

基于粒子系统的烟花动画设计

葛芳,张成,韦穗,程鸿

(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)

摘要: W. T. Reeves 提出的粒子系统模型是模拟不规则模糊物体最成功的一种图形生成方法。为了模拟出效果逼真的烟花,文中采用了粒子系统模型对烟花进行动画设计。在深入研究粒子系统的基本理论和对烟花燃放细节的分析的基础上,建立了基于粒子系统的烟花模拟模型,设计了一种五彩缤纷的烟花动画算法。在编程上,结合了 VC++ 编程语言和 OpenGL 的纹理映射技术,降低了渲染粒子的复杂度。实验结果表明,该算法快速、简单,视觉效果逼真,可以方便地应用到游戏和影视动画中。

关键词: 粒子系统;烟花粒子;烟花动画;OpenGL 纹理映射

中图分类号: TP391.41

文献标识码: A

文章编号: 1673-629X(2010)08-0180-04

Firework Animation Design Based on Particle System

GE Fang, ZHANG Cheng, WEI Sui, CHENG Hong

(Ministry of Education Key Lab. of Intelligent Computing and Signal Processing,
Anhui University, Hefei 230039, China)

Abstract: In the methods of irregular object modeling, the particle system model which was proposed by William T. Reeves is recognized as the most successful in the graphics generation algorithms. In order to simulate realistic fireworks effects, use the particle system model to design animation of fireworks. Establish a fireworks simulation model based a particle system in-depth study of the basic theory of particle system and the analysis of the fireworks booming details, and design an animation algorithm of multicolored fireworks. In the programming, combines VC++ programming language and OpenGL texture-mapping technology to reduce the complexity of rendering particles. Examples illustrate that algorithm is fast, simple, and has good reality, so can be applied conveniently to games, film and television animation.

Key words: particle system; fireworks particle; fireworks animation; OpenGL texture mapping

0 引言

近年来,随着计算机技术的发展,自然景物的模拟越来越多地出现在游戏和影视动画中。但是,由于自然景物的复杂性,对它们的模拟一直是计算机图形学的研究热点,尤其是对动态自然景物的模拟,如云、雾、烟花、水流等,更具有挑战性。1983年, W. T. Reeves 首次提出了粒子系统模型^[1],并用其模拟出了爆炸、烟花等效果。粒子系统采用的是一套完全不同于以往任何绘制的方法来模拟自然景物,它是一种以成千上万个不规则的、随机分布的、形状简单的微小粒子作为基本元素来表示自然景物的方法。其中,每个粒子都有

一定的生命周期,并且每时每刻都在不断运动着;这充分体现了自然景物的动态性和随机性。因此,粒子系统被公认为是模拟自然景物最成功的一种图形生成算法^[2,3]。1985年, William T. Reeves 和 Blau R. 采用“volume filling”基本单元模拟了随风飘动的花草树木^[4],很好地发展了粒子系统。此后,粒子系统从理论到技术都得到了进一步的发展,并广泛用于动画研究中。

目前,在模拟烟花方面,国内外已有一系列的研究成果。1992年 T. Loke 等人^[5]提出一种用粒子衍生法来表现烟花粒子轨迹的绘制算法,同时采用链表数据结构的存储方式,实现了多种烟花的特殊效果。陈利平^[6,7]、甘露^[8]等人通过深入研究烟花燃放过程的特点,给出了不同状态下的烟花的具体算法。可是用这种方法模拟的烟花,烟花粒子数目的多少很大程度上影响了模拟效果的实时性和逼真性。罗玉玲^[9]给出了一种基于粒子系统与纹理映射结合的烟花动态模型,

收稿日期: 2009-12-01; 修回日期: 2010-03-18

基金项目: 安徽省科技攻关计划项目(08010202118)

作者简介: 葛芳(1984-),女,硕士研究生,研究方向为计算机图形学及三维模拟;韦穗,教授,博士生导师,研究方向为计算机视觉、图像处理与分析、全息显示技术等。

模拟出的烟花动态视觉效果较为真实。李清畅^[10]等人将粒子系统与 BillBoarding 技术相结合,取得了逼真的烟花模拟燃放效果。

通过深入研究上述方法的优缺点,文中将粒子系统模型和 OpenGL 中的纹理映射技术相结合,提出了一种快速简单,可以模拟出不同形状、多种颜色的烟花生成算法,满足了烟花模拟的实时性和逼真性的要求。

1 粒子系统的基本理论

粒子系统并不是一个简单的系统。在粒子系统中,把运动的模糊物体看作由有限的具有确定属性的流动粒子所组成的集合^[11,12],这些粒子在它们所处的空间中不断运动。粒子彼此间相互独立,并不与场景中的任何其它物体相互影响。系统中的每个粒子并不是抽象的,它们都具有一系列的属性,如数量、形状、位置、速度、加速度、颜色、生命周期等等,其中颜色、速度等属性随着时间不断地发生变化。粒子系统中的每个粒子在其生命周期内,都会经历新生、活动和消亡三个基本阶段。

图1给出了利用粒子系统仿真不规则模糊物体的一般过程。

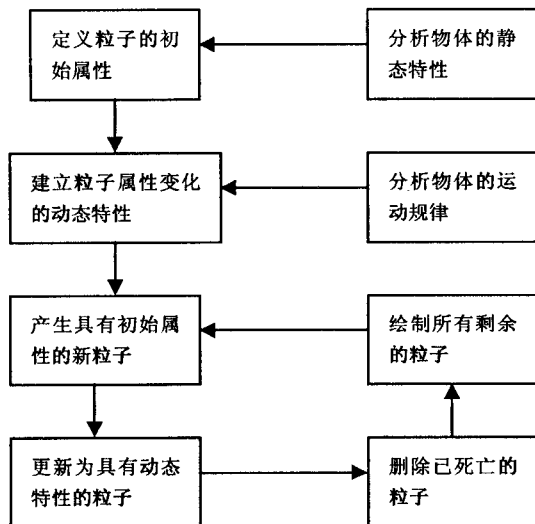


图1 粒子系统仿真不规则模糊物体的一般过程

2 OpenGL 中的纹理映射技术

纹理映射(texture mapping)是图形学中的一项基本的、重要的技术,同时也是一个相当复杂的过程,它是通过数字化的纹理图片覆盖或映射到物体的表面,从而为物体表面增加表面细节的过程。纹理图片可以通过拍摄等方式得到,也可以通过数学函数生成,或者将计算机中显示的内容输出为纹理图片。物体的很多表面细节通过多边形逼近或其它几何建模的方法是难

以表现的,因此纹理映射技术能够使得计算机生成的物体看起来逼真自然。

考虑到人眼对动态自然景物的辨认是有限的,人们更多关心的是能够在虚拟环境下实时模拟出自然景物,且场景看起来连续、生动。这就要求一个好的粒子系统模型必须具有实时性,然而,通常情况下,粒子数目越多,物体的真实程度才越高,在相同的硬件条件下,就降低了实时性。文中利用 OpenGL 中的纹理映射技术将一幅二维的 bmp 格式的烟花图片(如图2所示)叠加在物体表面上,从而降低了渲染粒子的复杂度,减少所需粒子的数量,在满足实时性要求的同时,使仿真出的烟花更加逼真。

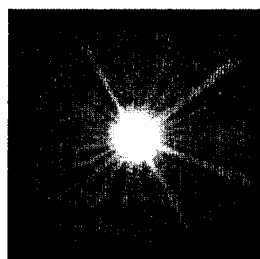


图2 烟花的纹理映射图片

3 烟花粒子系统的模型

3.1 烟花的分析

粒子系统为模拟烟花提供了有力的工具。为模拟出更加逼真的烟花,首先对烟花燃放的细节和特点进行分析:

* 每束烟花在空中的位置不同,且烟花在空中是慢慢消失的。

* 烟花的形状千姿百态,且烟花绽放的速度和持续的时间也会有所不同。

* 烟花的颜色五彩缤纷。不同束的烟花,颜色不同;同一束的烟花,颜色也会变化。

考虑到上述烟花的特点,用粒子系统方法就可以对烟花进行模拟了。文中设在整个烟花燃放的过程中,有两处产生烟花的点:一个是燃放烟花的位置,即燃放点;另一个是冲向空中后以不同形状爆炸开的点,即炸点,烟花的形状主要由爆炸点控制。

3.2 烟花粒子的数据结构

下面给出了烟花粒子的数据结构:

```

typedef struct
{
    float xPosition,yPosition,zPosition; //粒子在 x,y,z 方向的位置
    float xSpeed,ySpeed,zSpeed; //粒子在 x,y,z 方向的速度
    float xAcceleration,yAcceleration,zAcceleration; //粒子在 x,y,z 方向的加速度
    float red,green,blue; //粒子颜色的值
}
  
```

```

| Particle;
typedef struct
|
| float life; //烟花的生命周期
| float fade; //烟花生命的衰减速率;
| float fire_Speed; //烟花在空中爆炸后的速度
| struct Particle fire[MAX_PARTICLE][MAX_TAIL]; //流星系
| 统数组, MAX_PARTICLE 为烟花爆炸后的粒子数, MAX_TAIL
| 为烟花尾部长度
| Firework[MAX_FIRE];

```

3.3 烟花粒子的初始属性

对每个新产生的烟花粒子,系统都要赋予其一系列的属性。为了使模拟的烟花具有真实感的效果,文中采用多种属性,如数量、位置、速度、加速度、颜色、生命周期等等,对烟花粒子进行描述。

初始数量:在烟花粒子系统中,只在某一帧 f_0 产生烟花粒子。在燃点处,通过设置 MAX_FIRE 的值来确定燃点处随机出现的最大粒子数。在炸点处,通过设置 MAX_PARTICLE 的值来确定烟花粒子的数目,这个值的选定很关键,它决定烟花的密度。数目过少,模拟效果失真;数目过多,实时性受到影响。

初始形状:假设一个烟花粒子系统由一个头粒子和一组尾粒子组成,头粒子最大最亮,其它粒子逐渐变细变暗。程序中通过 MAX_TAIL 值的大小来显示粒子拖尾的长短,算法中取 MAX_TAIL = 30。

初始位置:初始位置反映了烟花粒子的燃点所在的位置,由(1)式来确定:

$$\begin{cases} xPosition = \text{rand}() \% k_{xp} \\ yPosition = \text{rand}() \% k_{yp} \\ zPosition = \text{rand}() \% k_{zp} \end{cases} \quad (1)$$

其中, $xPosition$, $yPosition$, $zPosition$ 表示烟花粒子的燃点位置; k_{ip} ($i = x, y, z$) 是用于控制烟花燃点的初始位置的参数,可根据所定义的坐标系的情况选取合适的值; $\text{rand}()$ 为随机函数。

初始速度和加速度:粒子的速度和加速度大小关系到粒子的运动轨迹,是系统必不可少的。假设烟花在燃点处发射的粒子,只受重力作用。当烟花粒子升向空中时,初始速度方向设置为 y 轴,则初始速度和加速度分别由(2)式和(3)式来确定。爆炸后粒子的瞬时速度用(4)式来表示:

$$\begin{cases} xSpeed = 0 \\ ySpeed = \text{rand}() \% k_{ys} + A_{ys} \\ zSpeed = 0 \end{cases} \quad (2)$$

$$\begin{cases} xAcceleration = 0 \\ yAcceleration = \text{rand}() \% k_{ya} + A_{ya} \\ zAcceleration = 0 \end{cases} \quad (3)$$

$$\text{fire_Speed} = \text{rand}() \% k_{fs} + A_{fs} \quad (4)$$

其中, A_{ya} , A_{fs} 为平均速度, A_{xa} 为平均加速度, k_{ys} , k_{ya} , k_{fs} 为控制粒子的速度和加速度的参数。

初始颜色:颜色是烟花显示的一个重要特征,每个粒子的初始颜色分量包括粒子的三原色值 Red, Green, Blue。在燃点处发出的粒子初始颜色由(5)式来确定:

$$\text{Inital}C_r = \text{Inital}C_g = \text{Inital}C_b = \text{rand}() \% k_{rgb} / k_{rgb} \quad (5)$$

其中, k_{rgb} 控制烟花的颜色的参数,实验中选取 $k_{rgb} = 1000$ 。

生命周期:在烟花粒子诞生时就赋予其固定的生命周期。不同的烟花粒子,生命周期可以不同,随着时间的进行,当某些粒子的年龄达到生命周期时,它们就死亡,从系统中删除。程序中设粒子生命周期为 1.0,用式(6)来表示:

$$\text{life} = 1.0 \quad (6)$$

衰减速率:烟花粒子爆炸后的衰减速率由(7)式确定:

$$\text{fade} = (\text{float})(\text{rand}() \% k_{f1}) / k_{f2} \quad (7)$$

其中, k_{f1} , k_{f2} 控制烟花粒子的衰减速率的参数。

3.4 烟花粒子的运动状态

烟花粒子的运动描述了粒子在三维空间中的状态变化过程,主要包括空间位置、速度、颜色等的变化。在文中,烟花在整个燃放过程分两个阶段:一个是燃点→炸点,一个是炸点→消亡。在燃点→炸点期间,为简化程序,只考虑粒子所受的重力作用,忽略空气浮力及其它力的作用,在 y 方向作匀减速运动,直至速度为零,发生爆炸。爆炸后的粒子在爆炸力的作用下,具有一个瞬时的速度,然后以不同的形状炸开,如圆形、三叶型、心型等,如图 3 所示。

粒子在三维空间中的运动可由用简单的(8)式来表示:

$$\begin{cases} V = V_0 + \int A dt \\ P = P_0 + \int V dt \end{cases} \quad (8)$$

式中 A 为加速度, V 为速度, P 为位置。烟花粒子状态的变化可用 Euler 方法来计算,如(9)式所示:

$$\begin{cases} V(t + \Delta t) = V(t) + A(t) \cdot \Delta t \\ P(t + \Delta t) = P(t) + V(t) \cdot \Delta t \end{cases} \quad (9)$$

为了仿真真实感更好的烟花,文中对爆炸后的烟花颜色设计了几种表示方式。程序中纯色的烟花颜色,可以用式(4)表示,五彩缤纷的烟花可由(10)式来确定:

$$\begin{cases} C_r = \text{rand}() \% k_r / k_r \\ C_g = \text{rand}() \% k_g / k_g \\ C_b = \text{rand}() \% k_b / k_b \end{cases} \quad (10)$$

式 k_r, k_g, k_b 中控制烟花的颜色。

3.5 烟花粒子的消亡

烟花粒子在爆炸后经过一定的时间间隔后,就会从系统中消亡。文中粒子消亡的条件是粒子的生命周期以 fade 速率递减到零时,粒子完全透明,消失不见。

3.6 烟花绘制

图3给出了实验中仿真出的五彩缤纷的烟花,具有很好的真实感。

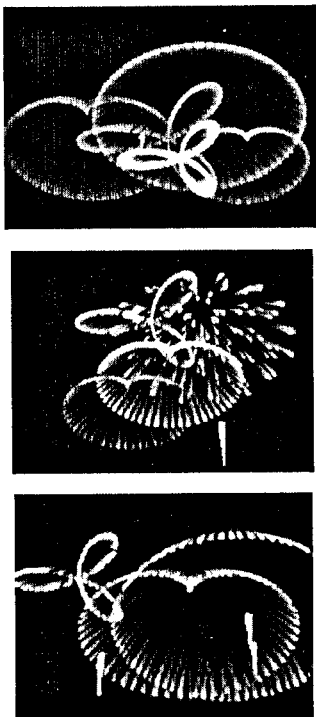


图3 仿真的五彩烟花

4 结束语

文中在分析粒子系统基本理论的基础上,采用粒

子系统对烟花进行了仿真。同时结合 OpenGL 图形库,仿真出了效果逼真的五彩缤纷的烟花,较传统的粒子系统仿真出的烟花视觉效果更好,而且速度快,简单。

参考文献:

- [1] Reeves W T. Particle systems - A technique for modeling a class of fuzzy objects [J]. ACM Transaction on Graphics, 1983,2(2):91-108.
- [2] 丁强,陈青林,左福强.基于 LOD 的火焰粒子生成技术[J].计算机应用,2005,25(12):257-259.
- [3] 庞新,王相海.基于 OpenGL 的礼花粒子系统模拟研究[J].计算机科学,2008,35(5):216-219.
- [4] Reeves W T,Blau R. Approximate and probabilistic algorithms for shading and rendering structured particle system[J]. Computer Graphics,1985,19(3):313-322.
- [5] Loke T, Tan D, Seah H. Rendering Fireworks Displays [J]. IEEE Computer Graphics and Applications, 1992, 12 (3): 33-43.
- [6] 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,2002,28(4):240-241.
- [7] 陈利平,王国才.基于粒子系统的蜡烛焰火实时模拟[J].计算机技术与发展,2006,16(5):186-188.
- [8] 甘露,王文永,孙博.基于粒子系统的烟花建模方法研究[J].长春师范学院学报:自然科学版,2008,27(1):63-66.
- [9] 罗玉玲.粒子系统与纹理映射相结合模拟礼花的研究[J].电脑知识与技术,2004,20:70-72.
- [10] 李清畅,杨高波,王小静.基于粒子系统的焰火建模及其算法仿真[J].系统仿真学报,2009,21(8):2179-2184.
- [11] 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,2006,19(7):28-30.
- [12] 王红霞,王文永,钟绍春,等.基于粒子系统的烟花仿真算法的改进[C]//第十四届全国图象图形学学术会议论文集.北京:清华大学出版社,2008:644-648.

(上接第99页)

计算机与数字工程, 2009,37(10):46-50.

- [5] 陈康,郑纬民.云计算:系统实例与研究现状[J].软件学报,2009,20(5):1337-1348.
- [6] 石永革,许建林,石峰.富客户端技术应用研究与实现[J].计算机工程与设计,2008,29(3):639-641.
- [7] 郭本俊,王鹏,陈高云,等.基于 MPI 的云计算模型[J].计算机工程, 2008,35(24):84-86.
- [8] Dean J. MapReduce: Simplified Data Processing on Large Clusters [C] // Proc. of the 6th IEEE Symposium on Operating System Design and Implementation. San Francisco, CA,

USA: [s. n.], 2004.

- [9] Lamell R. Google's mapreduce programming model - revisited [M]. Redmon, USA: Data Programmability Team Microsoft Corp, 2007.
- [10] 秦金磊,朱有产,李玉凯.基于网格计算的关键技术研究[J].计算机技术与发展,2006,16(11):103-105.
- [11] 万至臻,陈刚,寿黎但.基于 MapReduce 的并行计算平台的设计与实现[D].杭州:浙江大学, 2008.
- [12] 李樱,王永滨,王珂,等.广域网环境下分布式动漫渲染研究[J].微电子学与计算机, 2009,26(8):25-27.