

# L-系统生成树的计算机模拟

张全伙 范慧琳 吕俊白

[计算机科学(电脑)系]

**摘要** 讨论 L-树描述系统(简称 L-系统),建立了 L-系统生成树的算法及叙述其设计思想,并用 TURBO PASCAL 加以实现。

**关键词** L-系统,生长规则,字符串

## 0 引言

近年来,在计算机图像处理领域中,植物图像的生成问题逐步引起重视.植物图像的生成是计算机植物标本管理的基础,并在深入研究植物生长规律的基础上,能够进一步用计算机模拟植物的生长规律.目前,植物图像的生成在动画制作中也获得广泛的应用.因此,研究和探索计算机在植物学中的应用是一项极有意义的工作.

植物图像生成问题的研究主要有二种途径.一是通过拓扑学的方法,探讨植物的基本形态.一是通过几何学的方法进一步研究植物的成份,比如,枝的长度、角度、粗细等.植物的生长是通过反复应用一组生长规则形成更为复杂的结构.本文选择树作为探讨的对象,使用 L-系统对树进行描述,提出了 L-系统生成树的算法并详细讨论了该算法的设计思想.程序实现了从 L-系统描述直至图形输出的整个过程.

## 1 L-树描述系统

为了在计算机上对自然界中生长的各种树木的结构和生长规律进行描绘和模拟,首先必须选择一种合适的描述方法,L-系统便是较为成熟的描述方法之一.

### 1.1 L-系统简介

L-系统由以下三个基本元素构成:(1)一组字符  $V = \{a_1, a_2, \dots, a_n\}$ ; (2) $V$  中某些符号表示生长基枝 GW; (3)一组生长规则.例如,L-系统(a)

$$V = \{S, A\}$$

本文1992-10-07收到.

W: A

P1:  $A \rightarrow S[A] S[A] A$

P2:  $S \rightarrow SS$

用L-系统描述树的生长过程,主要在于生长规则的应用.其实质实际上就是字符的替换,生长规则就是字符的替换规则.我们采用的是并行替换的方法.在一棵树(也就是一串字符)中寻找符合生长规则的生长枝,根据生长规则,用生长枝的子枝(通常是字符串)替换原来的生长枝,从而构成一棵在原来的基础上进一步生长后的树.例如

W: A

P1:  $A \rightarrow I_0[I_0 F_0] A$

P2:  $I_i \rightarrow I_{i+1} \quad (i \geq 0)$

P3:  $F_i \rightarrow F_{i+1} \quad (i \geq 0)$

由上述L-系统生成的树的各个时期分别为

① A

②  $I_0[I_0 F_0] A$

③  $I_1[I_1 F_1] I_0[I_0 F_0] A$

④  $I_2[I_2 F_2] I_1[I_1 F_1] I_0[I_0 F_0] A$

如果  $I_i$  表示枝,  $F_i$  表示花,  $A$  表示枝芽,则对应于字符串①,②,③,④树的生长,分别如图1的a,b,c,d所示.  $I$  与  $F$  的下标逐渐递增表示树枝越长越粗且花越开越大.

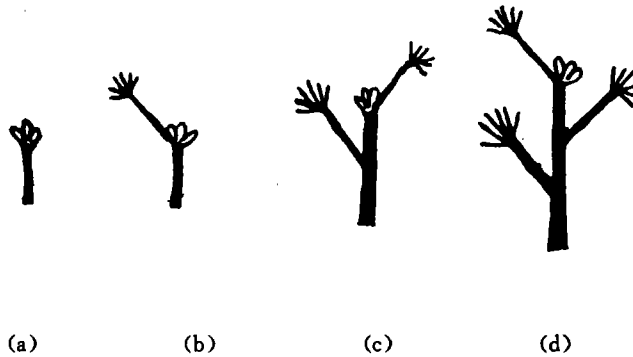


图1 树的生长

## 1.2 树的表示法

用带括号的字符串表示树的拓扑结构. 括号[ ]中的树枝与该括号左边的树枝相连,成为它的分枝. 如果该括号前是一对括号,则跳过这对括号. 最左端的字符表示树的根部,字符从左到右对应于树的根部至顶部. 例如,字符串  $S[A]S[A]A$  与  $ABC[DE][FG[HI][JK]L]MNO$  对应的树形如图2(a)(b)所示.

树木的描述仅限于拓扑法并不能逼真地再现自然界树木的景象,须进一步把树枝的几何形状,如枝的粗细、树叶、花、果等加入L-系统中进行描述,才能获得逼真的效果. 进行这类描述的符号,称为特征符. 例如,用“+”和“-”表示树枝相对于它的前枝的偏转角度,设“+”表示顺时针偏转 $20^\circ$ ,-表示逆时针偏转 $20^\circ$ ,则字符串  $A[+B[+C[-D]]]E[+F[-G][++H]]$  对应于如图3所示的树形. 同理,还可以加入其它的特征符表示树的其它特征.



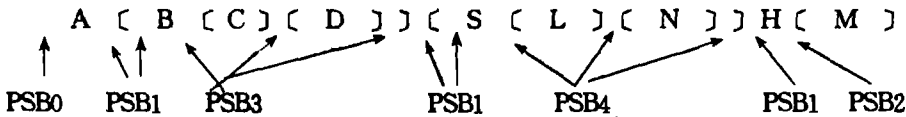
- < 比其前枝长一百分数;
- > 比其前枝短一百分数;
- \ 比其前枝细一百分数.

解释字符串时是从左到右进行的. 以上所给出的特征符可以改变当前关于树枝特征的描述. 我们用 PSB (present status of a branch) 记当前的树枝特征. 五个特征符统一记为 AO (Attribute operator). 操作符作用于状态则记为 AO # PSB. PSB 还包括树枝的起点, 树枝起点的变化与当前树枝的长度、角度有关.

在一串字符中, 设当前字符的起点为 (psx, psy), 长度为 P-l, 角度为 P-a, 如图5所示. 则下一与其相连的树枝的起点为

$$Psx' = Psx + P-l * \sin(P-a * \frac{\pi}{180}),$$
$$Psy' = Psy + P-l * \cos(P-a * \frac{\pi}{180}).$$

显然, 只要读一字符, (Psx, Psy) 就会随之改变. 字符串中的 “[”, “]” 两种符号具有特别的意义. 它表示一种拓扑结构, 即括号中的树枝是长在括号左边字符所代表的树枝上. 例如, 在 A[B[C][D]][S[L][N]]H[M] 中, 各枝的关系如图6所示, 由此可见, 在一对括号的两端 PSB 是不变的, 下式更清楚地展示了这一性质.



因此, 在处理 “[”, “]” 时, 把它们当作一对对当前状态 (PSB) 进行堆栈操作的算符. 即在进行字符串解释时 “[” 完成 Push 功能, 而 “]” 完成 POP 功能.

对树枝的描述是利用一组数来表达. 分别给出树枝的起点、长度、粗细、角度.

b-sp[k, 1] 表示起始 x 坐标,  
b-sp[k, 2] 表示起始 y 坐标,

b-lenth[k] 表示树枝长度,  
b-width[k] 表示树枝宽度,

b-angle[k] 表示树枝相对于 y 轴的角度, 顺时针方向为正, 逆时针方向为负. 其中, k 表示第 k 个描述树枝的字符, 或者说第 k 枝. 以上五个表示第 k 枝的参数统记为 b-exp(k).

为了在构成树林时, 每一棵树都不尽相同, 我们在字符串描述到数据描述转换过程中加入

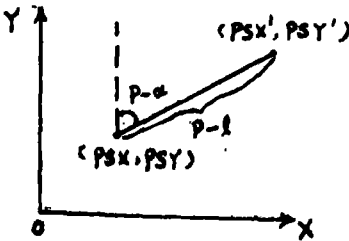


图5 起点、长度、角度示意

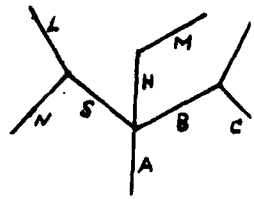


图6 括号的引用

随机因子. 形成  $b\text{-exp}(k)$  时不是简单地把当前状态 PSB 赋给  $b\text{-exp}(k)$ , 而是加入一定的随机偏差  $\delta$  后再赋给  $b\text{-exp}(k)$ . 图7给出了过程 transfer 的流程图. 其中, 入口参数为  $b\text{-stru}$  数组, 内容为字符串, 该串以“.”结尾. 图中, A、B、C、D 框实现了转换规则. 其中: ① A 框. 如前所述, 把树枝现状赋给  $b\text{-exp}$  时, 使用随机函数加入随机偏差  $\delta$ . 读入的字符转换成树枝后,  $(psx, psy)$  必须修正, 如图8所示, 指向  $sp$  点改为指向  $ep$  点, 即

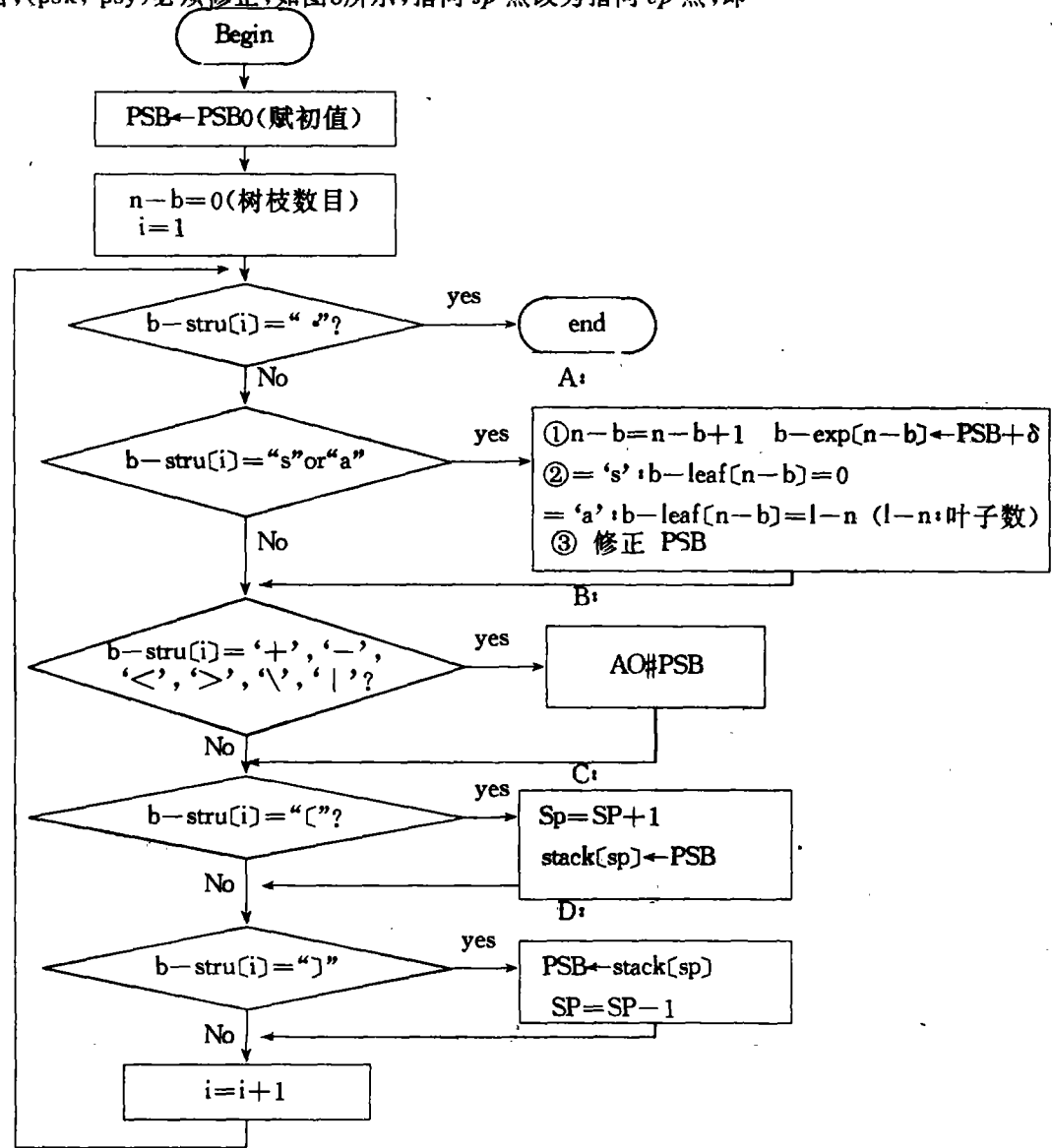


图7 transfer 流程图

$$psx = psx + p - l * \sin p - \alpha$$

$$psy = psy + p - l * \cos p - \alpha$$

②B 框. 按照定义的各操作符功能改变 PSB 为

- “+”  $P-\alpha=P-\alpha+\delta_1$
- “-”  $P-\alpha=P-\alpha-\delta_1$
- “<”  $P-l=P-l*(1+\delta_2\%)$
- “>”  $P-l=P-l*(1-\delta_2\%)$
- “\”  $P-\omega=P-\omega*(1-\delta_3\%)$
- “|”  $P\omega=\text{初始值}$

当然,还可以加入其它的特征描述符. ③C、D 框. 分别完成堆栈操作 Push PSB 和 POP PSB.

2.2 其余过程的功能

除核心过程 transfer 之外,本程序其余过程的功能,作如下简述. (1)linea (i,j,l;integer; a;real);以 (i,j) 为起点画一条长度为 l,与 y 轴夹角为  $\alpha$  的直线. (2)b-drawing(k;integer);以 (b-sp[k,1],b-sp[k,2]) 为起点画一条宽为 b-width[k],长为 b-lenth[k],与 y 轴夹角为 b-angle[k] 的树枝枝干. (3)l-shapel(ix,iy,size; integer); 画一形如图9(a)所示的叶子, size 为尺度, size 越大则叶片越大. (4)l-shape2(ix,iy,size; integer);画一形如图9(b)所示的叶子. size 意义同上. (5)l-drawing(k;integer),在第 k 枝末端画上 b-leaf[k]片叶子. (6)d-init;程序中使用了一些控制参数,如叶子大小、叶子数、叶子类型等. 该过程对数据作初始化. (7) sample 1画树的结构. (8) sample 2,画单棵树. (9) sample 3,画一片树林.

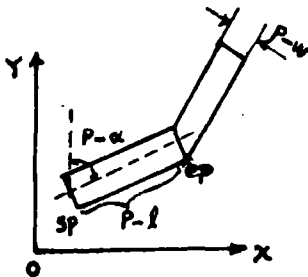
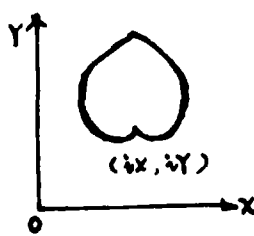
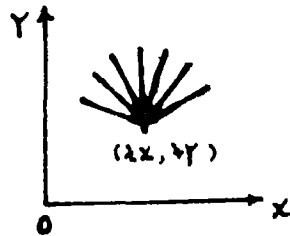


图8 枝干



(a)



(b)

图9 叶子

2.3 图形效果

有了过程 transfer 生成的数据 b-exp 之后,通过各绘图过程不难画出所定义的树的结构、单树以及加入随机因子后形成的树林,如图10,图11,图12所示.考虑二维图形的立体效果,作图时可相应作某些处理. 作图时,一般先画树干后画树叶,因此所有的树叶都处在树干前面(图11). 若能在画树干和树叶时交叉进行,那么某些叶子将挡住树干而另一些叶子将被树干遮挡(图12). 二种做法的视觉效果是不同的,后者显然优于前者,特别在彩色屏幕上后者更具立体效果.

生成树林时每棵树的位置是随机确定的. 但这样处理会出现一些问题. 一棵树的近根部分越“高”离我们便越“远”,若随机地确定树的位置,将出现“远”树挡住“近”树的错误. 如图13中



图10 结 构



图11 单 树

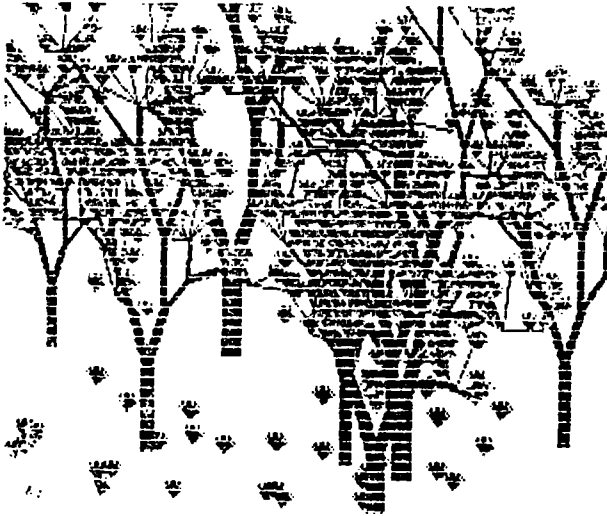


图12 树 林

的 A、B 二棵树便是如此. 为了既能正确地反映树林的相互位置关系又能用随机方法确定树的位置, 在画树林时先生成一批随机数, 然后对其  $y$  坐标进行排序, 按  $y$  坐标从大到小的顺序依次画树, 从而构成树林, 便能避免上述错误(图12).

### 3 结束语

作图程序 tree 具有一定的通用性, 在编制时采用自顶向下的设计方法, 结构清晰. 对程序中所用变量、常数以及一些参数说明详细, 易读易改. 特别是只要增加特征参数, 可以模拟出非常逼真形象的树和植物.

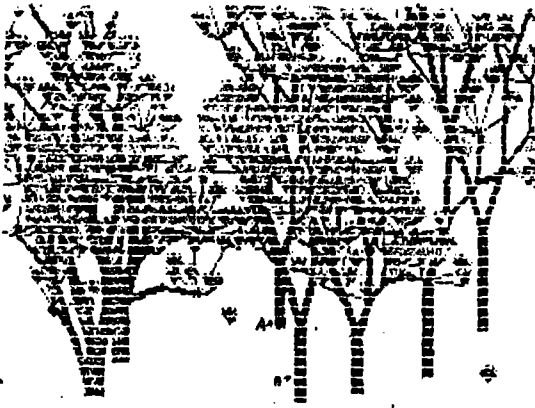


图13 树的互相遮挡

### 参 考 文 献

- [1] Smith, A. R. , *Plants, Fractals and Formal Language*, computer Graphics, 18,3 (1984).
- [2] Prusinkiewicz, P. , Hanan, J. , *Visualization of Botanical Structures and Processes using parametric L-systems*, Scientific Visualization and Graphics Simulation, (1990).
- [3] Prusinkiewicz, P. , *Graphical application of L-systems*, Proc. of Graphics Interface, (1986).
- [4] Aono M. , Kunii, T. L. , *Botanical tree image generation*, IEEE Computer Graphics & Application 4,5 (1984).
- [5] 罗杰斯, D. F. (梁有栋等译), *计算机图形学的算法基础*, 科学出版社, (1987).

## Computer Simulation of the generating Tree of L-System

Zhang Quanhua Fan Huilin Lu Junbai

(Department of Computer science)

**Abstract** Starting with a discussion about L-tree description system or L-system for short, the authors create an algorithm for the generation tree of L-system, with emphasis on the design principle of the algorithm; and implement the algorithm with TURBO PASCAL.

**Key words** L-system, rule of growth, character string