

快速构造植物几何结构的子结构算法

康孟珍^{1),2)} Philippe de Reffye^{4),5)} 胡包钢^{1),3)} 赵星^{3),4)}

¹⁾(中国科学院自动化研究所模式识别国家重点实验室,北京 100080)

²⁾(首都师范大学信息工程学院,北京 100037)

³⁾(中国科学院自动化研究所中法信息、自动化与应用数学联合实验室,北京 100080)

⁴⁾(法国国家信息与自动化研究院, BP105, 78153 巴黎, 法国)

⁵⁾(法国国际农业研究发展中心, BP5035, 34032 蒙彼利埃, 法国)

摘要 顺序地模拟符合植物学生物特征的复杂树结构会占用很多计算机资源。为减少对计算机资源的占用,提出了一种从最简单的子结构起,通过子结构的引用来形成复杂结构的方法。由于在该方法中重复出现的结构只被计算一次,因此不仅可以提高复杂结构的计算速度,而且可以降低几何信息的存储量。该几何信息不仅包括所模拟植物在各个生长周期的结构,而且包括了每个子结构的具体形态。尽管这是一种自上而下的算法,但由于子结构方法能够实现弯曲枝条的模拟,并能产生重复生长的结构和花序结构,因此具有通用性。

关键词 快速算法 子结构 植物几何结构 自动机 递归算法

中图分类号: TP391.41 TP391.9 文献标识码: A 文章编号: 1006-8961(2004)01-0079-08

Fast Construction of Geometrical Structure of Plant with Substructures Algorithm

KANG Meng-zhen^{1),2)}, Philippe de REFFYE^{4),5)}, HU Bao-gang^{1),3)}, ZHAO Xing^{3),4)}

¹⁾(National Laboratory of Pattern Recognition, Institute of Automation, CAS, Beijing 100080)

²⁾(College of Information Engineering, Capital Normal University, Beijing 100037)

³⁾(Sino-French Laboratory in Information, Automation and Mathematics (LIAMA), Institute of Automation, CAS, Beijing 100080)

⁴⁾(INRIA, BP105, 78153 Paris, France) ⁵⁾(CIRAD, BP5035, 34032 Montpellier, France)

Abstract Sequential simulation of 3D geometry of complex plant that is faithful to botany may occupy a lot of computer resources. In this paper, a method of building complex structures begun by the simplest stems is presented; the simpler substructures are pasted recursively to form more complicated branches. In doing that, the repetitive structure in plant is computed only once, and then can be reused according to its index, thus this method can increase computation speed of complex tree structure, and decrease storage size for the geometrical information as well. These information includes not only geometrical data of the simulated plant at each growth cycle, such as orientation and position of each botanical organ, but also that of each substructure. Although substructure method is a top-down algorithm that is inverse to growth sequence of plant growth, it is shown in this paper that bending branches can still be simulated in this way, as well as reiteration and inflorescence structures, which shows the generality of this method.

Keywords Fast algorithm, Substructure, Plant geometrical structure, Automaton, Recursive algorithm

1 引言

在计算机上进行植物形态构造的研究可以追溯到上个世纪60年代^[1]。迄今为止,世界上已经有了多种专门用于产生虚拟植物的自由或商业软件,可

用于研究、教育、商业、娱乐等领域^[2]。基于过程的动态植物模型,由于能够反映植物不同生长时期的生长变化过程,而且可以和植物的生理模型相结合^[3],因而具有更广泛的用途。

实现动态模型的一般方法是逐节顺序模拟的算法,即每次生成一个节,以及节上的花、叶等侧生器

官,例如在文献[4]中介绍的顺序算法。这种方法虽然对于草本植物或花序的模拟非常快速有效,而且易于理解,但是由于模拟时间与节间数目为线性关系,因此对于复杂的树结构,例如包含很多重复结构的松树、柏树等,其所需的模拟时间将迅速增长。

本文提出的算法是以子结构为单位来生成植物结构,即事先计算重复出现的子结构,当其他结构中出现这样的子结构时,则直接调用,而不再逐节模拟。由于重复的子结构只被计算一次,因此对于复杂植物可以有效地减少模拟时间。

子结构的调用关系与产生植物结构的模型有关。常见的动态模型有L系统^[5]和AMAP软件所用的模型^[6]。本文采用了双尺度自动机模型^[7]。由于其参数的物理意义明确,且模型结构清晰,使得每种子结构都具有定量的属性,因而便于调用。

该植物结构的构造是笔者等研究的整个名为“青园(GreenLab)”的植物模型的一个部分。

2 子结构算法

子结构是个相对的概念,又是个递归的概念,如植物主干上的侧枝是植物的子结构,当植物处在群落中时,则植物本身就成为群落的子结构。以子结构为单位进行植物形态构造的算法称为子结构算法。子结构算法是首先模拟出一个(对于确定性生长)或若干个(对于随机生长)具有相同属性的子结构,而当在其他子结构中出现这些子结构的时候,则直接复制已有的信息,不再重复计算。此外,由于模拟中可以只保留重复信息的索引,而仅根据需求来获取详细信息,因

此还可以达到降低存储量的目的。这种对单独子结构的保存和获取的思想自然地可以应用到植物器官库的构造上。事实上,花序不仅可以用于子结构方法模拟,而且其本身就是一个植物中的子结构。本文通过子结构算法不仅构造了具有一般性的植物结构,而且模拟了重复生长和花序结构,另外,通过对枝条初始角的离散化,使得该方法还可以实现柔软枝条的模拟。

子结构算法最早是在文献[8]中提出的,其主要用于植物器官个数的计算,其中虚拟植物的拓扑结构是由符合植物学原理的双尺度自动机模型^[7]产生的(如图1所示)。在这个模型中,每个生长周期,植物在轴上长出的生长单元,可以双尺度自动机的宏状态表示(见图中的实线椭圆框中的部分),而每个生长单元中的各个叶元则以双尺度自动机的微状态表示(见图中的虚线框中的部分)。其中一个叶元由一个节间和侧生的器官(如叶、花和腋芽等)组成,每个芽具有一种生理年龄,且由相同生理年龄的芽发育的枝条具有相似的形态特征。由于这种生理年龄相当于在植物生长中出现的次序,因此侧芽的生理年龄大于或等于主干。图1中每个宏状态的生理年龄以 p 表示,而自动机的开始状态为生理年龄为1的宏状态,其中每个数字表示每种状态的最大重复次数。当某种状态循环结束后,主轴的生长就进入箭头所示的下一状态,一直到最大的生理年龄5。由于生理年龄为5的宏状态为自动机的终止状态,这时即使植物的生长周期数增加,主轴上也不会再有新的生长单元出现。主轴上芽的生理年龄决定了对应的侧枝由哪种宏状态开始生长。由于主轴的顶芽和侧芽并行地生长,从而形成分枝结构。

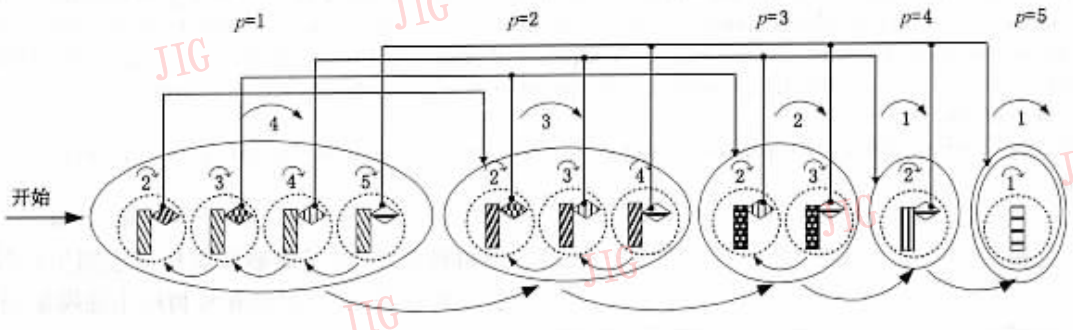


图1 具有5种生理年龄的双尺度自动机实例

对于双尺度自动机,顺序模拟算法从生理年龄为1的宏状态开始,然后在每个周期产生一个生长单元,并逐个地产生每个生长单元中的节间。这种顺

序模拟的过程与植物的生长过程较为接近。

2.1 基于子结构的构造过程

子结构的构造是通过轴和较简单子结构的组合

来实现的,显然植物中最简单的子结构为没有侧枝的单轴结构,而一般的子结构轴上只有侧芽产生侧生子结构。如果生长轴所对应的宏状态的循环结束,那么就进行状态跳转,产生生理年龄不同的顶端子

结构(如图 2 所示),其中每个子结构具有一个生长年龄 i (即子结构自出现以来经过的周期数)和一个生理年龄 p (即其生长轴的生理年龄)。由此可见,生长年龄和生理年龄是子结构的两个重要属性。

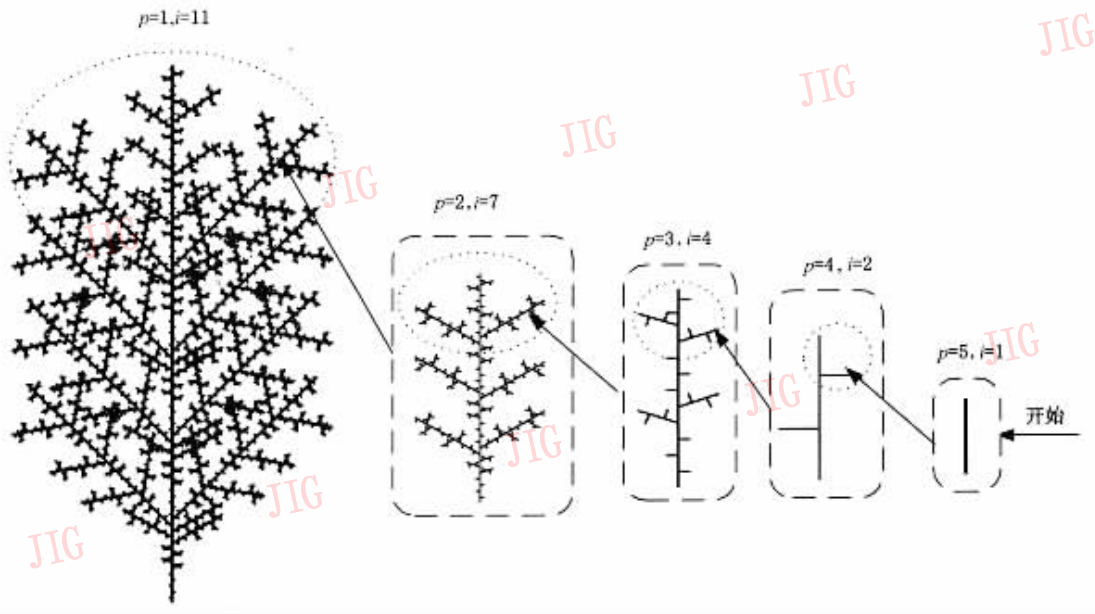


图 2 子结构算法示意图

通过以下两个步骤来进行子结构构造:

(1) 生长轴每个节间的位置、方向的确定 在“青园(GreenLab)”植物生长模型中,每个给定生长年龄和生理年龄的节间长度可以根据模型计算得到,也可以设置为任意值。在给定长度的情况下,每个节间的位置、方向与生长轴的弯曲形状有关,后者可根据重力影响计算,其相关的参数有轴的弹性模量、离垂直方向的初始方向角等。

(2) 轴上的所有子结构的粘贴 每个子结构的信息包括子结构的两个重要属性、空间方向、以及在轴上的粘贴位置。侧生子结构的生理年龄为对应的腋芽的生理年龄,而生长年龄则和其在轴上的位置相联系,即越靠近轴的顶部,出现得越晚,生长年龄越小,而顶端子结构的生理年龄即为如图 1 所示跳转后宏状态的生理年龄。

图 2 为与图 1 所示的自动机对应的子结构构造过程,其与顺序模拟不同的是,子结构算法是从最大的生理年龄开始,由于对应的生长单元中不可能有生理年龄更大的腋芽,因此这种子结构没有侧枝,是最简单的结构。在下一步构造生理年龄为 4 的结构时,由于其中有生理年龄为 5 的腋芽,子结构($p=2, i=2$)中还包含了($p=5, i=1$)的子结构,因此可以直接调用已生成的

结构。图 2 中箭头指出了子结构的调用关系,比如, ($p=1, i=11$) 的结构中包含了 3 个 ($p=2, i=7$) 的子结构,而后者又包含了 3 个 ($p=3, i=4$) 的子结构,分别见虚线框中的部分。此外,生理年龄为 1 的轴上,其实还直接调用了生理年龄为 3, 4, 5 的结构。

子结构的确定则和叶序角度、侧枝角度以及轴本身的方向有关(如图 3 所示),其中, α 表示侧枝角度,即子结构与生长轴的夹角, β 表示生长轴本身与 z 轴的夹角, γ 表示叶序角度,即围绕生长轴上下两片相邻叶子间的角度,这里为 180° ,而子结构在所属结构中的方向就是由这些角度联合决定的。

已生成的所有子结构均按统一的初始方向存放,当被调用到其他子结构中时,将经过由上述角度决定的旋转变换,旋转矩阵可由下式计算:

$$\mathbf{V} = \mathbf{V}_\beta \cdot \mathbf{V}_\gamma \cdot \mathbf{V}_\alpha = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \cdot \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \quad (1)$$

然后根据旋转矩阵和子结构在轴上的位置就可以对已有子结构的三维信息进行旋转平移操作。注

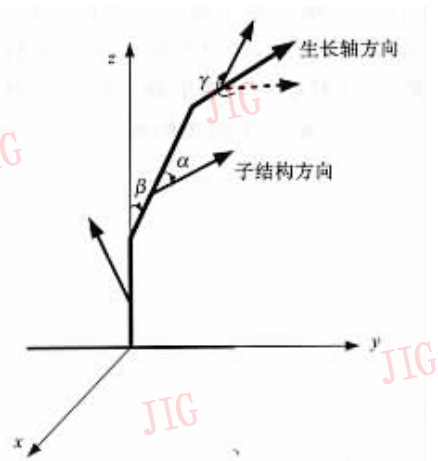
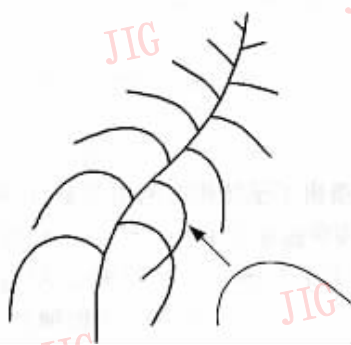


图3 子结构方向示意图

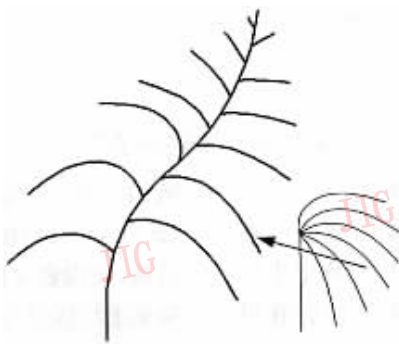
意这里没有对子结构进行拉伸变形。

2.2 弯曲枝条的子结构实现

如前一节所述,生长轴的弯曲形状与轴的初始



(a) 由单个子结构粘贴引起的
“错误”的弯曲



(b) 选取初始角最近的弯曲
子结构产生的正常弯曲



(c) 子结构算法产生的
弯曲枝条实例

图4 弯曲枝条的子结构方法模拟

尽管预设的初始方向越多,子结构的弯曲形状越具有真实感,但占据的时间和空间也都会增加,然而一些先验知识有助于设定初始角的变化范围以及离散化程度。

2.3 重复生长的模拟

当侧芽的生理年龄与生长轴相同时,其长出的侧枝具有与主轴相同的特性,在植物学上称这类生长为重复生长,而这种重复生长可以导致自相似分形结构。由于植物的重复生长并不是无限制进行下去的,因此在模型中具有参数“重复生长级数”,表示重复生长的最大次数。图5(a)、图5(b)、图5(c)分别是重复生长级数为2,1,0的结构。对于重复生长的子结构,模拟时,主轴只调用重复生长级数较低一级的侧枝,比如图5(b)中重复生长的侧枝调用的就是如图5(c)所示的子结构,而重复生长级数为0的

状态离垂直方向的角度有关,但是经过式(1)所述的变换后,子结构在所属结构中的实际方向是任意的。这意味着,如果某个子结构轴的弯曲预先按一个给定的初始角计算,那么在进行变换后,其形状会显得不自然(如图4(a)所示)。由于顺序算法是根据当前的方向实时地计算枝条几何形状,因此不存在这一问题。那么,怎样实现子结构算法下的枝条正常弯曲?解决方法是,对于相同属性的子结构,可事先计算若干个不同初始角的弯曲形状(如图4(b)所示)。图4(b)显示了根据7个不同的初始角计算的枝条弯曲形状。当子结构被调用时,可事先在已有的不同方向中找一个与实际的方向最接近的子结构,这样就能够避免出现枝条端点方向与垂直方向正向夹角大于 180° 的“错误”弯曲。利用这一方法,尽管子结构的变换是刚性的,但仍可以构造柔软的树结构。图4(c)给出了一个弯曲枝条模拟的三维实例。

结构则不再有生理年龄与轴相等的侧枝。图5(d)为重复生长级数为1的结构的一个三维几何形态。

2.4 花序的子结构模拟

花序是植物学上的重要特征,实践证明,双尺度自动机不仅可以模拟植物结构,也可以模拟花序结构^[9]。由于花序中有很多重复结构,因此子结构算法也可以有效地用于花序的构造。事实上,Prusinkiewicz曾利用“分枝映射(Branch mapping)”产生花序中的同级结构^[10],即认为短枝和长枝的顶部相同,这样虽然使得同级结构可以用相同的函数产生,然而对于长短枝则缺乏定量的描述,而在本文的模型中,结构是以生长年龄和生理年龄区分的,其不仅都是以相同的过程产生,而且方法可以应用到花序和复杂树结构的构造。

图6为由子结构方法构造的3种基本的花序结

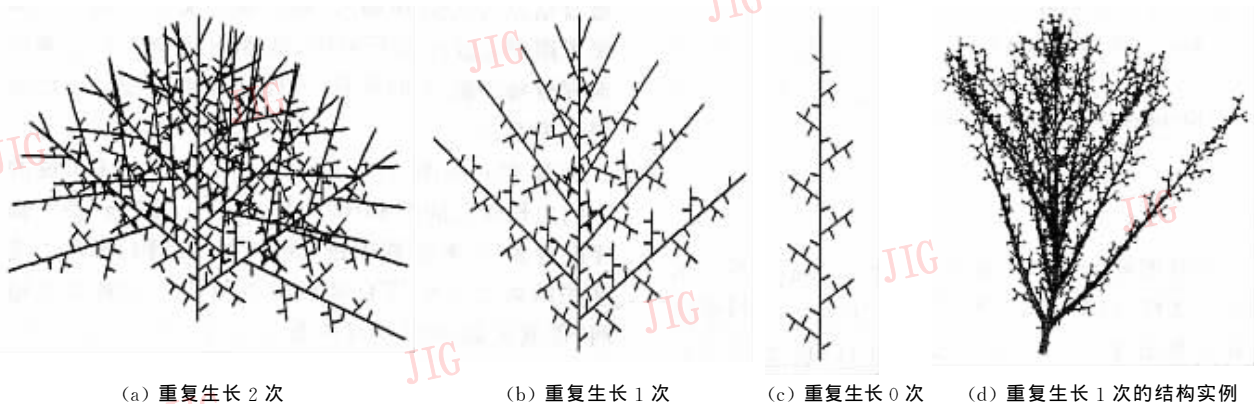


图 5 重复生长结构的子结构方法实现



图 6 花序结构的子结构方法实现

构。由图 6 可以观察到,伞形花序中包含了多个重复的单花,而复伞形花序又包含了多个伞形花序,这种花序中的重复结构,使其成为应用子结构方法很好的对象。

3 子结构的存储与检索

从更广的意义看,植物中的每个器官(例如花、叶、节间)都是子结构,只是子结构中仅包含一个元素。为进行区分,可为每个子结构增加标识,取值分别为“子结构”、“节间”、“叶子”、“雌花”和“雄花”,而每个子结构的特征有子结构的生理年龄、生长年龄、初始生长角、所属植物的生长年龄等等;几何数据有子结构的旋转矩阵、位置、大小等。统计结果表明,描述所有这些信息共占用 80 Byte 左右,而且其中包含的每个子结构也占用这么多空间。描述子结构的数据结构如下:

```
struct SUBSTRUCTURE
```

```
{
    int      symbol;      //子结构标识;
    int      length;     //包含的子结构个数;
```

```
    int      id;         //子结构特征;
    float    m;         //旋转矩阵;
    float    sz;        //对于器官为器官大小,而对于子结构则无意义。
    SUBSTRUCTURE s[length]; //所包含的子结构;
}
```

在模拟过程中,每当调用子结构时,如果直接将子结构数组中的数据进行旋转变换,然后再添加到当前结构中去,则会消耗大量的内存和 CPU 时间,而实际上,在这个过程中,可以只保存所用到的子结构特征。这样可首先将每个被调用的子结构的特征写入一个文件中,称为“子结构描述文件”,然后需要时,就可以根据输入的子结构特征,递归地从子结构描述文件中检索出该子结构各个器官的几何信息。当子结构的标识为一般器官时,则调用返回。子结构递归的检索过程如下:

```
Retrieve_Substructure (SUBSTRUCTURE S)
{
    if S.symbol<> '子结构'
        Output_Substructure(S); //输出器官的信息
    else
```

```

for  $i=1$  to  $S.length$ 
  Read-substructure( $S.s[i]$ ); //读取子结构数组中
                             每一个元素的特征
  Retrieve-Substructure( $S.s[i]$ );
end
end
}

```

器官的输出可以直接在屏幕上绘制,或者将结果写入文件。如果直接进行绘制,则不仅可以根据按生理年龄来区分节间的线条的颜色(如图 2 所示),也可根据器官大小与方向绘制三维的器官,器官的颜色由其生长年龄控制(如图 7 所示),而器官的形状则可以从一个标准的器官库中读取。如果将每个

器官信息写入到所谓的“器官描述文件”,那么就便于用专业软件进行渲染。显然,由于该文件是顺序地存放每个器官的信息,因此其大小与器官个数呈线性关系。

图 7(a)~图 7(d)给出了以子结构方法生成的不同生长阶段的植物图形,图 7(e)~图 7(g)是生长年龄为 20 的树结构中包含的一些子结构。虽然生成这些结果的双尺度自动机参数与图 1 中的基本相同,但其宏状态的重复次数需改为 $[10\ 5\ 3\ 1\ 1]$,以便使得植物能生长 20 个周期。根据植物学家的名字命名的 RAUH 模型^[11]就是一种自动机模拟的植物构造模型,也是一种很常见的结构。

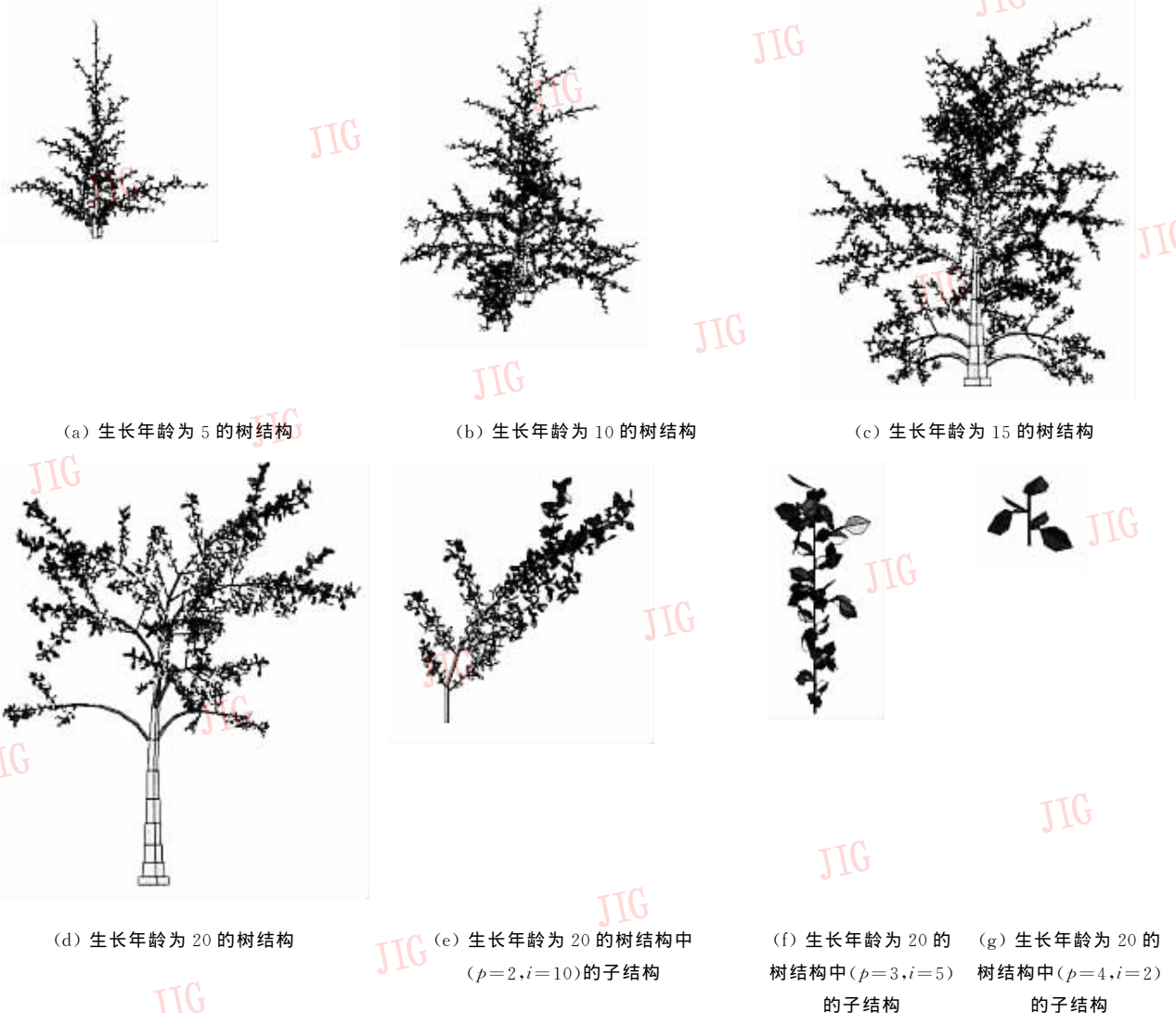


图 7 以子结构方法实现的部分植物和枝条三维结构

每个子结构包含的节间数计算方法在文献[8]中已有介绍。一般在植物模型中,器官个数用于生物

产量的计算,但在这里则用于分析算法的效率。本文在一次模拟中,生成了如图 7 中生长年龄

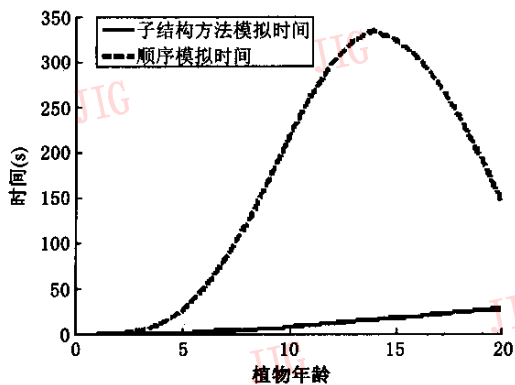
从 1 到 20 的所有植物结构,并将结果存入同一个子结构描述文件,计算时间为 18.2 s,生成扩展名定义为“.str”的子结构描述文件大小为 1548KB,而从这个文件得到的生长年龄为 1~20 的所有树结构的扩展名定义为“.lig”的器官描述文件共占用空间为 14.01MB,是“.str”文件大小的约 9 倍。

为进行进一步结果分析,可每次仅计算一个给定年龄的植物结构,这样生成的“.str”文件中仅包

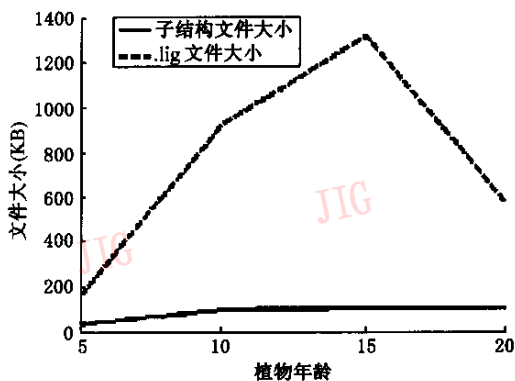
含一棵植物及其中所有子结构的信息,其结果见表 1 以及图 8。表 1 中节间的个数随着植物生长年龄的增加呈先上升又下降趋势,其原因是由于根据自动机的参数,当生长接近结束时,新的节间数目减少,而且已停止的子结构会自动脱落,因而图 7(d)中的结构比图 7(c)更为稀疏。图 8 则将表 1 的结构以图形的方式直观地显示出来,从中可以看出子结构算法在计算时间与存储空间上具有很大的优势。

表 1 以子结构方法计算几何结构的结果

树的生长年龄	包含的节间数	几何结构的构造时间(s)	产生的子结构描述文件大小(KB)	树结构的器官描述文件大小(KB)
20	5021	3.55	109	573
15	11 209	1.57	111	1 318
10	7 826	1.69	101	925
5	1 169	0.75	36	165



(a) 生长年龄 1~20 的模拟时间比较



(b) 生长年龄为 5, 10, 15, 20 的文件大小比较

图 8 子结构算法与顺序模拟算法性能比较

由图 8(a)可见,子结构算法的模拟时间明显比顺序算法模拟时间要少,其原因是由于顺序算法的耗时与节间数目呈线性关系,而子结构算法只对生长轴顺序模拟一次,其余工作就是子结构的粘贴,因此其耗时主要与子结构的个数有关;而图 8(b)中,如前所述,“.lig”文件大小也和节间数呈线性关系,但是由于“.str”文件是以子结构为单位来记录特征,因此压缩了数据信息。实际上,如果需要,还可以从文件中单独输出枝条结构(如图 7(e)、图 7(f)、图 7(g)所示)。

本文所有的插图和表格都是用作者自行编写的名为“GreenLab”的软件生成,软件的运行环境为 Matlab,操作系统为 Win2000,主机 CPU 为 Intel Pentium 4,系统内存 256MB。

4 结论与展望

本文系统地介绍了以子结构为单位计算和存储植物几何结构的算法,并讨论了对于重复生长、花序以及弯曲枝条的模拟,模拟结果说明,该方法具有通用性,本文还通过一个实例说明了该算法的有效性。尽管本文中植物的结构是基于双尺度自动机模型的,但是这一思想也可以用于基于其他模型的形态构造,例如 L 系统。当然,对于一棵简单植物结构,例如像向日葵、玉米这样的单枝结构,子结构方法虽然可以实现,但是没有明显的优越性。

由于文中的子结构是在确定性生长情况下得到的,因此属性相同的子结构只有一个,而实际植物的生长则或多或少具有随机性,比如某个节间上的侧枝的个数服从某种分布。尽管很多作者通过增加几

何参数的随机性来生成真实感的植物形态,但由于“GreenLab”模型是通过模拟芽的生长变化,比如死亡、休眠等来产生随机的拓扑结构,因而更加接近真实植物的生长。对这样产生的植物形态,可以将本文的方法加以推广,即用随机子结构算法实现快速构造。

参 考 文 献

- 1 Cohen D. Computer simulation of biological pattern generation processes[J]. *Nature*, 1967, **216**:246~248.
- 2 Virtual Terrain Project[EB/OL]. <http://www.vterrain.org/Plants/plantsw.html>
- 3 De Reffye P, Houllier F, Blaise F. Modelling plant growth and architecture: some recent advances and applications to agronomy and forestry [A]. In: Second International Symposium on Models for Plant Growth, Environmental Control and Farm Management in Protected Cultivation [C]. Wageningen, Netherlands. 1997.
- 4 De Reffye P, Dinouard P, Jaeger M. Basic concepts of computer plant growth simulation[A]. In: Computer Graphics: "Where do we go now that we've arrived? NICOGRAPH' 90[C], Tokyo, Japan, 1990: 219~234.
- 5 Prusinkiewicz P, Lindenmayer A. The Algorithmic beauty of plants[M]. New York: Springer-Verlag, 1990.
- 6 De Reffye P, Edelin C, Francon J, *et al.* Plant models faithful to botanical structure and development[J]. *Computer Graphics*, 1988, **22**(4): 151~158.
- 7 赵星, De Reffye P, 熊范纶, 等. 虚拟植物生长的双尺度自动机模型[J]. *计算机学报*, 2001, **24**(6):608~617.
- 8 Yan H P, Barci J F, de Reffye P, *et al.* Fast algorithms of plant computation based on substructure instances [A]. In: Proceeding of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision [C]. Plzen, Czech, 2002, **10**(3):145~153.

- 9 赵星, De Reffye P, 熊范纶等. 基于双尺度自动机模型的植物花序模拟[J]. *计算机学报*, 2003, **26**(1):116~124.
- 10 Prusinkiewicz P, Mundermann L, Karwowski R, *et al.* The use of positional information in the modeling of plants [A]. In: Proceedings of ACM SIGGRAPH 2001 [C], New York, ACM Press, 289~300.
- 11 Hallé F, Oldeman R, Tomlinson P B. Tropical trees and forest: An architectural analysis [M]. Berlin: Springer-Verlag, 1978.



康孟珍 1975年生,2003年获中国科学院自动化所博士学位,现为首都师范大学讲师。主要从事虚拟植物生长建模与科学可视化研究。



Philippe de Reffye 1946年生,1974年获法国国际农业研究发展中心博士学位,高级研究员,博士生导师,1998~2001年任中国科学院自动化所中法联合实验室法方主任,现在法国国家信息与自动化研究院工作。主要从事虚拟植物生长建模研究。



胡包钢 1955年生,1993年获加拿大McMaster大学博士学位,研究员,博士生导师,IEEE高级会员,中国科学院自动化所中法联合实验室中方主任。从事模糊控制、模式识别、植物生长的可视化研究、计算机图形学。

赵星 1974年生,2001年获中国科技大学博士学位,现在日本岩手大学从事博士后研究工作。研究领域包括植物生长的可视化研究、计算机图形学。