

6-1-2020

## Fast Tree Modeling Method Based on Simple Freehand Sketch

Ma Ling

*School of Computer and Information, Hefei University of Technology, Hefei 230009, China;*

Xiaoping Liu

*School of Computer and Information, Hefei University of Technology, Hefei 230009, China;*

Xiaoping Liu

*School of Computer and Information, Hefei University of Technology, Hefei 230009, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

## Fast Tree Modeling Method Based on Simple Freehand Sketch

### Abstract

**Abstract:** Aiming at problems such as complicated process and professional knowledge requirements in tree modeling, *a quick tree modeling method for non-experts was proposed, combining branch library with simple sketch to make non-experts create tree model quickly.* The method preprocessed the sketch to eliminate noise and error points and identify the strokes, the trunk and branch were reconstructed, child three-dimensional profile was built through the two-dimensional canopy profile, under the constraint of the child profile, suitable branches from library and insert were extracted into the branches, consequently the whole tree model was completed. Experimental results show that this method can make non-expert quickly create a complete tree model.

### Keywords

tree modeling, simple sketch, branch library, sketch-based modeling, fast modeling

### Recommended Citation

Ma Ling, Liu Xiaoping, Liu Xiaoping. Fast Tree Modeling Method Based on Simple Freehand Sketch[J]. Journal of System Simulation, 2017, 29(8): 1667-1676.

# 基于简单手绘的树木快速建模

马玲, 李书杰, 刘晓平

(合肥工业大学计算机与信息学院, 安徽 合肥 230009)

**摘要:** 针对树木建模过程繁琐、专业知识要求高等问题, 提出一种面向非专业用户的树木建模方法, 将简单手绘与枝叶库结合, 实现让普通用户快速创造树木模型的目的。对用户的手绘草图预处理, 消除笔画噪声点和错误点并完成笔画识别, 对主干和分支进行三维重建, 再依据二维轮廓构建三维子轮廓, 以此为约束, 从枝叶库中提取合适的枝叶模型插入到分支插入点上, 从而完成整个建模过程。实验表明, 该方法能让非专业用户在几十秒内建模出完整的树木模型。

**关键词:** 树木建模; 简单手绘; 枝叶库; 手绘建模; 快速建模

中图分类号: TP391.9      文献标识码: A      文章编号: 1004-731X (2017) 08-1667-10

DOI: 10.16182/j.issn1004731x.joss.201708005

## Fast Tree Modeling Method Based on Simple Freehand Sketch

Ma Ling, Liu Xiaoping, Liu Xiaoping

(School of Computer and Information, Hefei University of Technology, Hefei 230009, China)

**Abstract:** Aiming at problems such as complicated process and professional knowledge requirements in tree modeling, a quick tree modeling method for non-experts was proposed, combining branch library with simple sketch to make non-experts create tree model quickly. The method preprocessed the sketch to eliminate noise and error points and identify the strokes, the trunk and branch were reconstructed, child three-dimensional profile was built through the two-dimensional canopy profile, under the constraint of the child profile, suitable branches from library and insert were extracted into the branches, consequently the whole tree model was completed. Experimental results show that this method can make non-expert quickly create a complete tree model.

**Keywords:** tree modeling; simple sketch; branch library; sketch-based modeling; fast modeling

## 引言

树木是现实生活中最为常见的景观之一, 随着古文化保护、数字娱乐等虚拟场景应用的普及, 三维植物模型的建模方法成为虚拟现实领域广泛研究的课题之一<sup>[1-5]</sup>。在虚拟场景中, 通常需要能够快速建立具有较高真实感外观的树木模型。基于手绘草

图或测量数据(包括树木图像、点云等)建模是快速建立树木模型的常用方法, 例如 Livny 用点云数据重建现实中树木的三维模型<sup>[1]</sup>, 陈等人用手绘草图和树木数据库建立与手绘草图相似的树木模型<sup>[2]</sup>, Liu 面向缺乏设计知识的非专业用户, 综合利用树木图像和手绘快速获得树木模型<sup>[3]</sup>。从研究现状上来看, 虚拟漫游场景中的树木快速建模仍面临专业知识要求高、建模过程繁琐以及真实感不强等问题。本文介绍一种面向非专业用户的树木快速建模方法, 该方法仅需要用户绘制简单的几笔树木草图, 即可快速的生成与草图外形相似的三维树木模型。



收稿日期: 2015-05-12      修回日期: 2015-07-06;  
基金项目: 国家自然科学基金面上基金(61370167),  
国家自然科学基金青年基金(61305093), 国家科技支  
撑计划(2012BAJ08B01);  
作者简介: 马玲(1990-), 女, 安徽蚌埠, 硕士生, 研  
究方向为计算机图形学。

<http://www.china-simulation.com>

• 1667 •

基于观察发现, 决定树木主要外观形状的是树冠轮廓、主干及其分支, 而之后生长出来的细小枝叶, 只是起到丰富充实的作用, 对整体的外观形态影响不大。鉴于上述性质, 本文提出一种基于简单手绘的树木快速建模方法。该方法将树木结构中数量众多, 对树木形态不具备主要影响却耗费大量建模时间的细小枝叶预先建模出来存储在枝叶库中, 而由用户自己绘制能决定树木形态的树冠轮廓、主干及其分支, 这样用户就可以方便自主地控制树木形状。本文方法输入简单, 面向的用户群广泛, 不需要用户任何的专业技术, 就能够快速建出树木模型。本文方法流程如图 1 所示。

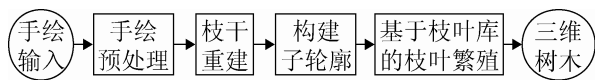


图 1 方法流程图

Fig. 1 The pipeline of method

本文提出的树木建模方法特色在于:

① 用户仅需给出树木的简单手绘并选择树种, 即可在枝叶库的支撑下获得具备用户手绘特点且真实感较好的树木模型。简单的输入要求和容错性较强的预处理步骤使得本文方法便于非专业用户使用。

② 由于采用了枝叶库, 使得树木建模过程中无需涉及以往算法中最耗时的末端细节树枝和树叶构造, 这是本文方法快速的关键所在。

## 1 相关工作

树木建模方法大致可以分为基于规则的建模方法、基于测量数据的建模方法和基于手绘的建模方法。

基于规则的建模方法是较早出现的经典建模方法, 有着悠久的历史与深厚的理论背景, 目前比较成熟的建模方法主要有 l-system 和参数集方法<sup>[4]</sup>。1968 年 Lindermay<sup>[5]</sup>提出了 L-system 字符串迭代替换系统, 其最初是用来表示细胞分裂行为, 后来发展为模拟植物的生长和分支结构<sup>[6]</sup>。1971 年 Honda<sup>[7]</sup>通过定义一系列的参数控制植物的形态, 生成了丰富的树形结构。1984 年 Aono<sup>[8]</sup>在 Honda

模型的基础上提出了构造植物形态的 A 系统。基于规则的建模方法制作出来的树木模型真实感强, 但是需要专业人员制定出模拟植物生长的规则或者描述植物形态的参数, 这对于普通用户来说, 基本很难完成, 而且用户对建模出来的模型很难有整体的掌控。

基于测量数据的建模主要有基于图像的建模和基于点云的建模。测量数据, 即图像和激光扫描数据, 是建模真实植物的可靠依据<sup>[9]</sup>。2001 年 Shlyakhter<sup>[10]</sup>利用多幅校准后的树木图像构建视见凸包, 并在其内部使用 l-system 生长枝叶实现树木的近似重建。2007, Nebert B<sup>[11]</sup>与 Deussen 等人使用图像构造引导粒子流的方向场, 利用粒子运动生长出植物骨架再利用异速生长规律构建三维模型。Livny<sup>[11]</sup>于 2010 提出基于三维点云数据的树木骨架自动生成算法, 随后又提出了树木轻量建模<sup>[12]</sup>概念, 利用采集的树木点云构建 lobes, 在其内部填充从枝叶模板库中选取合适的枝叶达到树木的重建。以上方法都是对真实树木的重建, 需要对树木进行信息采集, 存在采集设备价格昂贵、采集数据来源噪音大等问题。

基于手绘的建模方法是树木建模中较为创新和灵活的方法。2005 年, okabe<sup>[13]</sup>在论文中首次提出树枝生长间距最大化假设来作为 3 维子枝重建角度分布的规则, 利用该方法生成一棵树需要几分钟甚至十几分钟, 而使用本文的方法, 系统生成一颗完整的树木模型基本在几十秒时间内。2008 年 Tan<sup>[14]</sup>利用用户在树木图片上标注树冠轮廓和主干达到简化图像分割算法的目的, 从而实现树木的快速重建。同年 Chen<sup>[2]</sup>等利用无向图模型和预先定义的树木库推测用户手绘树木的树枝形态, 并通过树枝繁殖和叶子生长得到树木模型, 其方法中涉及到繁琐的细枝和叶子的繁殖过程, 耗时多, 本文方法利用预先定义的枝叶库避免该过程, 从而提高了速度。Wither 等<sup>[15]</sup>于 2009 年提出手绘设计树的新范式: 从轮廓推断结构。用户从主干到叶子逐级绘制轮廓, 系统再结合一定的推理和植物学规律生成

完整的树木模型, 该方法需要用户绘制多级轮廓, 输入麻烦。而我们的方法, 仅需要用户绘制一个整体的树冠轮廓, 即可自动计算出各个分支的轮廓, 更加方便快捷。

比较来说, 基于规则的建模方法规则难制定, 模型整体形态可控性差, 基于测量数据的方法, 测量数据获取困难并且处理繁琐, 基于手绘的方法灵活, 交互性好。然而大多基于手绘的建模方法, 都要求用户有一定的绘画基础或者绘制大量的树木结构细节, 面向的用户群狭窄, 而且对于用户的手绘草图预处理机制也比较简单, 很少对手绘过程中出现的问题进行总结和处理。针对以上问题, 本文提出了基于简单手绘的植物快速建模方法, 该方法输入笔画简单, 用户群广泛, 有较为完善的容错机制并通过预先定义的枝叶库完成细小枝叶的繁殖, 大大缩短了建模时间。

## 2 手绘预处理

本文提及的简单手绘包括树冠轮廓、树木的主干及其分支(称为一级分支)三个部分, 如图 2 所示。它是由单笔画绘制而成, 对于用笔刷绘制的多像素草图可以首先用文献[3]中的方法转化为单像素枝干骨架后再做处理。

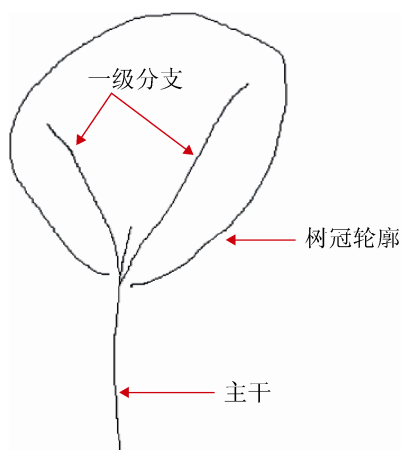


图 2 简单树木手绘的定义

Fig. 2 The definition of simple sketch of tree

本文输入的手绘草图是非专业用户快速勾勒而成, 存在一定的模糊性和随意性。为便于预处理,

对简单树木手绘做出如下合理限制及假设:

- ① 手绘中只包含一个主干;
- ② 手绘只包含一个整体树冠轮廓;
- ③ 手绘中仅可以包含一级分支, 不包含一级以上的分支;
- ④ 树冠轮廓和树枝不相交并且接近闭合;
- ⑤ 每个笔画均为连续画出, 不存在断点。

系统对用户手绘草图实时采集时, 会同时受鼠标采样率和用户绘制熟练度的影响, 导致采样点中包含大量冗余甚至错误点, 严重影响后期的建模效果, 所以需要在不明显改变形状的前提下, 对草图进行精简优化处理。将三维重建前对草图处理的一系列操作统称为手绘预处理, 预处理过程包括冗余点消除、手绘草图平滑、笔画识别与分支末端修正。

### 2.1 冗余点消除及草图平滑

设用户输入的手绘有  $m$  个笔画, 用  $S_0 \dots S_{m-1}$  表示, 将笔画  $S_j (j \in \{0, 1, \dots, m-1\})$  用其采样点集合  $\{p_0, p_1, \dots, p_{n-1}\}$  标记, 其中  $p_i = \{x_i, y_i\}$ ,  $p_0$  表示笔画起始点,  $p_{n-1}$  表示笔画终点,  $\text{dis}(p_i, p_{i+1})$  表示  $p_i$ 、 $p_{i+1}$  之间的距离。

若  $\text{dis}(p_i, p_{i+1}) \leq k (i \in [0, \dots, n-1])$ , 则称  $p_{i+1}$  为冗余点, 即下一个采样点与当前采样点的距离小于阈值  $k$ , 则判断为冗余点, 如图 3 中的  $p_{i+1}$ 。

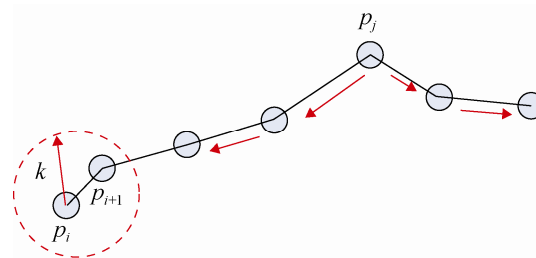


图 3 冗余点判断和采样点平滑示意图

Fig. 3 The schematic diagram for redundant point judgement and sampling point smoothing

对每条笔画从起始点  $p_0$  到终点  $p_{n-1}$  进行遍历, 将冗余点从采样点集合  $S_j$  中删除。算法复杂度为  $O(n)$ , 其中  $n$  为初始采样点数。本文设置阈值  $k=3$ 。

另外一方面,手绘草图中会出现因用户绘制过程中出现抖动等原因产生的噪声点,如图3中标记出的 $p_j$ 点。本文使用平滑滤波算法对这些噪声点进行修正,主要思想是利用中心点及其周围临近点的坐标进行加权平均,从而对中心点进行修正,达到平滑的作用。设当前点为 $p_i(i \in [2, \dots, n-2])$ ,选择点 $p_j(i-N < j < i+N)$ ,令:

$$p_i = \sum_{j=i-N}^{i+N} w_j p_j \quad (1)$$

本文 $N$ 设为2,权值模板设为0.07, 0.13, 0.60, 0.13, 0.07。

## 2.2 笔画识别与分支尾端修正

本文方法对用户绘制的笔画顺序没有要求,可以自动识别出用户绘制笔画表示的意义。具体识别方法如下:

- ① 包含最大纵坐标像素的笔画为轮廓;
- ② 包含最小纵坐标像素的笔画为主干;
- ③ 识别完主干和轮廓后,其余笔画均为一级分支;

为便于区别,将轮廓笔画记为 $O$ ,主干笔画记为 $M$ ,一级分支笔画记为 $B_i(t \in [0, k])$ , $k$ 为一级分支个数。

用户绘制一级分支靠近尾端时,这里的尾端是指靠近主干端,一旦出现抖动等情况,就极易形成明显的笔画错误,会导致生成的树木模型产生不符合自然规律的情况,如分支逆向穿过主干、分支和主干不相连等。

我们随机抽取多位无设计背景的非专业用户,每人分别绘制20副简单树木手绘。经统计分析,总结一级分支尾部出现的主要问题类型,分支笔画问题多为图4(a)、(b)、(c)、(d)四种或其组合。

图4(a)和(b)分别表示分离和交叉的情况,这是分支尾端最容易出现的两种问题。对于分离处理的机制是直接对尾端进行延长,求得和主干的交点,将交点插入到分支尾端作为新的尾点;对于交叉的情况,则直接求得分支和主干的交点,以交点将分支分成两部分,将长度短的部分去除即可。

图4(c)和(d)所示的勾和圈,属于聚点类型<sup>[16]</sup>,聚点多易出现在笔画提笔落笔或转折处,通常情况下,其附近采样点密度大于曲线其他部分。设分支笔画采样点数为 $N$ 、长度为 $L$ ,则平均采样点密度为 $N/L$ ;分支上任一采样点数为 $n$ ,长度为 $l$ 的折线段,其采样点密度为 $n/l$ ,其相对采样点密度为 $nL/IN$ 。从分支尾端开始,求线段相对采样点密度,若其大于给定阈值,则将该折线段上的所有采样点用其重心替换。用重心对聚点修正后还需对分支进行分离或交叉的判断和处理。

图4中分支问题情况经修正处理后对应的效果如图5所示。

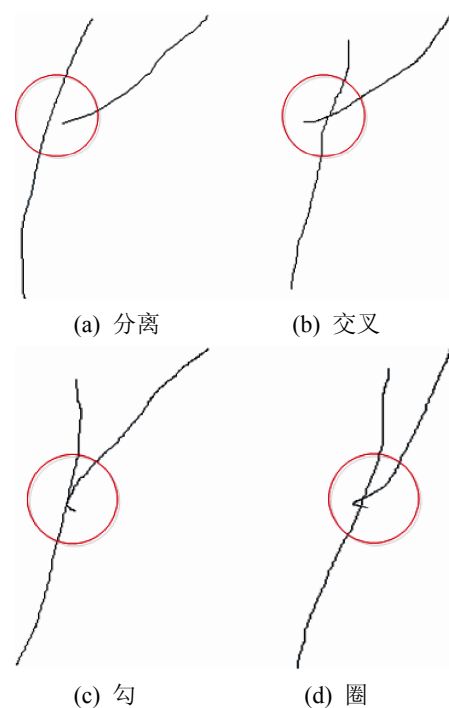
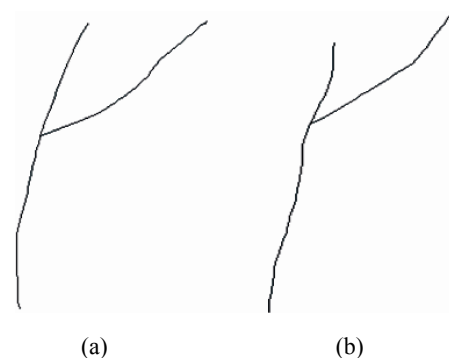


图4 一级分支笔画问题  
Fig. 4 The sketch problem of the first branch



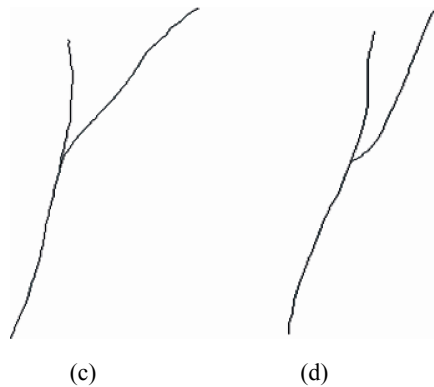


图 5 一级分支修正后效果图

Fig. 5 The sketch of the first branch after correction

### 3 三维重建

经过前期预处理后的手绘树木草图可以直接用来进行三维重建, 三维重建主要是依据用户的手绘二维树木草图推测三维结构, 包括枝干的重建和子轮廓的构建。

#### 3.1 枝干重建

枝干的重建是指将用户绘制的二维枝干形态转化为三维结构。主要涉及两方面工作: 一是给枝干赋予深度信息, 即  $Z$  轴的值; 另一方面是赋予枝干表示粗细信息的半径值。

用户手绘的枝干由二维采样点  $p_i(x_i, y_i)$  构成, 所有枝干均在  $XOY$  平面上, 通过给点  $p_i$  赋予深度信息  $z_i$  将其转化为三维枝干的节点  $p_i(x_i, y_i, z_i)$ 。对于主干, 使其在  $XOY$  平面上保持不动, 通过直接赋予深度值  $z=0$  映射到三维空间中。对于一级分支  $Bt$ , 将其在局部坐标系空间绕  $Y$  轴旋转, 局部坐标系以分支尾点  $bt_0$  为坐标原点, 具体旋转方式是将相邻两点构成的向量  $\vec{v}$  ( $\vec{v}$  由尾端指向顶端), 旋转后得到新的向量  $\vec{v}' = \vec{v} \times R_y(\alpha)$ ,  $R_y(\alpha)$  为  $3 \times 3$  矩阵, 如下所示:

$$R_y(\alpha) = \begin{bmatrix} \cos \alpha & 0 & -\sin \alpha \\ 0 & 1 & 0 \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix}$$

除了尾点  $bt_0$  保持不动外, 其余所有点依据向量旋转后的位置更新, 旋转后所有枝干节点均赋予  $z$  轴上的深度值。一级分支的旋转使最终

生成的树木模型在屏幕上的投影与用户的手绘二维树木形状产生偏差。我们通过制定旋转规则, 在保证树木模型真实感的前提下尽量减少偏差。旋转规则如下:

① 为了防止一级分支旋转造成树木模型与草图出现明显不符的情况, 对其旋转区域做出限定, 规定每一个一级分支随机旋转角度  $\alpha$  的范围  $[-60^\circ, 60^\circ]$ , 如图 6 所示。这样能保证旋转后树木模型的正视图和用户手绘的二维图形偏差不至于过大, 在用户视觉上可以接受的范围内;

② 通过设定角度  $\alpha=10^\circ$  作为分支之间的旋转最小角度差, 避免枝干集中旋转到一处。

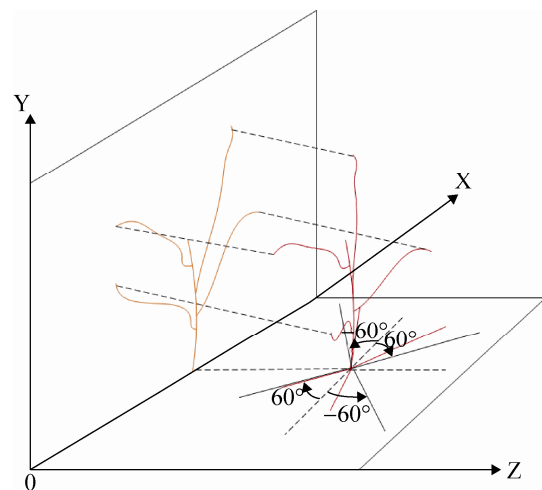


图 6 一级分支的旋转

Fig. 6 The rotation of the first branch

赋予枝干深度信息后, 还需要半径信息, 才能将线结构转化为体结构。上一节我们规定分支靠近主干的一端为尾端, 则另一端为顶端, 而主干则规定靠近地平面的一端为尾端, 另一端为顶端。现实中的树木枝干从尾端到顶端逐渐变细, 基于这一原则制定枝干半径生成规则。赋予枝干尾点的半径初始值  $R$ , 枝干中间某节点  $p_i$  处的半径  $R_i$ , 是由该点到顶点距离  $d$  与整个枝干长度  $l$  的比值和半径初始值决定的, 具体公式如下:

$$d/l = R_i/R \quad (2)$$

利用该公式生成的树枝半径从尾端到顶端逐渐递减, 主干半径设置最小值  $R/10$ 。

通过赋予节点半径值将线结构枝干绘制成六棱柱结构的三维枝干模型。具体绘制方式是以枝干上的节点  $p_i$  为中心，生成垂直于 Y 轴，半径为  $R_i$  的六边形，利用三角网格将相邻六边形顶点  $d_1 \sim d_6$  连接起来，生成由六棱柱表示的三维枝干。如图 7 所示。

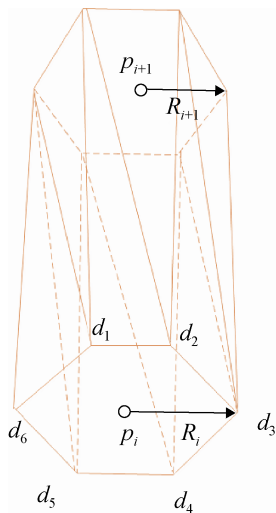


图 7 绘制三维枝干

Fig. 7 Drawing three-dimensional limb

### 3.2 构建子轮廓

用户手绘的二维整体树冠轮廓转化为围绕各个一级分支的三维子轮廓称为子轮廓的构建。子轮廓是作为后续计算一级分支上插入枝叶长度的约束条件。

首先获取手绘二维轮廓点序列，以轮廓最高点为中心点，在中心点左右两边取匹配点，依据每组匹配点构建截面域或截体域，在域的内部随机撒点，形成描述轮廓三维形状的点云集，再将该点云集依据点到一级分支距离最短原则分割成围绕各个级分支的子集，最后分别对各个子集使用 alpha shape 算法构建点云三维轮廓，以此作为一级分支的枝叶轮廓，简称子轮廓。

以轮廓中心点将轮廓分为左右两边，两边各取一点两两匹配，匹配原则是两点 Y 轴坐标最接近。若匹配点 Y 轴坐标相同，则构建垂直于 Y 轴，且经过两点，边长为两点 X 轴坐标距离差的正方形

截面域；若 Y 轴坐标不同，则构建以两点 Y 轴坐标差为高，以两点 X 轴坐标距离差为边长，且经过两点的截体域。整个草图构建出多个截面域或截体域，在所有域空间内随机撒点生成轮廓的三维点云集，记为 Cloud。Cloud 形成以后，将其依据一级分支定义距离函数，通过距离函数对点集进行分类，将 Cloud 分割成点云子集。

定义距离函数：

$$\text{dis}(c, Bt) = \min \{ \text{dis}(c, bt_i bt_{i+1}) \} \quad (3)$$

其中  $t \in [0, k)$ ， $i \in [0, n-1)$ ， $c(x, y, z)$  表示 Cloud 中的点； $bt_i bt_{i+1}$  表示分支  $Bt$  上相邻两点构成的线段； $\text{dis}(c, bt_i bt_{i+1})$  表示点  $c$  到线段  $bt_i bt_{i+1}$  的距离。公式(3)表示点云中的点到分支的距离函数，点  $c$  隶属于距离函数最小的  $Bt$  分支。

子集形成后，利用 alpha shape 算法提取子集外部边缘点并重建出三维网格模型。alpha shape 是由 Edelsbrunner, Kirkpatrick & Seidel (1983) 所定义的边缘轮廓提取算法，该算法与凸包提取算法相似，是其概念的一个泛化，它提取的轮廓形状不局限于凸形结构，因此适用于提取不规则的树冠轮廓。

对于同一个点集，通过调节 alpha shape 算法中关键参数  $\alpha$ ，可以得到不同的轮廓提取结果，当  $\alpha \rightarrow \infty$ ，提取的结果接近点集的凸包；当  $\alpha \rightarrow 0$ ，结果是点集自身。在本文方法中，若该值设置过大，将导致提取的轮廓过于平滑而无法较好的体现点集真正的外部形态；若设置过小，会造成轮廓内部有碎片(即与整体不连通的点、面或体)，而这些单独存在的碎片，会影响后期枝叶插入时轮廓相交性判断的有效性。通过多次实验发现，将  $\alpha$  的值定为最小值(即满足提取的轮廓为一个整体的最小的  $\alpha$  值)的 3 倍时，实验效果最佳。

图 8 展示子集轮廓构建过程中出现的效果图。(a)中红色的点是依据二维轮廓构建的三维点云，(b)中黄色和蓝色的点云子集是将初始点集根据最近距离原则分割后的两个子集，(c)是利用 alpha shape 算法提取的各个子集的轮廓。



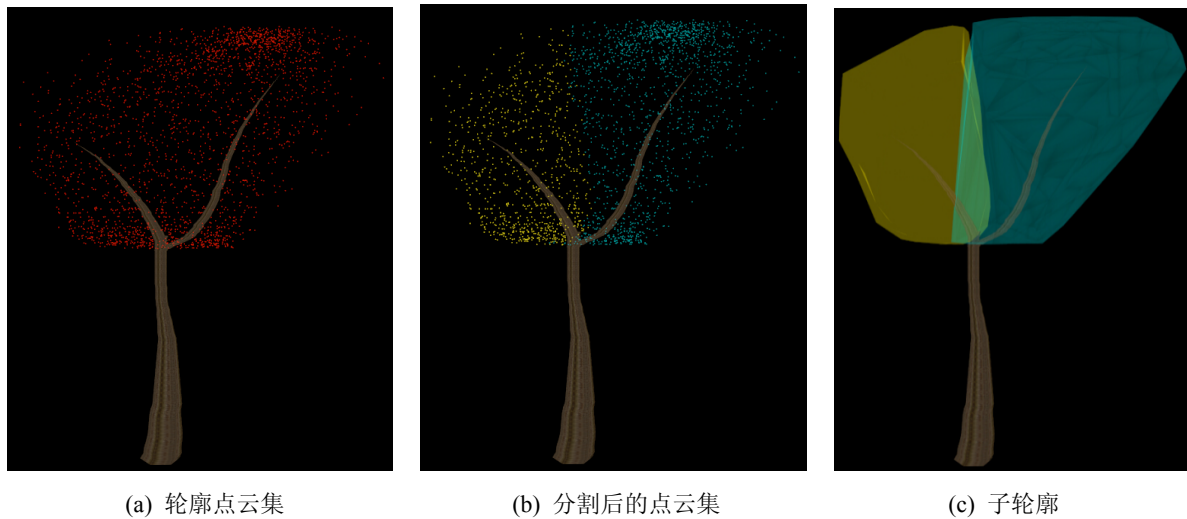


图 8 子轮廓构建过程

Fig. 8 Sub-contour construction process

## 4 基于枝叶库的枝叶繁殖

由用户手绘草图重建出来的树木模型仅包含主干及其上的一级分支, 而真实树木模型结构复杂, 大多包含多层枝干分支以及大量的树叶, 因此, 我们还需在重建的树木模型上进行枝叶的繁殖, 这里的繁殖是以插入枝叶的方式进行的。一级分支后的多级分支和叶子合并起来称为枝叶模型, 预先建模出大量的枝叶模型并存储在一起, 即形成枝叶库。枝叶库中存放着长度、半径各不相同的枝叶模型, 繁殖过程中根据筛选条件实时地将枝叶模型从枝叶库中调出来插入到一级分支上, 从而达到树木快速建模的目的。真实丰富的枝叶库是影响最终模型效果的关键因素。

### 4.1 构建真实感枝叶库

针对不同的树种, 从自然界拍摄或者网站下载









相应的树木图片作为模型制作的贴图素材和参照来源。

真实感的贴图制作主要有两方面, 一个是树干贴图, 树干由多个节段构成, 需将树干贴图制作成无缝贴图, 才能保证枝干整体的真实感; 另一个是树叶贴图, 树叶使用面片建模, 需制作树叶的平面透明贴图。

除了制作真实感贴图外, 还需要模拟真实枝叶形态结构。不同的树种, 在枝干形态和树叶排列模式上会有差异, 存在显著差异的是叶序模式, 不同的树种有其对应的叶序模式, 例如互生的银杏、簇生的广玉兰等。因此我们在模型制作时, 针对不同的树种, 参照其对应的真实树木图片构建枝叶形态。

建立出来的枝叶模型连同其长度信息、根部半径信息一起存储在枝叶库中。表 1 列出了部分枝叶库信息。

表 1 部分枝叶库  
Tab. 1 A part of foliage library

树种	枫树		银杏		杨树		广玉兰	
长度/cm	199	355	240	323	78	113	89	120
半径/cm	5	3	4	3	1	2	2	1
模型								

## 4.2 插入枝叶

插入枝叶是指从预先定义的枝叶库中筛选合适的枝叶插入到一级分支上,分为 4 个步骤:(1) 在一级分支上生成插入点;(2) 预先生成插入点处枝叶的朝向,称为插入点方向;(3) 以一级分支枝叶轮廓为约束,计算插入点处需放置的枝叶长度,称为插入点长度;(4) 根据插入点长度信息和半径信息制定筛选条件,选择适合的枝叶模型将其根部放置到插入点,并将其方向设置为该插入点方向。如图 9 所示,小圆圈表示插入点,黑色箭头表示插入点方向。

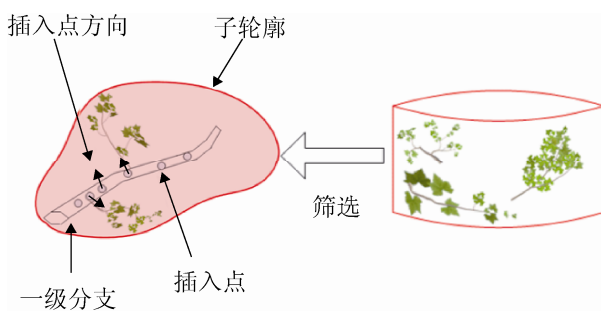


图 9 插入枝叶示意图

Fig. 9 The illustration diagram of inserting the foliage

### ①生成插入点

生成插入点是指在一级分支上选择合适位置的节点将其设置为插入点。为了使枝叶分布有自然的视觉效果,在确定插入点时需要同时考虑到均匀性和随机性。先将一级分支上的节点均匀分组,再从每个组内部随机选点,通过这种方式可以使得插入点的位置既能符合植物生长的随机性又符合均匀性。

### ②赋予插入点方向

树枝的生长具有随机性,因此插入枝叶模型的朝向也应具有随机性,但同时还必须进行一定的约束,否则会使得方向过于杂乱无章,反而有失真性,甚至可能会出现枝干逆向生长等问题。

插入点方向是以前一级分支的朝向(即一级枝干首尾两点构成的向量方向,由尾端指向顶端)为轴的一定角度范围空间内随机生成,即控制

插入的枝叶模型与父枝干的夹角范围。本文通过多次试验,得到实验效果较佳的范围空间 $[30^\circ, 70^\circ]$ 。

### ③计算插入点长度

使得三维树木正视图树冠形态接近用户手绘二维树冠轮廓的关键是依据子轮廓形状计算插入点枝叶长度。从插入点处发出沿插入点方向的线段,以此粗略模拟待插入枝叶的长度,子轮廓是围绕着一级分支的三角网格模型,由多个三角面片构成,利用线段和面片相交原理,探测能够与子轮廓相交的线段的长度作为插入点长度。

线段起始点定在待插入点的位置后,通过设置一个较小的递增长度  $\Delta l$ , 从线段起始点开始逐次递增一个  $\Delta l$ , 直到检测到该线段与子轮廓相交为止,此时记下线段的长度作为插入点枝干长度。本文  $\Delta l$  设为 1。

### ④插入枝叶

先通过插入点的半径和长度从枝叶库中筛选出适合该插入点的枝叶模型。设枝叶库中枝叶模型的长度为  $B_l$ , 根部半径为  $B_r$ , 插入点长度为  $l_i$ , 半径为  $R_i$ , 设置筛选条件:

$$\begin{cases} B_l < l_i \\ B_r < R_i \\ (B_l - l_i)^2 < \theta_l \\ (B_r - R_i)^2 < \theta_r \end{cases} \quad (4)$$

初始  $\theta_l = 5$ ,  $\theta_r = 1$ , 根据公式(4), 选择符合条件的枝叶模型,若没有搜索到符合条件,将初始  $\theta_l$ ,  $\theta_r$  适量放大,直到找到合适的模型为止。再将枝叶模型的根部放置到插入点,并在程序中实时地将模型方向设置为插入点方向。

## 5 实验结果与分析

为了验证本文方法的有效性,我们开发了基于 ogre3D 渲染引擎的手绘树木建模系统,该系统的软件配置: Windows XP 操作系统, Visual Studio 2008, Ogre1.6.5, CEGUI, 硬件配置: 2.67GHz CPU、4.0G 内存。我们邀请多名非专业用户对我们的方法进行实验,如图 10 所示是部分的实验效果图。

表 2 是对实验的数据进行分析, 我们展示的四组实验, 分别是杨树、枫树、广玉兰和银杏。系统根据用户的选择, 自动调用对应的枝叶库和树干纹理, 生成多个不同种类、真实感较强的树木模型。实验表明, 用户绘制的树木草图简单易画, 非专业用户也可以在短时间内完成, 而且生成的树木模型, 接近其绘制的二维形状, 模型生成的速度也十分快, 基本在几十秒的速度级别, 同时生成的树木模型也具有较高的真实感。

表 2 实验分析

Tab. 2 Analysis of experiment

实验组号	树种	笔画数	用户绘制时间/s	模型生成时间/s
第一组	杨树	8	18	33
第二组	枫树	5	14	26
第三组	广玉兰	7	20	41
第四组	银杏	5	15	22

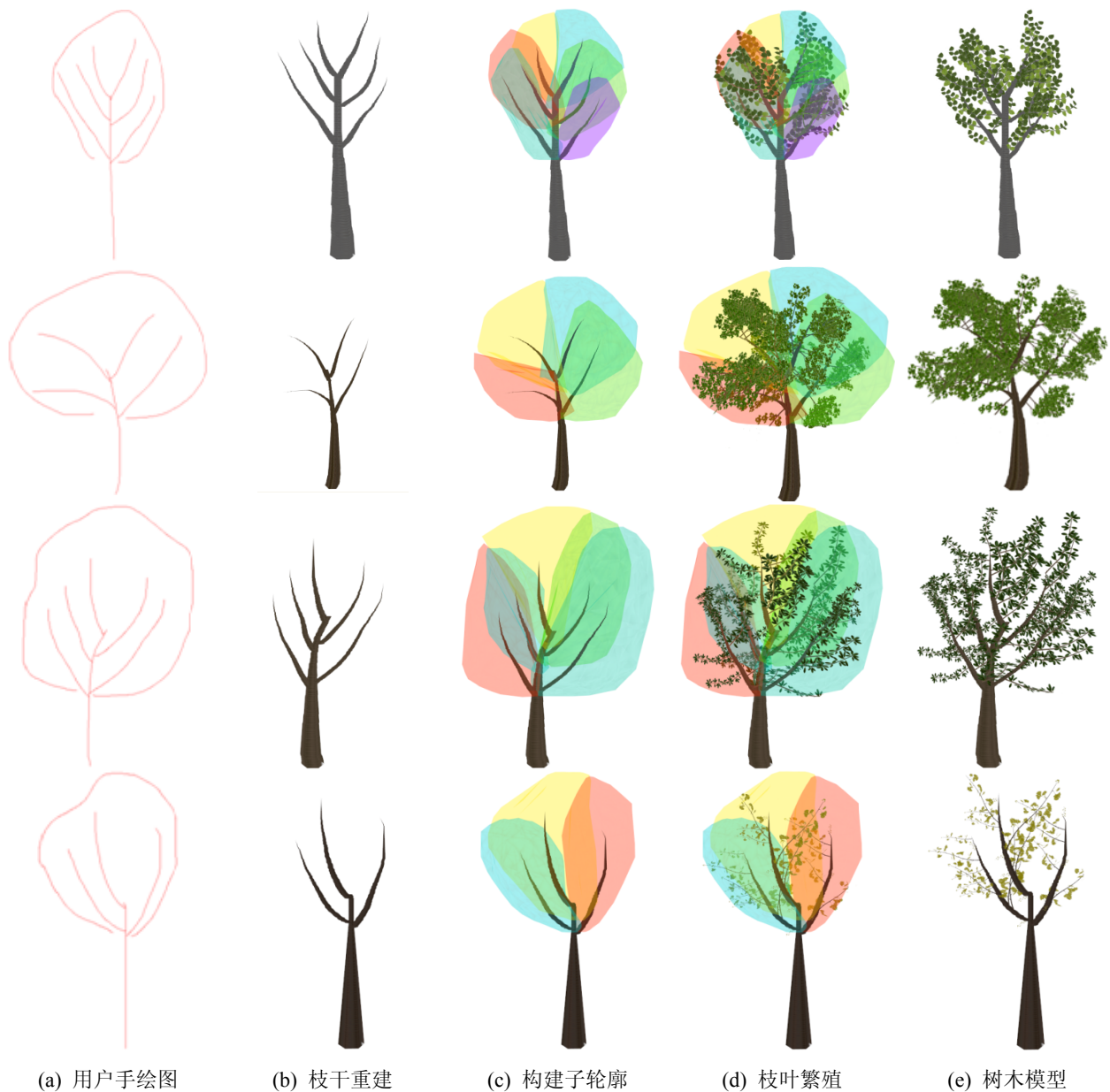


图 10 实验结果

Fig. 10 The results of experiment

<http://www.china-simulation.com>

## 6 结论

本文输入的草图笔画简单,没有树木细节的绘制,适合普通用户,且预处理有较强的容错机制,同时利用枝叶库达到快速枝叶繁殖的目的也是本文的一大特色。实验表明,利用本文的方法,非专业用户能在短时间内建模出自己设计的树木模型。

### 参考文献:

- [1] Livny Y, Yan F, Olson M, et al. Automatic reconstruction of tree skeletal structures from point clouds [J]. *ACM Transactions on Graphics (S0730-0301)*, 2010, 29(6): 81-95.
- [2] Chen X, Neubert B, Xu Y, et al. Sketch-based tree modeling using Markov random field [J]. *ACM Transactions on Graphics (S0730-0301)*, 2008, 27(5): 32-39.
- [3] Liu J, Jiang Z, Li H, et al. Easy modeling of realistic trees from freehand sketches [J]. *Frontiers of Computer Science (S2095-2228)*, 2012, 6(6): 756-768.
- [4] 谭云兰, 贾金原, 张晨, 等. 3D 树木建模技术研究进展 [J]. *中国图象图形学报*, 2013, 18(11): 1520-1528. (Tan Yunlan, Jia Jinyuan, Zhang Chen, et al. Survey on virtual 3D tree modeling technologies [J]. *Journal of Image and Graphics*, 2013, 18(11): 1520-1528.)
- [5] Lindenmayer A. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs [J]. *Journal of Theoretical Biology (S0022-5193)*, 1968, 18(3): 280-299.
- [6] Deussen O, Lintermann B. Digital design of nature: computer generated plants and organics [M]. Germany: Springer Science & Business Media, 2006.
- [7] Honda H. Description of the form of trees by the parameters of the tree-like body: Effects of the branching angle and the branch length on the shape of the tree-like body [J]. *Journal of Theoretical Biology (S0022-5193)*, 1971, 31(2): 331-338.
- [8] Aono M, Kunii T L. Botanical tree image generation [J]. *IEEE Computer Graphics and Applications (S0272-1716)*, 1984, 4(5): 10-34.
- [9] 马伟, 项波, 查红彬, 等. 基于测量数据的植物建模 [J]. *中国科学(F辑: 信息科学)*, 2009, 39(1): 134-144. (Ma Wei, Xiang Bo, Zha Hongbin, et al. Modeling plants with sensor data [J]. *Science China (F: Information Sciences)*, 2009, 39(1): 134-144.)
- [10] Shlyakhter I, Rozenoer M, Dorsey J, et al. Reconstructing 3D tree models from instrumented photographs [J]. *IEEE Computer Graphics and Applications (S0272-1716)*, 2001, 21(3): 53-61.
- [11] Neubert B, Franken T, Deussen O. Approximate image-based tree-modeling using particle flows [J]. *ACM Transactions on Graphics (S0730-0301)*, 2007, 26(3): article No.88.
- [12] Livny Y, Pirk S, Cheng Z, et al. Texture-lobes for tree modeling [J]. *ACM Transactions on Graphics (S0730-0301)*, 2011, 30(4): 76-79.
- [13] Okabe M, Owada S, Igarash T. Interactive design of botanical trees using freehand sketches and example-based editing [J]. *Computer Graphics Forum (S0167-7055)*, 2005, 24(3): 487-496.
- [14] Tan P, Fang T, Xiao J, et al. Single image tree modeling [J]. *ACM Transactions on Graphics (S0730-0301)*, 2008, 27(5): 32-39.
- [15] Wither J, Boudon F, Cani M P, et al. Structure from silhouettes: a new paradigm for fast sketch-based design of trees [J]. *Computer Graphics Forum (S0167-7055)*, 2009, 28(2): 541-550.
- [16] 孙正兴, 徐晓刚, 孙建勇, 等. 支持方案设计的手绘图形输入工具 [J]. *计算机辅助设计与图形学学报*, 2003, 15(9): 1145-1152. (Sun Zhengxing, Xu Xiaogang, Sun Jianyong, et al. Sketch-Based Graphic Input Tool for Conceptual Design [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2003, 15(9): 1145-1152.)