

Methods of Fractal Algorithm for Simulating the Vegetation in the Virtual Scene

JI Feng-quan¹, SH I Gui-gang²

1. Ministry of Education Key Laboratory of Integrated Regulation and Resource Development on Shallow Lakes, Hohai University, Nanjing, China;

2. Department of Surveying and Geo-Informatics, Tongji University, Shanghai, China

1. e-mail: finela@126.com, 2. e-mail: 21533758@qq.com

Abstract: To simulate the complex vegetation is the difficult field in computer graphics, and the fractal theory is adapted to resolve the simulation complex-shaped objects. Based on the principle of fractal geometry and algorithm of fractal sets, mapping the complex landscape is realized so as to effectively enhance the authenticity and vividness of the virtual scene. The L system and the random iteration algorithm have been implemented to generate the trees and plants; and VC++ programming is employed to provide examples of trees and flora scenes in the virtual scenes. It's shown by experimental results and application examples that trees and flora scenes created by the algorithm are vivid and high efficient. And it becomes more convenient to simulate complex large-scale landscape.

Keywords: Fractal; VR; Vegetation; Virtual Scene

虚拟场景中植物分形算法的实现

冀凤全¹, 施贵刚²

1. 河海大学浅水湖泊综合治理与资源开发教育部重点实验室, 南京, 江苏, 210098

2. 同济大学测量与国土信息工程系, 上海, 200092

1. e-mail: finela@126.com, 2. e-mail: 21533758@qq.com

【摘要】复杂景观中的植物模拟是计算机图形学研究的难点领域, 而分形理论适于解决复杂形态物体的模拟问题。根据分形几何原理和分形集生成的算法原理, 实现复杂景观的绘制, 有效增强了虚拟场景的真实性和逼真性。主要运用 L 系统和随机迭代算法生成树木和植物, 并利用 VC++ 编程, 给出了树木和植物群场景的例子。实验及应用实例表明该算法生成的树木和植物群场景形象生动、效率高, 可使大面积模拟复杂景观变得容易。

【关键词】分形; VR; 植物; 虚拟场景

1 引言

植物是虚拟场景中的重要组成部分, 复杂的植物形态给植物的模拟带来极大的困难。随着 GIS 技术和虚拟现实理论的研究在不断的深入发展, 从而要求三维环境中的地物地貌能得到准确的显示和模拟。对于三维场景中房屋、道路等的模拟已能十分形象, 但对如树木、花草、灌木盆景、路灯等复杂景观中的地物模拟难度很大, 已成为研究的热点^{[1][2]}。现在主要使用的方法有模型法、贴图法、分形法, 模型法和贴图法对于复杂的植物描述难以达到理想效果, 而分形理论能很好的解决复杂形态物体的模拟问题^[3]。分形法是通过分数维来描述地物现象随观测尺度变化的宏观结构动态变化和地物本身细节随观测尺度变化的微结构动态概括^[4]。在分形科学发展的初期, 因为地球的表面形状非常复杂, 研究者指出用一般的代数曲面来

描述往往是不合适的, 而用某些定义于二维空间的分形随机过程则能够取得相当满意的效果, 这就说明实际的景观具有分形性质, 从而导出了分形在地球科学中的应用^[5]。分形图形的生成是分形景观生成的核心, 在分形图形基础上, 利用图像复合, 纹理映射等技术, 在图形上加上植被等, 进而生成逼真的模拟景观。本文将力图在 VC++6.0 环境下对分形法绘制地物作详细的介绍。

2 分形和分数维

用几何定义自然界中的许多现象(植物)都是不规则的, 但它们在不同的尺度上都有自相似的形态和相同的复杂性。美国科学家曼德勃曼(B. B. Mandelborn, 1977)首先意识到自相似是宇宙中普遍现象, 提出了分形(fractal)的概念, 并建立了分形几何学。分

形几何是用来描述这样的体系：当扩大观察范围时，增加的细节是重复出现的，这些新的在更大尺度下重复出现的结构与在小范围内观察到的结果是相似的。

一般维数是整数，直线是一维的，平面是二维的，空间是三维的，相对论中时空是四维的，数学上存在 n 维空间 (n 为任意自然数)。一个几何对象 (线、面、体等) 的维数是由表示它的一个点所需的独立变量个数所确定的，这里的维数只能是整数。其实，一个几何对象的维数还可以从测量它的角度来定义。根据欧氏空间维数的一般定义可知维数与测量有关。测量就是用选作单位的标准体去覆盖被测对象，所需标准体的最少数目就是测量结果。因此可设几何对象 S ，以直径 R 的小球为标准体，覆盖被测对象 S 所需小球的最少数目为 N ，随着 R 的减少， N 将增大。如果有极限存在，那么定义 $D = \lim_{R \rightarrow 0} (\ln N / \ln(1/R))$ ，分形理论认为维数是连续的，可以是有理数也可以是无理数，即求出的通常不只是整数，也可是分数，因此称 D 为分数维。如 koch 雪花曲线的分数维 $D = \ln 4 / \ln 3 = 1.2618$ 他的另一种形式为 $N = (1/R)^D$ 。分数维是描述这种复杂自相似结构的最好概念。分数维刻划了分形结构的重要几何特征，可以定量地描述事物内部结构的复杂性。这里我们主要研究植物的模拟。

3 分形法绘图

3.1 最简单的 L 系统 (DOL 系统)

DOL 系统是最简单的 L 系统，也称“确定性的上下文无关的系统”，L 系统基于“符号重写系统”，即使用一组重写规则逐次地替换一个简单的初始对象的各个部分来确定一个复杂的对象，其中初始对象为一个字符串或一个字母，可以按一组生成规则来取代字符串中每一个字母。其基本原理详细见文献[4]。它的重要应用是对植物结构的模拟，设定好初始的参数，用 VC++ 描述的核心程序为：

```
Void CPlant::Drawplant()
{
    定义变量;
    for(条件){
        switch(c){
            case '+':op.alpha=op.alpha+delta;break;
            case '-':op.alpha=op.alpha-delta;break;
            case 'F':np.x=op.x+len*cos(op.alpha*PI/180.0);
                    np.y=op.y+len*sin(op.alpha*PI/180.0);
                    dx=(int)np.x;dy=(int)(maxy-np.y);lie
                    to(dx,dy);op.x=np.x;op.y=np.y;break;
            case '[':push();break;//参数进栈
            case ']':pop();//参数出栈
```

```
moveto((int)op.x,(int)(maxy-op.y));
break;
```

```
default: break;}}}
```

对于不同的初值的设计利用上面的程序可得到不同的植物。如设置初值：演变规则的个数为 5，角度增量为 22.5 度，生成规则为 F, F→F[+F]F[-F]F 可得带芽的植物；也可设置初值：演变规则的个数为 6，角度增量为 25.7 度，生成规则为 Y, Y→YFX[+Y][-Y], X→X[-FFF]F[+FFF]FX 可得开花的树等。

3.2 随机迭代算法绘图

由 IFS 的定理 1^[4]，设 $\{X; w_1, w_2, \dots, w_n\}$ 是一个 (双曲) IFS，其中每一个 w_i 具有一个概率 $p_i > 0$ ，且 $p_1 + p_2 + \dots + p_n = 1$ 。选取任一点 $x_0 \in X$ 为初始点，然后递归地随机选取下述集合中一个点作为 $x_n, n=1, 2, \dots$ ，于是有 $x_n \in \{w_1(x_{n-1}), w_2(x_{n-1}), \dots, w_n(x_{n-1})\}$ ，最终得到序列 $\{x_n\} \in X$ ，收敛于 IFS 的吸引子。在迭代过程中，落实到实际的程序算法时，应首先确定好最大迭代次数、仿射变换个数、存放 w_i 的仿射变换系数及其相应概率的一维数组，绘图起始位置及坐标方向的比例因子。当视窗选择得太大时，可以增加迭代次数，以提高图象质量。

其中利用 VC++ 实现的程序核心部分如下：

① 根据上面的原理易得树叶的模拟程序：

```
for(i=0;i<itn;i++)
{rn=rand()/32767.0;total=p[0];k=0;
while(total<rn){k++;total+=p[k];}
newx=a[k]*x+b[k]*y+e[k];newy=c[k]*x+d[k]*
y+f[k]; x=newx;y=newy;
if(I>count){dx=(int)(x+sx*x);dy=(int)(y+
sy*y;
putpixel(dx,maxy-dy,color)}}}
```

② 由上面树叶的算法和程序，再对坐标轴作适当的旋转和平移，代入确定的参数即可由 VC++ 生成如图 1 所示的图形。



Figure 1: simulated trees by random iteration algorithm in VC++6.0 environment

图 1 随机迭代算法 VC++6.0 环境下的模拟树木

③ 带凝聚的 IFS 模拟植物群场景

根据带凝聚的迭代函数系统的定义和定理^[4], 设 (X, d) 是一个距离空间, 给定 $C \in \mathcal{X}(X)$, 如果一个变换 $w_0: \mathcal{X}(X) \rightarrow \mathcal{X}(X)$, 使 $w_0(B)=C, \forall B \in \mathcal{X}(X)$, 那么称 w_0 为凝聚变换, 而 C 称为相伴凝聚集。假定 $\{X, w_1, w_2, \dots, w_N\}$ 是一个 (双曲) IFS, 具有压缩因子 $s = \max\{s_i: i=1, 2, \dots, N\}, 0 \leq s < 1$. 令 $w_0: \mathcal{X}(X) \rightarrow \mathcal{X}(X)$, 是一个凝聚变换, 则 $\{X, w_1, w_2, \dots, w_N\}$ 称为带凝聚的 (双曲) IFS, 压缩因子为 s . 由此可定义变换 $W: \mathcal{X}(X) \rightarrow \mathcal{X}(X), W(B) = \cup_{n=0}^N w_n(B) \forall B \in \mathcal{X}(X)$ 是完备距离空间 $(\mathcal{X}(X), h(d))$ 上的一个压缩映射, 即

$$h(W(B), W(C)) \leq s h(B, C) \quad \forall B, C \in \mathcal{X}(X) \quad (1)$$

其中 $0 \leq s < 1$ 且具有唯一的不动点 $A \in W(A) = \cup_{n=0}^N w_n(A)$. 对任一个 $B \in \mathcal{X}(X)$, 可通过迭代求得吸引子 A :

$$A = \lim_{n \rightarrow \infty} W^{(n)}(B) \quad \forall B \in \mathcal{X}(X) \quad (2)$$

如果定义一个完备距离空间 (X, d) , 给定 $L \in \mathcal{X}(X)$, 令 $\epsilon > 0$, 选择一个 (带凝聚的) IFS $\{X; (w_0, w_1, w_2, \dots, w_N)\}$, 具有压缩因子 $s: 0 \leq s < 1$, 使 $h(L, \cup_{n=0}^N w_n(L)) \leq \epsilon$, 其中 $h(d)$ 是 Hausdorff 距离, 则 $h(L, A) \leq \epsilon / (1-s)$, A 是 IFS 的吸引子。也等价于 $h(L, A) = (1-s)^{-1} h(L, \cup_{n=0}^N w_n(L)) \quad \forall L \in \mathcal{X}(X)$.

这样就提供了一个解决逆问题的方法, 即对给定的集合 L , 找一个 IFS, 使 L 为其吸引子。现在可设 $(R^2, Euclidean)$ 是距离空间, C 表示一个集合, 可以想象 C 为一个枯萎的松树站在原点上, 它的树干垂直于 x 轴。令 $w_1(x)=0.75x+0.25, w_1(y)=0.75y$, 凝聚变换 w_0 满足 $w_0(B)=C, \forall B \in \mathcal{X}(X)$, 要求 IFS $\{R^2, w_0, w_1\}$ 吸引子, 先按常规定义 W , 易知凝聚集 C 在 W 映射下的像是:

$$W(C) = w_0(C) \cup w_1(C) = C \cup w_1(C) \quad (3)$$

即由凝聚集 C 和 $w_1(C)$ 组成, C 是原枯萎的松树, $w_1(C)$ 是压缩以后的松树, 再经一次迭代得:

$$W^{(2)}(C) = W(w_0(C) \cup w_1(C)) = W(C \cup w_1(C)) = W(C) \cup w_0 \circ w_1(C) \cup w_1 \circ w_1(C) = C \cup w_1(C) \cup w_1 \circ w_1(C) \quad (4)$$

由三棵松树组成的集合。因此, 当 $n \rightarrow \infty$ 时, $W^{(n)}(C)$ 趋向于一排小松树, 这里凝聚集实际是由另一组防射变换生成的。根据上面的原理, 我们来确定中间的一棵树, 通过 2 个仿射变换迭代来生成植物群场景, 图 2 是 VC++ 编程实现的植物群场景的一帧。

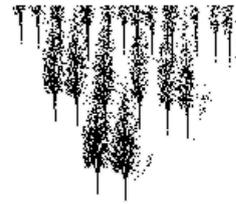


Figure 2: the simulated flora scenes by IFS with coherent transformations in VC++6.0 environment

图 2 VC++6.0 环境下带凝聚的 IFS 模拟的植物群场景

由图可知由此法生成的植物群场景形象生动, 直观简洁, 同时可以局部放大而不舍原小区域的形态, 即与分辨率无关, 可以解码到具有比原图象高的分辨率的输出设备上。由此可知非常简单的机制便能生成一幅复杂图象, 因此它的效率很高, 在虚拟场景中可以得到很好的利用。

4 结束语

根据以上图形的绘制, 再利用 VC++ 结合 OpenGL 来渲染虚拟环境, 就能得到很好的虚拟场景效果, 可大大加强其真实性、可观性。又 VC++ 开发应用程序的运行效率很高, 因此采用本文提出的方法或算法可使大面积模拟植物变得容易, 更好地满足实际需要。

References(参考文献)

- [1] DING Sheng-yan, TIAN Li, ZHAN Peng-fei, et al. Review on Researches of Dummy Landscape Ecology Environment Fabrication Based on L-systems[J]. *Journal of Henan University(Natural Science)*, 2008,38 (1) :59-64(Ch).
丁圣彦, 田莉, 展鹏飞, 等. 基于 L 系统的虚拟景观生态环境构建研究综述 [J]. 河南大学学报 (自然科学版), 2008,38(1):59-64.
- [2] GUO Run-qiu, Zhou lei, Lin Xiao-chun, et al. Simulation of Natural Landscape Based on Fractal Brownian Motion[J]. *Computer Simulation*, 2005,22 (12) :155-157(Ch).
过润秋, 周蕾, 林晓春, 等. 基于分形法在计算机上仿真自然景观方法的研究 [J]. 计算机仿真, 2005,22(12):155-157.
- [3] RUI Xiao-ping, YANG Chong-jun, CHENG Qi-min. A Simple Drawing Method for Tree in 3D Scene[J]. *Bulletin of Surveying and Mapping*, 2002(3)61-62(Ch).
芮小平, 杨崇俊, 程起敏. 3 维场景中的树的一种简易绘制方法 [J]. 测绘通报, 2002(3)61-62.
- [4] JIN Yi-wen, LU Shi-jie. Principle and Application of Fractal Geometry [M]. Hangzhou: Zhejiang University Press, 1998(Ch).
金以文, 鲁世杰. 分形几何原理及其应用 [M]. 杭州: 浙江大学出版社, 1998.
- [5] GUO Ren-zhong. Spacial Analysis. Beijing [M]. Beijing: Higher Education Press, 2001(Ch).
郭仁忠. 空间分析 [M]. 北京: 高等教育出版社, 2001.