

一种基于不完全四叉树的 LOD 生成算法

芮小平

(北京交通大学交通运输学院, 北京 100044)

摘要 为了实时地绘制大规模地形数据,提出了一种改进的实时连续 LOD 生成算法。该算法首先采用 Mortan 码的编码方式对地形数据进行简化,并利用不完全四叉树存储简化后的高程数据;然后根据视点位置和网格空间对象误差的关系建立基于不完全四叉树的 LOD 模型,同时采用逐层找邻法调整不同层次之间的裂缝,并给出了寻找不同类型邻居的实现过程;最后采用背面剔除算法将起伏地形的不可见部分去除。实际编程时,由于采用了 Hilbert 填充曲线方式存储四叉树结点,并采用隔层四叉树方式访问结点数据,从而提高了大规模地形的绘制效率。使用该方法描述荆江地区的地形,取得了良好的绘制效果。

关键词 多层次细节模型 地形 Mortan 码 四叉树 背面剔除

中图分类号: P208 TP391.9 **文献标识码:** A **文章编号:** 1006-8961(2005)09-1152-09

A LOD Algorithm Based on Unfull Quadtree

RUI Xiao-ping

(School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044)

Abstract In order to render massive terrain data in real time, this paper presents an improved LOD algorithm. This algorithm firstly simplifies the DEM data by using Mortan coding principle and stores the simplified data with an unfull quadtree. Then, it builds the real-time continuous Lod based on this unfull quadtree according to the relationship between viewer position and grid object space error. The authors bring out a method named "find the neighbors layer by layer" to patch the cracks between different layers and the procedures to find the different type neighbors are presented in this paper. Finally, it wipes off the invisible grids through back-culling algorithm. In order to improve the rendering speed, it uses Hilbert filling curve method to store and interleaved quadtree method to access the unfull quadtree nodes. The authors use the improved method to simulate the terrain of Jingjiang area and obtain a good effect.

Keywords level of details(LOD), terrain, Mortan coding, quadtree, back-culling

1 引言

数字地球中的虚拟现实系统是以海量高程数据作为支撑的,因此,大规模地形数据的可视化是实现数字地球中虚拟现实系统的关键技术之一。由航空航天摄影测量获取的地形数据生成的 3 维地形模型往往由上百万或更多的三角形面片组成,由于这种海量数据使得大规模地形环境的实时绘制和无缝漫游成为地形可视化领域的一个瓶颈,所以如何有效

地简化地形数据已经成为地形可视化领域的热点。目前,在不影响地形浏览真实感的前提下,人们主要采用模型层次简化技术和多分辨率技术,降低地形数据的几何复杂度和纹理数据的大小,以提高地形数据的绘制速度^[1,2]。常用的网格简化方法是建立多层次细节模型(level of details, LOD),即根据视点的位置决定调用网格的层次,如果离视点近,则调用高精度的网格,反之,则调用低精度的网格。通常有两种方法构建 LOD 模型,其中一种是离散的 LOD 技术,即将 DEM(digital elevation model)数据做成多

收稿日期:2004-12-10;改回日期:2005-03-10

第一作者简介:芮小平(1975~),男,2004 年获中国科学院遥感应用研究所理学博士学位,现为北京交通大学交通运输学院教师。研究方向为网络 3 维地理信息系统。E-mail:ruixp@yahoo.com.cn

个精度不同的层次,漫游时则根据视点的位置调用不同层次的数据;另外一种实时连续的 LOD 生成技术,即一次将 DEM 数据读入计算机内存,然后在 3 维场景中漫游时,可根据视点位置实时地调整网格的层次。目前,人们对地形数据简化算法的研究成果很多,现将其中比较经典的算法介绍如下:

Lindstrom 于 1996 年提出了一种可以对用规则网格表示的地形模型进行实时的细节层次删减和绘制的方法^[3]。该算法的基本思想是:首先进行粗简化,用来对曲面网格中一些块的具体细节层次进行选择;然后进行细化,用来对块中的顶点进行移去操作,并重新网格化,以便进一步简化。该算法利用了一个可变的屏幕-空间阈值来限定投影图像的最大误差。由于采用了规则网格,从而使所有操作都可以动态实时地完成。

Duchaineau 等人在 1997 年提出了“实时优化自适应网格”(real-time optimally adapting meshes, ROAM)算法^[4]。ROAM 算法的基础是用三角形二叉树来动态表示网格,因此可通过定义三角形二叉树和分裂、合并操作来保证增加或删除顶点时网格的连续性,其中分裂与合并操作用两个优先级队列来控制。这种方法先使用一个可以根据视点改变而灵活变化的误差矩阵,然后通过对矩阵进行优化来产生一个误差约束,即可以直接获取指定的三角形数目,并且运用帧与帧之间的相关性等其他一些优化措施来处理地形实时显示问题。

Cignoni 等人提出了一种利用门限函数来控制地形不同区域分辨率的可变分辨率(variable resolution)模型,其中门限函数由实际的应用问题确定。这种方法不仅可以实时地以任意分辨率恢复地形数据,还可以根据需要,在整个地形区域采用同一分辨率或不同分辨率,同时可以保证地形表面的连续性^[5]。

淮永建等在四叉树结构基础上进一步提出了一种基于自适应四叉树结构视相关的动态多分辨率地形简化算法,并讨论了地形简化中多分辨率地形网格生成、节点评价函数、多分辨率 LOD 转化时的图像跳跃以及多分辨率地形纹理映射等关键技术问题,该方法有效地解决了多分辨率地形网格拼接中的裂缝问题和层次细节转化中的图像跳跃现象^[6]。

王源等针对规则网格数据的特点,提出以直角三角形为基本单元,利用二叉树构建地形的层次结构来描述大规模地形,这种方法兼顾了网格数据与

不规则三角网的优点,由于它是通过地形面片自身的复杂程度、地形面片与视点距离以及地形面片与视线间的方向关系 3 方面来确定该面片的绘制分辨率,从而较好地提高了 LOD 生成的效率^[7]。

谭兵等采用一种约束四叉树来实时构建地形多分辨率模型,该算法引入了视相关的概念,同时给出了一种与视点相依赖的对地形结点误差进行评价的方法,并改进了 LOD 模型“裂缝”效应消除方法^[8]。

此外,小波变换也被用于多分辨率模型,它可分为以下两类:一类如 Zhu Zhi-fan 等提出的与视点有关的算法^[9],该算法首先确定视点感兴趣的区域,然后将地形划分为可变大小的块,以确定信息所在的位置,接着用小波变换对每个块进行编码,再将这些块按其视觉信息量的大小分为不同的等级,并将其中所含信息量最大的小波系数置入一个位流中进行累进传输和存取,这个方法的主要研究目的是为了进行数据的压缩、存储和传输,但也可用于地形的多分辨率实时绘制;另一类算法则是 Gross 等人提出的由地形表面特征决定其局部细节层次的算法^[10],该算法首先通过小波变换分解原始数据集,然后分析变换所得的系数,即先定义一个小波空间滤波器,然后通过对局部谱的估计来控制局部表面细节层次的近似程度,这个局部的谱特性是由地形表面数据的小波表达式决定的,且地形任何区域的细节层次都可以根据这个谱特性自适应地实时优化。这样,在山脊、山谷等曲率高的区域用高细节层次表示,而在那些近似于平面的区域则只需用较少的三角形就完全可以满足要求。这种方法不仅对地形有效,而且对其他定义在规则栅格上的数据集均很有效。

笔者认为,现有的 LOD 技术都是读取地形数据后直接构建模型的,这并不能减少原始数据文件的大小,由于在进行实时的细节层次变换和绘制时,有些区域的高程是固定不变的,在建立 LOD 模型时反而会在不同层次细节过渡时产生一定的误差,因此这些区域的数据没有必要完全参与到 LOD 的实时生成过程中。此外,在 3 维场景漫游过程中,当地形有起伏时,由于面的遮挡关系,背着视线的有些地形是不可见的,所以在最后绘制地形时,可以不画,这样就可以减少地形网格绘制的数目。针对现有 LOD 技术的缺点,本文提出一种实时连续 LOD 技术的改进算法,该算法是在 Lindstrom 提出的基于四叉树的实时连续 LOD 生成技术基础上,进一步提出了

“先压缩、后构模”以及“背面剔除”的思想,即在对数据进行无损压缩的情况下生成 LOD 模型,并在绘制地形时,采用背面剔除的算法来去除不能看见的地形网格,获取更加有效的地形数据,以提高绘制效率。下面将详细介绍该方法的实现过程。

2 改进的实时连续 LOD 生成算法

2.1 先压缩后构模的实现方法

所谓先压缩、后构模就是在用四叉树存储 DEM 数据之前,首先对原始的 DEM 数据文件进行压缩,并用四叉树存储压缩后的 DEM 数据进行构模^[11]。本文按照 Mortan 码编码规则对原始 DEM 数据进行无损压缩。所谓 Mortan 码就是在 $2^n \times 2^n$ 的网格中,按照“Z”字形的顺序对每一个网格进行编码的方

法,其数值为整数,从 0 开始,到 $2^n - 1$ 结束。一个 $2^n \times 2^n$ 的 Mortan 码如图 1(a) 所示。由于网格中每个 Mortan 码的值是唯一的,因此 Mortan 码含有表示网格所在位置的平面坐标信息,即 Mortan 码实际上为每个网格位置建立了索引。根据 Mortan 码的性质,笔者采用构建四叉树的方法自下而上地对 DEM 数据进行简化,该简化方法可以描述为:对网格高程数据按照 Mortan 码的顺序依次对每 4^n (n 从 1 开始,每一次遍历结束后增加 1) 个高程数据的子网格进行判断,如果相同,则进行合并;如果不完全相同,则将网格的高程数据输出,并将这个子网格的第 1 个高程数据赋一个特殊值。这样不断深入,直到对整个网格中的高程值都进行判断,即 4^n 等于网格的个数为止。图 1(b) 是一个原始网格高程数据的分布情况,经过简化后高程数据如图 1(c) 所示。

0	1	4	5
2	3	6	7
8	9	12	13
10	11	14	15

(a) 一个 $2^n \times 2^n$ 的 Mortan 码

0	0	1	1
0	0	1	1
2	1	0	0
0	2	0	0

(b) 原始网格中高程数据的分布情况

0		1	
2	1	0	
0	2		

(c) 简化后的高程数据分布情况

图 1 Mortan 码及其简化过程

Fig. 1 Mortan codes and their simplification procedure

为了在简化的 DEM 数据基础上生成 LOD 模型,需要根据简化后的 DEM 数据构建四叉树,对于图 1(c) 表示的 DEM 数据,可以用图 2 所示的一棵不完全四叉树表示。本文中,笔者采用图 3 所示的四叉树单元结点来存储 DEM 数据,并在此基础上构建 LOD 模型。

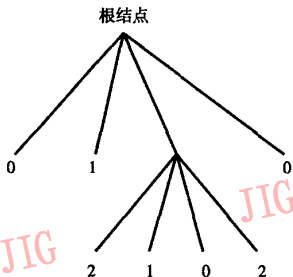


图 2 用四叉树存储简化后的 DEM 数据

Fig. 2 Using quadtree to store simplified DEM data

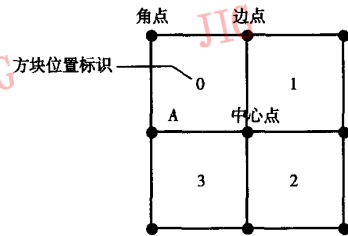


图 3 四叉树单元结点

Fig. 3 A node of quadtree

2.2 基于不完全四叉树的 LOD 技术

2.2.1 算法描述

通常来说,在四叉树基础上构建 LOD 的方式有以下两种,一种是自上而下的方式,即网格由粗到细逐渐细化,直到符合精度要求;另外一种是自下而上的方式,该方式网格由细到粗逐渐粗化,直到符合精

度要求。但无论采用何种方式调整网格精度,其网格变化的时机都是由视点与四叉树网格节点的距离以及与网格粗细变化时不同网格精度的对象空间误差(object-space error)对应的屏幕空间误差(screen-space error)两个因素共同决定的。本文采用自上而下,逐级细化的方法来构建 LOD 模型,即在绘制地形网格前,首先从存储 DEM 数据的不完全四叉树的根结点开始遍历;然后依次判断该结点的 4 个孩子单元能否被细化(图 4 为网格从四叉树根结点 O 开始细化的示意图),在图 5 中,当四边形 $ABCD$ 被细化为与四叉树结点 O 对应的四叉树单元时,则在近似网格和细化网格之间会引起对象空间误差,其大小为加入的边点或者中心点与其对应的近似点之间的距离(图 5 中 h_1, \dots, h_4 为四叉树的某个孩子细化为新的四叉树单元后 4 个边点产生的对象空间误差, h_5 为中心点产生误差),设细化后边点及中心点产生的误差最大值为 e_{max} ,则在实时漫游时,网格细化的条件为

$$e_{max} \times \lambda > d \quad (1)$$

其中, d 为视点到该四叉树结点的距离, λ 为一与投影方式相关的常数,一般取 $\lambda = \omega / \varphi$, ω 为视锥角 φ 范围内投影面上的像素个数;最后按照递归的方式遍历四叉树,当四叉树孩子结点满足细化条件且该结点不是四叉树的叶子结点时,则该结点被细化,否则该结点不再细化,并设置 $bRefinement = false$ 。

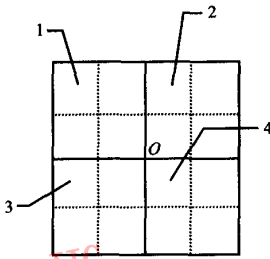


图 4 四叉树结点细化

Fig. 4 Subdivision of a quadtree node

2.2.2 裂缝消除方法

在漫游时,由于不同位置的网格层次是不同的,因此不同层次之间的网格会产生裂缝。为了避免裂缝的产生,还必须对网格中相关的点作调整,图 6 中,为了避免裂缝,需要将 P 点的高程值调整为 \hat{P} 。调整裂缝的方法是从四叉树根结点的 4 个孩子单元开始,用递归的方式依次判断各个结点的上下左右 4 个邻居的细化程度,如果细化程度比该结点单元

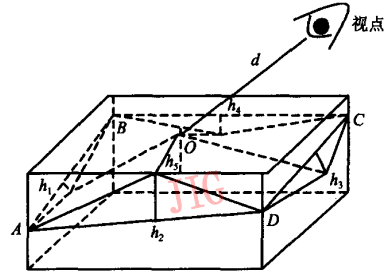


图 5 根据视点位置和屏幕空间误差决定是否细化示意

Fig. 5 Subdivision determined by viewer position and screen-space error

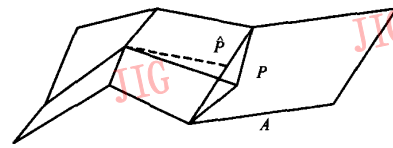


图 6 不同层次之间的裂缝修补

Fig. 6 Patch the cracks between different layers

高,则将该结点的邻居以及邻居的后代单元中相关的点作调整。一般可以将与某结点相邻的邻居分为 2 类,其中一类与该结点具有相同的父亲,另一类则反之。笔者以某结点的第 1 个孩子 A (位置标识为 0) 为例来介绍如下裂缝的调整方法(其他孩子以此类推):

(1) 从图 4 可以看到,图 3 中位置标识为 0 的孩子 A 在整个地形网格中可能有 4 种情形(即图 4 中位置标识为 1, 2, 3, 4 的 4 种情况),除了第 1 种情形外,其他 3 种情形孩子 A 的邻居与孩子 A 不具有相同的父亲,比如,当孩子 A 为第 2 种情形时,它的左邻居和它不具有相同的父亲,当这个孩子 A 的方块处于图中的第 1 种情形时(即孩子方块的第 1 个角点坐标与整个地形方块的第 1 个角点坐标吻合),其邻居为该四叉树结点中位置标识为 1 和 3 的孩子方块(图 3),则它们与孩子 A 具有相同的父亲,此时分别判断标号为 1, 3 的孩子的细化程度,如果孩子 A 孩子的细化程度高,则分别对标号为 1, 3 的孩子进行下一层次的判断,反之退出递归判断,不调整该结点;

(2) 对于标号为 1 的孩子,如果其细化程度比 A 孩子高,且还未达到 LOD 模型的最底层,则需要判断该结点的孩子中的 0, 3 孩子结点,如果该结点的孩子中的 0, 3 孩子结点可以继续细化,则重复判断这 2 个孩子结点的 0, 3 孩子,直到最底层,否则就调整该孩

子与 A 孩子相邻的点(包括 2 个角点);如果标号为 1 的孩子的细化程度比 A 孩子高,且已经达到 LOD 模型的最底层,则调整该结点孩子中与 A 孩子相邻的点(包括 2 个角点和 1 个边点),此时可以通过线性插值的方法来得到新的调整点,经过调整的点,需要将其 `ajustVertex` 或 `ajustCorner` 属性设置为 `true`,这样就可以根据 `ajustVertex` 或 `ajustCorner` 的属性来决定绘图时是使用原始点,还是使用调整点;

(3)对于标号为 3 的孩子,可采用与步骤(2)同样的方法对其 0、1 孩子进行判断,并对相关的结点进行调整。

当孩子 A 的方块处于图 4 中的第 2、3、4 情形时,可以看到,与孩子 A 相邻的结点并不都具有相同的父亲结点,其中对于和孩子 A 具有相同父亲的 1 和 3 孩子,可以采用与步骤(2)、(3)同样的方法进行判断,而对和孩子 A 有不同父亲的邻居,则首先要采用特殊的方法找到该邻居结点,例如在图 4 中,第 2 情形需要寻找其左邻居,第 3 情形需要寻找其上邻居,而第 4 情形则需要同时寻找其左邻居和上邻居。本文以寻找左邻居为例来进一步介绍寻找不同父亲邻居的方法(寻找其他邻居的方法可以类推):

根据以下条件,从四叉树根结点开始,对各个孩子结点进行判断:

- ①该孩子结点的左上角坐标小于等于 A 孩子的左上角坐标;
- ②该孩子结点的右下角坐标大于等于 A 孩子的左下角坐标;
- ③该孩子结点单元中心点的 y 坐标与 A 孩子中心点 y 坐标不同;
- ④该孩子结点的细化程度比 A 孩子低。

如果同时满足上述条件,则该结点可能包含 A 孩子的地形区域或者在 A 孩子区域的左边,需进一步判断;否则,不再对其子结点进行判断。对于满足以上条件的结点,可采用递归的算法,对其每个孩子继续与上述条件进行比较,当符合上述条件,并且该结点的右上角坐标与 A 孩子的左上角坐标相同、右下角坐标与 A 孩子的左下角坐标相同时,即找到了 A 孩子的左邻居。当左邻居细化程度比 A 孩子高时,则对这个左邻居采用与上述步骤(2)、(3)同样的方法,对其 1、2 孩子相关的点进行调整。

2.2.3 误差分析

当在不完全四叉树基础上建立 LOD 模型时,通过 Mortan 码方法简化的那部分地形是不产生投影

误差的,而与被 Mortan 码方法简化部分的不相邻的地形在随视点位置移动时,则由于要调用不同层次的 DEM 数据,因此会在较粗层次产生空间误差,空间误差的值经过投影变化后又在屏幕上产生屏幕误差,屏幕误差的大小以及视点位置的远近决定了具体调用 DEM 数据的层次。

图 7 为透视变换中各参数间的关系示意,设视点的张角和投影平面的边长分别为 α 和 L ,被投影线段的长度为 l ,视点与该线段中心的距离为 d ,设该线段与投影平面平行,显然,此时线段在投影平面上的长度最长,则屏幕投影误差公式^[8]为

$$e \approx \frac{k \times l \times L \times \lambda}{2 \times \tan(\alpha/2) \times d \times \varepsilon} \cdot e_{\max} \quad (2)$$

其中, k 为一可变系数,可以通过它对系统性能进行调整, L 为结点覆盖区域的边长, λ 为物体空间中的单位长度在投影平面上的像素数, d 为视点到地形结点中心点的距离, α 为视角, ε 为用户容许像素误差, e_{\max} 为四叉树结点单元精度变化时产生的最大误差。从式(2)可知,由于 e_{\max} 和 d 随着视点的变化会随时改变,而在特定的屏幕投影误差下,其他参数则可以视为常量,因此实际编程时,笔者先将式(2)简化成式(1),然后采用式(1)来判断调用何种精度的 DEM 数据。

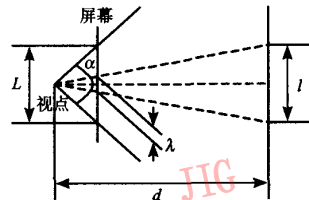


图 7 透视变换原理示意图

Fig. 7 Perspective transform on screen

2.3 背面剔除算法

由于人们在 3 维场景中漫游时,只能看到地形起伏的正面部分,而地形的背面部分被正面部分的网格面片遮挡,因此在绘制地形网格时,这部分网格可以不绘制。背面剔除算法的目的就是将这些看不到的背面网格去掉,该算法可以通过以下步骤实现:

(1)计算位于一个给定网格多边形平面上的某两个向量的矢量积,以便得到这个网格多边形的法向量,这两个向量可以通过多边形顶点的差分来得到,但在求解网格平面的法向量时,必须保证两个向量的

矢量积的方向朝外,否则无法得到正确的法向量值;

(2)计算视点观察方向与法向量之间标量积的符号,由此决定它们之间是否形成大于90°的角。视线与网格平面之间的关系如图8所示,当视线与网格平面法向量之间的夹角大于90°时,则表示这个多边形位于起伏地形的背面,需要剔除,否则不被剔除。

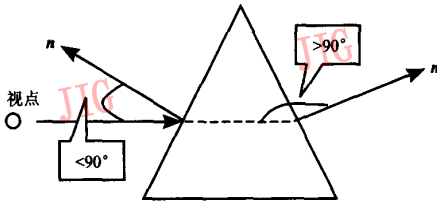


图8 视线与网格平面的关系

Fig. 8 Relationship between view direction and grid plane

在实际编程时,地形网格的法向量可以预先计算并存储在内存中,当漫游时,只需要直接计算视线与法向量的夹角就可以判断网格是否要被剔除。值得提出的是,本文所述的背面剔除算法并不同于一般的消隐算法,因为消隐算法需要绘制被遮挡的物体(即需要执行画图语句),但是不显示这部分被遮挡的物体,而本文所述的背面剔除算法则不绘制(不执行画图语句)被剔除的面。

2.4 高效的二叉树存储和访问机制

快速访问二叉树结点的基本思想是将存储

DEM数据的二叉树结点用1维数组来表示。目前主要有两种方法来提高二叉树结点的访问效率,其中一种是二叉树结点数据在1维数组中的存储方式,另外一种则是访问二叉树结点数组的方式,下面分别介绍这两种方法的实现过程。

2.4.1 二叉树结点的存储方式

将2维的DEM转化为1维方式组织的基本原理是:先将数据空间划分成大小相同的网格,再根据一定的方法将这些网格编码,使得每一个网格的标号是唯一识别的,并能在一定程度上保持空间邻近性,即相邻网格的标号也相邻,以便使一个空间对象由一组网格组成。二叉树结点的存储顺序通常用2维空间填充曲线的方法来确定,因为2维空间填充曲线不仅可以保证遍历空间中的每个数据点,而且填充曲线局部区段内的数据点在空间分布上也是相邻的。本文采用著名的Hilbert空间填充曲线来组织二叉树结点,因为Hilbert值的排序能很好地反映空间邻近性,图9给出了用Hilbert空间填充曲线组织DEM数据的初始几步网格顶点的索引号。DEM数据经过Hilbert空间填充曲线组织后,它自顶向下,由粗到细地将各个分辨率的二叉树结点存储到了一个1维数组中,这种方法实际上是把单一分辨率的数据变换到多分辨率数据后再进行排序,这在3维地形数据的可视化中是一种十分有效的数据组织方式,其使用户能够方便地访问二叉树结点数据。

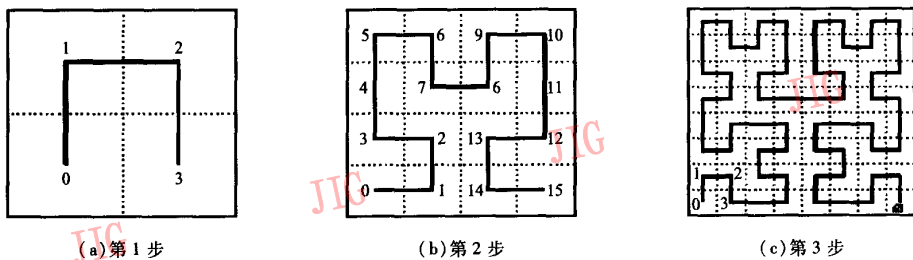


图9 Hilbert 填充曲线

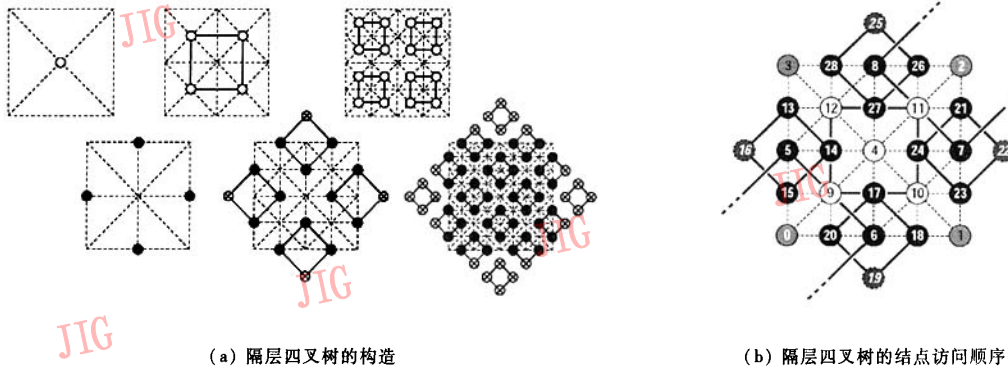
Fig. 9 Hilbert filling curve

2.4.2 二叉树结点的访问方式

二叉树结点访问有隔层二叉树和嵌入二叉树两种常用的方式。这里介绍一种利用隔层二叉树来快速访问二叉树结点数据的实现方法。

隔层二叉树(如图10(a)所示)是指在地形网格由粗往细的剖分过程中,把地形网格按层次分为白色二叉树和黑色二叉树两种,而且每种均为一棵完全的二叉树。这种黑白隔层表示顶点的方式是利

用二叉树的编码,并根据父子结点关系推导出顶点索引号,再按照自上而下、由粗到细的方式存储结点,这样就可以很容易地读取所需的数据信息。为了保证空间数据的邻接性,每一层的数据都需要保存。这样就使得在空间位置上相近的顶点在内存里面的存储位置也相近。然后就可以通过式(3)求解结点的存储位置,如父结点p的第k个孩子的存储顺序为



(a) 隔层四叉树的构造 (b) 隔层四叉树的结点访问顺序

图 10 隔层四叉树的构造及访问

Fig. 10 The construction and access of interleaved quadtree

$$c_{p,k} = 4p + k + m, k = 0, 1, 2, 3 \quad (3)$$

其中, m 是一个常数, 其取值取决于根结点的位置以及相邻两层的距离。使用这种方法, 可以使同一层所有顶点数据都放在一块。图 10(b) 为隔层四叉树头几层结点访问顺序的示意图。

3 应用实例

根据上面所述方法, 笔者在 VC++6.0 环境下, 以 OpenGL 作为 3 维图形绘制工具来模拟荆江地区

的 3 维地形。笔者所使用的微机环境为: P III 933、内存 256M、硬盘 40G、显卡为 32M GeoForce, 原始的 DEM 数据格式为 $129 \times 129, 257 \times 257, 513 \times 513, 1025 \times 1025, 2049 \times 2049$ 。本文对这几个数据文件分别进行了测试, 表 1 给出了对地形数据仅使用 Mortan 码简化、在 Mortan 码简化基础上使用 LOD 技术 ($\lambda = 300$ 时) 以及同时用 Mortan 码简化、LOD 技术和背面剔除算法时的网格简化率以及各种方法相应的绘制速度 (每秒绘制的帧数, 即 fps), 同时给出了不采用地形简化技术的绘制速度。

表 1 不同网格数目 DEM 数据采用不同算法的简化率以及绘制速度

Tab. 1 The simplification rate and the render speed of DEM with different grid number by different algorithms

DEM 数据	原始地形绘制速度 (fps)	Mortan 码简化		Mortan 码 + LOD 技术		Mortan 码 + LOD + 背面剔除	
		网格简化率 (%)	绘制速度 (fps)	网格简化率 (%)	绘制速度 (fps)	网格简化率 (%)	绘制速度 (fps)
129 × 129	97	20.7	98	57.4	95	63.7	95
257 × 257	73	21.3	84	59.8	91	64.4	84
513 × 513	39	24.3	51	60.7	63	66.3	57
1 025 × 1 025	18	25.7	22	65.3	45	69.5	37
2 049 × 2 049	8	33.6	17	73.6	35	78.2	28

从表 1 可以看出, 当网格数据超过 1025×1025 以后, 在微机绘制整个地形时出现明显停顿, 而使用本文所述的方法后, 不仅地形网格得到很大简化, 而且绘制速度仍然能够满足用户在 3 维场景中流畅漫游的目的。

为了和传统的四叉树 LOD 算法比较, 笔者在相同的屏幕误差条件下, 将上述数据分别用传统的四叉树 LOD 方法和本文算法进行了测试, 图 11 和图 12 分别给出了传统四叉树方法与本算法在地形绘

制效率和地形简化率方面的比较结果。从这两个结果图的比较可知, 在 Mortan 码简化地形的基础上直接使用 LOD 技术绘制地形的效率最高, 而且它的简化率也比传统的四叉树 LOD 技术要高; 使用背面剔除算法后虽然可使地形的简化率增加, 但是由于在绘制地形网格之前需要判断网格法向量与视线的关系, 因此影响了地形绘制的效率, 但总的来说, 它的效率与传统的四叉树 LOD 技术差不多。

图 13(a) 显示了一个经过 Mortan 码简化而没

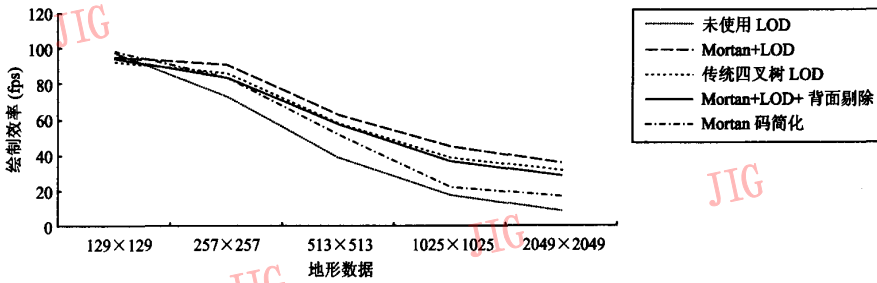


图 11 不同方法的绘制效率

Fig. 11 Different rendering speed with different methods

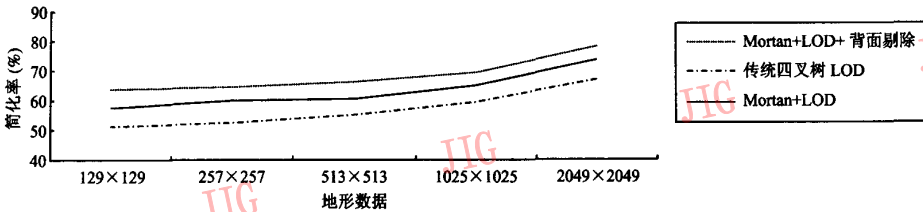
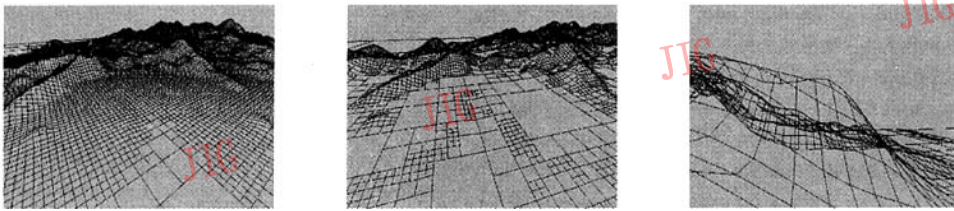


图 12 不同方法的简化率

Fig. 12 Different simplification rate with different methods



(a) 经过 Mortan 码简化后的地形

(b) 在简化地形上构建的 LOD 模型

(c) 进行背面剔除后的地形

图 13 几种不同方法绘制的地形

Fig. 13 Some screen shots generated by different algorithms

有使用 LOD 模型的地形 (257 × 257)。在此基础上使用 LOD 模型和背面剔除算法后,实际绘制地形所需要的方块数大大减少,图 13(b)为在简化地形上构建的 LOD 模型,图 13(c)为在漫游时,对地形起伏的部分进行背面剔除后的地形形态。通过本文所述方法简化网格后,用户就能够流畅地在 3 维场景内进行任意漫游。

4 结论与展望

众所周知,LOD 技术是描述大规模场景的常用方法。为了实时地绘制大规模地形场景,本文提出了一种基于四叉树实时连续 LOD 技术的改进算法,

该算法在传统的 LOD 技术基础上,进一步采用了“先压缩、后构模”以及“背面剔除”的思想来简化地形网格。通过研究可以看到,这两种思想在描述地形时是互相影响的,例如,当地形相对平缓时,虽然原始的 DEM 数据能够进行较大幅度的压缩,但背面剔除的网格较少;当地形起伏相对较大时,背面剔除算法虽能将众多不可见的地形网格去除,但是这部分地形能够压缩的比例往往很小。在实际应用中,用户可以先分别为这两种思想建立程序模块,然后根据地形的实际情况来确定使用哪种模块,这样就可以避免简化效果不理想,同时消耗计算机资源的情况产生。在本文中,笔者同时运用两种方法对荆江地区 DEM 数据进行简化,取得了较好的效果。

由于在四叉树遍历过程中大量使用了递归的算法,因此四叉树的存储格式将直接影响 LOD 构模的速度。一种有效的方法是,采用数组方式存储四叉树结点(而不是通常所用的链表结构),这样就可以利用计算机位移运算(左移和右移相当于结点位置乘 2 或者除 2)的特点来快速定位到需要计算的四叉树结点上,以加快四叉树遍历的速度。

此外,四叉树本身的特性决定了所用的 DEM 数据格式必须为 $(2^n + 1) \times (2^n + 1)$,而实际应用中,有大量的 DEM 数据都不符合这种格式,现在已经有专家提出采用逐层插值的方法来对非 $(2^n + 1) \times (2^n + 1)$ 格式的 DEM 数据构造四叉树(四叉树结构单元的中心点和边点通过插值获取,而其角点则使用原始的 DEM 数据)。如何优化四叉树的存储结构,并采用四叉树结构来描述任意网格的 DEM 数据是笔者正在进一步研究的问题。

参考文献 (References)

- 1 QI Min, HAO Chong-yang, TONG Ming-an. The research development of 3D terrain generation and real-time display techniques [J]. *Journal of Image and Graphics*, 2000, 5(4):269 ~ 275. [齐敏,郝重阳,佟明安. 三维地形生成及实时显示技术研究进展[J]. *中国图象图形学报*, 2000, 5(4):269 ~ 275.]
- 2 WANG Yong-ming. Overview of terrain visualization [J]. *Journal of Image and Graphics*, 2000, 5(6):449 ~ 456. [王永明. 地形可视化[J]. *中国图象图形学报*, 2000, 5(6):449 ~ 456.]
- 3 Lindstrom P, Koller D, Ribarsky W, et al. Real-time continuous level of detail rendering of height fields [A]. In: *Proceedings of SIGGRAPH '96* [C], New Orleans Louisiana, USA, 1996: 109 ~ 118.
- 4 Mark Duchaineau, Murray Wolinsky, David E, et al. ROAMing Terrain: Real-time optimally adapting meshes [A]. In: *Proceedings of Visualization '97* [C], Phoenix Arizona, USA, 1997: 81 ~ 88.
- 5 Cignoni P, Puppo E, Scopigno R. Representation and visualization of terrain surfaces at variable resolution [J]. *The Visual Computer*, 1997, 13(5):199 ~ 217.
- 6 HUAI Yong-jian, HAO Chong-yang, FAN Yang-yu, et al. A view-dependent multi-resolution terrain simplification based on adaptive quadrees [J]. *Journal of System Simulation*, 2002, 14(6):748 ~ 751. [淮永建,郝重阳,范养余等. 基于自适应四叉树视相关的多分辨率地形简化[J]. *系统仿真学报*, 2002, 14(6):748 ~ 751.]
- 7 WANG Yuan, LIU Jian-yong, JIANG Nan, et al. A dynamic triangulation algorithm for the view dependent and real-time LoD model of terrain [J]. *Acta Geodaetica et Cartographica Sinica*, 2003, 32(1):46 ~ 52. [王源,刘建永,江南等. 视点相关实时 LoD 地形模型动态构网算法[J]. *测绘学报*, 2003, 32(1):46 ~ 52.]
- 8 TAN Bing, XU Qing, MA Dong-yang. Real-time multi-resolution terrain rendering using restricted quadtree [J]. *Journal of Computer Aided Design & Computer Graphics*, 2003, 15(3):270 ~ 276. [谭兵,徐青,马东洋. 用约束四叉树实现地形的实时多分辨率绘制[J]. *计算机辅助设计与图形学学报*, 2003, 15(3):270 ~ 276.]
- 9 ZHU Zhi-fan, Machiraju Raghur, Bryan Fry, et al. Wavelet-based multiresolution representation of computational field simulation datasets [A]. In: *Proceedings of Visualization '97* [C], Phoenix, Arizona, USA, 1997:151 ~ 158.
- 10 Gross M H, Gatti R. Fast multiresolution surface meshing [A]. In: *Proceedings of Visualization '95* [C], Atlanta, Georgia, USA, 1995: 135 ~ 142.
- 11 RUI Xiao-ping, ZHANG Yan-min, YANG Chong-jun. Research on the terrain simplification based on Mortan codes [J]. *Journal of Chengdu University of Technology (Science & Technology Edition)*, 2003, 30(3):299 ~ 302. [芮小平,张彦敏,杨崇俊. 基于 Mortan 码的地形简化方法研究[J]. *成都理工大学学报*, 2003, 30(3):299 ~ 302.]