虚拟雪景物理建模与仿真

石 敏,胡海涛,赵 颖
SHI Min,HU Hai-tao,ZHAO Ying

华北电力大学,北京 102206
North China Electric Power University,Beijing 102206,China
E-mail:shimin01@ict.ac.cn

SHI Min,HU Hai-tao,ZHAO Ying.Physical modeling and simulation of snow scene.Computer Engineering and Applications,2010,46(6):188-190.

Abstract: A simplified model of movement that snowflakes float is established based on the dynamics principle.The virtual scene that snow float is rendered based on particle system is implemented by Direct3D technologies.A simplified model of particle launchers and similar wind model are proposed so that computing speed and simulation effect are improved.Then,a method of collision detecting is put forward.Experiments show that natural result and real-time speed can be achieved by this method.And,the method can also be used to other scene simulation,such as rain,smoke,fog,and so on.

Key words: virtual snow scene;particle system;Direct3D

摘 要:基于动力学原理建立了雪花飘落的简化运动模型,并在 D3D 技术基础上,基于粒子系统的原理,绘制渲染了雪花纷飞的动态虚拟场景。提出了简化粒子发射器模型和碰撞检测方法,并对风力模型作了近似,在保证仿真效果的基础上提高了运算速度。实验结果表明,该方法仿真效果较好,速度快,同时能容易地推广应用到其他诸如雨、烟、雾等虚拟场景的绘制中。

关键词:虚拟雪景;粒子系统;3D 图形函数库
DOI:10.3778/j.issn.1002-8331.2010.06.055 **文章编号:**1002-8331(2010)06-0188-03 **文献标识码:**A **中图分类号:**TP391.9

1 引言

近年来,随着虚拟现实技术的迅速发展,真实感三维虚拟场景的建模和绘制成为计算机图形学领域的一个研究热点。特别是在游戏、广告、影视和动漫等应用领域,除了需要绘制静态三维场景,诸如雨雪、风、云、火焰和爆炸等实时动态效果的展现也非常必要。然而,不同于静态场景的描述,这些动态自然景物具有不规则性、随机性和复杂性,用传统的方法对其进行建模和绘制十分困难。自从 William T.Reeves[1]用粒子系统模拟了行星被撞击后产生的爆炸和火焰效果之后,基于粒子系统模拟不规则模糊体的研究工作就蓬勃发展起来。Peachey[2]和 Fournier[3]等人在海浪模型的研究中运用粒子系统模拟了浪花。Goss[4]运用粒子系统实时地模拟了船行驶时所形成的轨迹。国内的万华根[5]等人采用粒子系统模拟了喷泉的水流运动。其他的研究者基于粒子系统进行了各种动态场景的模拟。

D3D 是一组 3D 图形函数库,可以进行三维图形的绘制和渲染。D3D 通过库中的接口函数设置渲染流水线的各种过程参数(包括:顶点的世界坐标、各种变换矩阵、光照和纹理参数等),然后启动渲染管道流水线,实现三维图形显示。D3D 以其强大的渲染引擎、良好的硬件兼容性以及友好的编程方式被广泛应用于图形系统,特别是三维游戏的开发上[6]。因此,它非常适合于三维虚拟场景的显示。

2 粒子系统

粒子系统是把模糊物体定义为由成千上万个运动的、不规则的、随机分布的粒子所组成的粒子集。每个粒子有一组属性,如位置、速度、颜色和生命周期。它们不断改变形状和运动,从而表现出景物的总体形态和特征的动态变化。一个粒子有什么样的属性,主要取决于具体的应用。

粒子通常由位于空间某个地方的粒子源产生,粒子的初值由随机过程控制。粒子系统是动态变化的,在生命的每一刻,都要完成 4 步工作:(1)粒子源产生新粒子;(2)更新粒子属性;(3)删除"死"粒子;(4)绘制粒子。

粒子系统首先需要产生任意数目的新粒子,它们的初始属性由随机过程控制。在粒子运动过程的每一帧数据中,系统根据运动规律更新粒子现有属性。每个粒子都有一个生命期,如果生命期结束,粒子变为"死粒子";而如果粒子具有无限长生命期,则意味着该粒子永远不会从粒子系统中消失。在运动过程的每一帧中更新粒子属性之后,系统对粒子的生命期进行检查,当生命周期结束时,将粒子从系统中删除。最后,渲染显示粒子系统中所有现存的粒子。

基于粒子系统的雪景模拟首先定义了粒子属性,建立了粒子物理模型,根据自然现象中的运动规律,模拟雪花受到的重力以及风力,计算出粒子瞬时速度和在场景中的坐标位置并将其在三维场景中绘制。

作者简介:石敏(1975-),女,讲师,博士研究生,研究方向为虚拟现实;胡海涛(1973-),男,博士,讲师,研究方向为智能交互。
收稿日期:2008-09-15 **修回日期:**2008-12-09

3　粒子结构和运动模型

3.1　粒子结构

一个粒子是一个非常小的物体,通常被模拟为数学上的一个点。一个点原(point primitive)是表示一个粒子的很好的候选方案。然而,点原被光栅化为一个像素时,不能满足不同粒子大小和设置不同材质的需要。通常可以采用一个公告栏(billboard)表示粒子。

采用 D3D 中的点精灵(point sprite)来适应粒子系统的需要。一个单个的点就可以描述一个点精灵,点精灵可以使用纹理贴图并且改变大小。粒子通常具有位置、速度、颜色、生命期等基本属性,基本粒子结构描述如下:

```
struct Particle{
    D3DXVECTOR3 m_vPosition;
    D3DCOLOR m_Color;
    float m_fLife;
    bool m_isalive;
    D3DXVECTOR3 m_vVelocity;
    float m_fSize;
    float m_fMass;
    D3DXVECTOR3 m_fFriction;
}
```

m_vPosition—世界坐标系中粒子的位置;m_vVelocity—粒子的速度;m_fSize—粒子的大小;m_fLife—粒子的生命期;m_fMass—粒子的重力;m_Color—粒子的颜色;m_isalive—true 表示粒子活着,false 表示它死了;m_fFriction—粒子受到的阻力。

设计了粒子属性结构之后,可以将雪花粒子系统中的大量粒子存储在粒子数组中,然后分块处理。

3.2　粒子运动模型

现实生活中,雪片和雨滴的降落轨迹是千变万化的。运动的雪花或者雨滴要受到重力、风力、空气阻力等各种外力的作用,而这些外力也不是一成不变的。因此,受这些外力作用的"小粒子"的运动方程是非常复杂的。在计算机模拟过程中,如果要将所有存在的作用因素都考虑到运动方程中几乎是不可能的。所以,在不影响仿真效果和保证运算速度的前提下,可以对复杂的运动方程作如下简化:

设粒子在运动中受到的外力为 $\boldsymbol{F}$,则:$\boldsymbol{F}=\boldsymbol{G}_{重}+\boldsymbol{f}_{空}+\boldsymbol{f}_{风}$。其中,$\boldsymbol{G}_{重}$为粒子所受重力,$\boldsymbol{f}_{空}$为空气阻力,$\boldsymbol{f}_{风}$为横向风力。然后根据牛顿第二定律 $\boldsymbol{F}=\boldsymbol{Ma}$ 求得粒子运动加速度,进一步计算出每一时刻粒子的运动速度 $\boldsymbol{v}$ 和位移 $\boldsymbol{x}$。运动方程如式(1):

$$\begin{pmatrix}\dot{\boldsymbol{x}}\\ \dot{\boldsymbol{v}}\end{pmatrix}=\begin{pmatrix}\boldsymbol{y}\\ \boldsymbol{M}^{-1}\boldsymbol{F}\end{pmatrix} \tag{1}$$

其中,$\boldsymbol{x}$ 和 $\boldsymbol{v}$ 分别为 3×1 位移向量和速度向量,$\dot{\boldsymbol{x}}$和$\dot{\boldsymbol{v}}$分别表示粒子位移和粒子速度的导数,$\boldsymbol{M}$ 是质量矩阵,它为 3×3 的对角阵,$\boldsymbol{F}$ 为 3×1 的力矩阵。

对上面运动方程离散化求解如式(2):

$$\begin{pmatrix}\boldsymbol{v}(t+\mathrm{d}t)\\ \boldsymbol{x}(t+\mathrm{d}t)\end{pmatrix}=\begin{pmatrix}\boldsymbol{v}(t)+\boldsymbol{a}\mathrm{d}t\\ \boldsymbol{x}(t)+\frac{1}{2}(\boldsymbol{v}(t)+\boldsymbol{v}(t+\mathrm{d}t))\mathrm{d}t\end{pmatrix} \tag{2}$$

其中,$\mathrm{d}t$ 为单位步长,t 为当前时刻,$\boldsymbol{v}(t)$为当前时刻速度,$\boldsymbol{x}(t)$为当前时刻位移,$\boldsymbol{v}(t+\mathrm{d}t)$为下一时刻速度,$\boldsymbol{x}(t+\mathrm{d}t)$为下一时刻位移。

4　雪景模拟算法实现

4.1　算法框架

在仿真中,首先需要设置一个粒子源,使用随机过程不断产生大量雪粒子,并设置粒子的初始属性。在运动过程的每一帧,根据运动方程计算雪花粒子的新速度和新位置,并对其进行相应的属性更新,如果满足条件,则进行绘制。整个算法流程如图 1 所示。

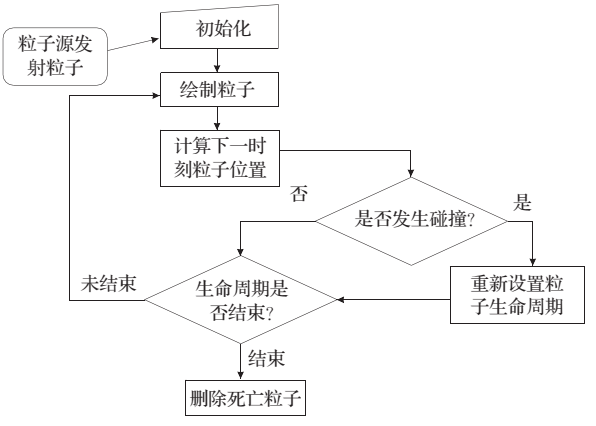

图 1　算法流程图

4.2　雪粒子产生

大量雪粒子是由粒子源发射产生的。通常的方法是将粒子源发射器设置为一个半径为 R 的圆。该文使用的方法是:将发射器设计成为一个半径为 R 的扇形,扇形中心角 θ 依赖于视点可以观察到的角度,发射器位置离地面具有一定的高度。为了保证视点变化时雪景模拟依然逼真,始终将粒子发射器和视点方向保持一致,即:视点转动时,发射器也进行相应角度的转动。粒子发射器位置和方向如图 2 所示。

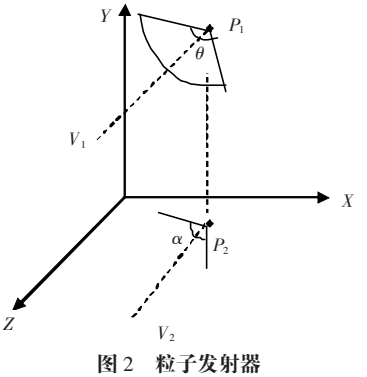

图 2　粒子发射器

图中,P_1 和 P_2 分别表示发射器和视点中心位置,θ 为发射角,α 为视角,V_1 和 V_2 分别为发射器和视点正对方向,二者方向相同。其中,$\theta=\alpha+k$,$k>0$,为补偿,以保证视点所见场景始终处于雪景范围。当发射器绕 Y 轴旋转时,视点需作相应旋转。

如此设置方式一方面减少了粒子数量,用尽可能少的粒子仿真大规模场景中的实时雪景,同时也保证了各种视点看到的场景都处于雪景范围之内。从而使得雪景模拟逼真、绘制速度更快。

4.3　雪粒子的运动与消亡

设世界坐标系中雪粒子的初始位置为 $\boldsymbol{P}$,粒子所受重力由粒子质量决定,初始速度 $\boldsymbol{v}$ 由随机函数确定,空气阻力 $\boldsymbol{f}_1$ 与速度方向相反,计算方法如式(3):

$$\boldsymbol{f}_1=-k\boldsymbol{v} \tag{3}$$

其中,k 为阻力系数,可根据经验取得。

可以考虑风力对粒子运动的影响。采用的一种近似方法:假定风力由风速大小决定,而风向与水平面 XZ 平行,则风力 $\boldsymbol{f}_2$ 可如下式(4)计算:

$$\boldsymbol{f}_2=-k_w\boldsymbol{v}_w \tag{4}$$

式中，$\boldsymbol{v}_w$ 为风速，k_w 为调节系数。设风向与 X 轴夹角为 β，则 $\boldsymbol{v}_w=[v_w\cdot\cos\beta,0,v_w\cdot\sin\beta]^{\mathrm{T}}$。

雪粒子受力图 3 所示。其中，P：粒子；$\boldsymbol{v}$：粒子速度；$\boldsymbol{f}_1$：空气阻力；$\boldsymbol{f}_2$：横向风力；$\boldsymbol{mg}$：粒子所受重力；虚线所示平面表示与粒子 P 处于相同高度且平行于水平面的平面，$\boldsymbol{f}_2$ 在平面内。

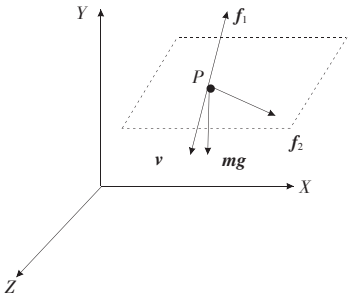

图 3　雪粒子受力分析图

确定了受力和初值，粒子每一帧的运动状态即可由上节所给出的运动方程确定。在雪粒子运动过程中，需要判断其是否与场景中的障碍物发生碰撞，一旦检测到碰撞发生，则根据衰减系数重新设置雪粒子的生命周期并置速度为 0。采用简单的碰撞检测方法：判断当前帧的雪粒子运动轨迹是否穿透包围盒的上边界，如果是，则碰撞发生，生命周期近似为 $life=life(1-\alpha)$ 其中，$0<\alpha<1$ 为生命衰减系数；否则认为没有发生碰撞，$life=life-k$，k 通常可设为 1。当生命数为 0，粒子变为“死”粒子。此方法对于三维空间中的正方体、长方体等外形简单规则的障碍物切实可行，但是当障碍物表面很复杂的时候，需要重新修改碰撞检测方法。

4.4　基于 D3D 的雪粒子绘制

在 D3D 中渲染粒子点时候，点的大小随着点与摄影机的距离自动调整。点大小计算公式如下。其中，*FinalSize* 是点精灵的最终尺寸，*ViewportHeight* 是视口的高度，*Size* 是根据渲染状态指定的值。

$$FinalSize=ViewportHeight\cdot Size\cdot\sqrt{\frac{1}{A+B(D)+C(D^2)}}$$

在视空间中，点精灵与摄像机的距离 $D=(x^2+y^2+z^2)^{1/2}$，其中 (x,y,z) 是点精灵在视空间中的位置。A，B，C 构成了关于 D 的二次系数。在视口和输入点大小已经知道的情况下，点的大小变化就由系数 A，B，C 决定。

点精灵是用一个正方形来渲染的，正方形的中心位置就是点精灵的位置。以点 $p(x,y,z)$ 为中心，s 为边长，可以确定出正方形 4 个顶点的坐标和纹理坐标，如图 4 所示。为每个图像像素点插入 Alpha 混合值，将方形的黑色部分镂空，将白色部分显示出来，就可实现一个圆形点。如图 5 所示。

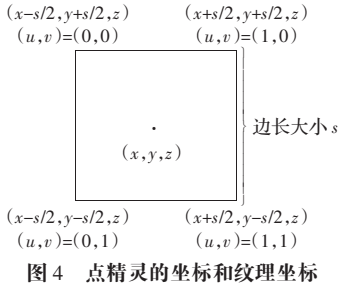

图 4　点精灵的坐标和纹理坐标

图 5　界定点的形状

设置源图像和目标图像的颜色混合因子，这样点精灵的黑色部分与背景色混合，白色部分为点精灵本身的颜色。将点精灵的位置和颜色数据写入创建的顶点缓冲区，进行渲染，就可以绘制出一系列颜色和形状自定义的点精灵。

粒子系统是动态的，需要在每一帧更新粒子状态。创建一个能够保存最大数量粒子的顶点缓冲，将粒子数组中活粒子的位置和颜色写入顶点缓冲区，然后进行点的渲染。为了提高系统效率，将顶点缓冲区分为几块，并且使顶点渲染和顶点数据写入并发进行。粒子数组、顶点缓冲区以及缓冲区分块之间的关系如图 6 所示。

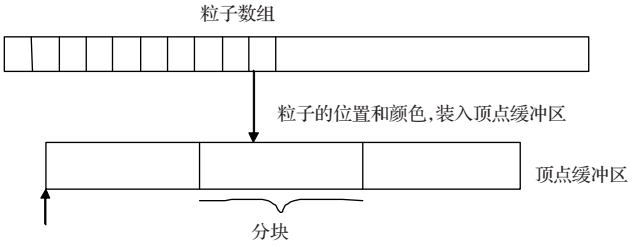

图 6　粒子数组与顶点缓冲区

5　实验结果

实验硬件环境为主频 2.26 GHz、内存 1 GB、显存 128M 的 PC 机，选择 VC++开发平台，利用 D3D 图形库进行模拟。实验结果如图 7 和图 8 所示。实验结果表明该文方法具有较好的模拟效果。

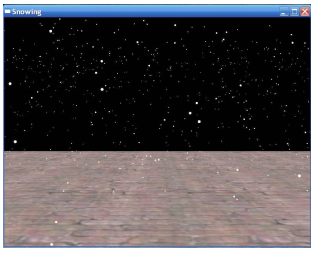

图 7　雪景模拟

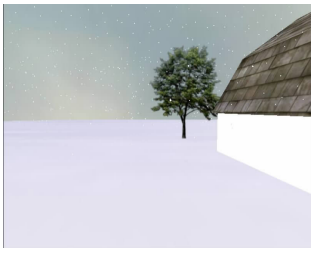

图 8　加入三维场景的虚拟雪景

6　结论

该文基于动力学原理建立了雪花飘落的简化运动模型，并在 D3D 技术基础上，基于粒子系统的原理，绘制渲染了雪花纷飞的动态虚拟场景。文中提出了简化粒子发射器模型，并对风力模型作了近似，在保证仿真效果的基础上提高了运算速度。实验结果表明，该文方法仿真效果较好，速度快。该文方法也能容易地推广应用到其他诸如雨、烟、雾等虚拟场景的绘制中。

在雪景模拟中没有实现雪花下落到地面后的堆积效果，在有起伏的地形上产生堆积需要地形分割和更精确的碰撞检测等方法的支持，这将是下一步研究工作。

参考文献：

[1] Reevs W T.Particle systems:A technique for modeling a class of fuzzy objects[J].Computer Graphics,1983,17(3):359-376.

[2] Reachey D.Modeling waves and surf[J].Computer Graphics,1986,20(4):65-74.

[3] Fourier A,Reeves W T.A simple model of ocean waves[J].Computer Graphics,1986,20(4):75-84.

[4] Goss M E.A real time particle system for display of ship wakes[J].IEEE Computer Graphics and Applications,1990,10(3):30-35.

[5] Wan Hua-gen,Jin Xiao-gang,Peng Qun-sheng.Physics based real time animation of fountain[J].Chinese Journal of Computer,1998,21(9):774-779.

[6] 陈卡.Directx9 3D 图形程序设计[M].上海：科学技术出版社，2003.