

融合遗传操作的2维分形递归算法研究

吴为丹 周凡

(中山大学计算机科学与技术系, 广州 510275)

摘要 分形算法是自然景物建模的一种有效技术, 普通分形算法往往存在部分与整体严格自相似的问题, 即使加入随机因素的扰动, 仍不能逼真体现自然景物的遗传生长过程。提出了以自然树生成为例, 吸收遗传算法的思想, 对树枝特征进行编码, 以树枝为单位对编码进行遗传操作, 通过父代树枝的遗传操作生成子代树枝, 使父代与子代之间保持相似又不严格相似, 更贴切地模拟树木的遗传生长过程。实验结果表明, 采用此算法生成的分形树随机性更强、效果更逼真。融入遗传操作, 能有效弥补普通分形算法的严格自相似问题和随机函数在模拟自然景物遗传生长过程方面的不足, 并可推广至其他分形递归算法。

关键词 分形树 随机扰动 交配 变异 参数控制 随机数产生器

中图法分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2008)08-1560-06

Research on 2D Fractal Recursive Algorithm Syncretized Genetic Operation

WU Wei-dan, ZHOU Fan

(Department of Computer Science and Technology, Sun Yat-Sen University, Guangzhou 510275)

Abstract Fractal is an effective technology in the modeling of nature scenery, but the strict self-similarity of the traditional fractal algorithms has limited their efficiency. Even when a random issue is incorporated, the process of genetic growth is still not simulated appropriately. In this paper, the concept of genetic operation are adopted in the traditional fractal-tree generating algorithms, the characteristic of the branch is encoded, and the cross-over, mutation, selection are operated on the code. The characteristic of the child branch is generated from the genetic operation of the parent branch; they are similar but not identical. As a result, an image that is self-similar but not strictly self-similar can be generated and the process of genetic growth can be simulated more appropriately. The case study and simulation shows the fractal-trees generated by the new algorithm are more stochastic and more real. The genetic operation can effectively remedy the strictly self-similar of the fractal algorithm and the limitation of the random issue in the modeling of natural scenery, and this algorithm could be extended to other fractal recursive algorithms.

Keywords fractal-tree, random perturbation, cross over, mutation, parameter control, random number generator

1 引言

自然景物的模拟一直是真实感图形学的一个研究热点。以 Mandelbrot 的分形理论为基础而提出的各种分形算法, 根据自然景物的部分与整体间的统计自相似, 通过构造若干简单的规则描述部分的特

征, 利用规则进行迭代生成整体, 可以产生非常逼真的自然物质形态和视觉效果, 从而提供了一种有效的方法用于描述缺少连续、光滑等基本性质的自然景物。

目前主要的分形算法包括 IFS (iterative function system, IFS) 迭代算法、文法构图算法、逃逸时间算法、粒子系统^[1,2]等等, 这类基本算法的本质都是通

过重复迭代来构图,但构造出来的图形往往具有严格的自相似结构。但是自然景物由于受到环境客观因素的影响,极少会有严格的自相似,因此,在自然景物的模拟中必须引入随机因素的扰动。通常是在迭代过程中引入随机函数,使参数在某一范围内随机变动^[1,3-4]。然而,随机分形过程与物种的自然生成过程差别很大”,“无法与自然界的物种遗传、物种生存环境的选择、进化和变异作用建立必然联系^[5]。因此,对于模拟具有遗传生长过程的自然景物,其生长过程应该以相似性为主,差异性为辅,引入普通的随机函数并非最科学的做法。

遗传算法是弥补这一缺陷的有效方法,通过对问题对象进行编码,对编码进行交配、变异、选择等遗传操作,通过模拟生物遗传进化的机制和复杂的适应过程来求解问题。因此,遗传操作可以代替原来分形算法中的随机扰动,弥补其在模拟生物遗传过程方面的弱点;同时,子代染色体与父代染色体保持相似但有所差异,可以弥补普通分形算法的严格自相似问题。

本文以分形树的生成算法为例,对树枝特征进行编码,通过遗传操作模拟自然树的分支生长过程,实现子代与父代相似又不严格相似的效果。通过 32 位编码细致刻画树枝长度、角度、类型以及分支个数、位置等信息。遗传操作以树枝为单位,通过对描述树干的父染色体进行遗传操作,将得出的子染色体用于描述其分支,体现了遗传生长的过程。同时,借鉴文献[2]的作法,对随机数生成器进行处理,将原来满足平均分布的随机数改为符合正态分布的随机数,后者能更有效地模拟自然景物的随机分布规律。最后,讨论了这种思想应用于普通分形递归算法的要点。

2 算法思想

2.1 分形理论介绍^[6]

分形是分形空间上压缩映射的不动点。其中的分形空间是指定义了 Hausdorff 距离的完备度量空间,这样一个空间常常是 \mathbf{R}^d 上的一个紧致子集,而 Hausdorff 距离定义如下:

假设 $D_{\text{dist}}(E, F)$ 为集合 E 到 F 的 Hausdorff 距离,则

$$D_{\text{dist}}(E, F) = \max \{ D_{\text{dist}}(e, F), e \in E \}$$

其中,

$$D_{\text{dist}}(e, F) = \min \{ d(e, f), f \in F \}$$

d 可以是欧氏距离或绝对值距离。而集合 E, F 的 Hausdorff 距离定义为 $D_{\text{dist}}(E, F)$ 和 $D_{\text{dist}}(F, E)$ 的较大者。

压缩映射是指距离空间 (X, D_{dist}) 上的一个映射 $f: X \rightarrow X$,如果存在一个 $s, 0 \leq s < 1$,使得 $D_{\text{dist}}(f(m), f(n)) \leq s \times D_{\text{dist}}(m, n)$ 。其中 s 称为压缩因子。特别地,对于 $\omega(F) = C$ (C 为 X 上常量)的映射称为凝聚映射, C 称为凝聚集。

分形算法的基本步骤是迭代,这来源于分形本身的迭代性质, Banach 的不动点定理保证完备度量空间上的压缩映射必有一不动点。因此,通过在一个分形空间 (X, D_{dist}) 上定义一有限压缩映射族 $\omega_i: X \rightarrow X, i = 1, 2, \dots, N$,令 $W = \bigcup_{i=1}^n \omega_i$,根据吸引子定理, X 上必有一个不动点 A ,使得对任意 $k \in X, A = \lim_{n \rightarrow \infty} W^n(k)$ 。也即是说,通过重复施加映射 W ,最终可以得到不动点 A 所代表的分形图。

例如,构造分形树可以基于下述图。通过在树干上重复画出分支,最终可形成树的图案。

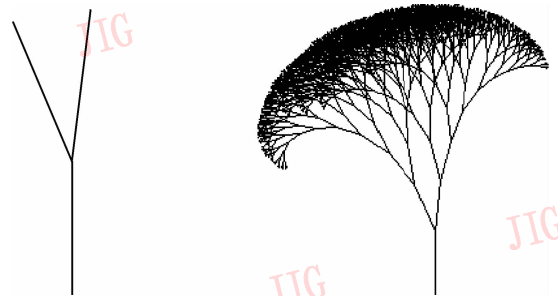


图 1 分形图

Fig. 1 Fractal graphics

其映射族可表示为 $(\omega_0, \omega_1, \omega_2)$,其中, ω_0 为凝聚映射,凝聚集为图中的主干部分,而

$$\omega_1(x, y) = (xrcos\theta + yrsin\theta, xrsin\theta + yrcos\theta + r_0)$$

$$\omega_2(x, y) = (xrcos\theta - yrsin\theta, xrsin\theta + yrcos\theta + r_0)$$

其中, r_0 为主干长度, r 表示当前所画分支的长度,在每一次迭代过程中将 r 缩小为 $r = r_0 \times s$ ($0 \leq s < 1$),保证分支长度小于干的长度^[9]。

ω_1, ω_2 将树干上的每一点映射到其分支上的一个点,压缩因子为 s 。算法重复地在树枝上画出其分支,直至 r 减小到某一阈值为止,便可得到图 1 所示的分形树。

由图 1 (b) 可以看出,它具有严格的自相似结构。要更加逼真地模拟自然树,需要对图元引入随

机因素,构造非严格自相似的图案。

2.2 加入遗传操作的分形算法

遗传算法对问题对象进行编码,通过选择、交配和变异等遗传操作模拟生物进化过程。由父代经过遗传操作产生的子代,既保持了父代的特征,又有一定的差异。将遗传操作加入分形算法中,能更真实地模拟自然树的生长过程。

一般的构造分形树的递归算法是在画出主干后修改相关参数,再递归画出其分支:(1)画主干;(2)修改参数;(3)递归调用画出各个分支。主函数初始化一系列控制参数后调用绘图函数一次即可画出图形,算法体现了分形的迭代性质,但存在尾递归,不利于遗传操作的应用,而且会耗费大量空间。

本文的算法用循环代替尾递归,采用一个队列保存树枝信息,从队列中循环取出待画树枝信息,进行遗传操作,生成各种绘制参数信息和分支信息,并将分支信息压入队列,然后根据参数信息画出这根树枝。程序流程可描述如下:

- (1)画主干并保存主干信息在队列中;
- (2)do{

取出待画树枝信息;
遗传操作;
保存分支信息;
画树枝;

while (尚有未画树枝);

同样,程序要初始化主干信息作为循环的起点。绘制一根树枝所需的控制参数包括起始点、长度和偏角。树枝信息结构定义如下:

```
struct info {
//各分支起始点,直接作为绘制信息之一
    CPoint p[ ];
//当代树枝特征染色体种群
    int population[POPSIZE];
//树枝角度,绘制信息之一
    int pangl;
};
```

染色体种群经过遗传操作产生分支长度、分支相对于父代树枝的角度,结合分支起始点和树枝角度用于控制分支绘制。同时产生的下一代分支位置、遗传操作后的种群以及分支角度构成分支信息保存于队列中。可以看出,“遗传操作”在生成分支信息方面的作用相当于普通递归算法的“修改参数”。

另外,本文生成的树枝类型包括合轴和单轴两种。前者的主干发育到一定阶段停止发育,由侧芽取而代之,形成强的侧枝接在主干上,因此没有明显的主干。后者则有明显的主干。如图所示:

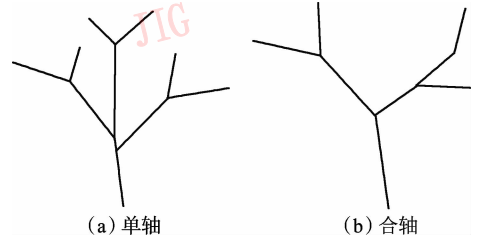


图 2 单轴与合轴

Fig. 2 Uniaxial and Composite-axis

本文的算法虽然用循环代替了尾递归,但两者是等价的^[7]。算法在本质上仍具有普通分形算法的迭代性质,不同之处在于只加入了遗传操作,使原来固定的图元变成了在一定范围内随机变化的图元。因此,压缩映射族 W 现在变为 W_{CA} ,其中,压缩因子 s 和角度 θ 根据每一次遗传操作的结果而定。对于合轴树枝,分支生长在主干末端,其压缩映射如上文所示。而对于单轴树枝,由于产生侧枝,在表示侧枝的映射中,须将干长度 r_0 改为侧枝起点距干起点的距离 \hat{r} ,即

$$\omega_1(x, y) = (xrcos\theta + yrsin\theta, xrsin\theta + yrcos\theta + \hat{r})$$

$$\omega_2(x, y) = (xrcos\theta - yrsin\theta, xrsin\theta + yrcos\theta + \hat{r})$$

显然,加入遗传操作的算法并没有改变原来分形算法的迭代性质,算法最终也会收敛到不动点 A_{CA} 。

2.3 算法中的遗传操作

遗传操作是本算法的要点,由于并非用于精确的寻优过程,遗传算子对算法收敛性等方面的影响不需像寻优问题那样细致考虑,在本算法中只是根据大致影响进行控制,保持父代与子代的相似,同时能够产生一定的差异即可。

(1) 编码

由于遗传过程的鲁棒性,对染色体编码要求并不苛刻,只要能简洁表示出问题特征即可。本文采用 32 位编码一根树枝,各二进制位划分如下:

- 0 ~ 4: 树枝长度,解码时乘以倍数因子 10;
- 5 ~ 9: 树枝相对于 90°的偏角;
- 10: 树枝类型(合轴/单轴);
- 11: 树枝类型(左分支/右分支);
- 12 ~ 13: 分支数(1 ~ 3);

14~31: 3 个分支位置,各 6 位,表示分支点距起点长度相对于树干总长的百分比。

解码时只需取出各字段并计算对应十进制数值即可。但对角度做一定的控制,限制在 $45^\circ \sim 60^\circ$ 或 $120^\circ \sim 135^\circ$ 的范围内。

(2) 选择操作

适应度是选择操作的基础,其计算方法要根据实际问题决定。通常情况下,实际问题都能够提供明确的适应度计算方法,如旅行商问题中路径越短则适应度越高,函数求最值问题中函数值越接近最值则适应度越高。但本文问题较特殊,树枝特征的差异只是适应度差异的基础,还须结合对生成树大致形状的期望来决定适应度的高低。

对每一个特征,指定最高适应值为 10,最低为 0,染色体适应值为各特征的适应值之和。例如,若指定合轴分支数量占 90%,左侧分支占 90%,则对于生成合轴分支和右侧分支的染色体,其树枝类型的适应值为 $9 + 1 = 10$ 。若希望生成树比较密集,则控制树枝长度较短、相对于干的角度较小,并且分支较多、分支间距离较小,可将对应的树枝特征的十进制数值均匀映射为 $0 \sim 10$ 之间的一个整数作为适应值。例如分支数取值 $0 \sim 3$,其映射函数定义为:

$$f(\text{count}) = \begin{cases} \text{count} \times 3 & \text{希望有较多分支时} \\ 9 - \text{count} \times 3 & \text{希望有较少分支时} \end{cases}$$

分支间距离适应值的度量则取所有分支距离平均值的倒数 d ,定义映射函数为

$$f(\bar{d}) = \begin{cases} \bar{d} \times 20 & \text{希望分支较密} \\ 10 - \bar{d} \times 20 & \text{希望分支较疏} \end{cases}$$

在此基础上,采用基于排挤的小生境遗传策略^[8]。这一策略模拟自然界中生物在特定生存环境下,在形状、特征相近的同类间交配、繁衍的现象,将种群分为若干小生境,在同一个小生境内,对适应度相近的两个染色体,排挤掉适应度低的,选出适应度高的若干个体作为代表。然后在小生境之内或者之间进行遗传操作。相比之下,简单遗传算法(SGA)在进化后期多数染色体往往集中于某个点,而小生境遗传策略能够更好地保持群体多样性,避免遗传操作的早熟现象影响算法的有效性。具体来说,本算法规定适应度差值在 5 以内的两个染色体为相近的染色体。

(3) 交配操作

本文的交配算子采用单点交配。考虑到分形算法是利用部分与整体的相似性来实现的,本算法的交配操作目的在于保持父代与子代的相似性,差异性次要的。单点交配的可达性较小^[9],这正是本算法所需要的。同时,选择二进制数的低位进行交配,对数值影响不大,从而也控制了父代与子代的差异性。另外,根据模式理论,交配操作对模式的破坏作用与交配概率有关^[8],在本文取较小的交配概率 0.4。

具体交配方法如下所示,对各字段分别应用单点交配法,随机生成交配点。

交配前:

$$\begin{array}{l|l} x1, x2, x3, x4, & x5, x6, x7 \\ y1, y2, y3, y4, & y5, y6, y7 \end{array}$$

交配后:

$$\begin{array}{l|l} x1, x2, x3, x4, & y5, y6, y7 \\ y1, y2, y3, y4, & x5, x6, x7 \end{array}$$

在对树枝长度的字段进行交配操作时,取压缩因子 s 为 0.7,即控制子代树枝长度在父代树枝长度的 0.7 倍以内。另外,为避免同侧分支距离太近,规定每隔主干长度 0.2 倍以下的范围内只能有一根分支。这样可保证生成树的形状符合自然规律。

(4) 变异操作

变异操作用来产生差异性,目的在于保持群体多样性,在本文中,采用随机变异法^[8],对各个字段分别用一个在其位数允许范围内的随机数代替。对代表树枝长度和分支位置的字段同样施加上述控制。随机数的产生经过适当处理,符合正态分布。

2.4 随机数处理

自然界中的许多现象都满足正态分布,本文借鉴文献[2]的作法,对随机数进行下述处理。

假设随机数发生器 $P = \text{rand}()$ 产生 $[0, Z]$ 之间的均匀分布的随机数,其中 Z 取值 $2^{15} - 1$,任意随机变量 P 减去其期望值再除以标准差即可将其标准化,

$$Q = \frac{P - E(P)}{\sqrt{D(P)}}$$

根据中心极限定理,设 Q_n 是 n 个符合相同分布的随机变量的标准化总和,则 Q_n 的概率分布当 $n \rightarrow \infty$ 时满足正态分布,让 Q_i 为 $\text{rand}()$ 返回的第 i 个值,则

$$E(Q_i) = \frac{1}{2}Z, D(Q_i) = \frac{1}{12}Z^2$$

$$\text{又 } E\left(\sum_{i=1}^n Q_i\right) = \frac{n}{2}Z, D\left(\sum_{i=1}^n Q_i\right) = \frac{n}{12}Z^2$$

$$\text{于是 } Q_n = \frac{\sum_{i=1}^n Q_i - \frac{n}{2}Z}{\sqrt{\frac{n}{12}Z^2}} = \frac{1}{Z} \sqrt{\frac{12}{n}} \sum_{i=1}^n Q_i - \sqrt{3n}$$

本文中, n 取 3。 $Y_n(p)$ 将产生 $[-p, +p]$ 范围内近似服从正态分布的随机数。

3 算法流程

算法主体流程如上文所述,由两重循环构成,由 $depth$ 和 $cur_generation$ 分别控制。 $depth$ 表示进化代数,小于 0 时结束算法。 $cur_generation$ 表示当前一代的树枝个数,下一代的树枝个数由 $next_generation$ 记录,在完成当代所有树枝的绘制后,结束内层循环,将 $next_generation$ 赋给 $cur_generation$,进入下一次循环。

在进入主体部分之前,需初始化主干信息作为主体部分循环操作的起点。程序流程如下所示。

```

初始化主干信息,保存分支信息并绘制主干;
cur_generation ← 主干分支个数;
while ( depth >= 0 ) {
    while ( cur_generation != 0 ) {
        取出树枝信息进行遗传操作;
        if ( 枝长 > 0 ) {
            if ( depth > 0 ) {
                保存分支信息;
                next_generation += 分支数;
            }
            绘制树枝; cur_generation -- ;
        }
    }
    cur_generation = next_generation;
    next_generation = 0; depth -- ;
}

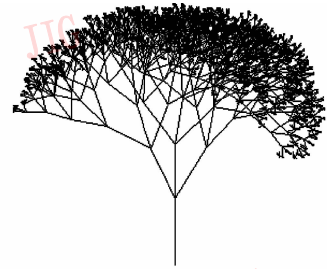
```

4 算法结果

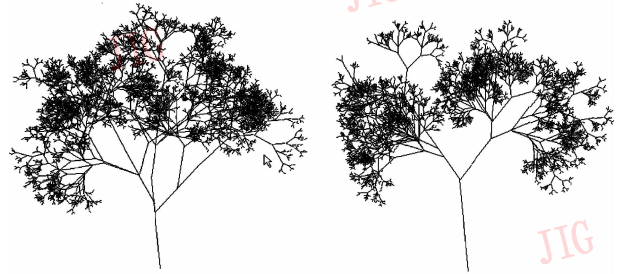
为验证算法效果,用 VC++ 6.0 实现了该算法,并与普通分形递归算法进行比较。新算法由于使用了遗传操作,克服了严格自相似的缺点,随机性和真实性大大增强,形态更接近自然树。

本算法用循环代替尾递归,节省了大量保存实参、形参的栈空间,只保存算法必须的树枝信息,从而提高了算法的时空效率。因此,虽然加入了遗传操作,在一定程度上使算法复杂化,但程序的运行效率却没有降低。

同时,由于加入遗传操作,原来尾递归实现的程序不能明显地体现树枝遗传生长的过程,程序必须先完成一个分支上的所有递归过程之后才能进行第 2 个分支的绘制,树枝信息、进化代数的保存较为复杂。相反,本文算法的循环过程能按照树枝生长的先后顺序进行遗传操作和绘制,贴切地体现树木的生长过程。



(a) 普通递归算法结果



(b) 本文算法结果

图 3 算法结果

Fig. 3 Result of the algorithm

更重要的是,由于本文算法对树枝特征进行编码,因此,通过对相应参数取值的控制,可以实现对整棵树形状的大致控制。指定有关的特征参数,通过选择机制控制进化方向,可以有效控制生成树的大致形状。

已实现的可控参数包括:单轴分支与合轴分支的比例、左分支与右分支的比例。而且由于单轴分支生长出来的树枝较多,通过控制单轴分支的比例,与进化代数相结合可以控制树的茂盛程度。下面几幅图采用比较极端的参数取值给出控制效果。

可以看出,采用本算法可以通过参数取值控制树的大致形状,进一步增加控制参数,如分支数量、位置等等,可以实现更细致的控制。

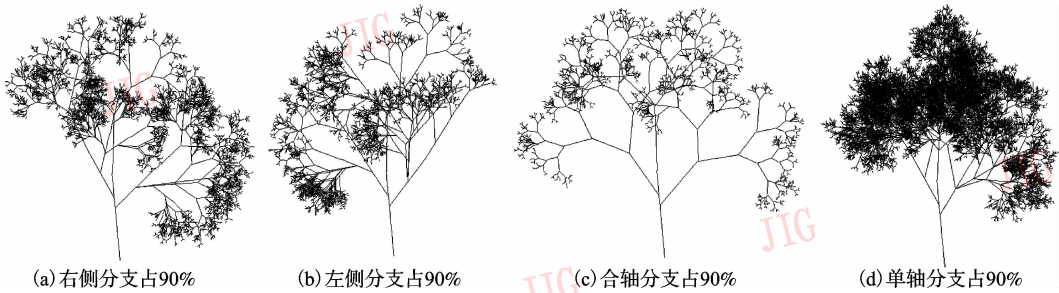


图4 参数调整

Fig. 4 adjust of the parameter

5 算法推广

从以上论述可知,融合遗传操作的分形算法具有以下特点:

- (1)对特征进行编码;
- (2)用遗传操作产生随机效果;
- (3)用堆栈或队列保存染色体编码,用循环代替尾递归;
- (4)染色体的后代数对应尾递归的个数。

许多类似的分形递归算法均可根据上述原则融入遗传操作。以康托集的构造为例,严格自相似的康托集对单位长度的线段三等分,去掉中间段,再对剩下的左右两段递归地重复上述过程。假设构造过程第 k 步的某一线段为 I_k ,分割后得左右两线段分别为 $I_{k+1,1}$ 和 $I_{k+1,2}$,则 $C_k = |I_{k+1,1}|/|I_k| = |I_{k+1,2}|/|I_k| = 1/3$ 。统计自相似的康托集则要求 C_k 为随机变量。

康托集的构造需要两个尾递归,相当于每一个线段生成两个长度为其 C_k ($0 < C_k < 1$)倍的后代,通过对线段特征编码,在队列中保存染色体,再通过循环依次取出即可构造。而遗传操作使 C_k 符合一定的随机分布。也就是说,遗传操作封装了分形递归算法中的随机函数。在模拟遗传生长的过程中,则进一步保证了相似性为主、差异性为辅的遗传生长过程。

6 结论

本文主要介绍了将遗传操作融入普通分形算法的自然树生成算法,并讨论了将其推广至普通分形递归算法的要点。遗传编码能够细致刻画树枝特征,并加入符合生物生长过程的遗传操作,因此,在不牺牲算法性能的基础上大大增强了随机性和真实性,消除了严格自相似的缺点,并克服了随机分形的不足,生成树更

逼近自然效果。同时,由于对树枝特征进行数字化编码,能够通过组合多个参数控制整棵树的生成。

但是,控制参数的精度和数量仍需进一步改进和完善,并可加入随机弯曲,增强真实性。而且,可进一步考虑对树枝的立体特征进行编码,如将树枝抽象为圆柱体,描述其底面半径,半径随树干高度的衰减系数等等,将本算法推广到3维的情况。

参考文献 (References)

- 1 Hearn Donald, Pauline Baker M. Computer Graphic [M]. Upper Saddle River, NJ, USA: Prentice Hall, 1998.
- 2 Barnsley M F, Devnney R L. Fractal Graphics [M]. He Feng Translation. Beijing: Ocean Press, 1995. [Barnsley M F, Devnney R L. 分形图形学[M]. 和风译. 北京:海洋出版社,1995.]
- 3 Peter E. Real time design and animation of fractal plants and trees [J]. ACM Siggraph, 1986, 22(4): 55~64.
- 4 Zhang You-hui. Fractal Simulation of Nature tree's Appearance [J]. Computer Engineering, 2000; 26(9): 112~114. [张有会. 自然树木形状的分形模拟[J]. 计算机工程, 2000, 26(9): 112~114.]
- 5 Li Yun-fei, Jiang Xiao-fen, Guo Zhao-xian. The application of genetic algorithm in creation of natural trees [J]. Microelectronics & Computer, 2005, 22(5): 95~97 [李云飞, 姜晓峰, 郭肇娴. 遗传算法在自然树生成中的作用 [J]. 微电子学与计算机, 2005, 22(5): 95~97.]
- 6 Liu Dan. Practical Fractal Graphics [M]. Dalian: Dalian Maritime University Press, 2001. [刘丹. 实用分形图形学[M]. 大连:大连海事大学出版社. 2001.]
- 7 Yan Wei-min, Wu Wei-min. Data Structure [M]. Beijing: Tsinghua University Press, 1997 [严蔚敏, 吴伟民. 数据结构(C语言版) [M]. 北京:清华大学出版社, 1997.]
- 8 Zhou Ming, Sun Shu-dong. Genetic Algorithms: Theories and Applications [M]. Beijing: National Defense Industry Press, 1999. [周明, 孙树栋. 遗传算法: 理论与应用 [M]. 北京: 国防工业出版社, 1999.]
- 9 Zhang Jun-ying. Attainability of genetic crossover operator [J]. Acta Automatica Sinica, 2002, 28(1): 120~125. [张军英. 遗传交叉运算的可达性研究 [J]. 自动化学报, 2002, 28(1): 120~125.]