

基于 L 系统的植物建模方法改进

张树兵 王建中

(华北工学院计算机系, 太原 030051)

摘要 为了克服将 L 系统代码用于编程指导时繁琐的缺点, 在简要介绍 L 系统表达机制的基础上, 对用 L 系统代码生成的各阶段图形的性质进行了分析, 利用分析结果, 给出了一种改进方法, 该方法首先把系统代码转换成简单递归表达式, 然后, 根据这个表达式编写具有递归结构的程序, 从而不仅避免了传统方法中庞大的链表结构和复杂的遍历过程, 而且使植物建模和实现更加简洁而高效. 最后给出该方法的一个具体应用实例.

关键词 L 系统 植物建模 分形

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2002)05-0457-04

Improvement of Plant Structure Modeling Based on L-System

ZHANG Shu-bing, WANG Jian-zhong

(Department of Computer, North China Institute of Technology, Tai Yuan 030051)

Abstract In 1968, Aristid Lindenmayer introduced a formalism for simulating the development of plant, subsequently named L-systems. This formalism was closely related to abstract automata and formal languages. But there is a shortage in programming by the L-system codes to construct plants graphic. The shortage is it's complicate and the low productive in its code because the program has to reuse the same code and traverse or modify the same linked list's each node to draw the plant graphic in every status. To make up this defect, this paper analyses and testifies a quality that those L-system creating graphics in different status are of fractal character. Furthermore, the paper advances an improvement method based on this quality, which means that generation of the plant graphic in the later status can directly make use of the plant graphic in the former status. In other words, the later status graphic only is the former status graphic's enlargement, translation and revolving. Concretely, the new method is let the L-system code into a simple recursion expression which can make the program code cut down and bring up the speed of draw a plant's graphic. A example of the new method is given at the end of the paper. So we can acquire a very useful conclusion in simulating plant with the L-system theory which turn the L-system code into a recursion expression.

Keywords L-system, Plant structure model, Fractal

0 引言

自然景物的计算机模拟一直是计算机图形学的研究热点, 其中, 以着重表达植物形态见长的 L 系统就是该领域诸多研究方法的重点之一. 它由美国生物学家 Aristid Lindenmayer 于 1968 年提出^[1], 后由 Smith 为模拟植物而将其引入计算机图形

学^[1]. L 系统侧重于植物拓扑结构的表达, 它试图用抽象出来的规则描述植物的形态及生长规律^[2,3], 该系统虽然具有定义简洁、结构化程度高、易于实现等优点^[4], 但是, 经研究发现, 用遵循并行重写机制的 L 系统代码来作为编程指导, 并不是最好的方法, 本文分析了 L 系统代码生成的各阶段图形性质, 给出了一个具体的改进方法, 它可以把 L 系统代码改写为一个简单递归式, 实践证明, 这种由递归式写出的

算法将比原算法高效得多。

1 L 系统表达机制简介

对于一株给定的树状植物, L 系统如何对它进行模拟呢? L 系统首先定义一个元素集 V , 集合中每个元素代表树中不同功能的节点或段落; 然后, 根据它的生长过程, 产生一个初始元素 W , 并且有一个生长规律 P . 这 3 个要素 (V, W, P) 就构成了一个 L 系统。

如图 1(d) 所示的树结构就可以认为是以图 1(a) 到图 1(d) 这个过程长大形成的. 其中, A 的功能是, 一定时间后长成图 1(b) 的样子; B 的功能是, 经过同样这段时间后伸长一倍. 从树苗 A 开始, 按照这个规律, 它必然会长成图 1(d). 这就是 L 系统的并行重写机制. 这一过程用 L 系统代码表示如下:

$$\begin{aligned} W: & A; \\ P_1: & A \rightarrow B[A][A]; \\ P_2: & B \rightarrow BB; \end{aligned}$$

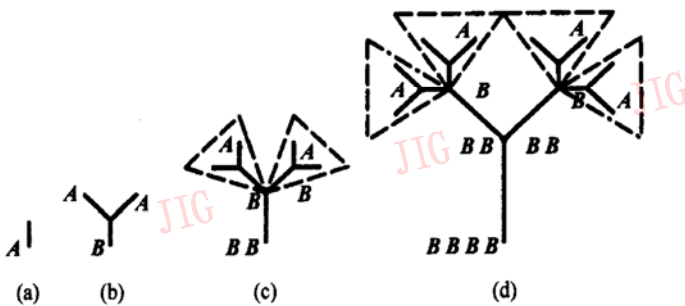


图 1 树苗 A 生长示意图

A, B 构成了集合 V ; 初始状态 W 是 A ; P_1, P_2 是两条生长规则; “[] ”按照 L 系统文法规定为枝条生长方向的偏转. 在整个建模过程中, P_1, P_2 是不变的, 凡遇到 A , 就按规则 P_1 生长, 凡遇到 B , 就按规则 P_2 生长.

这就是用 L 系统对树状植物进行模拟的方法. 根据这些代码编程, 就可以在计算机上实现植物图形的模拟.

2 对用 L 系统代码生成的各阶段图形性质分析

L 系统代码实质上表达了一个并行重写的过程, 但是如果仅仅注意到这一点, 并以这些代码作为编程的思路, 那就需要用—个链表结构来表示这棵

树, 而每个链表节点则代表树的相应节点, 节点的数据项包括: 节点属性、父节点或子节点指针、节点级数等内容. 按照 L 系统代码每迭代一次, 就需要遍历一次链表, 并对每一个链表节点进行修改, 还要增加新的节点. 最后一次遍历才画出整个图形, 但不修改链表. 由于这样的编程比较繁琐, 执行效率低, 因此为了改进这一不足, 本文对 L 系统代码生成的各阶段图形进行了分析, 结果发现, 它们具有自相似性, 下面给出这个性质及证明.

性质 1: 由 L 系统代码生成的各阶段图形必然具有分形的特征.

证明之前, 先分析一下分形的特征.

根据分形理论, 层层嵌套的自相似性是分形的基本特征. 构成分形整体的相对独立部分叫做分形元或生成元, 分形元在一定程度上是整体的再现和缩影. 假设一个分形整体 A , 它由分形元 A_0 按照一定的法则构成, 记作 $A = f_0(A_0)$; 而 A_0 又由自己的分形元 A_{-1} 按照一定的法则构成, 记作 $A_0 = f_{-1}(A_{-1})$, 那么, 所谓的自相似就是指由分形元 A_0 构成整体 A 的法则与由 A_0 的分形元 A_{-1} 构成 A_0 的法则一致, 即 $f_0 = f_{-1}$. 如此类推, 每一级分形元构成高一级的分形元的法则都是一样的.

性质 1 的证明:

(1) 设元素集合 V 只有一个子集 A ; 初始状态为 A ; 生长规律为 $P: A \rightarrow f(A)$.

由于集合 V 只有一个元素, 所以 $f(A)$ 不可能产生别的元素, $f(A)$ 只表示对 A 的各种几何变换, 如伸长、偏转等. 用 S_i 表示模型各阶段的状态, 则有:

$$\begin{aligned} S_0 &= A; \\ S_1 &= f(A); \\ \text{按照并行重写的原则可得:} \\ S_2 &= f(f(A)) = f(S_1); \\ S_3 &= f(f(f(A))) = f(S_2); \end{aligned}$$

$$\begin{aligned} \dots \\ S_n &= f(f(f(\dots f(A) \dots))) = f(S_{n-1}); \end{aligned}$$

显然, 这是一个分形的过程.

(2) 假设元素集合 V 有两个子集 A, B ; 初始状态为 A ; 则生长规律为

$$P_1: A \rightarrow Bg(A); \quad P_2: B \rightarrow h(B)$$

其中, $g(A)$ 和 $h(B)$ 与证明 (1) 中 $f(A)$ 一样, 分别只对所作用元素发生几何变换, 而不产生新的元素.

用 S_i 表示模型各阶段的状态, B_i 表示 B 变换后各阶段的状态, 则有:

$S_0 = A;$
 $S_1 = B_g(A); B_0 = B;$
 按照并行重写的原则可得:
 $S_2 = h(B)g(Bg(A)) = h(B_0)g(S_1); B_1 = h(B_0);$
 $S_3 = h(h(B))g(h(B)g(Bg(A)))$
 $= h(B_1)g(S_2); B_2 = h(B_1);$
 ...
 $S_n = h(B_{n-2})g(S_{n-1});$

显然, 这也是一个分形的过程.

同样道理, 不难证明对多个元素的集合, 以及各种复杂的生长规律, 都能得出同样的结论.

3 L 系统建模改进方法的提出及应用

对于任意一段 L 系统代码, 在上述性质 1 的指导下, 都可以变换成一个简单的递归表达式, 本文用这个递归式来指导编程, 实践证明, 这样将比用 L 系统代码要高效而简洁.

下面用一个例子来说明性质 1 的重要用途.

L 系统代码仍取文章开始的例子:

$W: A;$
 $P_1: A \rightarrow B[A][A];$
 $P_2: B \rightarrow B B;$

令 $h(B) = B B$, 用 S_i 表示模型各阶段的状态, B_i 表示 B 变换后各阶段的状态, 依据性质可直接写出:

$S_0 = A;$
 $S_1 = B[A][A]; B_0 = B;$
 $S_2 = h(B_0)[S_1][S_1]; B_1 = h(B_0);$
 $S_3 = h(B_1)[S_2][S_2]; B_2 = h(B_1);$
 ...;
 $S_n = h(B_{n-2})[S_{n-1}][S_{n-1}];$

现在, 根据这些递归式再画树的生长过程图, 如图 2 所示.

虽然结果还是一样, 但画的方法却明显简单而

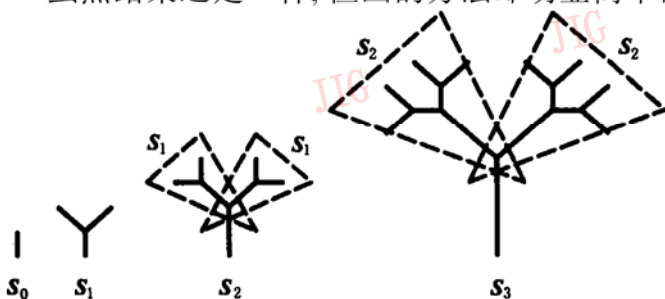


图 2 树的生长过程图

迅速. 这是因为, 当画完 S_2 时, 就增加了一条新的规则: $S_1 \rightarrow S_2$, 这样, 画 S_3 时, 就可以直接应用这条规则; 画完 S_3 时, 又增加了新规则: $S_2 \rightarrow S_3$, 且画 S_4 时, 又可以直接应用这条规则. 这种画法当然比按照 L 系统代码段顺序执行要好得多. 这样做, 实际上每画一步, 就把上一步生成的结果作为分形元, 同时生长规则也改变. 换句话说, L 系统代码中的 A, P 在该方法中是动态变化的.

本文把这种思想用于编程, 提出了一个新的递归方法用来实现树的三维造型模拟. 这里假设树的模型和上图类似, 只是分叉数为 3, 那么用文字表述算法步骤如下:

- (1) 从原点处生成一根竖直的枝;
- (2) 判断递归层数, 决定是否退出递归;
- (3) 以原点为球心, 分别向 3 个方向偏转一定角度, 这样, 3 个在原点相连的部分就形成一个整体;
- (4) 整体向上移动一段距离, 在叉节点和原点间生成一根新的树枝;
- (5) 将至此生成的所有树枝作为一个整体, 转向步骤 2;

在 OpenGL 里, 用上述方法实现递归非常方便. 同时采用相应的随机函数处理, 再加上 OpenGL 里的光照、透视等手段, 可以使生成树具有较强的真实感. 最后生成图形见图 3(a) 及图 3(b).

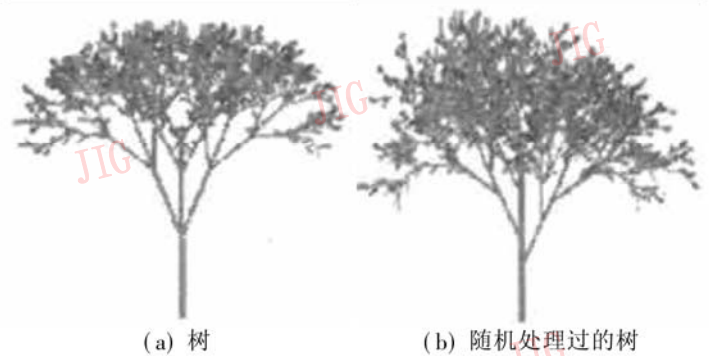


图 3 最后生成的树图

4 结 论

本文在简要介绍 L 系统表达植物形态的机制后, 针对系统代码用于建模时的不足, 首先从理论上证明了用 L 系统代码生成的各阶段图形具有分形的性质, 然后据此给出了改进方法——先由分形性质把系统代码转换成简单递归式, 再由递归式编写递归算法和程序. 从而简化了建模过程, 并提高了算法效率. 因为新方法中分形元、生长(迭代)规则是动

态可变的(这是由分形性质决定的),所以本方法应该可以推广到任何分形图形生成中。

参考文献

- 1 孙家广. 计算机图形学[M]. 北京: 清华大学出版社, 1998: 463~ 466.
- 2 Prusinkiewicz P, Lindenmayer A. The algorithmic beauty of plants[M]. New York: Springer-Verlag, 1990: 235~ 238.
- 3 Glassner A S. Geometric substitution: A tutorial. IEEE Computer Graphics and Application[J], 1992, 12(1): 22~ 36.
- 4 陈昭炯. 基于L系统的植物结构形态模拟方法. 计算机辅助设计与图形学学报[J], 2000, 12(8): 571~ 574.



张树兵 1971年生, 华北工学院99级研究生. 现主要研究方向为计算机图形学.

王建中 博士生导师, 教授. 现主要研究方向为图论、计算机图形学.