

# 三维地形板块构造模型的研究

喻春阳<sup>1</sup>,赵海滨<sup>2</sup>,赵海<sup>1</sup>

YU Chun-yang<sup>1</sup>,ZHAO Hai-bin<sup>2</sup>,ZHAO Hai<sup>1</sup>

1.东北大学 信息科学与工程学院,沈阳 110004

2.东北大学 机械工程与自动化学院,沈阳 110004

1.School of Information Science & Engineering,Northeastern University,Shenyang 110004,China

2.School of Mechanical Engineering & Automation,Northeastern University,Shenyang 110004,China

E-mail:chyyu@mail.neu.edu.cn

YU Chun-yang,ZHAO Hai-bin,ZHAO Hai.Study on 3D terrain model of plate tectonics.Computer Engineering and Applications,2010,46(28):10-12.

**Abstract:** This paper brings forward a plate tectonic model in the study on real-time and fast rendering of 3D terrain.To achieve real-time rendering between the terrain plates transition,it brings forward slot algorithm.By experiments this model does well in solving the rapid access and rendering of the very large scale terrain.And it is appropriate for distributed application.

**Key words:** 3D terrain;real-time;fast rendering;plate tectonics;slot algorithm;distributed application

**摘 要:**在三维地形实时、快速渲染的研究中,提出了一种板块构造模型。为实现地形板块之间的实时过渡渲染,提出了插槽算法。通过实验,该模型能够解决超大地形的快速访问和渲染问题,适合分布式应用。

**关键词:**三维地形;实时;快速渲染;板块构造;插槽算法;分布式应用

DOI:10.3778/j.issn.1002-8331.2010.28.003 文章编号:1002-8331(2010)28-0010-03 文献标识码:A 中图分类号:TP391.41

随着计算机软硬件的飞速发展,虚拟现实技术也随之迅速发展。其中,三维地形场景漫游有着广泛的应用,如计算机游戏、飞行模拟训练、驾驶模拟训练、军事演习、地理信息系统(GIS)等<sup>[1]</sup>。因此,地形渲染的相关算法研究多年以来一直是计算机图形学领域持续活跃的课题。随着图形处理器绘制能力的提高,以前只能在大型图形工作站上完成的处理任务,现在已经可以在普通PC机上进行,并且开始向手机、PDA、GPS导航仪等移动终端上移植。

地形渲染算法面临的主要问题是数据量巨大导致的存储和带宽限制。为了在计算机(或其他终端)上显示地形,需要为该地形创建网格模型。以一个大小为4 096×4 096的地形网格为例,需要占用至少192M显存空间,一次性渲染该网格模型需要向渲染流水线传输三角形个数为32M个。无论是传输带宽还是显卡的绘制能力,都无法满足实时快速渲染的要求,所以必须找到一些算法来解决这个问题。

## 1 相关工作

为了解决三角形数量的问题,人们提出了一系列LOD(Level-of-Detail)算法。

1996年Lindst Rom等<sup>[2]</sup>首先提出一种基于规整网格的续细节层次实时高度场绘制算法。该算法使用三角形进行自顶向

下二叉分割细分,采用屏幕误差判定条件,提出了具体计算公式。

1997年,Duchaineau等<sup>[3]</sup>在文献[2]的基础上提出了实时优化适应性网格(Real-time Optimal Adaptive Meshes,ROAM)算法,该算法是目前应用很广的一种算法。

1998年,Rittger等<sup>[4]</sup>给出了一种有用的裂缝消除算法。Rittger等使用了一个不同的差计算方法,该误差算法综合考虑了顶点到视点的距离文献[2]中使用的顶点高度差值。

1998年,Hoppe<sup>[5]</sup>将其视点相关递进网格(View-Dependent Progressive Meshes,VDPM)应用到地形绘制中,实现了大规模地形的实时漫游。

2000年,Jonathan<sup>[6-7]</sup>对ROAM做了重要改进。文献[6]抛弃了文献[3]中根据视点计算误差的方法,改为根据误差函数对每个顶点预先计算一个包围球,当视点进入球内时,就加入该顶点;否则就不加入。

2004年Losasso和Hoppe提出了Geometry Clipmap算法<sup>[8]</sup>,构建L层高度图数据保存在磁盘(金字塔型),按照优先级策略进行请求预测,从而实现大规模场景的地形实时漫游。

此外,国内的科研人员在该领域也进行了大量的研究<sup>[9-12]</sup>。

## 2 仍然存在的问题

通过上述介绍的各种算法,在一定的条件下可以较好地

基金项目:中央高校基本科研业务专项资金(No.90303002)。

作者简介:喻春阳(1977-),男,博士研究生;赵海滨(1979-),男,讲师;赵海(1959-),教授,博士生导师。

收稿日期:2010-06-22 修回日期:2010-08-26

解决超大地形快速渲染的问题。但仍然存在一些问题,比如早期的LOD算法,首先需要创建地形网格,将大量的顶点数据存入顶点缓存,使用LOD算法简化后,为需要绘制的顶点创建相应的索引缓存,再进行绘制。该类型的算法在面对更大的原始地形数据时,若显存溢出,即不能一次性存入显存,程序将无法运行。

为了解决这个问题,出现了如分块加载等算法,不再用原始的高度图直接创建地形网格数据,而是采用某个尺寸创建地形网格块数据。该方法的效率同可见区域地形块的数量成反比。同时,对于每个可见的地形块都要进行视点LOD计算,较大地限制了效率的提升。

在Geometry Clipmap算法中,采用L层精细度保存的高度图序列,需要占用更多的磁盘空间。并且不方便高度图的更新。在分布式应用中存在弊端,当前低端平台和低带宽环境下无法使用,且可移植性差。

### 3 板块构造模型

通过对与地形密切相关的虚拟现实具体应用的分析,可以得出一个重要结论:由于视野距离的限制,在超大地形中漫游,与超大地形本身拥有的海量数据是两个概念。以该结论为指导思想,针对使用这些应用的平台(如PC、手机、PDA及GPS导航仪等)的现有硬件资源的条件,并考虑到更新维护的方便性,虚拟三维地形应满足如下几个方面:

- (1)快速渲染,保持稳定的帧速率;
- (2)占用合理的硬件资源;
- (3)方便未来对地形更新和扩展;
- (4)方便移植到不同平台。

在分析和研究了上述算法的基础上,提出了一种板块构造模型,该模型可以很好地完成上述四个方面的任务。其模型构造如图1所示。

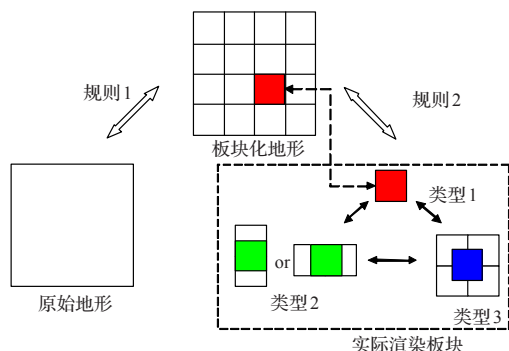


图1 地形板块构造模型图

#### 3.1 地形板块构造模型原理

(1)将用于创建原始超大地形的超大高度图分割成若干张尺寸满足 $(2^n+1) \times (2^n+1)$ 像素的高度图分图,确保相邻高度图分图相邻边的像素相同,即相邻边“重合”在一起。

(2)如图1所示,使用规则1对高度图分图创建索引并排列。

(3)根据用户视点位置,创建其所在板块的地形网格,渲染地形。

(4)随着用户视点的移动,当移动进入到该板块的临界区时,根据临界区的类型(见图3),使用规则2,开辟一个新的进程,预先创建将要“看见”的地形板块网格。满足条件时,用刚

创建的该板块网格进行渲染。

规则1可以采用不同方式对高度图分图进行排列。为了方便将来对地形的可持续扩展,文中将每个高度图分图按照自然数递增方式命名,采用逆时针旋转的方式将其值保存在一个二维数组之中,该数组如图2所示。

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

图2 索引值的保存方式

当用户视点处于某块地形板块中心区域附近时,只需创建该板块的网格数据,然后采用某种LOD算法进行处理,再送入渲染管线渲染。随着用户视点的持续移动,必然会“看到”与该地形板块相邻的板块,在特殊的区域(板块的4个角的区域)中将会“看到”3块相邻的地形板块。如果单纯采用加载并创建地形的的方法,不但需要多创建地形网格数据导致额外的显存消耗,而且还要分别对这几块地形进行LOD计算,这不是好的解决方案。因此,提出了插槽算法。

#### 3.2 插槽算法

插槽算法首先将地形板块划分为3个区域,如图3所示。

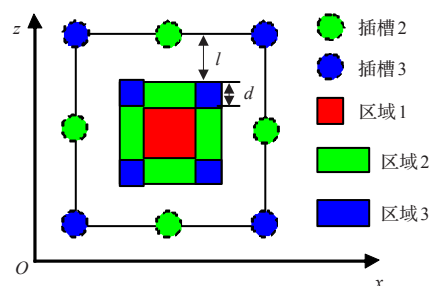


图3 地形板块示意图

图3中, $l$ 为视点的视野范围, $d$ 为缓冲区域的宽度,其值与视点移动的速度、地形板块的尺寸成正比。

接下来,描述插槽算法的实现原理:

(1)假设用户视点位于索引值为0的板块中,当视点在区域1内活动时,仅使用当前板块(索引值0)的网格数据渲染地形。

(2)当视点进入到区域2内时,假设由索引值0往索引值1的方向。开辟新进程,通过索引数组,读入高度图分图1的数据,按照图1中类型2方式创建过渡高度图分图,即使用高度图分图1的右半部分数据和高度图分图0的左半部分数据,相邻边由于数据相同(重合)只是用一次,创建一张过渡高度图分图,然后以插槽2(左侧)的位置为中心,生成相应的地形网格数据,再进行LOD预计算。

(3)当视点穿过区域2的下一帧,将已创建好的过渡地形网格数据送入渲染管线,实现实时、无缝过渡。

同理,当视点进入到区域3内时,假设由索引值0往索引值8的方向。开辟新进程,通过索引数组,读入与其相邻的3张高度图分图(索引值为1、8、7)的数据,按照图1中类型3方式创建过渡高度图分图。然后,以插槽3(左上)的位置为中心,生成相应的地形网格数据,再进行LOD预计算。当视点穿过区域3的下一帧,将已创建好的过渡地形网格数据送入渲染管线,实现实时、无缝过渡。

(4)针对实际应用情况,选取一个适当的消逝时间,超过

该时间后,释放原地形网格数据,节约资源空间。

插槽算法的核心思想是:使用较小的资源实现地形板块之间的无缝、平稳过渡。跨板块时的数据转换可用表1来描述。

表1 跨板块的地形数据转换			
板块类型	区域1	区域2	区域3
类型1	类型1	类型2	类型3
类型2	类型2	类型1、3	类型2
类型3	类型3	类型2	类型1

4 实验和分析

采用上述模型和算法,使用C++结合OpenGL开发了漫游程序。使用21×21共441块1025×1025像素(每个像素2 Byte)的高度图分图,共占用约882 MB硬盘空间。生成的网格点距为20 m,可以覆盖约184 900 km<sup>2</sup>的面积,与中国湖北省(185 900 km<sup>2</sup>平方公里)面积相近。运行机器环境为Windows XP SP2操作系统,CPU为PentiumD 2.0 GHz、内存1 GB、nVidia 128 MB显卡,窗口大小1 024×768,视距为4 km,视角为90°,采用基于四叉树的LOD算法,分辨率误差控制参数为4。对比使用插槽算法/直接加载算法,得到的测试数据如表2所示。

表2 实验数据信息			
板块类型	网格	读入高度分图	平均帧速率
类型1	12M/12M	2M	237.4/236.7
类型2	12M/24M	4M	218.7/163.3
类型3	12M/48M	8M	204.2/112.4

表2中的数据体现了漫游过程中,视点跨越板块时,创建不同类型地形时的情况。使用插槽算法时,创建不同类型的板块,需要额外读入不同数量的高度图分图,但三种类型板块的网格数据量是一样的。平均帧数率根据视点在不同类型地形中10 s内程序绘制的总帧数计算得到,由于LOD计算的结果导致存在的少量差值。而使用直接加载算法时,需要读入的高度图分图数量同插槽算法一样,但是创建的网格数据量却大大增加。同时导致帧速率下降。使用插槽算法运行时的地形网格模型效果如图4所示。

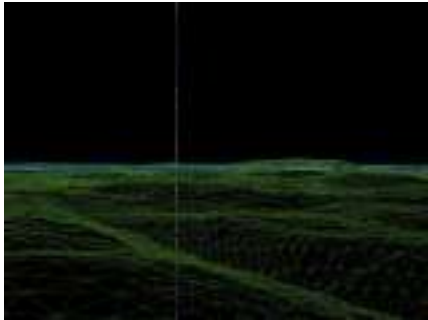


图4 程序运行截图

实验表明,板块构造模型面向应用级开发具有以下优势:

(1)渲染地形时能保持较高、稳定的帧速率;

(2)运行时占用硬件资源较少,针对分布式应用,客户端不需要保存全部高度图分图,使用合理的控制策略,必要时从

服务器端或P2P方式下载并加载即可(可删除长期未使用的高度图分图),可节约大量硬盘空间;

(3)方便高度图分图的修改,通过增加新的高度图分图可以很方便地对地形进行扩展;

(4)方便向手机、PDA、GPS导航仪等其他终端上移植。

5 结论

针对复杂场景中的海量地形数据,提出了一种板块构造模型。该模型可以满足超大地形实时、快速渲染的需求,并符合可持续发展观战略思想的要求。使用该模型在处理板块之间实时跨越问题上可以使用多种算法。本文提出的插槽算法相对占用的资源较少。该算法有一个约束条件,单个地形板块的边长要大于4倍的视野距离,这样,当用户视点在相邻地形板块间移动时,可以保证板块切换后,仍然位于新板块的区域1中。

参考文献:

[1] 赵友兵,石教英,周骥,等.一种大规模地形的快速漫游算法[J].计算机辅助设计与图形学报,2002,14(7):624-628.

[2] Indstrom P L,Koller D.Real-time continuous level of detail rendering of height fields[C]//Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, New Orleans, Louisiana, 1996:109-118.

[3] Duchaineauy M,Wolinsky M.ROAMing terrain: Real-time optimally adapting meshes[C]//Proceedings of IEEE Visualization, Phoenix A Rizona, 1997:81-88.

[4] Rittger S,Heidrich W,Slasallek P,et al.Real-time generation of continuous levels of detail for height fields[C]//Proceedings of the 6th International Conference in Central Europe on Computer Graphics and Visualization, 1998:315-322.

[5] Hoppe H.Smooth view-dependant level-of-detail control and its application to terrain rendering[C]//Proceedings of IEEE Visualization, Research Triangle Park, Worth Carolina, 1998:35-42.

[6] Jonathan B.Terrain rendering at high levels of detail[C/OL]//Proceedings of Game Developers' Conference, San Jose, California, 2000.http://www.bolt-action.com/papers/gdc2000-final.zip.

[7] Jonathan B.Terrain rendering research for games[C]//Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, New Orleans, Louisiana, 2000:39-44.

[8] Losasso F,Hoppe H.Geometry clipmap: Terrain rendering using nested regular grids[C]//Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, CA, USA, 2004.[S.l.]: ACM, 2004: 769-776.

[9] 陈刚,熊兴华.海量地形漫游中动态LOD算法研究[J].测绘通报, 2007(4):46-48.

[10] 杜莹,武玉国,王晓明,等.全球多分辨率虚拟地形的金字塔模型研究[J].系统仿真学报,2006,18(4):955-958.

[11] 戴晨光,张永生,邓雪请.一种用于实时可视化的海量地形数据组织与管理方法[J].系统仿真学报,2005,17(2):406-409.

[12] 郎兵,方金云,韩承德.复杂场景中海量外存地形模型的实时绘制算法[J].系统仿真学报,2009,21(20):6510-6514.