

基于手绘花瓣的花朵建模方法*

濮群, 曾兰玲, 张建明

(江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013)

摘要: 针对现有花朵建模方法形成的花瓣式样与形态单一的缺陷, 提出一种基于手绘花瓣的交互式花朵建模方法。该方法首先根据手绘花瓣的轮廓图提取花瓣模型的二维控制点, 然后通过弯曲控制函数得到花瓣的三维控制点, 继而利用三维控制点构造三维花瓣模型, 最后由经过真实感渲染的单个花瓣构建花朵。实验结果表明, 与其他建模方法相比, 该建模方法能够方便地生成各种裂瓣状花瓣的花朵且真实感更强。

关键词: 植物建模; 花朵建模; 手绘; 交互式; 控制点

中图分类号: TP391.4 **文献标志码:** A **文章编号:** 1001-3695(2012)05-1959-04

doi:10.3969/j.issn.1001-3695.2012.05.095

Modeling method of flower based on sketching petal

PU Qun, ZENG Lan-ling, ZHANG Jian-ming

(College of Computer Science & Communications Engineering, Jiangsu University, Zhenjiang Jiangsu 212013, China)

Abstract: Against the defect that the existing modeling method of flower could only form the petal with single pattern, this paper presented a modeling method of flower which was based on sketching the petal. At first, it extracted the two-dimensional control points based on the profile curve of sketching petal. Then, it received the three-dimensional control points using bending control function, and constructed the three-dimensional model of petal using these three-dimensional control points. At last, after photorealistic rendering the single petal, the flower could be gotten. The experimental results show that, comparing to the other modeling methods, this method can expediently generate kinds of flowers with crack-like petals, and these flowers have better sense of reality.

Key words: plant-modeling; flower-modeling; sketching; interactive; control points

0 引言

植物建模是计算机图形学研究的重要领域之一, 花卉建模是植物建模的重要分支。花卉建模的研究成果可用于虚拟场景模拟, 对农林业发展也有深远意义。目前对花卉建模的主要思想是基于L-系统^[1]和分形方法^[2]。L-系统建模的关键是根据植物的形态特征与生长规律提炼L-系统文法产生式, 其产生过程相对困难, 且系统生成后操作复杂, 必须要求掌握相关图形知识, 通用性不强。对于分形方法建模, 目前的研究成果只限于单层花瓣, 且花瓣与花瓣之间只是简单地复制与迭代, 无法表达花卉的千姿百态。

交互式建模^[3]的思想来源于人机交互, 具有简单方便、操作性强的优点。交互式思想也曾用于植物建模中。目前已有许多学者对花瓣的交互式建模方法进行研究, 并取得了一定的成果。日本东京大学的Ijiri等人^[4,5]开发的交互式花瓣建模系统可以根据手绘2D草图生成3D花卉模型, 但其操作复杂, 生成的花瓣形式单一且多为内包型花朵, 很少做到对裂瓣状花瓣和向外弯曲型花瓣进行建模。浙大的丁展等人^[6]开发的基于手绘的交互式建模将手绘的思想融入建模方法中, 但花朵的建模只是证明系统通用和可行性的一个实例, 没有就花瓣形态的多样性进行深入研究。

为弥补建模得到的花瓣形态单一以及生成外弯型花瓣的真实感较差的缺陷, 本文提出基于手绘花瓣的交互式花朵建模方法。该方法通过手绘曲线轮廓和弯曲函数控制花瓣形态, 可以得到不同类型的花朵, 并且在建模过程中, 根据不同的真实花瓣采用不同的渲染方法, 使生成的花瓣具有较强的真实感。

1 花瓣模型控制点的确定

图形学中模型的建模与控制形变实质是点的变动。目前对于花瓣建模方法中, 很少从点入手, 这导致根据模型生成的花朵形态不逼真。其次, 目前基于点的建模方法在提取点时存在提取过程相对复杂的缺陷^[7]。本文提出基于手绘的花瓣建模方法, 首先通过手绘提取花瓣外围轮廓点, 得到构成花瓣平面的二维控制点, 最后通过弯曲函数得到构成花瓣形态的三维空间点, 从而将花瓣形态离散成若干三维控制点, 通过对这些点的处理即可形成不同种类的花瓣。

1.1 花瓣模型二维控制点

为了方便定位花瓣, 把花瓣空间形态映射到三维坐标系, 如图1所示。令花瓣最低点为坐标系中的原点(0,0,0), 从原点出发竖直向上为y轴方向, 水平向右为x轴方向, 垂直于xoy面向外为z轴方向。

基于手绘建模方法的关键是控制点的选择。为了模拟不

收稿日期: 2011-09-15; **修回日期:** 2011-10-27 **基金项目:** 江苏省自然科学基金资助项目(BK2009199); 公安部科技创新项目(2010YYCXGDST066); 江苏大学高级专业人才培养基金资助项目(10JDCG073)

作者简介: 濮群(1986-), 女, 江苏人, 硕士研究生, 主要研究方向为计算机图形学、花卉建模(691525127@qq.com); 曾兰玲, 女, 江苏人, 副教授, 主要研究方向为计算机图形学、植物建模等; 张建明(1964-), 男, 江苏人, 教授, 博士, 主要研究方向为图像处理、模式识别等。

同类型的花瓣,首先需要对花瓣形态特征进行分类。

从外形看来,花瓣可以划分如下两种:利用左右两侧曲线可近似勾勒的规则花瓣;上端呈现一定波浪形状的不规则花瓣。

为确定规则花瓣的二维控制点,首先手绘花瓣边缘轮廓图,并将此轮廓分为左边和右边曲线。通过捕捉鼠标事件,分别获得两条曲线上按 y 轴等分的点。将同一 y 坐标的左右两端曲线上点相连并等分得到水平方向的控制点,此时即可确定花瓣模型的二维平面控制点,如图2所示。

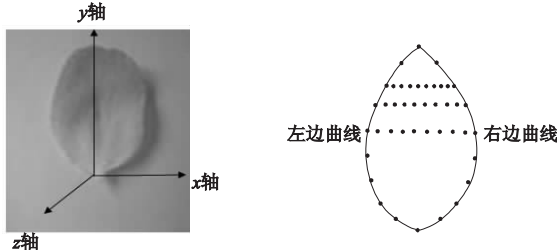


图1 现实花瓣与三维坐标系统的映射 图2 确定规则花瓣的二维控制点

对于不规则花瓣,首先通过手绘方式绘制花瓣轮廓,包括三条曲线、左右边界线和波浪状上边界线,将左右边界与上边界线分开处理。通过处理鼠标事件,得到左右边界线上以 y 轴均分的若干控制点。对于上边界,为了准确查找曲线弯曲处的拐点,绘制曲线的过程中需要尽可能记录下较多的点。遍历上线记录的控制点 p_i ,如果 p_i 的左右导数同时满足式(1)和(2),则此点为所求的控制点,如图3所示。

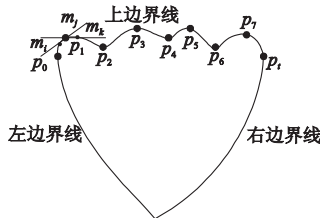


图3 确定不规则花瓣的二维控制点

$$F'(p_i^+) > 0 \quad (1)$$

$$F'(p_i^-) < 0 \quad (2)$$

经上述处理,得到控制点集合 $\{p_0, p_1, \dots, p_t\}$,按下式对控制点 p_0, p_t 作归一化处理:

$$p_0 \cdot y = p_t \cdot y \quad (3)$$

归一化后,控制点划分为式(4)(5)所示的子集:

$$p^+ = \{p_i | p_i \cdot y > p_0 \cdot y\} \quad (4)$$

$$p^- = \{p_j | p_j \cdot y < p_0 \cdot y\} \quad (5)$$

对于任意相邻的点 $p_{i1} p_{i2} = p^+$,连接 p_{i1}, p_{i2} ,然后按式(6)(7),得到点 n_{i1}, n_{i2} 。

$$p_{i1} n_{i1} \cdot p_0 p_t = 0 \quad (6)$$

$$p_{i2} n_{i2} \cdot p_0 p_t = 0 \quad (7)$$

$$n_c = (n_{i1} + n_{i2})/2 \quad (8)$$

然后,按式(8)计算控制点 n_c ,依次得到 p^+ 集合中所有点的控制点,如图4(a)所示。

为使曲线更光滑,在所选的拐点 p_i 间插值选取若干控制点,以迭代的方式找到相应关键控制点 p_i, n_i, t_i 。波浪线上选取的控制点越多,细分连接的折线越平滑接近曲线,如图4(b)所示。对于 p^- 而言,由于满足规则花瓣要求,可按照确定规则花瓣二维控制点的方法来处理。

1.2 花瓣模型三维控制点

得到花瓣的二维控制点后,根据花瓣的弯曲特征给每个点

赋予相应的深度值,使得这些分布在空间的点能构成三维空间中的花瓣。

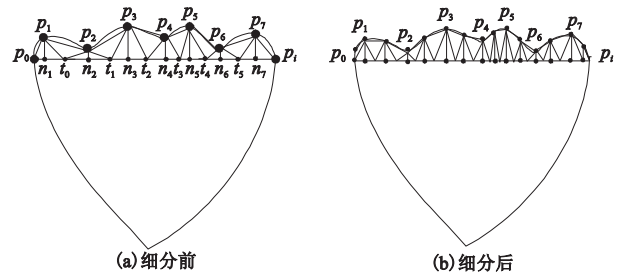


图4 获取上线控制点

首先找到花瓣的横纵轴线,如图5所示。有些形如荷花的花瓣,纵轴线的弯曲度不大,花瓣呈现内包形状,主要由横轴的弯曲控制花瓣的形变,将这类花瓣简称为基于横轴控制形变的花瓣。而有些形如牡丹花的花瓣,横轴线弯曲不大,花瓣呈现向外延伸的特性,主要由纵轴的弯曲控制花瓣的形变,将这类花瓣简称为基于纵轴控制形变的花瓣。

1.2.1 基于横轴确定三维控制点

对于以横轴为基准弯曲的花瓣,以荷花花瓣为例,可以利用系统设置的相关弯曲函数,将二维控制点与弯曲横轴上的控制点一一映射。三角函数在一个周期内的弯曲特性与花瓣的弯曲特性相似,因此可以将三角函数作为弯曲度控制函数并通过改变其参数来展现不同的弯曲效果。

如图6所示,花瓣轮廓左边缘点对应 $\sin$ 函数中的原点 $(0,0)$,花瓣轮廓右边缘点对应 $\sin$ 函数中的 $(\pi,0)$ 。对 $\sin$ 函数进行同样的等分,使手绘花瓣上每个等分控制点与 $\sin$ 函数中控制点对应,如式(9):

$$\text{CtrPoints}[i][j].z = A \times \sin[j \times (\pi/n)] \quad (9)$$

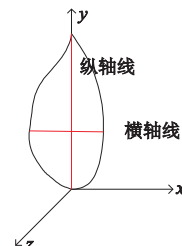


图5 确定花瓣的横纵轴线

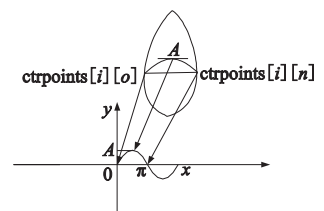


图6 花瓣深度值与三角函数的映射关系

参数 A 调整花瓣的弯曲度, A 值越大,曲面越弯曲,且 A 的值可以交互式调整。

考虑到不是所有花瓣的弯曲度都形似 $\sin$ 函数,更多的花瓣是任意弯曲,变数较大。Bezier 曲线段^[8]可以拟合任何数目的控制点,如三点可以生成一个抛物线,四点生成一个三次曲线,以此类推。如图7所示,花瓣的弯曲度可用四点控制,其中 p_0 和 p_3 对应花瓣二维轮廓上相同 y 坐标的横轴线的最左端和最右端的 x 坐标, p_1 和 p_2 是控制曲线弯曲度的控制点,对 p_1 和 p_2 的交互式设置,能很好地控制花瓣横轴线的弯曲度。

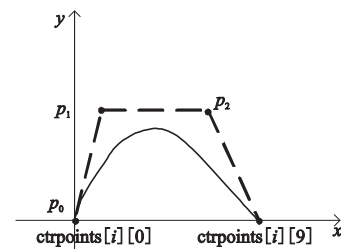


图7 花瓣深度值与Bezier曲线的映射关系

由于曲线计算复杂,会给程序增加负担,因此,对于不同花瓣,可交互式选择不同的弯曲控制函数。

z 坐标形成后,将所有的控制点连接可构成花瓣的立体轮廓。对于规则花瓣,循环遍历横轴线上的控制点,连接成三角形面片,实现过程如图8所示。

对于不规则花瓣,上部分连接成三角形面片,下半部分按照规则花瓣方法连接成四边形面片,如图9所示。

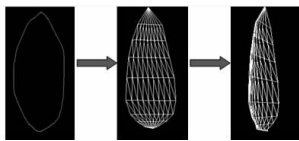


图8 规则花瓣的生成

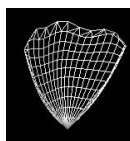


图9 不规则花瓣网格面

1.2.2 基于纵轴确定三维控制点

对于以纵轴为基准的花瓣,以典型的百合花瓣为例:首先在手绘花瓣轮廓草图得到花瓣的轮廓边缘,如图10(a)所示,此时如手绘荷花花瓣曲线一样,仅仅得到轮廓边缘二维控制点。为得到花瓣的三维深度信息,可在交互式界面上手绘出弯曲程度近似的花瓣纵轴线,如图10(b)所示。

此时根据鼠标响应得到一系列点 (x_i, y_i) ,即为位于 xoy 平面上曲线的二维控制点,而中心轴的深度值在花瓣中是出自三维坐标系中的 xoz 面,因此将最低点的 y 值设为 0.0, x 值相应地转换为深度 z 值,即形成 $(x_i, y_i) - (z_i, |y_i - y_0|)$ 的一一映射,并按照荷花花瓣控制点获取方法得到百合花瓣的内部控制点,将 z 值赋给相应的控制点,得到花瓣的三维控制点,连接成四边形面片,如图11所示。



(a)花瓣轮廓曲线 (b)花瓣弯曲曲线

图10 手绘曲线轮廓和纵轴线

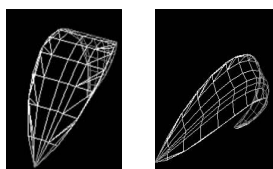


图11 以纵轴为基准的花瓣网格面

2 花朵生成

2.1 花瓣真实感渲染

单个花瓣网格模型生成后,需对建模花瓣进行真实感渲染,如给花瓣渲染颜色,或者添加真实花瓣纹理^[9]等,达到外形逼真的效果。针对花瓣的色彩不同可以交互式选择不同的渲染方法。

色彩渲染方法得到的花瓣颜色灿烂,且可根据用户喜好选择一定的色彩,如一些渐变色的渲染。利用这种方法模拟色彩需要为每个顶点设置颜色数组,以荷花花瓣为例,花瓣由底端白色向上渐变成粉色,每个顶点的颜色如式(10):

$$\text{gradient}[i][j] = \text{bottom. color} + (\text{top. color} - \text{bottom. color}) / m \times i \quad (10)$$

其中:bottom. color 表示最底端的颜色值;top. color 表示最顶端的颜色值;gradient $[i][j]$ 记录第 i 层第 j 个点颜色值。

为了使花瓣模型具有三维效果,着色后还需要给花瓣模型添加一定的光照。由于花瓣曲面是由多个三角形或者四边形逼近的,多个三角形面片共享一个公共顶点,如果仅仅用一个面的法向量作为一个顶点的法向量,则面与面之间不连续,光照后得到的面不光滑。方法通过解析表面或利用多边形数据求均值法得到所有控制点的平均法向量。

若 p_0, p_1, p_2 决定平面 $\pi, p_{12} = p_2 - p_1, p_{01} = p_1 - p_0$;若 $p_{12} \parallel \pi \wedge p_{01} \parallel \pi$,则面的法向量 $n_x = p_{12} \times p_{01}$ 。求得 n_x 后,将每个多边形的法向量归一化求平均值,求得公共点的法线。如图12所示为四个面共点 H 。对 H 点的法向量求解见式(11):

$$n = (n_1 + n_2 + n_3 + n_4) / |n_1 + n_2 + n_3 + n_4| \quad (11)$$

由于三点可以确定一个面,所有的法向量的求解放在三角形网格中,对于四边形的网格,可以添加辅助线链接成三角形网格,如图13所示。对于任意控制点 ctrpoints $[i][j]$,按照逆时针方向与相邻的六个控制点:ctrpoints $[i][j-1]$, ctrpoints $[i+1][j-1]$, ctrpoints $[i+1][j]$, ctrpoints $[i][j+1]$, ctrpoints $[i-1][j+1]$, ctrpoints $[i-1][j]$ 连接构成的三角形面片共享控制点 ctrpoints $[i][j]$ 。求出这六个三角形的面法向量再归一化求均值,即得到控制点 ctrpoints $[i][j]$ 点的法向量。对于某些边缘控制点,若不存在某些相邻的控制点,则将相邻的三角形法向量设置为 0,方便统一求解。

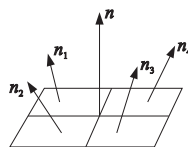


图12 多面共点的法向量求解

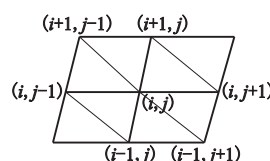
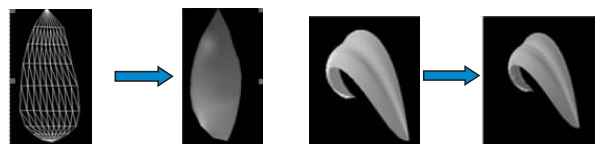


图13 公共点法向量

根据顶点数组、颜色数组、法向量数组产生的花瓣效果如图14所示。

由于控制点已知,为了体现真实感,还可以对所绘制的曲面进行纹理贴图,达到仿真效果。如图15所示是添加纹理前后的花瓣比较。



(a)着色前花瓣 (b)着色后花瓣 (a)添加纹理前花瓣 (b)添加纹理后花瓣

图14 平滑着色

图15 添加纹理

2.2 花朵的生成

单个花瓣形成后,按照一定的排列方式可以形成花朵。花瓣在花芽中的排列方式分为三种:镊合状——花瓣各片的边缘彼此相接触,但不覆盖;旋转状^[10]——花瓣每一片的一边既覆盖着相邻一片的一边,而另一边又被另一相邻片覆盖,菊花花瓣按此方式排列;复瓦状^[10]——与旋转状相似,只是各片中有一片或两片完全在外,另一片完全在内,月季花瓣按此方式排列。

花瓣在花朵之中的排列并不是简单地叠加与分布,通过观察发现,花瓣的层与层之间,每一层的瓣与瓣之间都有统计意义的自相似性,且满足一定的规律。

对于单层花,花瓣的个数一般按照斐波拉切数列变化,即 $\langle 1, 2, 3, 5, 8 \dots \rangle$ 。而对于多层花瓣,层与层之间自相似,是一种分形迭代产物,其中每层之间也是按照斐波拉切数列进行排列^[11]。

由于花瓣之间的自相似性,使得从一个简单的图形出发,通过迭代产生复杂图形成为可能。本例中通过对单片花瓣的若干仿射变换得到整朵花。每一朵花的生成可以通过如下四个属性进行描述 flower{ rotation, scale, deviation, inclination },通过对这四个参数的调整以产生整朵花,如图16所示。

假设所做的花有 m 层,每层有 n 个花瓣,每一层相邻花瓣之间的旋转角度为 rotation (rotation = $2\pi/n$)。Scale 表示不同

花瓣的放缩程度,可对 scale 变量进行 x 、 y 、 z 轴三个方向的放缩。为了灵活设置参考点,变换花瓣类型,可以设置缩放变换矩阵:

$$\begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ (1-s_x) \times x_f & (1-s_y) \times y_f & (1-s_z) \times z_f & 1 \end{bmatrix}$$

其中: (x_f, y_f, z_f) 是比例变换的参考点; s_x, s_y, s_z 是沿 x 、 y 和 z 方向上的缩放量; deviation 表示花瓣层与层之间的偏离角度,即花瓣面绕 xoz 平面的旋转角度; inclination 是附加参数,用来描述花瓣自身的倾斜度。可以通过设置如下旋转矩阵来对花瓣进行旋

转变换: $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & \sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos \varphi & 0 & -\sin \varphi & 0 \\ 0 & 1 & 0 & 0 \\ \sin \varphi & 0 & \cos \varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$

$$\begin{bmatrix} \cos \omega & \sin \omega & 0 & 0 \\ -\sin \omega & \cos \omega & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}。其中: \theta、\varphi、\omega 分别表示在右手坐标$$

系下绕 x 、 y 、 z 轴旋转的角度。花朵生成算法可以用如下伪代码表示:

```
void draw_flower(lever, number)
{
    if lever == 0;
        return ;
    else {
        draw_flower(petal, flower_bud)
        for(i = 1; i < number; i++)
        {
            modify( rotation, scale, deviation, inclination)
        }
        lever--;
    }
}
```

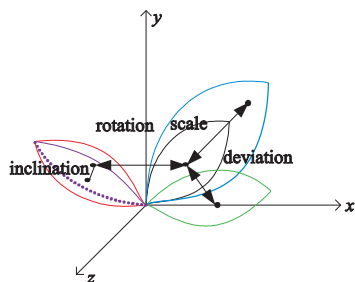


图16 花瓣迭代模型

3 实验结果与分析

为验证方法的有效性与通用性,本文选择荷花、百合、牡丹、桃花四种典型花朵进行实验建模。以荷花的建模为例,如图 17(a)所示为建模平台,建模过程是首先选择相应的真实花瓣模型;荷花。由于荷花是规则花瓣,且弯曲函数由横轴控制,因此选择基于横轴建模的手绘平台,在此平台拖动鼠标绘制花瓣的大体轮廓后(图 17(c)),选择相应着色方式(荷花建模选择渐着色),得到荷花如图 17(d)所示。

按照上述实验步骤,该系统还可以进一步生成花瓣向外弯曲的百合,具有裂瓣特性花瓣的牡丹等,实验结果如图 18 所示。

实验证明本文提出的建模方法能很好地完成交互式建模,

且生成的花朵种类繁多,弥补了目前花瓣建模系统所生成的花瓣形态单一的缺憾。

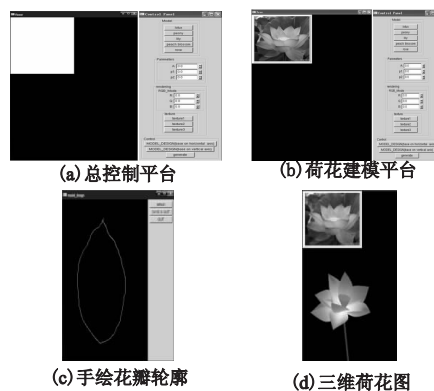


图17 建模过程



图18 最终结果图(从左至右依次为荷花、百合、牡丹、桃花)

4 结束语

本文提出了一种基于手绘花瓣的交互式花朵建模方法。通过手绘和弯曲函数确定花瓣构成的三维离散点,将对花瓣的处理转移到对点的处理。实验表明,与其他方法相比,本文的方法适用范围广、实用性强、操作简单易懂,且得到的结果理想真实。交互式思想的引入更好地达到了人机交互的理想状态。

参考文献:

- [1] 李文杰. 基于 L-系统植物树建模方法的研究[D]. 合肥:合肥工业大学, 2009.
- [2] 丛波,王琰. 基于分形构造三维花卉的算法[J]. 沈阳理工大学学报, 2008, 27(6): 14-20.
- [3] 曾兰玲,汪国昭. 交互式植物建模系统[J]. 计算机辅助设计与辅助图形学学报, 2009, 21(12): 1814-1820.
- [4] IJIRI T, OWADA S, OKABE M, et al. Floral diagrams and inflorescences: interactive flower modeling using botanical structural constraints[J]. ACM Trans on Graphics, 2005, 24(3): 720-726.
- [5] IJIRI T, OWADA S, IGARASHI T, et al. Seamless integration of initial sketching and subsequent detail editing in flower modeling[J]. Computer Graphics Forum, 2006, 25(3): 617-624.
- [6] 丁展,叶修梓. 基于手绘的交互方式与三维建模研究[D]. 浙江:浙江大学, 2008.
- [7] ANASTACIO F, PRUSINKIEWICZ P, SOUSA M. Sketch-based parameterization of L-systems using illustration-inspired construction lines[J]. Computers and Graphics (Pergamon), 2009, 33(4): 440-451.
- [8] SHREINER D, WOO M, NEIDER J. OpenGL programming guide[M]. 北京:机械工业出版社, 2009.
- [9] 黄秀常. 基于 OPENGL 的树木场景渲染分析与实现[J]. 计算机仿真, 2010, 27(5): 231-234.
- [10] 王良,曾兰玲. 基于分形三维菊花建模算法[J]. 辽宁工程技术大学学报, 2007, 26(3): 410-411.
- [11] SMITH R S, KUHLEMEIER C, PRUSINKIEWICZ P. Inhibition fields for phyllotactic pattern formation: a simulation study[J]. Canadian Journal of Botany, 2006, 84(11): 1635-1649.