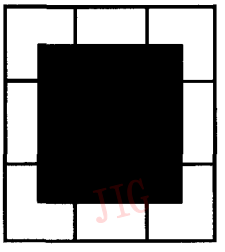


图 2　插值重叠区域

Fig. 2　Overlapping area for interpolation

由于边界块不存在 8 个相邻块,不能形成如图 2 所示的完整的扩充区域,其插值结果会存在一定的误差,因此会对后续块的插值产生影响。为了降低边界

块对插值结果的影响,算法对地形模型的插值过程应从地形模型的中心块开始,向四周扩散插值(图 3 所示为一分为 25 块的地形的插值顺序)。

21	13	5	15	22
14	6	1	7	16
8	2	0	4	12
17	9	3	11	20
23	18	10	19	24

图 3 块的插值顺序

Fig. 3 Interpolation order of subblocks

另外,如果等高线过密,插值块中某点可能对应等高线上多个高程值。针对这种情况,可采用求均值的方法,即如果插值块中某点对应 n 条等高线,则将与这 n 条等高线对应的高程值相加后再进行平均,即为该点的高程值。

算法过程如下:

(1) 定义块大小,将地形分块。

由本插值算法所获得的规则格网模型,在地形可视化平台中还需进一步进行简化,而该简化方法要求地形块大小为 2 的整数次幂,因此插值算法将地形块大小定义为 $2^M \times 2^M (M \in [1,2,3])$。当地形的点阵大小不满足 $2^N \times 2^N (N \geq M)$ 大小,则先将非完整地形块(点阵大小不满足 $2^M \times 2^M$)补齐,然后将等高线分解到各个地形块内。

(2) 将地形中心块插入队列,形成初始块队列。

(3) 取出位于块队列中心的块作为当前插值块,并将其周围 4 个邻块(上、下、左、右)插入队列,如果周围块中某块已经进行插值,则该块不插入队列,可对该当前插值块进行以下处理:

① 将当前插值块内,等高线上已知采样点的高程值写入块内点阵中相应位置;按直线 DDA 算法计算等高线上非采样点的位置,并将该等高线的高程值写入块内点阵中的相应位置。

② 对当前块进行扩充,获得当前块的扩充块。其中,扩充带点阵宽度定义为 Addition_Size,其依据当前插值速度、精度要求和块大小而设定。如果扩充区域的所属块已完成插值,则找到其所属块中该扩充区域的值,并写入当前扩充块的点阵;否则对扩充区域所属的周围块也进行①操作,即可获得该扩充区域的高程值。

③ 用逐点内插法对扩充块进行递归插值。对扩充块内规则格网点阵的每个点先查找其控制点,再求得该点的高程值。对于等高线比较稀疏的区域,由于会出现与待插点对应的控制区域内不存在控制点的情况,因此算法引入了待插点链表。

插值时,首先对块内点阵的各点依次进行处理,若该点有控制点,则直接进行插值计算,以获得该点的高程值,若该点不存在控制点,则将该点插入待插点链表;然后对待插点链表中的点进行递归插值,并从链表中删除已完成插值的点,直到链表里面为空,块插值完成。

④ 将插值结果以块为单位连续保存在外存中。

⑤ 判断该块的周围的 8 个块是否已经完成插值。若已完成,则该块数据在后续插值中,不会再被使用,可从内存中释放该块;否则,保留该块。

(4) 递归执行步骤(3),直到块队列为空。

4 实验结果

算法在 Pentium 1.7GHz,内存为 256M 的机器上执行。

本文对图 4 所包含的 302 条等高线、采样点总数为65 165个、数据规模为7 573 × 5 445的地形模型进行了插值计算,并分别对 3 种分块方式下算法的执行情况进行了测试。其中,块大小分别为 128 × 128、256 × 256 和 512 × 512,块数由等高线数据的长度、宽度和块大小决定。测试的结果如表 1 中(a)栏所示。

将原始数据分割成若干个 2 × 2 的点阵,并将每个 2 × 2 的点阵内的数据取平均值作为新的地形数

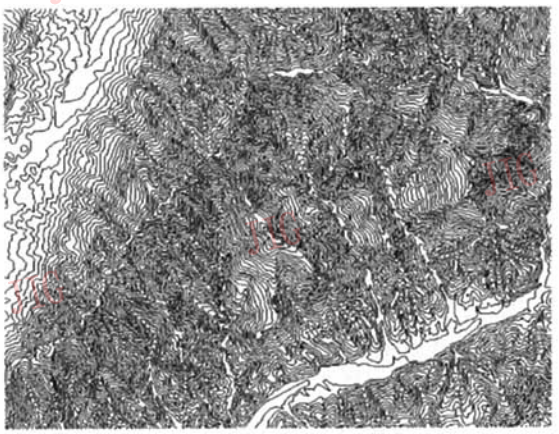


图 4 等高线地形图

Fig. 4 The contour map

表 1　算法测试结果

Tab. 1　Testing results on DEM data with the presented algorithm

块大小	(a) 直接分块逐点内插						(b) 2×2 的点阵采样分块逐点内插					
	块数①	生成点阵	64② 时间(s)	64② 内存峰值(M)	32② 时间(s)	32② 内存峰值(M)	块数	生成点阵	64 时间(s)	64 内存峰值(M)	32 时间(s)	32 内存峰值(M)
128×128	60×44	7 680×5 632	367.098	15	213.988	12	30×22	3 840×2 816	71.673	15	43.713	15
256×256	30×22	7 680×5 632	206.347	21	164.046	21	15×11	3 840×2 816	61.961	16	51.30	19
512×512	15×11	7 680×5 632	201.440	52	185.336	49	8×6	4 096×3 072	59.035	40	53.597	40

注:①块数为地形数据被分割成的子块的数量,由等高线数据的大小和块大小决定。②是块的重叠区域,64 是指将周围 8 个地形块中与当前块相邻的宽度为 64 的区域扩充入当前块,32 是指将周围 8 个地形块中与当前块相邻的宽度为 32 的区域扩充入当前块。

据点阵的一个格网点的值,这样数据规模就缩小为原等高线模型规模的 1/4。表 1 中(b)栏为对该组新的较小地形数据进行测试的结果。

从表 1 可知,利用本文所提出的算法进行等高线地形模型的插值,数据规模大小对插值时间有较大影响,但是对内存空间的占用却影响很小。无论数据规模大小如何,当定义块大小为 128×128 时,内存峰值最小,即所耗内存最少,当定义块大小为 512×512 和 256×256 时,其插值时间相当。结合算法执行速度和所耗内存大小两方面的考虑,当块大小定义为 256×256 时,插值效率较高。设定重叠区域宽度为 32 时比设定重叠区域宽度为 64 时进行的插值效率要高,但是具体采用多大宽度的重叠区域进行插值结果最优跟等高线数据的具体情况有关。

5　结　论

由等高线生成规则格网模型的内插方法已经比较成熟,一些软件也提供了这种功能,但是有些方法过于复杂,且不实用。对于 TIN 模型比较常用的是双线性插值,因该算法较为简单实用。此外,在大规模地形表现上,加权平均方法是等高线模型转换成规则格网模型普遍采用的方法。本文结合分块内插与逐点内插方法,先对地形模型分块,然后对当前插值块进行扩充,使其周围 8 块的部分区域参与插值,然后采用逐点内插的加权平均方法对扩充块进行插值,其结果不仅消除了块与块之间不平滑的过渡,能基本满足地形实时可视化的要求,而且视觉效果良好。

参考文献(References)

1　Li Zhi-lin, Zhu Qing. Digital Elevation Model[M]. Wuhan: Wuhan University Press, 2002. [李志林,朱庆. 数字高程模型[M]. 武汉:武汉大学出版社,2002.]

2　Lu Shou-yi, Tang Xiao-ming, Wang Guo-sheng. Practical Course for the Geography Information System [M]. Beijing: China Forestry Publishing House, 1998. [陆守一,唐小明,王国胜. 地理信息系统实用教程[M]. 北京:中国林业出版社,1998.]

3　Wang Jian-yu, Teng Shu-qin. The method of making DEM based on contour line[J]. Computer Application, 2002, **22**(8): 30~32. [王建宇,滕树钦. 一种基于等高线生成 DEM 的方法[J]. 计算机应用, 2002, **22**(8): 30~32.]

4　Qui Wei-ning. Generating DEM based on contour line[J]. Journal of Wuhan Technical University of Surveying and Mapping, 1994, **19**(3): 199~203. [邱卫宁. 根据等高线建立数字高程模型[J]. 武汉测绘科技大学学报, 1994, **19**(3): 199~203.]

5　Tomas Akeninemoeller, Eric Haines. Realtime Computer Graphics [M]. Pu Jian-tao Translate, Beijing: Press of Peking University, 2004. [实时计算机图形学[M]. 普建涛译. 北京: 北京大学出版社, 2004.]

6　Platings M, Day A M. Compression of large-scale terrain data for real-time visualization using a tiled quad tree[J]. Computer Graphics Forum, 2004, **23**(4): 741~759.

7　Peter Lindstrom. Out-of-core simplification of large polygonal models [A]. In: Proceedings of SIGGRAPH[C], New Orleans, Louisiana, USA, 2000: 259~262.