

三维植物的计算机模拟算法研究

张明书¹, 张 燕¹, 王维民¹, 梁立凯²

(1. 辽宁石油化工大学 信息工程学院, 辽宁 抚顺 113001;
2. 山东大学 威海分校, 山东 威海 264209)

摘要:针对 L 系统的数学模型和构图原理提出对具有自相似结构的植物在三维空间内进行计算机模拟的算法研究。将 L 系统对植物的模拟的应用从二维空间转变到三维空间, 定义 3 个空间坐标轴上的余弦值 X, Y, Z 与 3 个给定的 3 个坐标轴上的旋转矩阵的乘积作为植物在 3 个坐标方向上旋转参数, 使绘制出的植物产生明显的三维效果。并且以 VC++6.0 作为运行环境, 结合 OpenGL 函数库提供的强大绘图功能, 对算法进行计算机上的实现。探讨了通过增加植物的绘制次数, 改变观察者视点的位置, 实现对树林的模拟。通过引入随机数产生器使行、列间距离产生随机变化, 同时也能够随机调用不同的字符串复写规则使树林中产生不同类型、颜色、大小的树木, 提高了对自然景物模拟的逼真度。

关键词:分形; L 系统; 三维; 植物; 计算机模拟

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2006)10-0108-03

Computer Simulation Algorithm Research of Three-Dimensional Plant

ZHANG Ming-shu¹, ZHANG Yan¹, WANG Wei-min¹, LIANG Li-kai²

(1. Sch. of Info. & Eng., Liaoning University of Petroleum & Chemical Techn., Fushun 113001, China;
2. Weihai Branch, Shandong University, Weihai 264209, China)

Abstract: Based on the model of mathematics of L system and composition principle, put forward computer simulation algorithm investigation about the plant of the similar structure in three-dimensional space. Change the application of simulation of the plant of L system from two-dimensional space to three-dimensional space. Define product of the cosine value on three space coordinate axes X, Y, Z and three given definitely rotation matrix in the respective coordinate axis as the rotation parameter of plant in the three coordinate axes, and make the plant drawn out to produce the obvious three-dimensional result. And with VC++6.0 combines the strong drawing function offered in OpenGL function storehouse on this basis as the running environment, carry on the realization on the computer to the algorithm. Probe into through increasing the number of times of drawing the plant, change the position of observer's view point, realize the simulation of the woods. Finally through lead into random number generator make row, rank rooms of distance produce at random changing, can transfer different character clusters of facsimile rule make woods produce different kinds of at random at the same time, the color, the trees of the size. The fidelity to natural scene simulation is improved.

Key words: fractal; L-system; three-dimension; plant; computer simulation

0 引言

随着计算机图形学的发展,人们一直探讨在计算机上模拟自然景物的可能性。尤其是近年来,自然景物的计算机模拟一直是计算机图形学中极具挑战性的问题之一。植物树木是比较常见的自然景物,其结构特征很强,形状复杂且长势千姿百态。用计算机模拟生成时,其建模是很困难的。早期的植物模拟多数是基于迭代函数系统进行的。利用迭代系统生成的植物是从几何角度去模拟的,缺乏基于植物生理特征的描述。相比之下,利用植物形态结

构的分形性质(结构自相似性)生成植物图形的 L 系统在植物结构模拟方面是比较成功的。

1 L 系统的数学模型及构图原理

L 系统又称为字符串重写系统或正规文法系统,是由美国植物学家 Lindenmayer 在 1968 年提出的,简称 L 系统^[1,2]。它是用形式语言的方法来描述植物的生长过程,其基本思想可解释为:从一条树枝(种子)开始,发出新的芽枝,而发过芽枝的枝干又都发新芽枝……最后长出叶子^[3]。

1.1 字符串替换

L 系统的核心思想是“字符串替换”机制。字符串替换可以定义为根据一组改写规则或产生式(production)依次替换一个简单初始物的每一部分^[4~6]。

收稿日期:2005-12-19

基金项目:辽宁省自然科学基金资助项目(20022161)

作者简介:张明书(1980-),男,辽宁抚顺人,硕士研究生,研究方向为计算机图形学;张 燕,博士,教授,研究方向为计算机图形学、计算机辅助设计。

例如: ω :ab(初始物); $P1:a \rightarrow ab$ (产生式 1); $P2:b \rightarrow a$ (产生式 2)。根据产生式去替换初始字符串中的每一个字符,得到如下推导序列: $ab \rightarrow aba \rightarrow abaab \rightarrow abaababa \rightarrow \dots$ 。

1.2 龟解释模型

龟解释模型的主要思想是:设 (x, y) 为龟所在位置的坐标, a 为头指向的角度, (x_0, y_0, a_0) 是龟的初始状态, d 是前进增量, δ 是角度增量参数, 则龟按以下命令动作:

F : 向前移动一步, 步长为 d , 龟状态为 (x', y', a) ; $x' = x + d * \cos a$; $y' = y + d * \sin a$; 从 (x, y) 到 (x', y') 画一条直线;

$+$: 向左旋转 δ 角, 则龟下一个状态为 $(x, y, a + \delta)$, 规定正角是逆时针方向, 负角是顺时针方向;

$-$: 向右旋转 δ 角, 则龟下一个状态为 $(x, y, a - \delta)$, 规定和 $+$ 相同;

经过这样反复重写, 最后可以得到分形图。

2 基于 L 系统的三维植物计算机模拟及实现

2.1 模拟三维植物的算法

为了使图形更加形象逼真, 可以将二维 L 系统推广到三维 L 系统, 用 3 个矢量 X 表示向前, Z 表示向右, Y 表示向上, 指示龟的朝向。这些矢量具有单位长度, 互相正交, 并且满足方程 $H * L = U$ 。龟的旋转可以表示为:

$$[X' Y' Z'] = [XYZ] * R$$

其中, R 为 $3 * 3$ 的旋转矩阵, 绕 X 逆时针方向旋转 θ 角度的旋转矩阵用 $R_X(\theta)$ 表示, 绕 Y 及绕 Z 的旋转矩阵分别用 $R_Y(\theta)$ 和 $R_Z(\theta)$ 表示, 分别有如下形式:

$$R_X(\theta) = \begin{bmatrix} c & s & 0 \\ -s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}, R_Y(\theta) = \begin{bmatrix} c & 0 & -s \\ 0 & 1 & 0 \\ s & 0 & c \end{bmatrix},$$

$$R_Z(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix} \quad (1)$$

其中, $c = \cos \theta$, $s = \sin \theta$ [7]。

2.2 模拟三维植物的算法的计算机实现

应用 L 系统的原理在计算机上实现三维植物模拟的步骤如下:

step1 定义绘图函数: $\text{drawTree}(\text{int num}, \text{double len}, \text{int control})$; 和旋转函数: $\text{rot}(\text{int n}, \text{double del}, \text{double * R})$ 。

step2 定义两个字符串型数组: $\text{strStart}, \text{strNew}$; strStart : 存放起始字符串和上一次重写的结果字符串; strNew : 存放每一次重写的结果字符串。定义起点: x_0, y_0, z_0 ; 终点: x_1, y_1, z_1 ; 定义旋转角 θ 和在 3 个坐标轴 X, Y, Z 方向上的初始角度 $\text{angle-}X, \text{angle-}Y, \text{angle-}Z$ 。

step3 根据替换规则按照上文中的字符串替换规则做字符串替换并保存到数组 strNew 中。

step4 实现绘图函数 $\text{drawTree}(\text{int num}, \text{double len},$

$\text{int control})$ 。# num 为迭代次数; len 为所要模拟植物茎的长度; control 为控制变量。

```
for(int k=0; k<strNew.GetLength(); k++)
{
    t=strNew[k];
    switch(t) # 判断字符串数组中每个字符所代表的符号
```

```
    {
        如果是'F'则给定初始的 3 个坐标轴上的方向余弦值  $X=0.0; Y=1.0; Z=0.0$ 。确保要模拟的植物在没有进行坐标变换前始终指向  $Y$  轴方向; 根据  $R\_change$  值调用旋转函数  $\text{rot}(\text{int n}, \text{double del}, \text{double * R})$ ; 计算变化后的 3 个坐标轴上的方向余弦  $X, Y, Z$ ; 然后计算出新位置的坐标  $x_1 = x_0 + \text{len} * X; y_1 = y_0 + \text{len} * Y; z_1 = z_0 + \text{len} * Z$ ; 从  $(x_0, y_0, z_0)$  到  $(x_1, y_1, z_1)$  利用 OPENGL 辅助函数库提供的函数  $\text{auxSolidCylinder}(r, \text{height})$  绘制半径为  $r$ , 长度为  $\text{height}$  的圆柱; 其中  $\text{height}$  为从  $(x_0, y_0, z_0)$  到  $(x_1, y_1, z_1)$  的距离;
```

```
    如果是其它符号则  $+, -$  分别表示绕  $Y$  轴向左、右转  $\theta$  度, 旋转函数控制变量  $R\_change$  赋值为 2;  $\&$ ,  $\sim$  分别表示绕  $X$  轴向下、上转  $\theta$  度, 旋转函数控制变量  $R\_change$  赋值为 1;  $<, >$  分别表示绕  $Z$  轴向内、外转  $\theta$  度, 旋转函数控制变量  $R\_change$  赋值为 2;  $|$  表示绕  $Y$  轴向左转 180 度, 旋转函数控制变量  $R\_change$  赋值为 2;  $[$  表示将当前状态下的所有参数压入栈  $a$ ;  $]$  表示将栈  $a$  最顶端的元素弹出;
```

```
    step 5 实现旋转函数  $\text{rot}(\text{int n}, \text{double del}, \text{double * R})$ ; # 计算经过坐标变换后的  $X, Y, Z$ ;  $n$  为绘图函数中的  $R\_change$  值, 判断旋转时需要调用的旋转矩阵;  $\text{del}$  为传入函数中的当前的角度值;  $R$  是一个指向一维数组的指针。
```

```
{
    double X = *(R + 0); double Y = *(R + 1);
    double Z = *(R + 2);
    # 把当前  $X, Y, Z$  轴上的余弦值赋给变量  $X, Y, Z$ ;
    double x, y, z;
    switch(n)
```

```
{
    case 1: 此时龟的旋转可表示为:  $[x y z] = [XYZ] * R_X(\text{del})$ ,  $R_X(\text{del})$  为绕  $X$  轴逆时方向旋转  $\text{del}$  角度的旋转矩阵
```

```
    case 2: 此时龟的旋转可表示为:  $[x y z] = [XYZ] * R_Y(\text{del})$ ,  $R_Y(\text{del})$  为绕  $Y$  轴逆时方向旋转  $\text{del}$  角度的旋转矩阵
```

```
    case 3: 此时龟的旋转可表示为:  $[x y z] = [XYZ] * R_Z(\text{del})$ ,  $R_Z(\text{del})$  为绕  $Z$  轴逆时方向旋转  $\text{del}$  角度的旋转矩阵
```

```
{
    * (R + 0) = x; * (R + 1) = y; * (R + 2) = z;
    # 将旋转后的方向余弦值重新赋值给数组
```

```
{
    #  $R_X(\text{del}), R_Y(\text{del}), R_Z(\text{del})$  为 3 个  $3 * 3$  的旋转矩阵, 定义见式(1)。
```

step6 利用 OPENGL 提供的绘图函数确定观察者的视点,然后调用 step4 中定义过的绘图函数 drawTrec(int num, double len, int control),根据所给的字符串规则进行植物模拟。

2.3 三维树木的绘制实例

图 1~3 就是利用上述方法在计算机中模拟生成的,其中 num 为字符串复写次数,(1)中的图 1 及(2)中的图 2 显示了随 num 增加植物所呈现的变化。

(1)当字符串复写规则为: $F \rightarrow F[-\& < F][< ++ \& F] || F[-\& > F][+ \& F]$,观察者的视点设置为从正前方观察,同时设置光源为绿光并且从正前方进行光照时所得的图像;图 1 显示了当 num 取值 1,2,3 时所生成的图像。

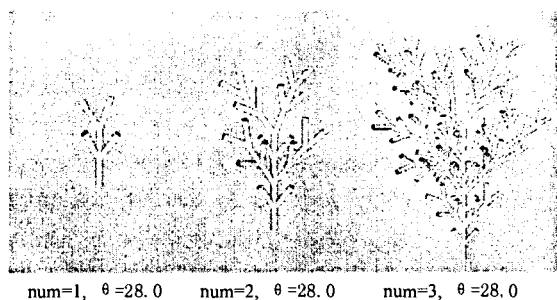


图 1 树木 I

(2) 当字符串复写规则为: $F \rightarrow F[\& + F]F[- > F][- > F][\& F]$,观察者的视点设置为从正后方观察,同时设置光源为绿光并且从正后方进行光照时所得的图像;如图 2 显示了当 num 取值 1,2,3 时生成的图像。



图 2 树木 II

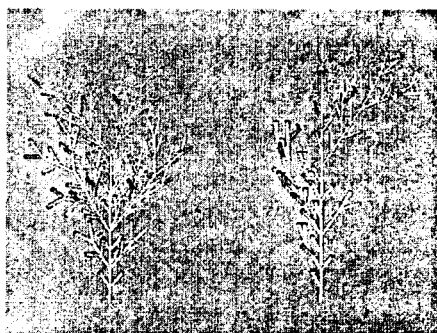
(3) 为了使树木具有多样化针,对图 1,2 中的树木图像做修改,当调用 auxSolidCylinder(r , height)绘图时候使 r 随字符串复写逐渐递减,形成由树木根部到树木顶端逐渐变细的效果,更逼真反映树木的自然形态,图 3 是改进后的树木生成图像(注: r 为圆柱半径)。

3 树林的绘制

在应用中,可以在三维植物基础上对 L 系统进行改进,适当结合 OPENGL 的强大功能实现对树林的模拟。图 4~6 就是在 VC 环境下调用 OPENGL 函数通过改变视点位置及增加植物的绘制次数实现的对树林的模拟。

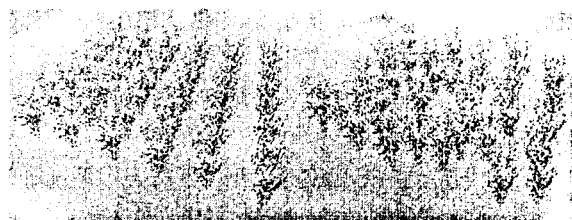
(1)图 4 中所模拟树林中的每棵树木基于 2.3(1)中

图 1 的生成规则,在绘制中可以通过调用双层循环分别控制行数与列数,其中设置树林为六行七列,并且在所模拟的树林引入了随机数产生器,使行与列的间距产生随机变化。使树木的排列呈现规则排列和不规则排列的效果。



$$r = r * 0.93$$

图 3 树木 III

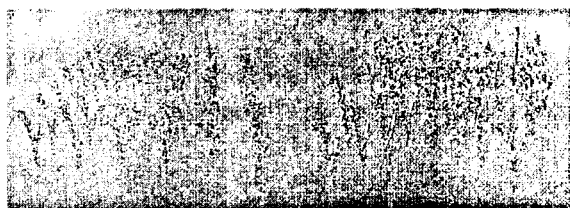


树林->规则排列

树林->不规则排列

图 4 树林 I

(2) 图 5 中所模拟树林中的每棵树木基于 2.3(2)中图 2 的生成规则,与图 4 相似也分别产生规则排列和不规则排列两种效果。

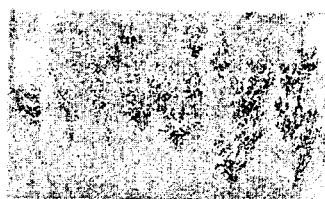


树林->规则排列

树林->规则排列

图 5 树林 II

(3)图 6 中先随机产生 1 和 2 这两个数字,当产生数字 1 时绘制的树木基于 2.3(1)中图 1 的生成规则;当产生数字 2 时绘制的树木基于 2.3(2)中图 2 的生成规则;同时在调用双层循环控制行数和列数时,也设置为六行七列并且引入随机数产生器使行列间隔产生不规则效果,同时在绘制时候利用产生的随机数使每棵树木的高度、颜色



树林->引入随机数产生器

图 6 树林 III

(下转第 117 页)

- [3] 张荣斌. 基于嵌入式 Linux 的实时操作系统研究[D]. 阜新:辽宁工程技术大学,2003.
- [4] 毛德操,胡希明. Linux 内核源代码情景分析(上,下册)[M]. 杭州:浙江工业大学出版社,2001.
- [5] 李善平. Linux 内核 2.4 版源代码分析大全[M]. 北京:机械工业出版社,2002.
- [6] 董晓峰. 嵌入式 Linux 及其调度策略的研究[D]. 西安:西安电子科技大学,2004.
- [7] Dankwardt K. Real Time and Linux[EB/OL]. <http://www.linuxdevices.com/articles/AT5997007602.html>,2005-01.
- [8] Vista M. Linux Real - Time Performance[EB/OL]. <http://www.mvista.com/dswp/realtime.pdf>,2004-12.
- [9] Williams C. Which is better the preempt patch, or the low - latency patch? [EB/OL]. <http://www.linuxdevices.com/articles/AT8906594941.html>,2005-03.
- [10] 陈敏,周兴社. 基于嵌入式 Linux 的实施优化方案[J]. 计算机应用研究,2005(3):235-237.
- [11] 刘彦忠. 嵌入式 Linux 操作系统实时控制的改进[J]. 齐齐哈尔大学学报,2005,21:64-65.
- [12] 中原电子. RTOS 的优先级反转问题[EB/OL]. <http://www.ccelec.com/Docs/RTOS/RTOSPriorityReverse2.htm>,2004-01.
- [13] 钟汉如,王创生. 嵌入式中调度算法的实现及优化[J]. 计算机工程与科学,2002,24(3):5-7.
- [14] 黄廷辉,农毅. 嵌入式的实时性能[J]. 桂林电子工业学院学报,2002,22(1):72-75.
- [15] 李庆诚,顾健. 嵌入式实时操作系统性能测试方法研究[J]. 单片机与嵌入式系统应用,2005(8):19-21.
- [16] Heursch A C, Cramhow D, Horstkotte A, et al. Steps Towards A Fully Preemptable Linux Kernel[J]. Real - time Programming,2003,3(1):17-20.
- [17] Williams C. Linux Scheduler Latency[Z]. Redhat Inc,2002.8-10.

(上接第 113 页)

```
if(isNull(form.area_name,"Area Name is required!"))
    return false; }
</script>
</html:html>
```

4 结束语

Struts 提供了一个非常好的 MVC 框架,它的 taglib 标记库灵活性大,在 Web 应用开发中把界面和业务逻辑进行分离,有利于网页设计者和程序设计者的分工,便于组件式的开发,有助于提高代码的可重用性和灵活性,大大提高了开发效率。同时它把业务逻辑放在一个配置文件中,有利于大型 Web 应用的工程化开发。利用 Action. Servlet 配合 config. xml 实现对整个系统导航,使系统脉络更加清晰,增强了开发人员对系统的整体把握,提高了系统的可维护性和可扩充性。运用 Struts 技

(上接第 110 页)

也随之变化,提高了模拟的逼真度。

4 结论与探讨

通过以上的基于 L 系统理论对树木和树林进行三维模拟的实例可以看出在对具有自相似结构的植物进行计算机生成的时候,L 系统是比较理想的选择。在以后的应用中,还可以考虑利用随机插值算法生成三维地形图^[8],在此基础上生成不规则地貌上的三维植物群落,进一步提高模拟自然景物的逼真度。

参考文献:

- [1] 曾文曲,王向阳. 分形理论与分形的计算机模拟[M]. 沈阳:东北大学出版社,1993.134-143.

术,快速构建了一个结构清晰、可复用、可扩展的会员服务体系。

参考文献:

- [1] 张娅,杜友福,张南平. 用 Struts 建立 MVC 模式的 Web 应用[J]. 微机发展,2003,13(12):105-107.
- [2] 丁振国,任新洁. 基于 Struts 的 Web 应用开发研究[J]. 微机发展,2004,14(1):90-92.
- [3] Cavaness C. Programming Jakarta Struts[M]. 北京:电子工业出版社,2002.
- [4] 陈建峡. Struts 框架在 Web 程序设计中的应用[J]. 湖北工业大学学报,2005(3):219-221.
- [5] Price J. Oracle9i JDBC 程序设计[M]. 北京:机械工业出版社,2003.
- [6] Moss K. Java Servlet 开发人员指南[M]. 北京:清华大学出版社,2003.

- [2] Kenneth J F. 分形几何 - 数学基础及其应用[M]. 曾文曲,刘世耀译. 沈阳:东北大学出版社,1991.173-185.
- [3] 胡瑞安,胡纪阳,徐树公. 分形的计算机图像及其应用[M]. 北京:中国铁道出版社,1995.113-137.
- [4] 杨光,杨树红,程晓青. 随机分形及其在自然景物模拟中的应用[J]. 试验技术与试验机,1996(2):118-120.
- [5] Edgar G A. The forest fractal puzzle[J]. Computers & Graphics,2000,24:133-141.
- [6] 李峰,陈天滋. 自然景物的模拟[J]. 计算机应用研究,1999(2):76-77.
- [7] 高旭,姜楠. 分形 L 系统理论与植物图像的计算机模拟[J]. 扬州大学学报,2000(3):71-74.
- [8] 郭维红,徐伟忠,谈正. 基于分形几何学的自然景物虚拟实现的研究[J]. 西安交通大学学报,1995(5):77-84.