

基于 OpenGL 的分形山三维模拟生成技术

张 艳

(安徽财经大学, 安徽 蚌埠 233041)

摘 要:采用 OpenGL 技术与 VC++ 6.0 结合分形几何的方法来模拟绘制三维分形山。利用分形几何理论建立几何模型;依据分形插值理论设计相应的算法,得到模拟对象的几何数据,绘制出三维山和云的图形;利用 OpenGL 的图形处理及渲染功能对生成的图形进行色彩、光照、纹理等方面的处理,得到三维分形山的模拟图形。

关键词:OpenGL;分形山;三维模拟

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2007)08-0243-04

Fractal Mountains' Three-Dimensional Simulation Based on OpenGL

ZHANG Yan

(Anhui University of Finance and Economics, Bengbu 233041, China)

Abstract: In this paper, OpenGL and VC++ 6.0 are applied with fractal geometry to protract three-dimensional fractal mountains. Firstly, the geometry models are built by the fractal geometry theory. Secondly, the corresponding algorithms are designed to gain the geometry data of these objects according to fractal interpolation algorithm theory so as to protract the graphics of mountains and cloud. Finally, graphics disposal and render functions of OpenGL are used to deal with the aspects of their color, illumination and texture. Consequently, three-dimensional simulation fractal mountains are obtained.

Key words: OpenGL; fractal mountains; three-dimensional simulation

0 引 言

虚拟自然环境的生成是计算机图形学和计算机仿真领域研究的热点,由于用传统欧几里得几何学不能描述自然界中一大类复杂无规则的几何对象,例如:弯弯曲曲的海岸线,起伏不平的山脉,粗糙不平的断面,变幻无常的浮云,九曲回肠的河流,纵横交错的血管等。于是,出现了用分形技术来模拟自然景物的方法^[1]。

OpenGL 对三维图形的处理功能十分强大,将 OpenGL 应用于三维图形的开发,大大简化了从底层开发的难度,OpenGL 作为一个性能优越的图形应用程序设计接口,适用于广泛的计算机环境。以 OpenGL 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 可实现与 VC++ 6.0 无缝连接,可保证算法的正确性和可靠性^[2]。OpenGL 应

用领域十分宽广,如军事、电视广播、CAD/CAM/CAE、娱乐、艺术造型、医疗影像、虚拟世界等。它使用方便、效率高^[3]。文中利用 OpenGL 与 VC++ 6.0 无缝连接实现分形山的三维模拟生成。

1 分形山的三维模拟生成

文中所生成的三维模拟系统是基于分形技术生成算法并结合 OpenGL 的强大功能(主要包括:建模、几何变换和投影变换、光照和材质设置、双缓存动画、纹理映射、颜色模式设置,位图显示和图像增强等功能)而生成的,本系统可以参数化输入、实现漫游、改变山形颜色,使之可以表现春夏秋冬变化、改变山的背景即晴天和阴天等功能。

1.1 分形山的设计

分形山的设计过程分为几何建模和渲染两部分。首先,在山脉轮廓结构的基础上,利用分形算法生成山体的细节地貌结构;然后,通过对所得到的三角面片的绘制,得到山体的网格结构^[4]。在完成分形山的几何建模之后,运用 OpenGL 对其进行光照、材质、纹理映射等一系列处理,得到具有真实感的三维分形山效果图^[5]。

收稿日期:2006-10-25

基金项目:安徽财经大学 2007 年度青年科研资助项目(ACKYQ 07472C);安徽财经大学信息工程学院青年教师资助项目(xgky2006015)

作者简介:张 艳(1979-),女,河北衡水人,硕士,讲师,主要研究方向为计算机图形学。

1.2 分形山的算法实现

分形山建模方法有多种,大致可分为泊松阶跃法(Poisson faulring)、傅里叶滤波法(Fourier filtering)、中点位移法(Midpoint displacement)、逐次随机增加法(Successive random additions)和带限噪声累积法(Summing band limited noises)、小波变换^[4]等。文中采用随机中点位移法来模拟山脉,使生成的图形简便快捷、形象美观。

1.2.1 一维分形山的实现

用随机中点位移法生成一维分形山,基本步骤:

第一:将一条线段作为初始元;

第二:让这条线段的中点向上或向下移动一段随机距离,这样就形成两条线段;

第三:将这两条线段的中点又分别向上或向下移动一段随机距离;

第四:以此类推,直到完成预定的递归次数。

1.2.2 三维分形山的实现

三维分形山的实现,是在一维轮廓图形的理论基础上的,其算法采用分形插值法中的 Diamond-Square^[1]算法。对随机中点位移法在构网方式及算法上进行改进,用它来模拟三维分形山脉。

首先绘制分形山的线框图,然后用 OpenGL 对图形进行处理,生成山脉的三维效果图。在程序设计中,通过增加网格中心点的高度来产生山峰。

采用递归的方法生成分形山,具体算法如下:

第一:在 X-Y 平面上绘制一个 $n \times n$ 的正方形网格,并对 4 个角点在 Z 方向上分别设置初始高度 ha, hb, hc, hd ,得到 A, B, C, D 四个点,用黑点表示(见图 1(a));

第二:取相邻 4 个点组成的正方形 ABCD,在其中点生成一个随机值 $\triangle h_1$ 。平均 4 个顶点的高度值再加上这个随机量 $\triangle h_1$,得到中点的高度值。即正方形 ABCD 网格中点 M 的高度为:

$$hm = (ha + hb + hc + hd)/4 + \triangle h_1 \quad (1)$$

从而得到 M 点(见图 1(b));

第三:根据角点 A, B, C, D 和中点 M 以及网格外一虚拟点(其高度为 0),计算边中点的高度,即:

$$he = (ha + hb + hm + 0)/4 + \triangle h_1 \quad (2)$$

$$hf = (hb + hc + hm + 0)/4 + \triangle h_1 \quad (3)$$

$$hg = (hc + hd + hm + 0)/4 + \triangle h_1 \quad (4)$$

$$hh = (hd + ha + hm + 0)/4 + \triangle h_1 \quad (5)$$

其中, $\triangle h_1$ 为与第二步相似的随机量,从而得到 E, F, G, H 这 4 个点(见图 1(c));

第四:递归第二步和第三步,使正方形网格逐步细化,直至达到预期的递归深度,然后连接每个正方形网

格点。

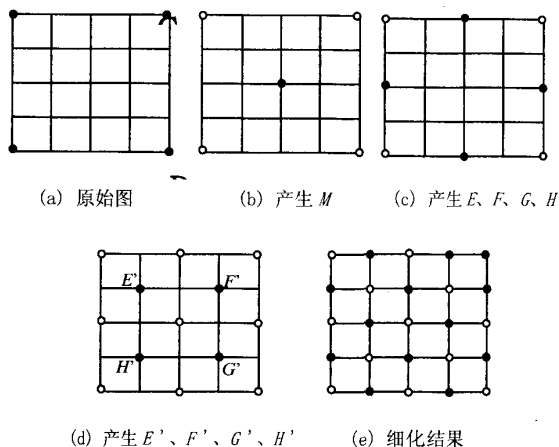


图 1 三维分形山的算法实现过程

这样,如果已经生成了一个种子正方形并经过单独一次细分过程,得到 4 个正方形,第二次经过该过程得到 16 个正方形,第三次得到 64 个正方形,增长得很快。正方形数目等于 2^{2i} ,其中, i 为递归经过细分过程的次数。

经过第一次运算(见图 1(a)、(b)、(c)),得到 A, B, C, D, M, E, F, G, H 这 9 个点的几何数据。用线将这 9 个点连起来,就可以得到一个线框图(见图 2)。

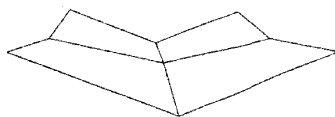


图 2 第一遍递归后生成的线框图

接下来进行第二遍递归过程。再次从第二步开始。第二遍与第一遍有些不同:

首先,现在有 4 个四边形面,它们分别是 EBFM, MFCG, HMGD, AEMH, 因此要分别计算这四个方面的中心 E', F', G', H' (见图 1(d)):

$$he' = (he + hb + hf + hm)/4 + \triangle h_2 \quad (6)$$

$$hf' = (hf + hc + hg + hm)/4 + \triangle h_2 \quad (7)$$

$$hg' = (hh + hm + hg + hd)/4 + \triangle h_2 \quad (8)$$

$$hh' = (ha + he + hm + hh)/4 + \triangle h_2 \quad (9)$$

然后,分别计算这 4 个正方形边中点的高度(见图 1(e))。

其次,也是关键的一点,生成随机数的范围已经减小了。在第三步中,4 个 $\triangle h_1$ 是不一定相等的,因为它们都是随机数,但是它们的取值范围相同。同样,在第二遍递归过程中,4 个 $\triangle h_2$ 也不一定相等,但是它们也有相同的取值范围。 $\triangle h_2$ 的取值范围要比 $\triangle h_1$ 的取值范围小,这样设计是为了满足视觉上的需要,同时也可以使模拟出的自然景物更加真实。

经过第二遍运算,图中 25 个元素都已经生成,可以得到分形山的线框曲面(见图 3)。



图 3 第二遍递归后生成的线框图

用分形插值算法生成山脉,中点高度值是平均其相邻 4 个顶点的高度值再加上一个随机量得到的。由于不同细分阶段产生的点在相邻区域中计算时,会得到不同的结果,这样就会留下一道明显的裂痕。为了解决这一问题,在设计程序时,将一个区域内的边界值拷贝给其相邻区域边界,以实现图形的无缝连接。

分形插值算法具有为已有形状增加细节的功能,增加迭代次数,可以给曲面增加细节。迭代次数为 5 时,分形山的线框曲面图见图 4。

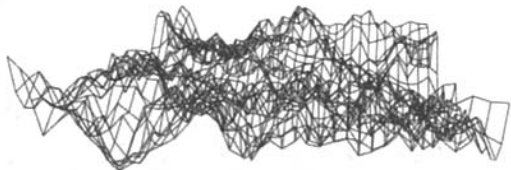


图 4 多次使用分形插值算法的结果

1.3 分形山立体效果图的设计

分形山的线框图已经生成,下面要生成分形山的立体效果图。首先,要判断山体中每个面的可见性,把可见面显示出来,隐去不可见面;然后对可见的表面进行纹理映射。

在设计过程中,用三角形来绘制山体表面,利用三角形的三个顶点的数据,计算出该面的法向量(见图 5),平面法向量 N 的计算公式为:

$$N = (P_0 - P_1) \times (P_2 - P_1) \quad (10)$$

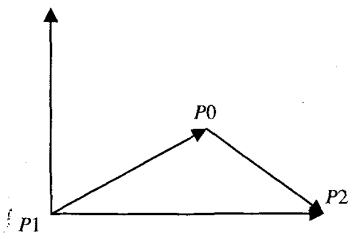


图 5 三角形平面法向量

表面法向量是垂直于相应平面指向平面外的矢量,观察方向向量是一条起始于表面法向量的起点指向观察点的矢量,见图 6。

$$\theta = \arccos\left(\frac{N \cdot V}{|N| \cdot |V|}\right) \quad (11)$$

有了三角面的法向量,再求出观察者对该面片构

成的方向向量,根据公式(11),通过观察方向向量与表面法向量的数量积的正负,就可以判断出该面片是否可见:为正即可见,为负不可见。对于不可见的面片就省去了以后的处理工作,提高了绘图效率。

在生成了山体表面的消隐图后,用 `glTexCoord2f()` 函数进行纹理映射,将在 OpenGL 中生成的纹理图映射到山体表面,便生成具有真实感的三维分形山脉效果图。

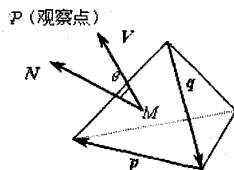


图 6 凸多面体可见面的判别

2 云的生成

云属于气体现象,其外观形状极不规则,没有光滑的表面,而且极其复杂和随意,所以用经典的欧几里得几何学对其描述非常困难。考虑到纹理映射技术具有快速建模的优点,结合分形理论和映射技术,用分形插值的方法生成模型的基础数据,通过转换得到云图纹理,最后利用映射技术将纹理图映射到天空模型上,实现蓝天白云效果图的绘制。其设计过程如下:

1) 云纹理图的生成。

采用递归的方法,通过分形插值算法,生成云的纹理图。具体算法如下:

第一:在平面上划分正方形网格 $n \times n$ (如图 1 所示),随机给出 A, B, C, D 这 4 个角点的颜色。

第二:平均这 4 个角点的颜色,再加上一个随机生成的颜色值,计算出中点 M 的颜色。

第三:计算正方形 4 个边中点的颜色:根据 A, B, M 点和网格外一虚拟点(颜色值为 0)求平均,加上一个随机生成的颜色值,得到边中点 E 的颜色;根据 B, C, M 点和网格外一虚拟点(颜色值为 0)求平均,加上一个随机生成的颜色值,得到边中点 F 的颜色;根据 C, D, M 点和网格外一虚拟点(颜色值为 0)求平均,加上一个随机生成的颜色值,得到边中点 G 的颜色;根据 A, D, M 点和网格外一虚拟点(颜色值为 0)求平均,加上一个随机生成的颜色值,得到边中点 H 的颜色。

第四:根据小正方形 $EBFM$ 4 个角点颜色的平均值,求出小正方形中点以及小正方形的 4 个边中点的颜色;然后再沿用上述规则,分别计算小正方形 $MFCG, HMGD, AEMH$ 的各个中点和边中点的颜色。

第五:递归第四步,使其正方形网格不断细化,直

至达到预期的递归深度。

2) 蓝天白云的绘制。

运用 OpenGL 来完成纹理贴图,经过投影变换及真实感效果处理后,得到蓝天白云的最终结果。绘制过程包括下面两个部分:

(1) 在三维空间中定义八面角锥作为天空模型。

调用 `glBegin` 函数来标识图元的开始,在调用函数时,需要传入一个参数,以告诉 OpenGL 我们将绘制什么类型的图元。传入 `GL_TRIANGLES` 表明将要绘制三角形。

```
glBegin (GL_TRIANGLES);
    glTexCoord2f (0.f, 0.f);
    glVertex3f (0.f, 2.f, 0.f);
    glTexCoord2f (-TMMAX45, -TMMAX45);
    glVertex3f (-10.f, -1.f, 10.f);
    glTexCoord2f (0.f, -TMMAX);
    glVertex3f (0.f, -1.f, 15.f);
    glTexCoord2f (TMMAX45, -TMMAX45);
    glVertex3f (10.f, -1.f, 10.f);
    glTexCoord2f (TMMAX, 0.f);
    glVertex3f (15.f, -1.f, 0.f);
    glTexCoord2f (TMMAX45, TMMAX45);
    glVertex3f (10.f, -1.f, -10.f);
    glTexCoord2f (0.f, TMMAX);
    glVertex3f (0.f, -1.f, -15.f);
    glTexCoord2f (-TMMAX45, TMMAX45);
    glVertex3f (-10.f, -1.f, -10.f);
    glTexCoord2f (-TMMAX, 0.f);
    glVertex3f (-15.f, -1.f, 0.f);
    glTexCoord2f (-TMMAX45, -TMMAX45);
    glVertex3f (-10.f, -1.f, 10.f);
}
```

(2) 利用 OpenGL 提供的纹理映射功能,将生成的云图纹理映射到天空模型上,得到云的效果图。

3 实现漫游

在山脉和云生成以后,利用双缓存和坐标的平移及旋转实现漫游功能。实现整体效果图的漫游,是利用分形自身的特点,其算法内部的参数变化会直接影响图形的外观,而且外观对参数的变化也是非常敏感的,当在程序中将坐标进行连续平移及旋转变化时,程序所生成的图形就会产生漫游的效果。利用 OpenGL 支持双缓存的技术,即创建两个缓冲区,一个用来绘制图形,一个用来显示图形。在任意时刻只有两者中的一个内容能被显示出来。当前可见视频缓存称为前台视频缓存,不可见的正在画的视频缓存称为后台视频

缓存。当后台视频缓存中的内容被要求显示时,OpenGL 就会将它拷贝至前台视频缓存。显示硬件则不断地读可见视频缓存中的内容,并把结果显示在屏幕上。当完整的画面在后台视频缓存中画出后就调用 `SwapBuffers()` 函数^[6],使其成为可见的视频缓存,以此来实现整体效果的动态漫游显示。

4 分形山三维模拟软件

在说明了模拟对象的算法之后,用 Visual C++ 6.0 开发该软件的用户界面及功能。该软件实现的主要功能有:模拟山脉轮廓,绘制山体的网格曲面图形,绘制山脉的纹理图,绘制云彩的纹理图、山脉及蓝天白云的整体渲染效果图(见图 7)。

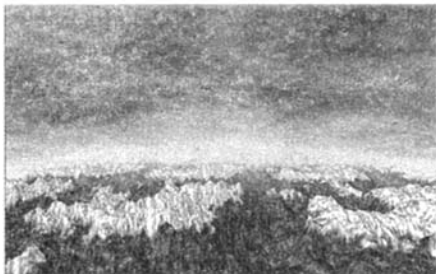


图 7 效果图

5 结 论

采用 OpenGL 技术结合分形几何的方法来模拟绘制三维分形山。首先,根据模拟对象的不同特点,利用分形几何理论建立它们各自的几何模型;然后,依据分形插值理论设计相应的算法,得到模拟对象的几何数据,绘制出平面图形;最后,为了得到较好的视觉效果,利用 OpenGL 的图形处理及渲染功能对生成的图形进行色彩、光照、纹理等方面的处理,从而得到具有真实感的三维自然景物模拟分形图形。

参考文献:

- [1] Mandelbrot B B. 大自然的 fractal 几何学[M]. 陈守吉, 凌复华译. 上海: 上海远东出版社, 1998.
- [2] 尚游, 陈岩涛. OpenGL 图形程序设计指南[M]. 北京: 中国水利水电出版社, 2001.
- [3] Hill F S Jr. Computer Graphics Using OpenGL[M]. Second Edition. [s.l.]: Prentice Hall/Pearson, 2004.
- [4] 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): 269-274.
- [5] Hearn D, Baker M P. Computer Graphics with OpenGL[M]. Third Edition. [s.l.]: Pearson, 2004.
- [6] 郭燕利, 胡建军. 利用 OpenGL 三维图形库进行三维实体造型[J]. 微型电脑应用, 1998(6): 93-96.