

# 基于粒子系统的心形烟花模拟

吴国宝<sup>1,2</sup>, 汪继文<sup>1,2</sup>

(1. 安徽大学 计算智能与信号处理教育部重点实验室, 安徽 合肥 230039;

2. 安徽大学 计算机科学与技术学院, 安徽 合肥 230039)

**摘要:** 计算机动画模拟的目标是实现仿真的实时性与逼真性, 粒子系统作为模拟不规则物体运动的有效算法, 在自然现象的仿真方面有着突出的贡献。为了弥补使用填充方式模拟心形烟花的不足, 增强模拟的效果, 文中提出了一种基于粒子系统的模拟方法。基本原理是采用阿基米德螺旋函数来控制爆炸瞬间粒子的属性, 同时结合纹理贴图技术绘制烟花粒子。编程上基于 OpenGL, 采用 Visual C++ 实现了“心”形烟花的模拟。实验结果证明提出的绘制方法具有快速、简单的特点, 达到了预期的效果。

**关键词:** 粒子系统; “心”形烟花; 纹理映射

**中图分类号:** TP391.9

**文献标识码:** A

**文章编号:** 1673-629X(2012)11-0089-04

## Heart-shaped Fireworks Simulation Based on Particle System

WU Guo-bao<sup>1,2</sup>, WANG Ji-wen<sup>1,2</sup>

(1. Key Lab. of Intelligent Computing & Signal Processing, Anhui University, Hefei 230039, China;

2. Science and Technology School, Anhui University, Hefei 230039, China)

**Abstract:** The goal of computer-animated simulation is to achieve real-time simulation and realistic, effective algorithm of the particle system as a simulation of irregular movement of objects, outstanding contributions to the simulation of natural phenomena. To compensate for the lack of fill simulate the heart-shaped fireworks, and enhance the effect of simulated, presents a simulation method based on particle systems. The basic principle is using Archimedes spiral function to control the explosive properties of the particle moments, combined with texture mapping draw fireworks particles. Programming is based on OpenGL, using Visual C++ implementation of the "heart"-shaped fireworks simulation. Experimental results show that the proposed drawing method is fast, simple to achieve the desired results.

**Key words:** particle system; heart-shaped fireworks; texture mapping

## 0 引言

近年来,随着计算机技术和动画技术的发展,自然景物的模拟越来越受到科研工作者的关注。实时性和真实性是我们追求的目标,然而自然景物中物体构造复杂,例如火焰、水、雨、雪、烟花等,在形态上具有不规则性,在运动上具有随机性,很难用传统的几何模型和行为模型建模。1983年, W. T. Reeves<sup>[1,2]</sup>提出了粒子系统,并成功地模拟了自然景物。它的优点是可以利用非常简单的体素来构造不规则的物体,而不是用传统线、曲面或者多边形来再现自然景物,因此粒子系统被公认为是模拟不规则物体最成功的图像生成算

法<sup>[3]</sup>。目前,国内外已经有许多研究者采用粒子系统对自然景物进行成功地模拟。万华根<sup>[4]</sup>等通过对 N-S 方程的求解并结合粒子系统来模拟喷泉,同时利用圆球实现水滴,采用光线跟踪算法绘制。T. Loke<sup>[5]</sup>等人采用粒子衍生法来绘制烟花粒子的轨迹。目前,粒子系统已经广泛用于烟花模拟,得到了逼真的效果<sup>[6]</sup>。然而,心形烟花模拟采用的填充方式,不能达到较好的实时性和动态性。文中通过在研究上述方法的基础上,使用粒子系统结合纹理贴图技术增强了烟花模拟的实时动态效果。

## 1 基本原理及方法

### 1.1 粒子系统基本原理

一个粒子系统由大量称为粒子的简单个体构成。每个粒子都有一组属性,包括位置、速度、大小、生命值、颜色等,这些属性随着时间的推移在不断的变化<sup>[7]</sup>。系统中的每个粒子在整个应用中都是不断变化

收稿日期:2012-03-05;修回日期:2012-06-10

基金项目:安徽省自然科学基金(090412045)

作者简介:吴国宝(1983-),男,安徽怀远人,硕士研究生,研究方向为计算机模拟技术;汪继文,博士,教授,博士研究生导师,研究方向为计算流体力学。

的,这个变化的过程,称之为粒子的生命期。通常在生命期内,每一时刻,粒子系统要完成以下几个步骤<sup>[8]</sup>:

1) 粒子系统初始化。初始化包括粒子数目,粒子大小、颜色、生命期等属性的预处理。

2) 更新现有粒子属性。随着时间的推移,粒子的颜色、速度、位置、透明度已经发生变化,需要更新粒子的属性。

3) 移除“消亡”的粒子。粒子本身有自己的生命期,该步骤用于检测每个粒子的生命期,若为 0 则需要将粒子从系统中删除。

4) 绘制粒子。前面的步骤是粒子的动态处理过程,要展示粒子的运动轨迹及各属性的变化,以形成动画的效果,就需要将每一帧的粒子绘制在屏幕上。

5) 重复 2)、3)、4) 步骤,构成了一个如图 1 的完整粒子系统。

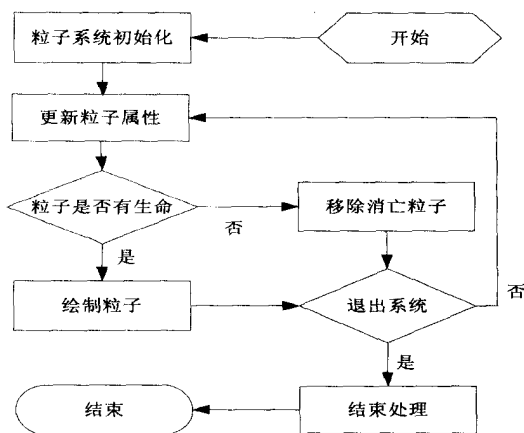


图 1 粒子系统流程图

## 1.2 心形烟花的模拟方法

烟花粒子爆炸时,产生的大量小粒子具有和爆炸前粒子相同的位置。由于粒子的速度及方向控制了粒子运动轨迹,因此采用粒子系统模拟爆炸后每个粒子的速度是模拟心形烟花的重要部分。由阿基米德螺线函数图像(如图 2)。

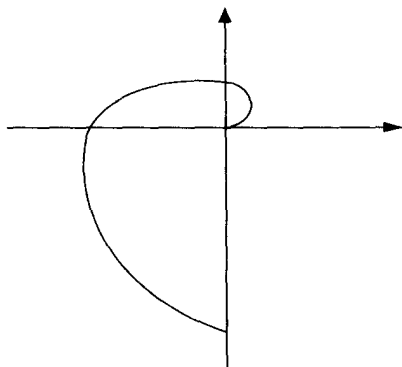


图 2 阿基米德螺线

可知将两个适当角度的螺线可以拼接成心形

状。再由阿基米德螺线函数在极坐标下控制烟花爆炸后左边粒子的速度为:

$$\begin{cases} v_x(t) = \cos\theta * \theta * r \\ v_y(t) = \sin\theta * \theta * r \\ v_z(t) = v_z(t - \Delta t) \end{cases} \quad (1)$$

根据对称性右边粒子的速度为:

$$\begin{cases} v_x(t) = -\cos\theta * \theta * r \\ v_y(t) = \sin\theta * \theta * r \\ v_z(t) = v_z(t - \Delta t) \end{cases} \quad (2)$$

其中  $\theta$  为三维坐标系下  $x, y$  平面控制在一定范围内的弧度值,使得爆炸后的粒子构成一个闭合的区域,  $r$  为随机函数值,用于微调形状,增强真实感。 $Z$  方向的速度与爆炸前粒子速度保持一致。在得到爆炸时粒子的位置和速度后,通过欧拉公式可知粒子的下一帧位置和速度为:

$$\begin{cases} s_x(t + dt) = s_x(t) + v_x(t) \Delta t \\ s_y(t + dt) = s_y(t) + v_y(t) \Delta t \\ s_z(t + dt) = s_z(t) + v_z(t) \Delta t \end{cases} \quad (3)$$

$$\begin{cases} v_x(t + dt) = v_x(t) \\ v_y(t + dt) = v_y(t) + g_y \Delta t \\ v_z(t + dt) = v_z(t) \end{cases} \quad (4)$$

其中  $g_y$  为  $y$  方向上的重力加速度,  $\Delta t$  为两帧间的时间间隔。为简化模型的复杂度,这里假设粒子只受到重力的作用,忽略空气阻力等外力作用。

## 2 心形烟花粒子系统的建模

由于烟花的形状多种多样、千姿百态,因此要实现

对烟花的模拟,需要建立起烟花粒子系统的模型。

### 2.1 应用场景模型

在使用粒子系统前,需要对模拟的应用做相应的分析。包括该应用中粒子需要具有的属性、运动过程的分解、粒子的多少,并建立起三维空间坐标系<sup>[9]</sup>。

粒子的多少,影响着系统的性能,粒子数太多,需要渲染和更新的次数也就越多,占用系统的资源也就越多。相反粒子的数目较少,占用的系统资源也就越少,速度也就越快,但是视觉上的效果显得不够理想,因此合理地选择粒子的数量是模拟的一个关键<sup>[10]</sup>。烟花从上升到爆炸、产生预定的形状、最后在空中消失,是烟花模拟的整个过程。因此烟花粒子的属性包括当前位置、颜色、透明度、速度、加速度、大小、生命值等,可表示为  $\text{ParticleAttr} = \{ \text{Position}, \text{Color}, \text{Fade}, \text{Speed}, \text{Acceleration}, \text{Size}, \text{Life} \}$ , 其中位置、速度、加速度是在空间坐标系下的三维向量。

### 2.2 粒子数据结构

由 2.1 对粒子的属性描述可知,粒子的数据结构

可表示为:

```
typedef struct Particle{
    GLfloat xpos; //(xpos,ypos,zpos)为粒子的位置
    GLfloat ypos;
    GLfloat zpos;
    GLfloat xspeed; //x方向的速度
    GLfloat yspeed; //y方向的速度
    GLfloat zspeed; //z方向的速度
    GLfloat r; //(r,g,b)为粒子的颜色
    GLfloat g;
    GLfloat b;
    GLfloat life; //粒子的生命值
    GLfloat fade; //粒子的衰减速度
    GLfloat size; //粒子大小
}
```

### 2.3 粒子系统的初始化

烟花粒子系统的初始化,是赋予烟花粒子属性初始状态。

#### 2.3.1 场景演示空间初始化

场景演示空间初始化,在 OpenGL 中使用视景体来定义空间的可视范围。视景体可以看作是一个有限空间体,在空间内的物体将会展示在画面上,超出空间范围的物体将被裁减到屏幕之外。与视景体相对应的是投影变换,包括透视投影和正投影。在正投影下,视景体被定义一个平行的长方体。使用 glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far) 创建视景体, A(left, bottom, -near), B(right, top, -far) 分别映射到窗口的左下角和右上角。根据实验演示需求,我们分别初始化 A(-10,0,0), B(10,20,10),即将窗口可视范围定义为水平方向范围  $x(-10,10)$ ,垂直方向  $y(0,10)$ ,深度  $z(0,10)$ 。

#### 2.3.2 位置及速度初始化

烟花粒子从窗口底端  $y=0$  处燃放,  $x, z$  方向使用随机函数分别控制位置在  $(-5,5)$ 、 $(0,3)$  范围。烟花燃放后,为控制烟花向靠近  $yz$  平面,可以通过烟花燃放的位置来控制烟花速度方向,如果  $xPose=0$  则初始化  $xSpeed$  为 0,如果  $xPose>0$  则初始化  $xSpeed$  为小于 0 的关于  $xPose$  的随机数,相反,如果  $xPose<0$  则初始化  $xSpeed$  为大于 0 的关于  $xPose$  的随机数。加速度初始化,运动的物体不仅受到重力的作用,同时也会受到空气阻力等外力的作用,所有外力的和力决定了物体受到的加速度大小。

### 2.4 粒子的运动

粒子系统中最重要的部分在于确定每一时刻当前粒子的运动规律,包括该时刻粒子的位置、速度、生命值等。而这些属性需要通过上一帧属性状态和物理知

识来分析。根据牛顿运动定律可知,粒子下一帧的位置为:

$$\begin{cases} s_x(t+dt) = s_x(t) + v_x(t) \int dt \\ s_y(t+dt) = s_y(t) + v_y(t) \int dt \\ s_z(t+dt) = s_z(t) + v_z(t) \int dt \end{cases} \quad (5)$$

$$\begin{cases} v_x(t+dt) = v_x(t) + a_x(t) \int dt \\ v_y(t+dt) = v_y(t) + a_y(t) \int dt \\ v_z(t+dt) = v_z(t) + a_z(t) \int dt \end{cases} \quad (6)$$

其中  $(s_x(t+dt), s_y(t+dt), s_z(t+dt))$  为下一帧的位置,  $(v_x(t+dt), v_y(t+dt), v_z(t+dt))$  为下一帧的速度,  $dt$  为连续两帧间的时间间距。

### 2.5 粒子的消亡与重生

烟花粒子在整个烟花系统中有着自己的生命值,粒子从生成开始到消失所经历的时间为烟花粒子的生命周期。初始化粒子的生命值为 1,为区分不同烟花粒子生命周期,提高真实感的效果,需要为每个烟花粒子定义在一定范围内的随机衰减因子,衰减因子越大,粒子生命周期越短,相反衰减因子越小,粒子生命周期越长。随着时间的推移,单个烟花粒子的生命值也在逐渐减少,直到为 0。此时,该粒子完成了整个模拟的流程,需要移除粒子系统,同时告诉系统当前的粒子总数,用以判断是否要生成新的烟花粒子。如果烟花粒子的总数小于预期定义的烟花粒子数目,系统将在燃放点处初始化一颗新的烟花粒子,完成了烟花粒子的“重生”。

### 2.6 烟花粒子的渲染

烟花粒子系统模型中,烟花粒子上升到最高点,粒子产生爆炸,将产生大量的粒子。每个粒子含有位置、速度、加速度、颜色等属性,初始化大量的烟花粒子,并跟踪计算,存储每个粒子的属性变化,需要占用大量的内存,如果机器的性能不是很好,难以达到烟花模拟实时性的效果。减少烟花粒子数目会影响模拟的真实感,因此解决该类问题,也不能单纯从减少烟花粒子的方面入手。为了降低渲染粒子的代价,需要在系统中使用纹理贴图技术。纹理贴图允许把一副烟花粒子图像映射到一个多边形的表面上,贴图能够保证当这个多边形变换或者渲染时,映射到多边形表面的图像也能够表现出正确的行为。利用纹理贴图使多边形看上去就像是自然世界的物质<sup>[11]</sup>。这样一颗烟花粒子就可以表示为通过纹理贴图的四边形(如图 3),从而降低了需要渲染的烟花粒子总数,提高了粒子渲染的速度。

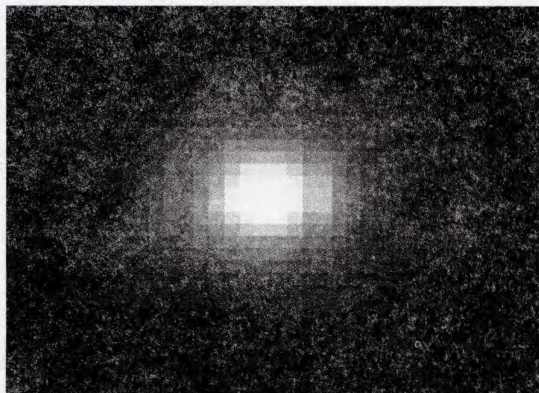


图 3 烟花粒子纹理

### 3 实验结果分析

文献[12]中提到心形形状的控制使用预先准备好的黑白心形纹理图片,通过查询 RGB 分量确定需要填充的像素部分,然后在该处填充烟花粒子,从而产生“心”的形状(如图 4)。

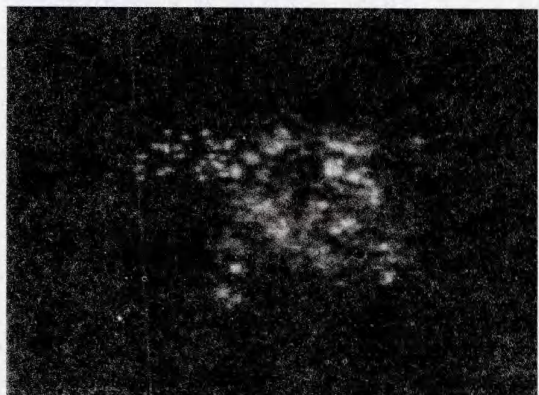


图 4 填充式心形烟花效果

然而填充方式有其自身的缺点,比如:

- 1) 确定烟花的填充位置,需要查询整个屏幕的 RGB 分量,对系统的性能开销很大,影响模拟的实时性。
- 2) 填充未能均匀分布烟花粒子,导致心形边缘稀疏、中心稠密、形状不明显。
- 3) 形状是由静态的烟花粒子填充的,整体上缺乏动态效果。

粒子系统模型,无需填充整个图形,减少了需要渲染的粒子个数,消除了 RGB 查询对系统的开销,提高了模拟的实时性。在极坐标系下控制心形边缘粒子运动属性,包括每个粒子的速度、方向、加速度等,增强了模拟的真实感。对比填充方式分别给出粒子系统实验中烟花在不同阶段的效果图(如图 5、图 6 所示)。

### 4 结束语

文中给出了在粒子系统下实现“心”形烟花的燃

放模拟方法,改进了心形烟花仿真效果,结合 OpenGL 纹理贴图技术增强了烟花模拟的真实感,提高了模拟的实时性。考虑到系统的完美性,系统的中还可以加入声音,或者烟雾,这将在以后的研究中作为考虑的方向。

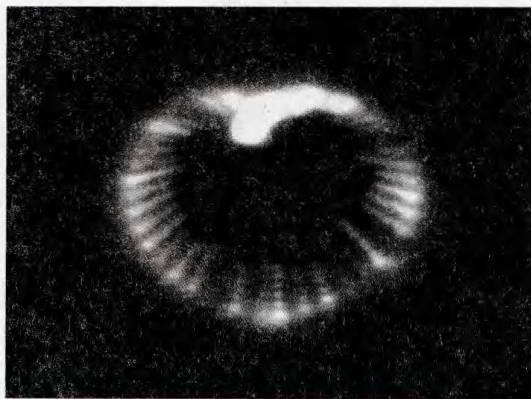


图 5 爆炸初期

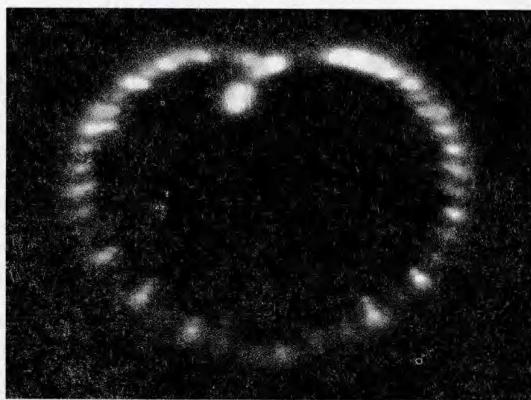


图 6 爆炸后期

### 参考文献:

- [1] Reeves W T. Particle system: a technique for modeling a class of fuzzy objects[J]. Computer Graphics, 1983, 17(3): 360-375.
- [2] Reeves W T. Approximate and probabilistic algorithms for shading and rendering structured particle system[J]. Computer Graphics, 1985, 19(3): 313-322.
- [3] 徐阳东. 基于粒子系统的不规则景物建模研究[D]. 济南: 山东师范大学, 2009.
- [4] 万华根. 基于物理模型的实时喷泉运动模拟[J]. 计算机学报, 1998, 21(9): 772-778.
- [5] Loke T, Tan D, Seah H. Rendering Fireworks Displays[J]. IEEE Computer Graphics and Applications, 1992, 12(3): 30-41.
- [6] 尹星云, 胡长俊. 基于 OpenGL 的烟花粒子系统设计[J]. 电脑开发与应用, 2006(7): 28-30.
- [7] 谢剑斌, 郝建新, 蔡宣平, 等. 基于粒子系统的雨点和雪花降落模拟生成[J]. 中国图象图形学报, 1999, 4(9): 734-

(下转第 96 页)



高 Turbo 码性能。在低信噪比环境下,三种交织器对 Turbo 码的性能影响几乎相同,而质数交织器在高信噪比环境下性能比行列交织器和随机交织器有一些优势,不仅误码率相对较低,而且性能比较稳定。

在不交织、使用行列交织器、随机交织器以及质数交织器时,对 Turbo 码进行编译码仿真,序列长度均为 500,迭代次数为 5 次,信噪比为 1dB,使用 Max-Log-MAP 译码算法仿真结果如图 4。从图中可以看出,质数交织器虽然带来了比一般的随机交织器和行列交织器更多的复杂度,但它对不同的交织长度特别是交织长度比较大时有比其它交织器更好的性能。

由于交织方案必须随数据一起传输,致使传输效率减小,而且译码器也依赖交织方案的准确传输,所以随机交织器在理论上性能优异,但不实用<sup>[11]</sup>。

质数交织器的交织方案固定不变,不需要随数据一起传输<sup>[12]</sup>。虽然算法比其他交织器要复杂,相应的,交织时延会比较大,但考虑到在实际应用中,输入交织器的序列长度往往比较固定,所以交织器的输出顺序可以不需要反复计算,对时延几乎不会造成影响,而质数交织器相比其他的交织器上的性能优势可以在实际应用中显现出来。

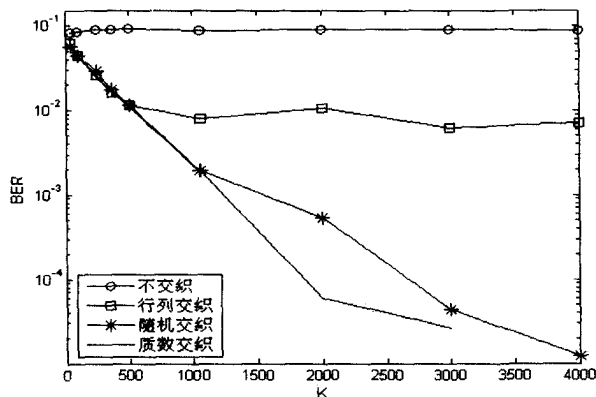


图 4 交织长度对不同交织器的影响

## 4 结束语

通过对 WCDMA 中 Turbo 码质数交织器算法理论

的研究,并通过 MATLAB 仿真实验结果分析可见随着交织长度的增加,译码器的纠错性能得到提高,WCDMA 中的质数交织器支持较大范围的各种帧长(40 ~ 5114 间任意整数),且均能保持良好的误码率。这一优异性能也使其特别适合作为多媒体传输时 Turbo 码的内交织器。

## 参考文献:

- [1] Berrou C, Glavieux A, Thitimajshima P. Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes [C]// Proc. of ICC'93. [s. l.]: [s. n.], 1993.
- [2] Berrou C, Glavieux A. Near optimum error correcting coding and decoding: Turbo-codes [J]. IEEE Trans. on Commun., 1996, 44(10): 1261-1271.
- [3] 刘东华. Turbo 码原理与应用技术 [M]. 北京: 电子工业出版社, 2004: 236-238.
- [4] 3GPP TS 25.212, 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Multiplexing and Channel Coding (FDD) (Release 6) [S]. 2006.
- [5] 吴江, 赵春明. WCDMA 中 Turbo 码算法及质数交织器的研究 [J]. 信息技术, 2002(4): 38-42.
- [6] 杨小辉, 熊建设, 徐洪梅, 等. Turbo 码及交织技术在 WCDMA 中的应用 [J]. 微计算机信息, 2006(33): 135-137.
- [7] 黄旭方, 覃团发, 陈跃波. WCDMA 系统中 Turbo 码编译码器交织长度的研究 [J]. 广西大学学报(自然科学版), 2002, 27(3): 210-212.
- [8] 吴运波, 罗汉文, 赵宇. WCDMA 中 Turbo 码译码算法和仿真 [J]. 移动通信, 2004(S2): 47-49.
- [9] 王视环, 王振宇, 周月臣. 一种应用于 ARQ 系统的新型 Turbo 码译码方案 [J]. 南京邮电学院学报(自然科学版), 2003, 23(2): 50-53.
- [10] 邱金刚, 陈自力, 何金义. AWGN 信道中 Turbo 码的性能和仿真 [J]. 军械工程学院学报, 2005, 17(2): 10-12.
- [11] 周炯槃, 庞沁华, 续大我, 等. 通信原理(合订本) [M]. 北京: 北京邮电大学出版社, 2006: 440-441.
- [12] 吴伟陵. 信息处理与编码(修订本) [M]. 北京: 人民邮电出版社, 2003: 306-309.

(上接第 92 页)

738.

- [8] 向世名. OpenGL 编程与实例 [M]. 北京: 电子工业出版社, 2000.
- [9] 张芹, 吴慧中, 张健. 基于粒子系统的建模方法研究 [J]. 计算机科学, 2003, 30(8): 144-146.
- [10] 丁纪云, 陈利平, 李思昆. 基于 OpenGL 的烟花动态模拟方

法的研究与实现 [J]. 计算机工程, 2002, 2(3): 240-243.

- [11] Shreiner D. OpenGL 编程指南 [M]. 北京: 机械工业出版社, 2010.
- [12] 张妍妍. 烟花粒子系统模拟的设计与实现 [D]. 合肥: 安徽大学, 2010.