

基于 OpenGL 的礼花粒子系统模拟研究^{*}

庞新¹ 王相海^{1,2}

(辽宁师范大学计算机与信息技术学院 大连 116029)¹

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)²

摘要 粒子系统是一种能够有效模拟不规则运动的景物或现象的图形生成算法。本文将粒子系统的关键方法和 OpenGL 纹理映射技术相结合,设计了一种礼花粒子系统的模拟生成算法。该算法利用粒子系统对礼花的属性和运动特征进行建模,再利用纹理映射技术显示计算得到的每个礼花粒子。实验结果表明,本文所提出的礼花模拟生成算法具有简单快速的特点,并且达到了较好的逼真度,适合实时性较强的应用。

关键词 实时,礼花模拟,粒子系统,纹理映射

Study on an OpenGL-based Simulation of Firework Particle System

PANG Xin¹ WANG Xiang-hai^{1,2}

(College of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)¹

(State Key Lab. of CAD&CG, Zhejiang University, Hangzhou 310027, China)²

Abstract Particle system is a graph generation algorithm which can efficiently simulate the sceneries and phenomena characterized by irregular motion. Combining the key techniques of the particle system and the OpenGL texture mapping technology, this paper designs a simulation algorithm of firework particle system. The proposed algorithm first models the properties and motion characters of firework using the particle system, and then display each computed particle through texture mapping. The simulation results show that this algorithm is simple, fast, and has good reality so that it is very suitable for real-time applications.

Keywords Real time, Firework simulation, Particle system, Texture mapping

1 引言

近年来,计算机游戏和动画等各种领域的不断发展,要求对动态自然景物,如云、雾、烟、火、雨、水等的运动进行模拟,这已经成为计算机图形学领域的研究热点。由于云、雾、烟、火等具有不规则的几何形状,并且其运动还具有不确定性,容易受到周围环境的影响而时时改变,动态性和随机性非常强,进而导致很难用常用的几何建模和数学建模方法来对其进行有效的模拟。1983年,Reeves提出了粒子系统^[1],采用许多形状简单的微小粒子作为基本元素来表示不规则模糊物体,充分体现了不规则模糊物体的动态性和随机性,被公认为模拟模糊物体最为成功的一种图形生成方法^[2]。目前,已有许多利用粒子系统方法模拟自然现象的研究工作,能够很好地模拟雨、雪、云等三维复杂自然景物。如 Peachey^[3]和 Fournier^[4]等人运用粒子系统模拟了风吹海浪所形成的水花;Goss^[5]运用粒子系统实时地模拟了船行驶时所形成的轨迹;万华根^[6]等采用粒子系统,通过求解流体力学中的 Navier-Stokes 方程的特例,模拟了喷泉的水流运动;Simon^[7]等人从光线在海水中传播规律出发,引入一个复杂的光照模型来模拟海面。但在模拟礼花的运动方面,国内鲜见相关的研究成果。王静秋^[8]和丁纪云^[9]等通过分析礼花燃放过程的特点,建立了模拟礼花的模型。可是利用这种方法生成的礼花,

如果粒子数多就会影响生成的速度,粒子数少则效果不好。考虑到礼花模拟的实时性和逼真度的要求,本文在深入研究上述方法的基础上,将粒子系统和 OpenGL 中的纹理映射技术相结合,提出了一种礼花特效的实时模拟生成算法。

2 粒子系统及其实现原理

粒子系统的基本思想是将许多简单形状的微小粒子作为基本元素聚集起来形成一个不规则的模糊物体,从而构成一个封闭的系统——粒子系统^[10]。用大量粒子来描述不规则景物的形状属性及其变化情况,每个粒子都有形状、大小、颜色、透明度、运动速度、运动方向、生命周期等属性。这些粒子是随着时间的推移处在不断变化中的粒子集合体,不是一个静态的整体。其中粒子群的分布可以改变,各个粒子的位置可以移动,新的粒子不断生成,同时旧的粒子不断死亡。

同其它描述不规则物体的方法相比,粒子系统具有以下三个特点:(1)对物体的描述不是通过原始的具有边界的面元(如多边形集合)来描述,而是通过一组定义在空间的原始粒子来描述;(2)粒子系统不是一个静态实体,每个粒子的属性均是时间的函数;(3)由粒子系统描述的物体不是预先定义好的,其形状位置等属性均用随机过程来描述。

自然界中的礼花是没有统一固定形状,没有千篇一律的颜色,礼花微粒的运动也没有提前设计好的轨迹,所有这些自

^{*} 本文受辽宁省高等学校优秀人才支持计划(RC-04-11)、辽宁省自然科学基金(2003205)和辽宁省教育厅科学技术研究项目(20060486)的资助。庞新 硕士研究生,研究方向为计算机图形学及动画技术;王相海 博士,教授,CCF 高级会员,主要研究领域为 CG、CAGD、图像编码和多媒体数字水印技术。

由随机不确定的属性和特征正适合粒子系统的方法来模拟。

通常用粒子系统绘制一幅画面需要如下步骤：

- Step 1. 分析物体的静态特性,定义粒子的初始属性;
- Step 2. 分析物体的运动规律,建立粒子属性变化的动态特性。
- Step 3. 在系统中生成具有一定初始属性的新粒子;
- Step 4. 根据粒子属性变化的动态特性改变其属性值;
- Step 5. 删除系统中已死亡的粒子;
- Step 6. 绘制所有剩余的粒子。

其中,Step 3-6 反复执行,就能够模拟物体的动态变化过程。

3 基于 OpenGL 的礼花模拟

3.1 纹理映射方法

根据经典粒子系统理论,要对礼花粒子进行三维建模和模拟生成的过程,其时间复杂度是比较高的。其原因在于,尽管粒子的基本形状为球、长方体等简单物体,但由于整个系统中含有大量的粒子,并且每个粒子都被作为一个点光源,这样就需要根据光照模型计算画面上每一像素的光亮度值,整个过程非常复杂,很难满足实时性要求。本文用贴图纹理模拟礼花粒子,利用 OpenGL 的纹理映射方法、Alpha 融合功能和深度测试功能,降低了粒子复杂度,减少了所需的粒子数量。这样,既能保证礼花的真实感效果,也能满足实时要求。

目前,有两种在屏幕上显示对象的基本方法^[11]。一种是将对象作为几何对象建模显示,另一种是将对象作为像素块显示,但是这两种方法都存在着一定的局限性。纹理映射技术集成了每种方法的最佳特性,将像素和几何对象组合,能够提供较为复杂的图像的同时,又不会带来构建大量几何模型所需的开销。一般的纹理映射方法中,主要是正向纹理映射和反向纹理映射方法^[12],表 1 对这两种方法的优点和不足进行了分析。

表 1 两种纹理映射方法的比较分析

纹理映射类型	图形显示算法	优点	不足
正向纹理映射	不依赖图形算法,直接针对纹理空间中一个元素计算其在图像空间中的图像元素	节省内存	容易引起纹理混淆,形成孔洞,图像失真,影响纹理映射速度。
反向纹理映射	依赖图形显示算法,如扫描线算法、Z 缓冲算法、光线跟踪算法等	实现纹理映射	要求计算机有大量内存

3.2 对贴图的处理

贴图纹理是一个 32×32 的四边形(如图 1 所示),用它代替粒子,一个平面片可以代替几百个粒子,从而可以大大提高实时性。然而随着观察者视点的移动,当视点正好处在平面片的侧面时,就不能看到粒子,影响系统的逼真度。然而在 OpenGL 坐标系下,只要始终保持粒子平面的法线与观察者视线之间夹角在 0°~45°之间,而超过 45°时就自动转成 0°角处理(0°角时粒子平面的法线与观察者视线平行),当就可以有效地解决上面的问题。设粒子沿 Y 轴的转角 Ry、沿 X 轴的转角 Rx,同时在刷新粒子属性时提供视点的坐标信息 ViewP,这样就能在绘制每个粒子时,实时计算它相对于视点的方向,通过与旋转矩阵相乘,从而使得观察者不会看到粒子

平面的侧面。具体算法如下:

设当前新生粒子面的法线 $f = \{0, 0, 1\}$, 粒子面片的位置为 $lp = \{x_p, y_p, z_p\}$, 视点的位置为 $ViewP = \{x_v, y_v, z_v\}$ ^[13]。则此时粒子面片与观察者之间的方向矢量为

$$V = \{x_p - x_v, y_p - y_v, z_p - z_v\},$$

粒子与视点连线在 XOZ 平面上的投影距离为

$$dy = \sqrt{(x_p - x_v)^2 + (z_p - z_v)^2},$$

为了使粒子面片的法线与 V 重合,必须在 X、Y 轴上做适当旋转,即先以沿 X 轴旋转 Rx 角度,再沿 Y 轴旋转 Ry 角度,且 $Rx = -\arctan((y_p - y_v)/dy)$, $Ry = 90 - \arctan((z_p - z_v)/(x_p - x_v))$ 。其中,arctan 表示反正切函数。

这种方法比用大量粒子节约了很多计算时间和存储空间,大大提高了显示速度。



图 1 礼花粒子的贴图纹理

4 礼花模型设计

4.1 礼花粒子的结构设计

4.1.1 礼花粒子的数据结构

```

struct particle
{ float t; // 粒子的寿命,随机产生一个 0~1 之间的整数
  float v; // 粒子的运动速度
  float d; // 粒子的运动方向
  float x, y, z; // 粒子的位置坐标
  float r; // 颜色 Red 的值
  float g; // 颜色 Green 的值
  float b; // 颜色 Blue 的值
  float xd, zd, yd; // X, Y 和 Z 方向上的增量
  char type; // 粒子的状态(移动或淡化)
  float a; // 粒子淡化的 alpha 值
  struct particle * next, * prev; // 双向链表前插法生成粒子
};
    
```

4.1.2 礼花粒子的存储结构

模拟过程中,根据粒子系统理论,每一帧都要产生新的粒子或者删除死亡的粒子,不时得对系统中的粒子进行更新,所以整个系统采用 P 组双向链表来存储。这样遍历、插入、删除粒子非常方便,提高了系统效率。粒子系统存储结构如图 2 所示,图 2 仅仅给出一组粒子结构,其它(P-1)组粒子结构与其相同。本结构采用前插法插入节点。



图 2 粒子系统的双向链表存储结构

4.2 粒子属性

为了保证礼花生成效果的真实感,本文的算法采用多种属性对礼花粒子进行描述,包括:初始数量、初始位置、初始颜色、亮度、初始速度和加速度等。

4.2.1 初始化粒子数量

粒子数量是影响礼花系统实时性和逼真度的重要因素。粒子数太多礼花实时性差,而且效果不一定好;粒子数少则实时性高,但过少就逼真度太差。人们通常对粒子数量的控制分固定控制和动态控制。经过大量实验,发现无论是哪种控制针对礼花而言,60~120 个粒子时逼真度比较高,而实时性也很好。文中采用一个粒子产生器,当初始化时随即产生 60

~120 之间的一个整数作为礼花粒子数。

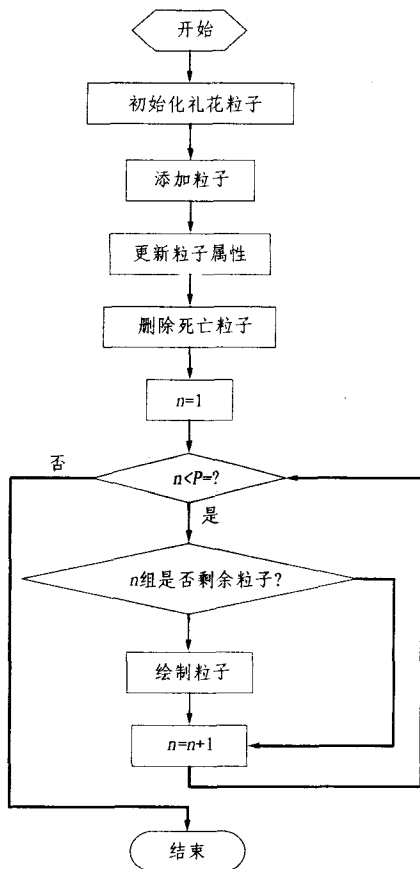


图3 模拟礼花生成过程的流程图

4.2.2 粒子初始位置

为了保证礼花的形状,程序中规定了粒子产生和死亡的

区域。例如要生成一个“天女散花”的礼花,则粒子活动区域为一个半径为 R 的一个球形区域。而粒子必须在此区域中随机产生,设初始化粒子位置 $P(x, y, z), x = rand() \% R, y = rand() \% R, z = \sqrt{R^2 - x^2 - y^2}$ 。

4.2.3 初始颜色

颜色也是礼花的一个重要特征,程序中将每个粒子的颜色用 (r, g, b, a) 表示, r 为红颜色的值, g 为绿颜色的值, b 为蓝颜色的值, a 为亮度。由于现实中礼花的颜色比较有限,程序中将 12 种常见礼花颜色存到数组中,当添加粒子时,就随机在数组中选择一种颜色作为粒子颜色,提高了礼花的实时性。

4.2.4 亮度

礼花显示的另一个重要特性是亮度,当礼花爆炸后,粒子最亮,然后逐渐变暗,最后消失。程序用 alp 表示粒子的亮度, $alp \in [0.0, 1.0]$, 亮度值为 1.0 表示粒子的颜色最亮, 0.0 表示颜色最暗。当 alp 逐渐减小到 0.0 时,礼花也由明亮逐渐变暗最后消失。

4.2.5 初始速度和加速度

礼花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。其中,速度方向是随机的,由 $v = rand() \% M$ 确定,其中 v 代表产生粒子的速度, M 代表一个定值。加速度分别表现在 X, Y, Z 三个方向上, X 方向加速度为 xd , Y 方向加速度为 yd , Z 方向加速度为 zd 。本文中, xd, yd 和 zd 都取一个定值 N 。

4.3 礼花的运动过程

4.3.1 粒子生存期的变化

粒子生存期表征了其在屏幕上停留的时间,越长则粒子运动的轨迹越远。粒子生存期直接关系到礼花外形的大小和粒子消失的快慢。设 t 表示粒子的生存期,且 $t \in [0.0, 1.0]$ 。当产生粒子时随机赋予粒子一个 0.0 到 1.0 之间的数。随着时间的推移, t 逐渐减小。当为 0.0 时,说明粒子已经死亡,从链表中将其删除,并且不再显示此粒子。

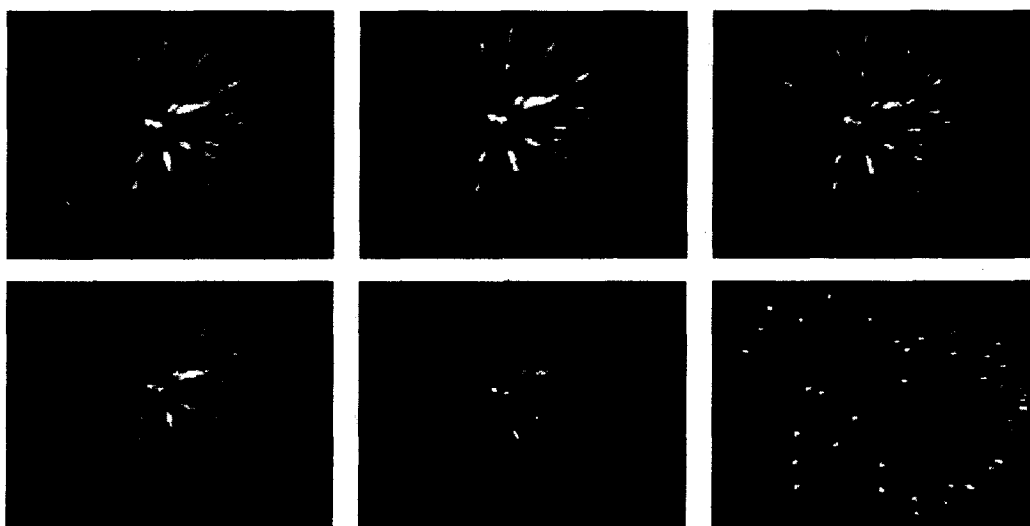


图4 礼花模拟效果图

4.3.2 力场中的运动

礼花粒子在空中要受到多个力的作用,有竖直向下的重力,竖直向上的浮力,还有风力等其它外力。这些力组成的力场会对礼花粒子产生重要的影响。由于实际上礼花粒子非常轻,非常小,受到的重力略大于空气浮力,所以看上去是缓缓

下降的,而不是自由落体运动。在本文的算法设计中,对力场进行了简化,仅仅考虑重力和浮力作用,而忽略风力等其它外力作用。当然,这样做是在能够保证礼花系统具有较高的逼真度的前提下进行的。

4.4 礼花模拟过程

根据上述的分析,利用粒子系统模拟礼花生成的过程就是对 P 组礼花粒子的属性进行不断更新的过程。为了保证礼花的模拟过程具有实时性和真实感,需要以一定的时间间隔更新每组粒子的属性。图 3 所示为本文算法在一个时间单位内对每组粒子所进行的操作。

5 实验结果与分析

为了验证所提出算法的有效性,对模拟礼花的过程进行了实验,实验中,参数 $M=100, N=0.05, P=60$ 。模拟是在配置为 Pentium(R)4 CPU2.66GHz、DDR256M 内存及 GeForce 5200(64M)显卡的 PC 机上进行的。实验结果如图 4 所示。从图中可以看出,在模拟礼花时能具有较逼真的效果,并且实时性也较好,画面流畅达到 30 帧/秒。

结束语 对于粒子系统而言,用传统的方法建立粒子,粒子的显示已经很消耗时间和资源,加之采用传统画面绘制算法将导致复杂的消隐计算。当粒子数很大时,实时性往往达不到。本文设计合适的二维粒子纹理面片,应用 OpenGL 的纹理映射技术等对礼花进行模拟,能够实时显示礼花,在兼顾实时性与逼真性方面取得不错的效果。考虑到较强的真实感效果,应该在以后的工作中把风力等外力作用考虑进去,使这个系统更逼真。

参考文献

[1] Reeves W T. Particle Systems-A Technique for Modelling a Class of Fuzzy Objects[J]. Computer Graphics, 1983, 17(3):

359-376

- [2] 张芹. 基于粒子系统的建模方法研究[J]. 计算机科学, 2003, 30:144-146
- [3] Peachey D. Modeling waves and surf. ComputerGraphics[J]. 1986, 20(4):65-74
- [4] Fournier A, Reeves W T. A simple model of ocean waves[J]. Computer Graphics, 1986, 20(4):75-84
- [5] Goss M E. A real time particle system for display of ship wakes [J]. IEEE Computer Graphics and Applications, 1990, 10(3): 30-35
- [6] Wan Hua-Gen, Jin Xiao-Gang, Peng Qun-Sheng. Physics based real time animation of fountain[J]. Chinese Journal of Computers, 1998, 21(9): 774-779(in Chinese)
- [7] Simon P, Michael A. Rendering natural waters[C]// Proc. of Pacific Graphics. 2000: 23-30
- [8] 王静秋,钱志峰,基于粒子系统的焰火模拟研究[J]. 南京航空航天大学学报, 2001, 33(2):166-170
- [9] 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, 275
- [10] 唐荣锡,汪嘉业,彭群生,汪国昭. 计算机图形学教程. 科学出版社, 1990
- [11] Angel E. OpenGL 程序设计指南. 李桂琼,张文祥,译. 清华大学出版社, 2005
- [12] 张英杰,赵汝嘉. 一种实现正向纹理映射的新方法[J]. 工程图学学报, 1997(1):161-165
- [13] 刘耀周,张锡恩. 基因粒子系统的导弹飞行航迹及烟雾的特效生成[J]. 计算机工程, 2004, 30:174-176

(上接第 215 页)

5) 调用角色状态分配模块,搜索角色状态分配集 ROS,并判定用户所属角色在当前客体状态是否有效;若无效,则返回用户请求失败信息,否则进行下一步;

6) 调用权限状态分配模块,查找状态权限分配集 POS 中当前客体状态下允许的权限操作与访问请求是否匹配;若不匹配,则返回用户请求失败信息,否则进行下一步;

7) 实时监控模式以观察者模式为设计思想,即以客体为被观察的目标对象,以角色和权限为其观察对象。当客体的状态发生改变时,它的观察者们将得到通知,并自动、连锁地做出相应的响应行为。在整个会话过程中,如果客体状态没有发生转换,则依原序执行用户请求的权限操作;否则,由实时监控模块检测到的客体状态变化信息将被及时返回给同构化客体状态适配模块,同时更新客体状态集 OS 中的状态信息,连锁激活相关状态,并通知角色状态分配模块及权限状态模块,而与之对应的角色和权限也将被重新适配,系统进程将返回本小节步骤“4”)。

至此,完成一个用户访问的会话监控过程,也就完成了该角色与其权限的系统资源动态适配。

结束语 本文依据同构化基本原理,提出了一种基于同构化客体状态适配监控的 RBAC 扩展模型。它从角色访问控制的客体(即被访问对象)出发,在其角色与权限之间新增设一个基于客体状态的同构化适配监控层,使不同客体状态所对应的角色及其权限建立起关联和适配机制,可在客体状态发生转换而引起其相关角色和权限变更时,通过观察者模式正确地及时捕捉、自动回应和动态适配这些转换和变更,以

尽可能地避免访问冲突、阻绝非法访问;从而,能对各客体在不同状态下的安全访问与访问安全,施行更有序、高效的实时敏捷控制。本模型仅为客体状态概念模型,它意在突出不同客体状态的访问控制方式,侧重于客体状态的设置、状态转换时的信息流动及转换后的权限调度与适配监控。对可能引起适配调度失败的各类原因(诸如:权限不足、客体访问冲突、客体状态约束等)及其解决方案,本文尚未作深入讨论;而这些正有待我们同更多学者一起进行其后续研究。

参考文献

- [1] Sandhu R, Coyne E, Feinstein H, et al. Role-based Access Control Models[J]. IEEE Computer, 1996, 29(2): 38-47
- [2] American National Standard 359-2004 is the Information Technology Industry Consensus Standard for RBAC[EB/OL]. http://csrc.nist.gov/rbac/, 2005-10-20
- [3] 耿晖,王海波. 基于 XML 的角色访问控制(RBAC)[J]. 计算机应用研究, 2002, 19(12):14-15, 42
- [4] 洪帆,赵晓斐. 基于任务的访问控制模型及其实现[J]. 华中科技大学学报(自然科学版), 2002, 30(1):17-19
- [5] 刘益和,刘嘉勇. 一个基于角色的信息流模型及应用[J]. 四川大学学报, 2004, 36(5):236-243
- [6] 孙鹏,杨德婷,等. 网格环境下动态访问控制模型的研究与实现[J]. 计算机工程与应用, 2007, 43(11):132-134
- [7] 韦超鹏,韩继红,王亚弟. 一种基于域分层概念的 RBAC 模型[J]. 微计算机信息, 2006, 22(03):296-298
- [8] 周启海. C++同构化对象程序设计原理[M]. 北京:清华大学出版社,北方交通大学出版社, 2004, 2
- [9] Gamma E, Helm R. 设计模式:可复用面向对象软件的基础[M]. 北京:机械工业出版社, 2000, 2