

Rendering Cloud Scene Based on Fractal Technology

Runchuan Liu

High School Attached to Xi'an University, Xi'an Shaanxi
Email: liurunchuan@yeah.net

Received: Feb. 14th, 2019; accepted: Feb. 25th, 2019; published: Mar. 4th, 2019

Abstract

Fractal geometry can effectively describe natural scenes that Euclidean geometry can't express. In order to improve the efficiency and speed of cloud scene modeling and real-time rendering, a GPU-oriented cloud scene rendering architecture is proposed, and the specific algorithm of cloud scene modeling and rendering is given. Combining different shapes of clouds and using different geometric models to render, the realistic visual effect of cloud scenes is greatly improved. The simulation results show that the proposed algorithm achieves a smooth, fast and real-time screen display, and can meet the dynamic requirements of sky background in game scenes.

Keywords

Fractal Geometry, Cloud Scene, Rendering, Reality

基于分形技术的云场景设计与实现

刘润川

西安电子科技大学附中, 陕西 西安
Email: liurunchuan@yeah.net

收稿日期: 2019年2月14日; 录用日期: 2019年2月25日; 发布日期: 2019年3月4日

摘要

分形几何可以有效地描述欧氏几何无法描述的自然景物。本文研究了基于Diamond-Square的分形算法与云场景渲染模式。为了改善云场景建模与实时渲染的效果和速度,提出了一种面向GPU的云场景渲染架构,给出了云场景建模与渲染具体过程。结合云朵不同形态,采用不同的模型渲染,较大地提升了云场景的真实感视觉效果。仿真实验表明,本文云场景建模与渲染算法具有画面显示流畅、快速和逼真特点,能够满足游戏场景中实时渲染天空背景的需求。

关键词

分形几何, 云场景, 渲染, 真实感

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

云彩是众多自然景观的重要背景之一, 建筑物配景、园林设计、生态环境仿真、动画制作中都需要云场景。因此, 计算机对云彩有效、快捷地模拟具有重要的研究意义[1] [2]。然而, 云彩和许多自然景物一样, 形态各异, 结构不规则, 欧氏几何无法描述。分形几何的引入, 为描述自然景物开辟了新途径。分形几何从自然景物的不规则性入手, 在不规则的结构中探寻其相似性。这深深地吸引了许多数学科学家对自然景物的几何特性作深入研究, 提出了描述自然景物的许多分形算法。经典分形算法有递归法、L 系统、IFS 法和 Diamond-Square 方法。递归法是利用自然景物的局部与整体的相似性、反复调用描述其局部算法的过程。递归法生成的自然景物比较生硬, 单一且呆板[3]; L 系统是某种符号的重写过程, 以初始符号为基础, 根据一定生成规则不断迭代。L 系统生成的自然景物缺乏骨质感, 纹理信息不足[4]。IFS 法利用仿射变换, 对局部特性进行放大、缩小、旋转或平移, 最终产生一个局部复制品。原则上说, 任何图形都可以用一组仿射变换生成, 火焰算法是 IFS 方法的一个杰出创新。IFS 法描述的自然景物形态差异小, 具有千篇一律性[5]。Diamond-Square 算法描述的自然景物逼真度高, 编程实现比较容易。

本文用 Diamond-Square 算法[6]建立云场景模型, 即生成一个矩形域上的高度数据组。该模型表现为真实感的云场景, 还需要 OpenGL 场景模式渲染。为此, 本文设计了一种面向 GPU 的渲染架构, 通过用户交互指定参数, 可以生成颜色不同、多种形态的云场景。实验表明, 本系统在交互操作下可以获得令人满意的云场景效果。

2. 云场景生成算法

2.1. 云场景建模

Diamond-Square 算法又称正方形 - 正方形细分法, 可以用来生成云彩、山脉等自然景物。算法描述如下:

(a) 在一平面上划分为 $n \times n$ 正方形网格, 四个角点的高度(已知)用 A, B, C, D 表示, 图 1 划分为 4×4 的正方形网格。

(b) 计算正方形对角线交点 M 的高度值: 取 4 个角点的高度平均值, 作为中点 M 的高度。

(c) 计算正方形每条边中点的高度: 根据 A, B, M 点和正方形外一点(高度值任意), 取平均得到 AB 边中点 E 的高度; 根据 B, C, M 点和正方形外一点(高度任意), 取平均得到 BC 边中点 F 的高度; 根据 C, D, M 点和正方形外一点(高度值任意), 取平均得到 CD 边中点 G 的高度; 根据 A, D, M 点正方形外一点(高度值任意), 取平均得到 AD 边中点 H 的高度。

(d) 将 $n \times n$ 的正方形细分为四个 $n/2 \times n/2$ 小正方形, 重复(b)、(c)步骤。

(e) 递归步骤(d), 使其正方形网格不断细化, 直至达到递归深度。

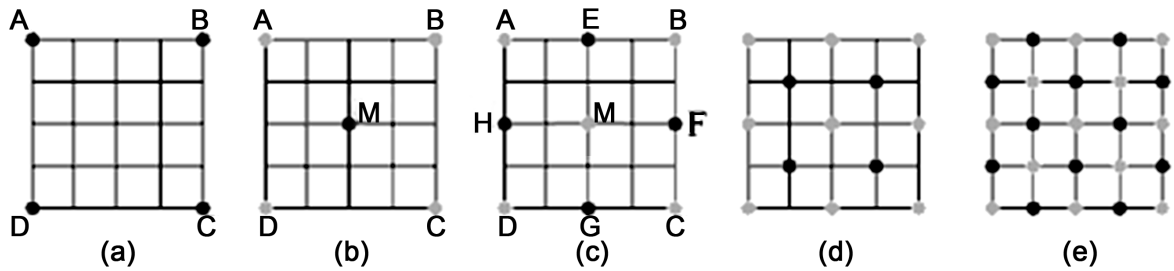


Figure 1. Diamond-Square algorithm

图 1. Diamond-Square 算法

Diamond-Square 算法生成矩形域上的高度数据，存放在一高维数组内，这些数据并不能表示云彩。要把高度值转换为颜色值，如用最小数值代表最蓝，最大值代表最白，才能显示出云彩图像(图 2)，从图 2 可以看出，直接映射生成的云彩图像与自然界中真实的云彩差别很大。

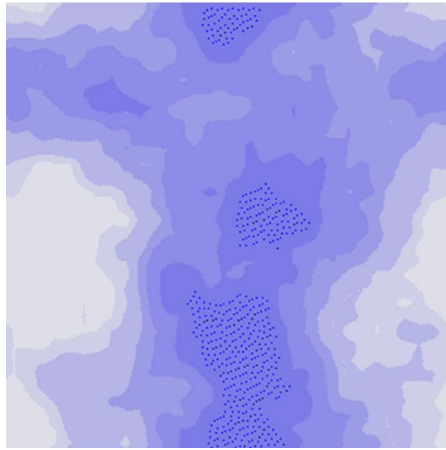


Figure 2. Cloud model based on Diamond-Square algorithm

图 2. Diamond-Square 算法建立的云彩模型

2.2. 云场景生成

上述模型还需要天空建模、纹理映射和雾化等操作才能得到真实感的云场景，这些操作用 OpenGL 编程实现。

2.2.1. 天空模型

自然界的天空看上去是一个苍穹，像个半球笼罩在地面上。用数学中的球面方程得到球上各个顶点坐标并存储于 Vertices [i]数组。通过 OpenGL 的三角形条带(GL_TRIANGLE_STRIP)方式，下面代码可以建立天空模型[7]。

```
glBegin(GL_TRIANGLE_STRIP);
for(i=0;i<NumVertices; i++)
{
if(Vertices[i].flag==1) //最后一圈顶点
glColor3f(0.95f,0.95f,1.0f);
else
glColor3f(0.2f,0.5f,1.0f);
```

```

    glVertex2f(Vertices[i].u, Vertices[i].v); //u, v 纹理坐标
    glVertex3f(Vertices[i].x, Vertices[i].y, Vertices[i].z); // x,y,z 球面图形坐标
}
glEnd();
//闭合底部
glBegin(GL_POLYGON);
for(i=0;i<NumVertices;i++)
{ If (Vertices[i].flag==1)
    {   glVertex2f(Vertices[i].u, Vertices[i].v);
        glVertex3f(Vertices[i].x, Vertices[i].y, Vertices[i].z);
    }
}
glEnd();

```

2.2.2. 云彩纹理映射

纹理映射过程是把图片(或者说纹理)映射到 3D 模型的一个或多个面上, 纹理可以是任何图片。使用纹理映射可以增加云场景的真实感。在 2.1 节, 用 Diamond-Square 算法已生成一个云彩模型, 现在, 通过使用 OpenGL 的纹理映射功能将矩形域上的高度数据映射为真实感的云彩, 使云彩的视觉效果更加真实。设 Cloud 为纹理图像的指针, 用下面的语句可以完成纹理映射过程。

```

//启用 2D 纹理
glEnable (GL_TEXTURE_2D);
m_textureOne =newCTargaImage;
//加载纹理图像
if(!m_textureOne->Load ("Cloud.tga")) returnfalse;
//创建纹理对象
glGenTextures (1, &m_textureObjectOne);
//绑定纹理对象
glBindTexture (GL_TEXTURE_2D, mtextureObjectOne);
//设定缩放器的过滤参数
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
glTexParameteri (GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
//为纹理对象指定纹理图像数据
glTexImage2D (GL_TEXTURE_2D, 0, GL_RGB, m_textureOne->GetWidth(),0, GL_RGB,
GL_UNSIGNED_BYTE, m_textureOne->GetImage());

```

2.2.3. 雾化

雾化可以使云场景更为真实, 远处的云彩产生朦胧的效果, 从而使云彩看起来更加自然。在 OpenGL 中, 雾化处理需要指定雾的属性, 包括雾的颜色、强度、远近、雾化效果等, 这些可以通过系列雾化函数 `glFog*()` 来设置。本文要实现具有深度感的雾化效果, 将雾化效果模式默认设置为 `GL_LINEAR`, 雾的颜色设置为白色。

```
GLfloat fog_color[4]={1.0f, 1.0f, 1.0f, 1.0f}; //雾色为白色
```

```

glEnable(GL_FOG);//启动雾化效果
glFogf( GL_FOG_MODE, GL_LINEAR);//设置雾为深度感效果
glFogf(GL_FOG_DENSITY, 0.6f);//设置雾浓度
glFogf( GL_FOG_COLOR, fog_color); //设置雾色
glFogf( GL_FOG_START, 10.0); //设置雾化效果起始距离
glFogf( GL_FOG_END, 20.0); //设置雾化效果结束距离

```

2.2.4. 光照模型设定

自然界中，云彩效果和光照强度、光源位置和材质的关系非常密切，光照强度可以改变云彩的温度，影响云彩的颜色和密度。环境光的特点是没有方向性，可以照射到全部空间；散射光一般来源于某个特定方向，从物体表面向各个方向反射，与太阳光比较接近，本文采用散射光来代替太阳光。光照效果主要有以下几个因素决定：

➤ 光源种类

环境光设置：

```

GLfloat ambientLight[]={0.8f, 0.3f, 0.3f, 1.0f}; //光的颜色为偏红色
glLightfv(LIGHT0, GL_AMBIENT, ambientLight);

```

其中，LIGHT0 是 OpenGL 提供的八种光源之一，默认情况下提供温和的白光；GL_AMBIENT 是环境光强度；如果模拟一个晴朗的天空和白云，需要将 ambientLight[] 设为 {1.0f, 1.0f, 1.0f, 1.0f}；如果模拟晚霞，则将环境光设置为偏红，ambientLight[] 可以为 {0.8f, 0.3f, 0.3f, 1.0f} 或相近数值。

散射光设置：

```

GLfloat diffuseLight[]={1.0f, 1.0f, 1.0f, 1.0f};
glLightfv(LIGHT0, GL_DIFFUSE, diffuseLight);

```

其中 GL_DIFFUSE 是散射光的强度；本文用散射光代替太阳光，diffuseLight[] 始终取为 {1.0f, 1.0f, 1.0f, 1.0f}。

➤ 光源位置

光源的位置和发光方向由下面两行代码设定：

```

GLfloat lightPosition[]={0.0f, 0.0f, 1.0f, 0.0f};
glLightfv(LIGHT0, GL_POSITION, lightPosition);

```

其中 GL_POSITION 表示光源位置，由四个值(X, Y, Z, W)表示。如果第四个值 W 为零，则表示该光源位于无限远处，前三个值表示了它所在的方向。这种光源称为方向性光源，通常，太阳可以近似的被认为是方向性光源。如果第四个值 W 不为零，则 X/W, Y/W, Z/W 表示了光源的位置。这种光源称为位置性光源。文中选取 lightPosition[]={0.0f, 1.0f, 0.0f, 0.0f}，即定义了一个来自于 y 轴正方向的方向性光源。

➤ 材质的选定

为云彩模型的正面设置环境光和散射光的材质属性：

```

glMaterialfv(GL_FRONT, GL_DIFFUSE, matDiff);
glMaterialfv(GL_FRONT, GL_AMBIENT, matAmbient);

```

其中 GL_FRONT 表示云彩模型的正面，matDiff[]={1.0f, 1.0f, 1.0f, 1.0f} 和 matAmbient[]={1.0f, 1.0f, 1.0f, 1.0f} 表示散射光和环境光中的红、绿、蓝分量将全部被表面反射，即反射白光。

2.2.5. 三维图形变换

现实中风场的变化会影响云团走向及形状，云团形状时刻变化，有时分裂有时聚集。还有许多其它

因素影响云彩的形态，比如云团内部微粒间的相互作用。本文只考虑外界因素的影响，将云团运动看作是整体运动，通过系列三维图形变换(平移、旋转、缩放和投影)，模拟云彩动态效果。调用函数 `glTranslatef(x, y, z)`，其中 x 、 y 和 z 分别表示沿坐标轴移动的数值。本文取 x 轴方向平移值为 $-1 \sim 1$ 之间， y 和 z 轴方向平移值均为 0。用表达式 $x = x \pm 0.0001$ 控制云彩沿 x 轴方向移动的速度，从而获得到云彩水平方向移动的动态效果。

3. 实验

算法使用 OpenGL 编程实现，在操作系统为 Windows XP 的 PC 机上进行了测试实验。交互界面控制参数见表 1，如 F1 按键控制环境光开启，云彩颜色通过 F2 按键调节。图 3(a)为该系统控制参数默认情况下生成的云场景效果；图 3(b)为增加云朵数目的云场景；图 3(c)为调整雾浓度和雾化距离两参数生成的云场景；图 3(d)为调整光源环境光参数的云场景。测试结果表明，文中算法可以实现云场景真实感绘制，绘制算法是有效的，绘制过程可以在普通微机平台上流畅运行。

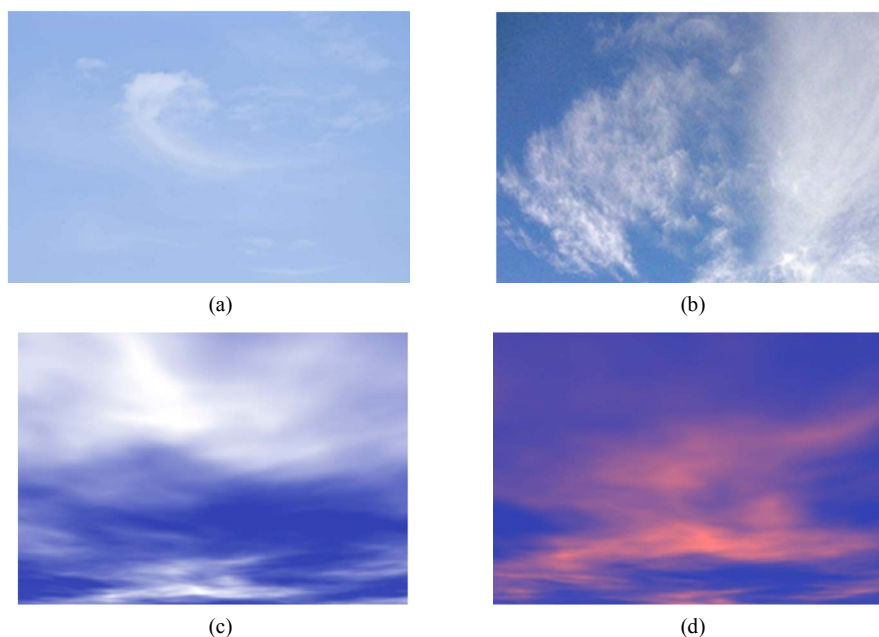


Figure 3. Cloud scene rendering: (a) Cloud scene in initial state; (b) Cloud scene in clear day; (c) Cloudy sky scene; (d) Sunset cloud scene

图 3. 云场景效果图: (a)初始状态的云场景; (b)晴天云场景; (c)阴天云场景; (d)晚霞云场景

Table 1. Control parameters of the program

表 1. 程序控制参数

控键	操作	控键	操作
UP	雾化距离增大	F1 + UP	光源环境光红色量增强
DOWN	雾化距离减小	F1 + DOWN	光源环境光红色量减弱
LEFT	云彩向左移动	F2	云朵颜色控制
RIGHT	云彩向右移动	F2 + UP	云朵个数增多
F1	环境光开关	F2 + DOWN	云朵个数减小

4. 结论

本文用分形算法 Diamond-Square 生成云场景模型, 结合 OpenGL 技术实现对云彩的真实感绘制。这种方法克服了传统方法绘制云彩的生硬、逼真度不足等问题。通过设置模型数组参数、云场景建模、定义纹理、纹理映射, 雾化和光照模型设置等过程, 最终生成具有真实感的云场景。分形算法的优点在于能够很好地描述自然景物的自相似性, 此文将分形方法用于云彩绘制研究, 云场景表述取得了较好的视觉效果。将分形算法与 OpenGL 技术结合, 更好地仿真了自然环境中的天空, 使其更加生动自然逼真。

参考文献

- [1] 邱航, 杨珂, 陈瑜, 陈雷霆. 云的真实感模拟技术综述[J]. 计算机科学, 2011, 38(6): 14-19.
- [2] 杨勋才, 余贞侠, 谢依妨. 真实感云的快速渲染算法的研究与实现[J]. 成都信息工程学院学报, 2014, 29(4): 353-359.
- [3] 邵绪强, 杨晓丹, 李继荣, 阎蕾. 大规模流体场景的真实感与实时模拟[J]. 中国图象图形学报, 2016, 21(7): 913-922.
- [4] 邵绪强, 宋雨. 固流交互中真实感溶化现象的实时模拟[J]. 系统仿真学报, 2017, 29(3): 502-508.
- [5] 蒋秉川, 万刚, 范承啸, 夏青. 战场环境体素模型适应性分析及体要素分类方法研究[J]. 系统仿真学报, 2017, 29(2): 241-247.
- [6] 任民宏. 基于 OpenGL 和分形技术的真实感云彩生成[J]. 微计算机信息, 2007, 23(12): 292-294.
- [7] Revive and Strive. OpenGL 中实现天空顶(SkyDome)的类: CSkyDome [EB/OL]. <https://www.cnblogs.com/donati1981/archive/2009/08/31/1557229.html>, 2018-09-01.

知网检索的两种方式:

1. 打开知网页面 <http://kns.cnki.net/kns/brief/result.aspx?dbPrefix=WWJD>
下拉列表框选择: [ISSN], 输入期刊 ISSN: 2161-8801, 即可查询
2. 打开知网首页 <http://cnki.net/>
左侧“国际文献总库”进入, 输入文章标题, 即可查询

投稿请点击: <http://www.hanspub.org/Submission.aspx>
期刊邮箱: csa@hanspub.org