

基于 OpenGL 的三维地形的模拟

邹海, 徐军, 褚维翠

(安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

摘要: 三维地形模拟一直是虚拟现实、地理信息系统(GIS)等领域的研究热点。提出了一种基于不同层次的多边形网格模型的地形生成算法,采用多重纹理映射技术,通过插值对应网格基本元素的位置实现模型之间的平滑过渡并生成地形连续 LOD 模型。通过该技术可以简化场景的复杂度,并对图形的真实感程度损失比较小。该算法是在 VC++6.0 平台上利用 OpenGL 编程技术实现的。实验表明:该算法具有较低的时间和空间开销,适合大规模三维地形的模拟。

关键词: 虚拟现实;多边形网格模型;纹理映射;连续层次细节

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2011)06-0239-03

3D Terrain Simulation Based on OpenGL

ZOU Hai, XU Jun, CHU Wei-cui

(School of Computer Science and Technology, Anhui University, Hefei 230039, China)

Abstract: 3D terrain simulation is always a hot topic in geographic information system(GIS), virtual reality and so on. An algorithm for constructing polygon meshes based on different levels is presented, which uses texture mapping to terrain LOD and interpolate the location of element in meshes to realize smooth transitions between models. The technology could reduce the complexity of scene properly and the distortion is also lessened. This algorithm is finished on VC++6.0 platform with OpenGL programming technique. The testing results show the efficiency of algorithm is high on both time and memory and the algorithm supports simulation of large scale 3D terrain.

Key words: virtual reality; polygon meshes model; texture mapping; continuous levels of detail

0 引言

三维地形模拟^[1-3]近年来应用领域非常广泛,涉及到地理信息系统、系统仿真、虚拟现实^[4,5]环境,因此受到广泛的关注。它的核心问题就是要解决海量地形数据构成的复杂地形模型与图形硬件有限的绘制能力之间的矛盾。为了能得到完美的视觉效果和计算机处理速度,在场景中使用一定的技术对其模型进行管理就显得很重要,之前有很多人采用不同的技术对场景中的模型进行管理,比较常用的技术有:LOD 技术^[6-8]、脱线预计算、内存管理技术以及单元分割法。文中采用 LOD 技术对地表模型进行简化,采用模型的分辨率不同来显示场景中的不同物体,使在生成的三维地形质量损失很小的情况下提高绘制效率,进而实现视点相关的连续 LOD^[9]三维地形的生成和绘制。

1 OpenGL 简介

OpenGL^[10,11](Open Graphics Library)是一个三维

的计算机图形库,它最初是美国 SGI 公司为其图形工作站开发的 IRIS GL。最初是个 2D 图形函数库,后来慢慢演化为这家公司的高端 IRIS 图形工作站使用的 3D 编程接口,之后在跨平台移植过程中通过改进和提高发展成为 OpenGL。它是一个开放的三维图形软件包,独立于窗口系统和操作系统,以它为基础开发的应用程序可以十分方便地在各种平台间移植;OpenGL 可以与 Visual C++ 紧密接口,便于实现机械手的有关计算和图形算法,可保证算法的正确性和可靠性。文中就是利用 OpenGL 与 Visual C++ 的接口来实现算法。

2 LOD 技术简介

当要生成具有相当真实感的场景时,由于场景本身的复杂性,要实现实时性交互往往不太可能。需要从场景本身的几何特性入手,通过适当的方法来简化场景的复杂性,LOD 技术就是在这种情况下应运而生。它是一种符合人的视觉特性的技术。当距离很远观察物体的时候,它们经过观察、投影变换后在屏幕上只是几个像素,没有必要为该物体绘制所有的细节,因此只要在不影响绘制画面视觉效果的前提下,对景物的表面细节逐次简化来减少场景的几何复杂性,这样

收稿日期:2010-11-03;修回日期:2011-02-20

基金项目:国家科技重大专项资助项目(2008ZX05039-004)

作者简介:邹海(1969-),男,安徽寿县人,硕士研究生导师,研究方向为数据挖掘与信息检索、工作流理论与技术。

在一定程度上可以提高绘制算法的效率。该技术通常是对一个原始多面体模型建立几个不同相近程度的几何模型,因此,恰当地选择细节层次模型可有效提高解决场景复杂与设备渲染能力有限的问题。

然而,在这里需要注意,当视点连续变化时,两个相邻模型的层次不同,它们在过渡时就存在一个明显的跳跃,为了达到相邻层次的模型之间光滑的视觉过渡,在生成的图像序列中真实感程度上有光滑的视觉过渡就显得尤为重要。文中的研究主要是相对于原始网格模型,怎样建立不同层次细节的模型以及相邻层次的多边形网格模型之间的几何形状过渡。

3 关键技术及解决方案

针对场景中原始网格模型,建立不同层次细节的模型,这里假定场景中的模型都是用三角形网格表示,那么根据网格的几何及拓扑特性,可以有三种化简操作,如图 1 所示。

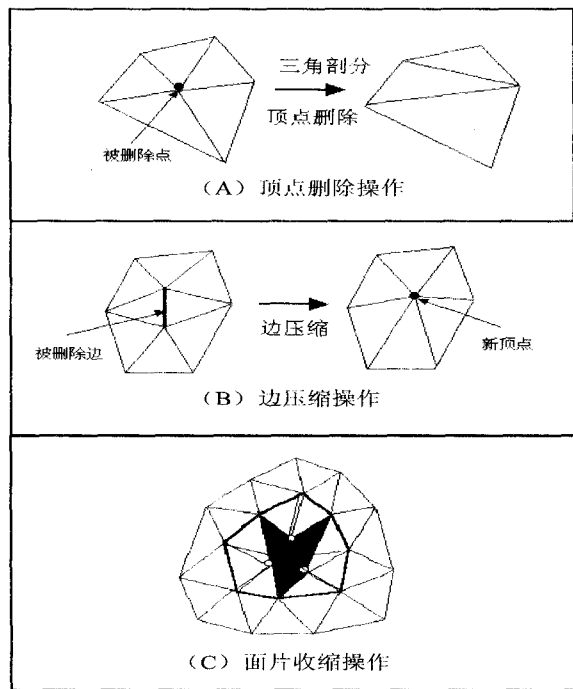


图 1 LOD 化简的基本操作

(1)删除顶点:将网格中的一个顶点删除,那么空洞就会出现在该顶点相邻的三角形上,需要对这些空洞进行三角剖分,这样网格模型的拓扑结构可达到一致。

(2)压缩边:把网格模型中的某条边压缩到一个顶点,该边上邻近的三角形就会收缩,并且这条边上的两个端点融合为一个新顶点。

(3)收缩面片:将网格中的一个面收缩为一个顶点,那么该网格中的三角形和与它相邻的三个三角形也同时收缩,该面片的三个顶点合并为一个新顶点。

利用上述的化简步骤,再确定这些步骤给网格场景所带的误差,再用这个误差代替计算方法,计算出原始模型网格中的每个基本元素的误差,并将误差作为权值,插入到队列中,该队列是按权值升序排列的。接着就可以对网格循环进行上述基本化简步骤。在每次循环中,需要从队列中选择队首权值最小操作来运行,对变动的网格信息进行刷新并重复计算已变动的网格元素的误差,将误差插入到队列中进行下一个循环。如果队列的最小误差不大于用户最初设定的阈值或是得到了用户所期望的网格简化数量,循环结束。通过此种途径,可以建立原始场景有差别的层次细节的模型,建立的模型相对于原始网格,它们之间的误差是逐步递增的。

对于邻近层次的网格模型,如何实现它们之间的模型过渡,一般的方法就是在对应的网格基本元素的位置对其插值来实现平滑过渡,而怎样得到两个相邻网格模型的基本元素之间的相互对应关系才是关键。如果进行面片收缩操作和顶点删除,只需要在被操作的对象与对象邻近的基本元素之间建立相对关系;而对于边压缩操作,只需要在压缩后的新点和压缩边上的两个点之间建立对应关系。确定了这些对应关系后,就可以通过插值法实现相邻网格模型之间的光滑过渡。

4 三维地形模拟关键代码及仿真结果

4.1 设置不同的层次结构

首先需要定义一个全局变量 level,该变量表示不同网格模型的层次参数,再根据不同的层次参数值和绘制点与视点的距离来计算各种参数,代码参考如下:

```
void CLod::setup_qtree(int x, int z, int width)
{
    int v[3];
    v[0] = x;
    v[2] = z;
    int width1;
    width1 = width / 2;
    //定义不同的层次结构
    if( DISTANCE(v, location) < width * level && (width > 1))
    {
        qtree[ qut(x, z) ] = node;
        qtree[ qut(x - width1, z - width1) ] = edge;
        qtree[ qut(x + width1, z - width1) ] = edge;
        qtree[ qut(x - width1, z + width1) ] = edge;
        qtree[ qut(x + width1, z + width1) ] = edge;
        setup_qtree(x - width1, z - width1, width1);
        setup_qtree(x + width1, z - width1, width1);
        setup_qtree(x - width1, z + width1, width1);
        setup_qtree(x + width1, z + width1, width1);
    }
    else
```

```

    } qtree[ qut(x, z) ] = edge;
}

```

4.2 连续 LOD 地形的绘制

在绘制地形时,主要通过计算不同层次的网格点的数据来绘制,采用三角形来模拟地形的绘制,针对不同模式的地形,三角形绘制函数也分别不同。

```

void CLod::tangle1(int x1, int z1, int x2, int z2, int x3, int z3)

```

```

{
    glBegin( GL_LINE_LOOP );
    glVertex( x1, z1 );
    glVertex( x2, z2 );
    glVertex( x3, z3 );
    glEnd();
}

```

```

void CLod::tangle2(int x1, int z1, int x2, int z2, int x3, int z3)

```

```

{
    glBegin( GL_TRIANGLES );
    glTexCoord2f( x1/(texScale), z1/(texScale) );
    setColor( x1, z1 );
    glVertex( x1, z1 );
    glTexCoord2f( x2/(texScale), z2/(texScale) );
    setColor( x2, z2 );
    glVertex( x2, z2 );
    glTexCoord2f( x3/(texScale), z3/(texScale) );
    setColor( x3, z3 );
    glVertex( x3, z3 );
    glEnd();
}

```

4.3 模拟结果

下面给出了文中提出的算法在上述解决方案下的实验结果。实验平台是在 VC++6.0 上利用 OpenGL 编程技术来实现。如图 2 所示,给出了两种模式下的地形模拟。可以看出,在线框模式下利用文中提到的三种化简操作对网格进行处理,在不影响视觉效果的前提下,一定程度上提高了绘制效率,并且在纹理模式下也取得了较好的渲染效果。

5 结束语

文中提出了一种基于不同层次的多边形网格模型的地形生成算法,采用纹理映射技术^[12],通过选择不同层次的网格模型来实时有效地生成三维地形,并实现了不同模型之间的平滑过渡,实验取得了比较好的效果,适合用于大规模三维地形的模拟。今后还要继续研究大规模三维地形的可视化问题,除了文中采用的 LOD 技术,还有如单元分割法、脱线预计算等,并致力于采用并行技术来实现海量数据的处理和绘制。

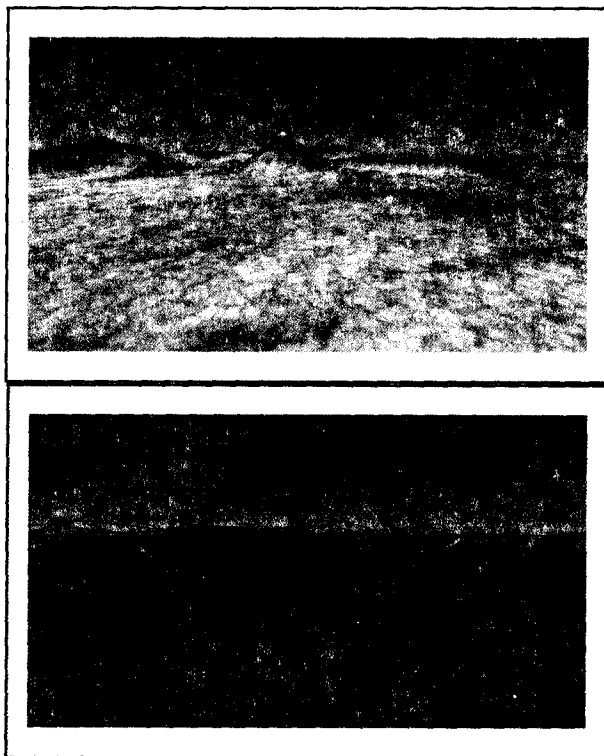


图2 地形模拟(纹理模式(上)和线框模式(下))

参考文献:

- [1] 平西建. 基于 OpenGL 的三维地形可视化技术研究[J]. 计算机工程与设计, 2008(1): 193-195.
- [2] Kruger J, Kipfer P, Konclratieva P, et al. A particle system for interactive visualization of 3D flows[J]. IEEE Transactions on Visualization and Computer Graphics, 2005, 11(6): 744-756.
- [3] Stachniak S, Stuerzlinger W. An Algorithm for Automated Fractal Terrain Deformation[C]//Proc of Computer Graphics and Artificial Intelligence. [s.l.]: [s.n.], 2005: 64-76.
- [4] 胡少林. 基于 DEM 数据的三维地形建模方法研究与实现[D]. 长沙: 国防科技大学, 2002.
- [5] Jiang Xuezh, Li Zhonghua. Domestic and Foreign Virtual Reality Technology Research Present Situation[J]. Liaoning Engineering Technology College Newspaper, 2004, 23(2): 238-240.
- [6] Watt A. 3D 计算机图形学[M]. 包宏, 译. 北京: 机械工业出版社, 2005.
- [7] 何援军. 计算机图形学[M]. 北京: 机械工业出版社, 2010.
- [8] 李颖, 陈业斌. 基于四叉树的三维地形绘制算法[J]. 安徽工业大学学报(自然科学版), 2008, 25(1): 80-82.
- [9] 杨晓霞, 齐华. 一种大规模地形的高效绘制算法[J]. 计算机工程与应用, 2005(14): 229-232.
- [10] OpenGL Architecture Review Board, Shreiner D, Woo M, et al. OpenGL 编程指南[M]. 徐波, 译. 北京: 机械工业出版社, 2008.
- [11] 刘飒. 基于 OpenGL 的三维地形可视化技术研究[D]. 大庆: 大庆石油学院, 2007.
- [12] 周永前. 面向低空飞行与突防的三维演示系统研制[D]. 南京: 南京航空航天大学, 2007.