

一种大场景环境下降雪和积雪仿真的新方法

贺怀清^{1,2)} 薛军涛²⁾

¹⁾(天津市智能信息与图像处理重点实验室, 天津 300300) ²⁾(中国民航大学计算机科学与技术学院, 天津 300300)

摘要 在分析已有的降雪和积雪模拟方法的基础上,提出了一种大场景环境下降雪和积雪模拟的新方法。将降雪和积雪过程看成雪景模拟的两个阶段,分别用不同的方法进行模拟。在降雪方面,引入粒子的旋转,粒子动态纹理和动态颜色等,提高了降雪模拟的真实性和灵活性;在积雪方面,引入噪声技术,解决了已有积雪模拟方法不能用于大场景模拟的问题。实验表明,降雪积雪模拟方法模拟效果真实,实时性良好,适合大场景尤其是飞行模拟系统中。

关键词 粒子系统 降雪模拟 积雪模拟 噪声

中图分类号: TP391.9 **文献标识码**: A **文章编号**: 1006-8961(2007)09-1676-06

A New Approach for Simulation of Snow Falling and Accumulating for Large-scale Terrain

HE Huai-qing^{1,2)}, XUE Jun-tao²⁾

¹⁾(Tianjin Key Laboratory for Advanced Signal Processing, Tianjin 300300)

²⁾(College of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300)

Abstract Based on the analysis of traditional simulation methods, this paper presents a new approach for simulation of snow falling and accumulating in large-scale terrain. The snow simulation is divided into two stages-falling and accumulating. In the former stage, rotation, dynamic texture, and dynamic color of particles are taken into account, which greatly improves the reality of simulation. In the later stage, image noise is introduced to a large-scale terrain, which solves the problem that the traditional methods cannot be scaled up to the large-scale terrain. Experiment shows this method improves the in snow simulation, keeps real-time, and adapts to the large-scale terrain, especially for the scene with flight simulation.

Keywords particle system, falling snow, accumulating snow, image noise

1 引言

随着计算机图形图像技术的高度发展,可视化系统的应用日益广泛,人们对可视化系统的模拟逼真度的要求也越来越高。由于雪会给各种机动设备的驾驶员带来许多视觉上的障碍,从而带来相应的操作困难,所以无论从计算机图形学的角度,还是从模拟训练的角度,在视景中创建或研究雪的计算机

模拟均有重要的意义和广泛的应用。

对下雪等复杂自然现象的模拟一直是计算机图形学中具有挑战性的研究方向之一^[1]。根据自然环境中降雪的过程,可以把对雪的模拟分为两个阶段:一个是降雪过程,即雪降落到地面前的过程,另一个是积雪过程,即雪降落到地面形成积雪的过程。这两个阶段都有其难点,前一个阶段主要特点是:由于雪本身的随机性和离散性,雪花降落速度复杂性,运行轨迹变化不一,无法用公式简单地模拟,因此这

基金项目:国家自然科学基金项目(60572169)

收稿日期:2006-06-12;改回日期:2006-07-17

第一作者简介:贺怀清(1969~),女,副教授,副院长。2001年获哈尔滨工业大学计算机系工学博士学位。研究方向为图形图像与虚拟现实。E-mail: huaiqinghe@yahoo.com.cn

个阶段主要根据雪花的受力情况,包括:重力、浮力、阻力等以及风的作用,建立动力学方程,实时控制雪花移动;后一个阶段主要特点是:在积累的过程中,不但要做降雪数量的统计,而且要考虑各种受力、地形、环境等多方面因素。

在雪模拟方面,研究者们基于粒子系统^[2]已经做了一定量的研究工作,有的研究者只考虑降雪本身,如:

谢等人^[3]提出了基于粒子系统的雪花降落的实时模拟生成算法。文章中并没有分析雪花粒子的受力情况,只是用简单的随机函数动态控制雪花粒子的运动,在显示上引入视觉感知心理学的处理,这一切都是为了简化模型,提高实时性。在当时计算机软硬件、软件相对落后的情况下,这个算法也达到非常好的实时效果。

王等人^[4]对雪花粒子的形状进行了改进,对雪花粒子的寿命进行精确的控制,从一定程度上提高了模拟的真实度和渲染速度。

陈等人^[5]提出给雪花粒子增加碰撞检测。

有的研究者则侧重于积雪模拟,如:

Paul^[6]提出计算积雪的方法,他把积雪的计算分成两步:第一步根据雪花的飘落,计算每一个小面上积雪的厚度;第二步对降落在地面上的雪花粒子进行稳定性判断,不稳定的雪花,继续控制它移动,直到稳定。这个算法产生的积雪效果非常细致,积雪非常有层次。但是这个算法渲染速度非常慢,不适合实时性要求高的降雪场景模拟。

Haglund等人^[7]也提出实时计算积雪的方法。这个方法的思想是把整个物体表面分成2维的网格,落在网格内部雪粒子的数量代表这个网格中积雪的厚度,然后,根据这个厚度,在每一个网格上再覆盖一层面,这层面就是后来的积雪。他这个方法非常灵活,可以根据要求的显示精度,调整网格密度,实时性比较好。这个技术的难点就是对网格的划分必须根据物体的形态分布,因此这个工作必须在建模型的时候完成,而且网格划分必须非常随机,太规则了,积雪就不够真实。

Moeslund^[8]不但提出了如何根据物理条件真实的建立雪粒子模型,而且提出了一种计算积雪的方法。首先,在物体表面设计粗网格,然后随着降雪的过程,计算落入每一个网格内部粒子的数量,然后对网格进行动态调整,使得粒子数量比较多的地方的网格相对比较密集,形成细网格,粒子数量少的地

方,网格之间相互合并,形成粗网格。这个方法的模拟效果也非常得好,可是他要动态调整,影响系统的实现速度。

本文对降雪和积雪都进行了研究,降雪方面,虽然也采用传统的粒子模型,可是对粒子进行了进一步的优化,考虑更多的因素,提高了降雪模拟的真实度;在积雪方面,引入噪声纹理技术,解决了已有积雪模拟方法不能用于大场景模拟的问题。

2 降雪模拟

2.1 算法思想

本文在降雪模拟方面仍采用粒子方法。传统方法中,雪粒子为不带纹理的白色椭圆面,在运动过程中,粒子的颜色始终是白色,而且代表粒子的椭圆面是始终垂直用户视线的 Billboard。这种方法中,粒子颜色不会随着天气、光照的变化而变化。在模拟中,大雪时,随着椭圆面增大,模拟效果就不够真实。因此本文采用的降雪模拟方法,在以下几个方面进行了改进:

(1) 粒子运动过程考虑多种速度影响和受力。粒子受力和速度描述如图1。

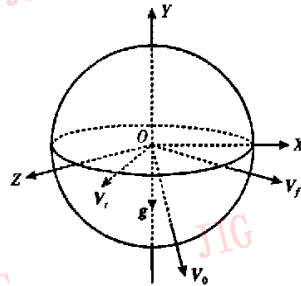


图1 雪粒子受力和速度图

Fig. 1 Particle's stresses and velocities

图1中, V_0 为初始速度矢量(常值), V_f 为风速矢量, V_r 为随机速度矢量, g 为重力加速度矢量,由于雪花的形状特殊,因此,受到空气阻力而在空中无规则飞舞,我们引入随机速度矢量 V_r ,在简化阻力的同时描述雪花的这一运动,取 V_r 为模取定值而方向任意的矢量。

$$V_i = V_0 + V_f + V_r + \int g dt \quad (1)$$

式中, V_r 为*i*帧时的随机速度矢量。

$$S_i = S_{i-1} + \int V_{i-1} dt \quad (2)$$

式中, S_i 表示第 i 帧的位置。

在实际渲染中,用两帧之间的时间差 Δt 来代替积分,因此,式(1),式(2)就转化成:

$$V_i = V_0 + V_i + V_i' + g\Delta t \quad (3)$$

$$S_i = S_{i-1} + V_{i-1}\Delta t \quad (4)$$

(2) 考虑到天气、光照和环境等对雪花粒子颜色的影响,我们用带有颜色的矩形,贴透明雪花纹理来模拟雪花粒子,采用融合技术,最终粒子的颜色就是矩形颜色和雪花纹理的融合,因此,只要调节矩形的颜色,就可以控制雪花粒子的颜色,使雪花适应周围环境变化。融合函数选择:

```
glBlendFunc(GL_SRC_ALPHA,
            GL_ONE_MINUS_SRC_ALPHA)
```

同时,为了提高真实度的同时维持渲染速度,我们采用的透明纹理也是根据粒子大小来选择不同精细度的纹理,如图 2 所示就是我们采用的雪花纹理。当粒子距视点足够远的时候(约 3.5m),就可以忽略其纹理,采用传统的纯白色,这样可以提高渲染速度。



图 2 不同的雪花透明纹理(纹理大小依次增大)

Fig. 2 Snowflake's textures

(3) 把旋转引入雪花粒子运动过程中。同样,当粒子距视点足够远的时候(约 3.5m),也可以忽略其旋转运动。

(4) 粒子密度调节。由于能见度的影响,降雪过程中远处的雪花是不易看清楚,因此,采用与距离相关的概率函数代替传统方法中的等概率函数,这样,模拟相同效果就只需要少量的粒子。如图 3 所示为暴雪模拟中采用的概率密度函数。

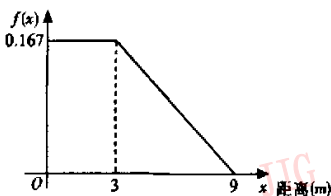


图 3 概率密度

Fig. 3 Probability density

综上所述,粒子的属性实现如下:

```
struct particle
{
    float size;           //粒子的大小
    Color4f color;       //包含 Alpha 的粒子颜色
    Vector3f pos;        //粒子实时位置
    Vector3f v0;         //初始矢量速度 V_0
    Vector3f vf;         //风速 V_f
    float rot;           //旋转角速度
    float angle;         //当前角度
    bool start_assign;   //标记粒子是否已指定属性
};
```

2.2 算法描述

降雪模拟算法如下(以下坐标都是针对 OpenGL 环境):

(1) 根据指定粒子系统的数量给每一个粒子分配存储空间,并设定每一个粒子的 start_assign 属性为 FALSE。用变量 t_0 记录下系统当前时间(毫秒级)。

(2) 以当前视点为中心,设定一个长方体:Left, Right, Top, Bottom, Near, Far,所有的粒子的运动将限定在这个长方体内,其中 Bottom 可以指定为水平地面,Top 为向上视点刚好看不到的位置。

(3) 对每个粒子的 start_assign 属性判断,如果为 TRUE,用 t_i 变量记录下当前系统时间,则 $\Delta t = t_i - t_0$,然后根据式(3)、(4)更新每个粒子当前帧的 pos 和 angle 等属性。如果为 FALSE,用随机函数初始化粒子 size、pos、rot 和 angle 属性,pos 的 $X \in (\text{Left}, \text{Right})$, $Y = \text{Top}$, $Z \in (\text{Left}, \text{Right})$, color、 V_0 和 V_f 属性由用户指定。

(4) 根据每个粒子的 pos 属性判断它是否在长方体内部,如果不在,则将其 start_assign = FALSE。

(5) 将所有 start_assign = TRUE 的粒子,根据它们的 size 属性选择对应的透明纹理,然后根据其 pos 和 angle 属性,采用融合技术,进行绘制处理。

(6) t_0 记录当前系统时间,交换缓存,然后转第 2 步进行下一帧的绘制。

3 积雪模拟

3.1 算法思想

在引言部分提到关于积雪模拟的研究中,Paul 方法用单个雪粒子堆积来模拟积雪,因此,模拟效果最真实,但实时性最差;Haglund 模拟效果最差,实时性最高;Moeslund 方法模拟效果和实时性介于

Paul 和 Haglund 之间。可是无论是 Moeslund 还是 Haglund 算法,他们的实时性都不能满足大场景模拟的要求,因为他们都是将降雪和积雪紧密联系在一起,都要根据地面上降落雪花粒子的数量来确定雪层的厚度,假如虚拟场景是 $1\text{km} \times 1\text{km}$ 的,那么同时空中飘舞的雪花数量就代表了实时要计算粒子的数量,而这个数量已经无法用计算机进行模拟,因此,这 3 种方法都不适合大场景积雪模拟。

现在涉及到大场景积雪模拟的系统,都会提前设计两个场景,一个是无积雪场景,一个是被积雪覆盖的场景。当降雪时,就直接切换到带积雪的场景,他们之间没有动态的过渡,也就没有积雪的过程,如 Flight Simulator 2004^[9](微软模拟飞行)。

我们考虑到在大场景的积雪模拟中,对于积雪的细节模拟要求不严格,尤其是在飞行模拟中,视点距雪层的最短距离也要 20 多米,因此,在大场景积雪模拟中,只要满足视觉上地面渐白的效果就足够。根据这个特点,本文将积雪和降雪分开,引入噪声技术,实现了地面渐白的积雪模拟。

3.2 算法描述

我们首先将整个地形划分为若干子面,本实验中子面平均为 800m^2 ,然后在对每一个子面进行纹理映射^[10]的时候,引入噪声,对其纹理进行干扰。具体实现算法描述如下:

(1) 从物体的表面中读取一个子面。假如子面顶点为 $A \rightarrow B \rightarrow C$,根据图形学规定,每个子面顶点的定义顺序采用逆时针,那么子面的形状如图 4 所示。

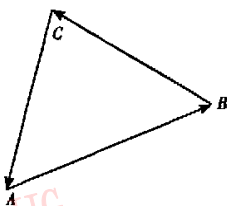


图 4 子面
Fig. 4 Surface

(2) 比较子面的法向量和向上向量夹角 θ 。子面法向量的方向向量为 $\alpha = (\vec{AB} \cdot \vec{BC}) / (|\vec{AB}| \cdot |\vec{BC}|)$ 。设 β 为向上的单位向量, $\beta = (0, 1, 0)$, 因此 $\cos(\theta) = \alpha \cdot \beta$ 。

(3) 根据 θ 大小决定是否对该面进行噪声干扰。如果 $\theta > 90^\circ$, 表明该面的正方向非朝上,不会有积雪效果,则转第 6 步,否则转第 4 步。

(4) 确定噪声干扰的强度。场景中一大子面也许只对应一小块(粗)纹理,而一小子面也许对应一大块(精)纹理,如果对它们采用同样的噪声干扰,则积雪就不均匀,严重影响真实度,因此,我们增加噪声干扰强度 φ 。 $\varphi = S_{\text{纹理面积}} / S_{\text{场景面积}}$ 。设 A, B, C 3 点的 UV 坐标^[10]分别为 a, b, c , 那么 $S_{\text{纹理面积}} = (\vec{ab} \cdot \vec{bc}) / 2, S_{\text{场景面积}} = (\vec{AB} \cdot \vec{BC}) / 2$ 。

(5) 根据噪声干扰强度 φ 对纹理进行干扰。本文提出的积雪模拟方法是将积雪和降雪分开的,因此,设定 $\gamma, \gamma \times \varphi$ 来控制干扰强度,用户根据降雪的大小,实时调整 γ , 就可以使得降雪和积雪同步。

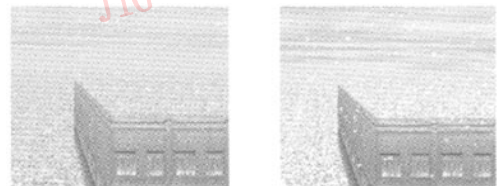
(6) 对该子面进行纹理映射。

对每一个子面持续进行噪声干扰,就形成了地面雪花逐步增多的过程。

4 实验与分析

4.1 几组实验

(1) 由于比传统方法增加了粒子的动态颜色和动态纹理,改进方法模拟的降雪和积雪更适应天气、光照等环境变化,如图 5 所示。



(a) 昏暗环境下降雪 (b) 明亮环境下降雪

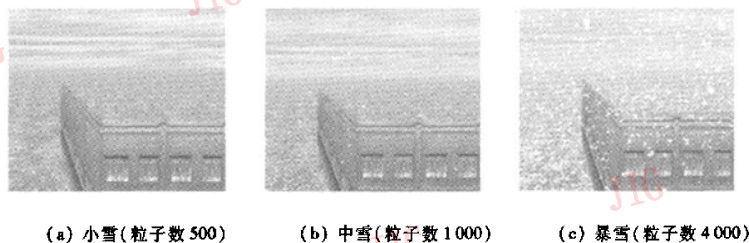
图 5 环境适应性

Fig. 5 Environmental adaptation

(2) 通过增加粒子的数量和改变粒子的数量来模拟不同大小的降雪,而中雪和暴雪中粒子相对较大,如果再按照传统方法,粒子平行于屏幕运动(Billboard 模型),那么真实度就非常差。而改进方法中引入粒子的旋转,就可以用比传统方法更少的粒子,模拟出更真实的中雪和暴雪(如图 6 所示)。

(3) 传统方法对风速的设定比较死板,所有粒子的风速相同且不变化;改进方法中,每个粒子都有自己独立的风速,而且都实时变化。实验中为了突出这点,可以让雪花在实时风的作用下形成龙卷风(如图 7 所示)。

(4) 模拟积雪过程如图 8 所示。



(a) 小雪(粒子数 500) (b) 中雪(粒子数 1000) (c) 暴雪(粒子数 4000)

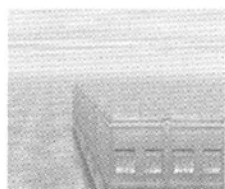
图 6 不同级别的降雪效果

Fig. 6 Different levels of snowing

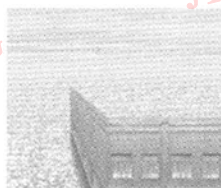


图 7 风的影响

Fig. 7 Wind-blown effect



(a) 无降雪



(b) 中雪 30min 后



(c) 中雪最终结果

图 8 积雪过程

Fig. 8 Process of accumulating

4.2 实验结果分析

在降雪方面,新方法比传统方法考虑到了更多因素,因此模拟更加真实,尤其是对中到大雪的模拟,缺点就是前期工作比较复杂,尤其是设计雪花的透明纹理。

在积雪方面,新方法成功地在大场景中模拟出了积雪过程,在保持系统实时性的基础上提高了模拟效果,该方法有以下优点:第一,实时性好,该方法

在原纹理映射的基础上增加纹理的噪声干扰,因此,对系统渲染速度影响较小,从表 1 可以看出,在降雪的基础上增加积雪模拟,系统速度变化很小;第二,操作灵活,通过调整噪声干扰方式就可以实现各种不同的效果,而且控制地面渐白的速度;第三,稳定,因为只是对纹理进行噪声处理,所以它不会影响物体特性,如地面的起伏不会受影响。

表 1 系统实时性比较

(CPU:1.6G,内存:256MB,子面平均为 800m²)

Tab. 1 Real-time comparison of different scales

场景规模	粒子数量	不同条件下模拟的帧速率对比 (fps)		
		无降雪 无积雪	有降雪 无积雪	有降雪 有积雪
2km × 2km	500(小雪)	89	65	60
	1000(中雪)	89	46	44
	3000(暴雪)	89	31	29
10km × 10km	500(小雪)	72	34	32
	1000(中雪)	72	28	26
	3000(暴雪)	72	22	19
100km × 100km	500(小雪)	58	32	28
	1000(中雪)	58	23	21
	3000(暴雪)	58	12	9

5 结论

本文对降雪和积雪模拟方法进行研究,在降雪方面,考虑更多的因素,提高了降雪模拟的真实性;在积雪方面,引入噪声技术解决了大场景无法进行积雪模拟的难题。实验表明,本文提出的降雪积雪模拟方法简单,模拟效果真实,而且实时性良好,适合大场景尤其是飞行模拟系统中。

参考文献 (References)

- 1 Wu Jia-zhu, Dang Gang, Liu Hua-feng, *et al.* Scene simulation technology and application [M]. Xi'an: Xidian University Press, 2001: 70~71. [吴家铸, 党岗, 刘华峰等. 视景仿真技术及其应用[M]. 西安: 西安电子科技大学出版社, 2001: 70~71.]
- 2 Reeves W T. Particle system—A technique for modeling a class of Fuzzy object[J]. Computer Graphics, 1983, 17(3): 359~376.
- 3 Xie Jian-bin, Hao Jian-bin. The real time simulating algorithm of rain and snow descending based on particle systems[J]. Journal of Image and Graphics, 1999, 4(9): 734~737. [谢剑斌, 郝建斌. 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): 734~737.]
- 4 Wang Run-jie, Tian Jing-quan, Ni Zheng-guo. Realtime simulation of rain and snow based on particle system[J]. Journal of System Simulation, 2003, 15(4): 495~496. [王润杰, 田景全, 倪政国. 基于粒子系统的实时雨雪模拟[J]. 系统仿真学报, 2003, 15(4): 495~496.]
- 5 Chen Lei, Chen He-xin, Wang Run-jie, *et al.* Real time snow simulation in flight simulator [J]. Journal of Jilin University (Engineering and Technology Edition), 2005, 35(2): 164~166. [陈蕾, 陈贺新, 王润杰等. 在飞行模拟器中实现雪的实时模拟[J]. 吉林大学学报(工学版), 2005, 35(2): 164~166.]
- 6 Paul Fearing. Computer modeling of fallen snow [A]. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques[C], New Orleans, LA, USA, 2000: 37~46.
- 7 Haglund, Mattias Andersson, Anders Hast. Snow accumulation in real-time[A]. In: SIGRAD2002, The Annual SIGRAD Conference, Special Theme—Special Effects and Rendering [C], Norrköping, Sweden, 2002: 28~29.
- 8 Moeslund T B, Madsen C B, Aagaard M. *et al.* Modeling falling and accumulating snow [A]. In: Proceeding of Second International Conference on Vision, Video and Graphics [C], Heriot-Watt University, Edinburgh, UK, 2005: 61~68.
- 9 Microsoft, Flight Simulator 2004: A Century of Flight[CP]. <http://www.microsoft.com/games/pc/flightsimulator.aspx>, 2003-07-10.
- 10 Xue Jun-tao, He Huai-qing, Zhang Yu-xiang, *et al.* Research on approach of texture mapping implementation [J]. Computer Engineering, 2005, 31(7): 211~214. [薛军涛, 贺怀清, 张宇翔等. 典型纹理映射实现方法研究[J]. 计算机工程, 2005, 31(7): 211~214.]