

大规模真实感雪景实时渲染

王 纲¹, 季振洲¹, 张泽旭²

(1. 哈尔滨工业大学计算机科学与技术学院, 黑龙江哈尔滨 150001; 2. 哈尔滨工业大学深空探测基础研究中心, 黑龙江哈尔滨 150008)

摘 要: 本文提出一种大规模真实感雪景实时渲染方法. 首先建立雪花模型及其在降雪过程中的运动模型; 其次将风的运动速度分为平均风速与随机风速, 简化了风场计算, 通过 Perlin 噪声生成了随机风, 并将计算结果存储在 3D 纹理中, 减轻粒子系统的计算负荷; 第三, 建立积雪与融化模型, 实现雪的积累和融化的模拟; 最后建立基于 GPU 的粒子系统, 提高粒子系统的渲染效率. 仿真结果表明本文的方法适合大规模雪景实时渲染, 并已成功应用于某直升机模拟器中.

关键词: 雪景模拟; 风场模型; 粒子系统; 实时渲染

中图分类号: TP391.9 **文献标识码:** A **文章编号:** 0372-2112 (2012)09-1746-06

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2012.09.008

Large Scale Realistic Snow Scene Real-Time Rendering

WANG Gang¹, JI Zhen-zhou¹, ZHANG Ze-xu²

(1. School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang 150001, China;

2. Deep Space Exploration Research Center, Harbin Institute of Technology, Harbin, Heilongjiang 150008, China)

Abstract: A real-time rendering method of realistic snow scene with large scale is presented. Firstly, a snowflake physical model and a snowflake moving model in the snowfall are taken into account. Secondly, the wind velocity is separated into the average velocity and the random velocity, which simplifies the computation of wind field. The random velocity is yielded by Perlin noise and the results are stored in a 3D texture, which effectively alleviates the computing cost of the particle system. Thirdly, the snow accumulation model and snow melt model are presented and simulated. Finally, a particle system based on modern GPU technologies is implemented which improves the performance of rendering. The simulating results show that the method is suitable for large scale snow sceneries rendering. This method is applied to a helicopter simulator successfully.

Key words: snow scene simulation; wind field model; particle system; real-time rendering

1 引言

自然景物随着四季与地理位置的不同而各具特色, 诸如雨雾^[1]、大气介质^[2]等对自然场景的渲染、分类^[3]等也有重要影响. 在高纬度或高海拔地区, 雪景是最常见的自然景象. 由于雪的累积与融化受风、温度、雪的厚度等诸多外界条件的影响, 模拟雪景的变化成为具有挑战性的研究课题.

雪景的模拟可分为降雪、累积和融化三方面. 降雪常采用基于粒子系统的方法^[4]. Sims 等提出了粒子系统的并行算法^[5]; Wei 等人采用 LBM (Lattice Boltzmann Model) 建立的风场模型也用于降雪的实现^[6]; 大规模粒子系统还可以通过 GPU 实现^[7]; 然而采用诸如 IFFT (In-

verse Fast Fourier Transform) 和基于纹理技术等方法在模拟降雨、降雪方面也有一定优势^[8~10].

雪的累积和融化过程更复杂. 依据场景的几何特性, 文献[11~13]基于过程或物理的方法建立了降雪和积雪模型, 文献[14~16]还综合考虑了风的影响. 现有的积雪方法大致可分为三类: 基于粒子系统、基于表面位移技术和基于表面细分技术. 基于粒子系统的方法主要是计算雪花被风吹起后的运动及其在地面的堆积, 较早的这类模型通过设置粒子的分布采用元球实现^[17]; Fearing 等通过跟踪随机建立的雪花粒子的向上发射, 递归的检测雪花降落到地面时的稳定性^[11]; 此方法的一些改进主要是对雪花在风作用下的运动方式的改进, 这类改进通过求解风的动力学特性进而得到了复杂的雪

景效果,其方法则是通过求解 Navier-Stokes 方程或 Boltzmann 方程^[14~16].基于表面位移的方法主要是通过计算局部区域的可达性和遮掩性(occlusion)求解积雪的厚度^[18];一些加速算法实现了大地形场景的积雪模拟^[13,19,20];表面位移法的另一种实现称为高度跨越映射(height span maps)^[21],用于实现颗粒材质如沙等的模拟,已被证明是一种高效的积雪模型^[12,22].表面细分技术则是通过将积雪的表面进行细分,从而使得积雪的表面更加平滑^[23].上述文献主要是对积雪进行了研究,文献[24,25]则基于热传导的物理模型同时建立了积雪和融化的模型.

此外,文献[23,26]提出了一种建立雪景的综合模型,其目标是实现电影级效果,但并不适于实时渲染.本文重点是提出一种实时渲染大规模雪景的方法,以视觉效果的真实感和渲染效率为主要研究目标.文献[27,28]的技术还可用于并行渲染.

2 降雪的模式

降雪模拟可以细分为雪花模型的建立以及风场模型的建立.本节对两种模型分别进行研究,雪花的模型基于文献[23]的研究成果.

2.1 雪花的模型

2.1.1 雪花的大小与密度

雪花主要由空气或云中的水分经冷凝结成的冰晶形成,而冰晶如何结成最终的雪花取决于所处环境的温度以及水分的饱和度.尽管雪花的形状千差万别,但冰晶大致可以归纳为几大类型,如星状、盘状等,此外雪花的大小也遵循一定的规律,其直径与其所处的环境温度构成如下的函数关系:

$$D = \begin{cases} 0.015 \times |T|^{-0.35}, & T \leq -0.061 \\ 0.04, & T > -0.061 \end{cases} \quad (1)$$

式中, D 为雪花的平均直径(m), T 是环境温度(°C).这一函数关系是根据长期观察结果的统计规律得出的,尽管存在着高达±50%的偏差,但是足可满足本文的应用,而且这一偏差在生成雪花粒子时,可以通过一个随机量与平均直径相加,来模拟雪花尺寸的变化.

雪花的密度与雪花的直径成反比

$$\rho = \frac{C}{D} \quad (2)$$

式中, C 为与雪花干湿程度相关的常量,对于干雪来说, $C = 0.170\text{kg/m}^2$,而对于湿雪来说, $C = 0.724\text{kg/m}^2$.雪花的密度对雪花运动特性的求解非常重要,设雪花的密度为 ρ ,则

$$m = \rho V = \rho \frac{4}{3} \pi \left(\frac{D}{2} \right)^3 = \frac{1}{6} \rho \pi D^3 \quad (3)$$

式中, m 为雪花的质量, V 为雪花的体积.

2.1.2 雪花的运动模型

雪花在从开始形成直至其降落到地面形成降雪的过程中,一直受到四种力的作用:浮力、重力、升力以及阻力.浮力与重力方向相反,两者皆为常量,升力与阻力则随着风向的变化而变化.

浮力与雪花所排开的相同体积大气的重量相等,重力则是雪花所受到的地球引力的大小,由于雪花所受到的浮力与其重力相比非常小,对于雪花的运动影响不大,可以忽略不计,雪花受到的力可表示为

$$\mathbf{F} = \mathbf{F}_b + \mathbf{G} + \mathbf{F}_l + \mathbf{F}_d \approx m\mathbf{g} + \mathbf{F}_l + \mathbf{F}_d \quad (4)$$

式中, \mathbf{F}_b 为浮力, \mathbf{G} 为重力, \mathbf{F}_l 为升力, \mathbf{F}_d 为阻力, m 为雪花的质量, \mathbf{g} 为重力加速度.

升力 \mathbf{F}_l 是使雪花在下落过程中沿着不规则环形路径运动并不断绕质心旋转的主要原因,这主要是由于雪花不规则的空气动力学外形及其在下落过程中形成的紊流引起的.在风速比较大的情况下,升力的作用可以忽略不计,而在风速较小或无风的情况下,升力的作用则必须予以考虑.由于本文采用一个简单的透明纹理来模拟雪花的外观,因而雪花绕质心的旋转没有模拟.

阻力 \mathbf{F}_d 主要是雪花在运动过程中受到的空气作用力,同时也是雪花能够沿着风的运动方向运动的重要原因.阻力的大小可以通过下式求得

$$\mathbf{F}_d = \frac{V_f^2 m \mathbf{g}}{V_{pmax}^2} \quad (5)$$

式中, m 、 \mathbf{g} 分别为雪花的质量和重力加速度, \mathbf{V}_f 为由雪花引起的空气运动速度,其方向与 \mathbf{F}_d 的方向相同, V_{pmax} 为考虑风阻时的最大垂直速度,对于干雪, $V_{pmax} \in [0.5, 1.5]\text{m/s}$,而对于湿雪, $V_{pmax} \in [1.0, 2.0]\text{m/s}$.当生成雪花粒子时,可以根据温度在上述范围内对 V_{pmax} 进行随机采样.式(5)中 m 、 \mathbf{g} 和 V_{pmax} 对于给定的雪花均为常量, \mathbf{V}_f 则是由两种因素引起的变量:风以及雪花与空气的摩擦,正是后者导致雪花随着风的运动

$$\mathbf{V}_f = \mathbf{V}_w - \mathbf{V}_s \quad (6)$$

式中, \mathbf{V}_w 为风速, \mathbf{V}_s 为雪花的运动速度.由上式可见,风对雪花的运动有比较明显的影响.

2.2 风场模型

风场模型的建模方法通常可以分为两大类,一类是基于连续流的模型,典型的如 Navier-Stokes 方程组^[23].另一类方法是 LBM 法^[15],这类方法通过对粒子群运动规律的研究替代对个体粒子的研究,在连续介质中,LBM 法的物理学意义比 Navier-Stokes 更加直观.这两类方法均为基于物理的方法,可获得真实的风场

模型.但这两类模型都需要较高的计算开销,在目前的技术条件下难以有效地实现实时计算.

由于风的运动所特有的复杂性,因此风场中某一点处的风速并不是一成不变的,对于风场中任一点在某个给定时刻的风速与该点在下一时刻是不同的.因而可以假定风速 V_w 由平均风速和随机风速两部分组成,即

$$V_w = \bar{V}_w + \tilde{V}_w \quad (7)$$

式中, \bar{V}_w 为平均风速, \tilde{V}_w 随机风速.上式中真正感兴趣的是随机风速 \tilde{V}_w ,对于给定的平均风速 \bar{V}_w ,可以通过下式得到 \tilde{V}_w

$$\tilde{V}_w = \mu \bar{V}_w \sin \alpha \quad (8)$$

式中, μ 为一个与风速大小相关的随机比例系数,一般可取为不大于平均速度的模 $|\bar{V}_w|$ 的 3% ~ 5%; α 为随机初始相位角,且 $\alpha \in [0, \pi]$.

尽管随机风速是由紊流产生的,但仍然在时间和空间上是连续的,因而初始相位角 α 也必须是连续的,这与 Perlin 噪声的连续性和随机性是一致的^[29],因而 Perlin 噪声可作为产生初始相位角的随机函数.由于风速一般按水平方向给出,其垂直分量依然可以通过 Perlin 噪声产生.将预先计算出的 $\sin \alpha$ 值以及垂直风速一并存储进一张 3D 纹理中,作为三维风场中随机风的模型,在后续的计算中使用.

3 雪的累积与融化

雪的累积主要是由其所处的大气条件和地形的高度、斜率等决定的,这些因素共同作用,决定了积雪的厚度及其外观.雪的融化主要是由可利用的热能决定的.如果有足够的热能,融化过程将会持续,否则融化后的雪水会再次结成冰,阻止雪的进一步融化,雪的融化过程比其累积要漫长得多.无论是雪的累积还是融化过程,基于物理的方法无疑都不利于实时应用.因此本文采用基于观测以及经验数据的方法.

在雪的累积计算中,积雪的厚度 D_s (mm) 可通过如下公式计算

$$D_s = C_a \times t \quad (9)$$

式中, C_a 为降雪的速度 (mm/day), t 为降雪过程的时间 (day).

在雪的融化计算中,雪的融化速度 M (mm/day) 一般可计算为

$$M = C_m \times (T_{\text{air}} - T_m) \times t \quad (10)$$

式中, C_m 为雪的融化率系数 (mm/°C day), T_{air} 为大气温度 (°C), T_m 为使雪融化的温度门限值 (°C),一般为 0°C, t 为融雪过程的时间 (day). C_m 与植被遮挡、雪的反照率、太阳的辐射等有关

$$C_m = k_m k_v R_l (1 - A) \quad (11)$$

式中, k_m 为比例常量, k_v 为植被遮挡引起的太阳辐射传递系数, R_l 为太阳的辐照度因子, A 为雪的反照率.

一般情况下

$$k_v = e^{-4C_v} \quad (12)$$

式中, C_v 为植物遮挡的密度.

$$A = 0.4 \times (1 + e^{-k_e t}) \quad (13)$$

式中, $k_e \approx 0.2/\text{day}$ 为时间常量.

4 基于 GPU 的粒子系统

典型的粒子系统的每一个粒子都包括众多属性:位置、速度、加速度、尺寸、颜色(包括透明度)、形状以及生命等,这些属性在粒子的整个生存周期内的每一次渲染,都要保存其当前的状态.在进行降雪模拟的过程中,这些属性中的大部分在粒子的整个生命周期中或保持不变(尺寸、颜色、形状),或可以通过参数等方法进行设置(速度、加速度),只有位置和生命是变化量,因而基于粒子系统的降雪模拟非常适合通过 GPU 来实现. DirectX 10 级别及其以上的现代商用 GPU 均支持多重渲染目标、渲染至顶点缓存、纹理数组以及浮点纹理等技术,这是粒子系统能够通过 GPU 实现的前提.雪粒子的位置和生命可以存储在顶点缓存对象或浮点纹理数组中.选用二维纹理时,其宽和高 S_i 为

$$S_i = \text{ceil}(\sqrt{N_p}) \quad (14)$$

式中, N_p 为粒子数, ceil 为向上取整函数.一张 1024×1024 的二维纹理可存储上百万个粒子的属性.本文方法的渲染效果如图 1 所示.

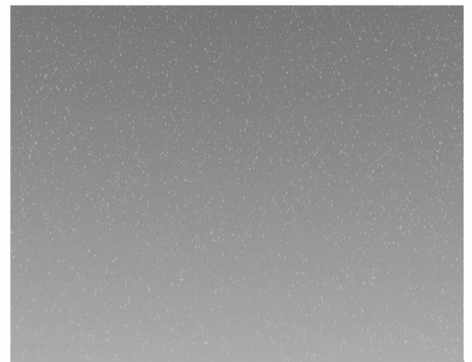


图1 GPU粒子系统降雪效果

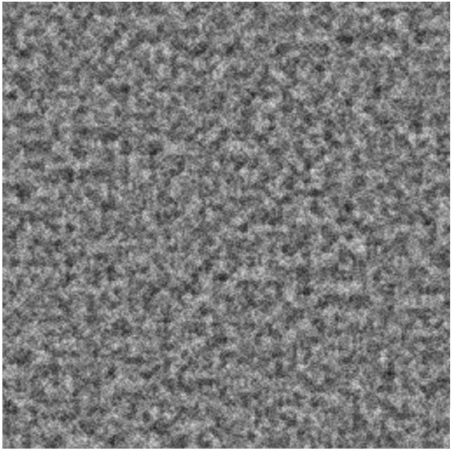
5 渲染实现

雪的累积与融化的物理过程极其复杂,因而通过对其视觉效果模拟既可以获得较好的真实感,又可以保证渲染的实时性.本文基于上述方法,采用如图 2 所示的两张 2D 纹理,其中图 2(a)为雪纹理,图 2(b)为

蒙板纹理,实现了雪景的实时渲染.



(a) 雪的纹理



(b) 蒙板纹理

图2 GPU粒子系统降雪效果

雪纹理为单通道纹理,蒙板纹理为两通道纹理,其红色分量存储雪融化蒙板,绿色分量存储雪累积蒙板,分别用于对雪的融化和累积效果进行模拟.对于一个给定的雪景区域 S ,设雪在积累过程中覆盖的比例为 A_S ,融化的区域比例为 M_S ,蒙板纹理为 T_M , A_M 为累积蒙板作用结果, M_M 为融化蒙板作用的结果,则

$$\begin{cases} A_M = T_M \cdot g - A_S \\ M_M = T_M \cdot r - M_S \end{cases} \quad (15)$$

式中 A_S 和 M_S 可以通过式(9)、(10)计算后获得.

该结果通过渲染结果的 Alpha 通道起作用,设雪纹理为 T_S ,给定的颜色为 C_{in} ,渲染结果为 C_{out} ,则渲染结果的颜色分量

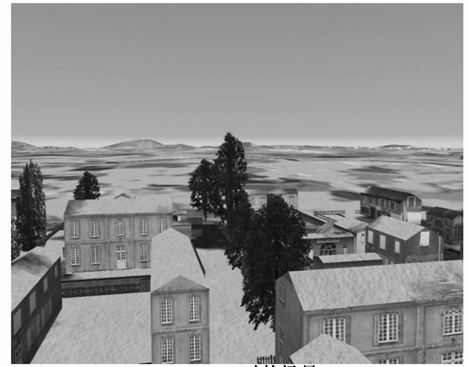
$$C_{out} \cdot rgb = (C_{in} \cdot rgb \times A_M \times T_S \cdot r) \times C_{in} \cdot a \quad (16)$$

式(16)给出了雪景绘制的红绿蓝三个分量计算方法,由上式可以看出,随着 C_{in} 的变化,最终的输出颜色也会变化,因而该方法同样可以用于诸如沙尘的累积与消散.

下面给出 Alpha 分量的计算方法,当雪累积时

$$a = 1.0 - A_M \quad (17)$$

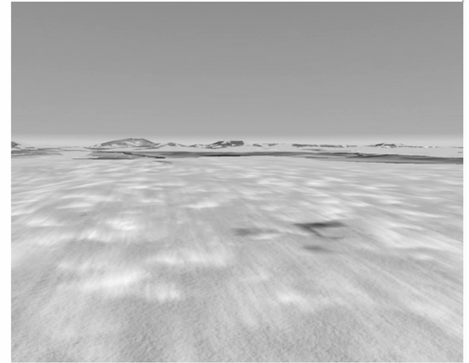
当雪融化时,融化的雪水会使地面变暗,称之为雨暗,设雨暗系数为 R_d ,则考虑融化及雨暗后



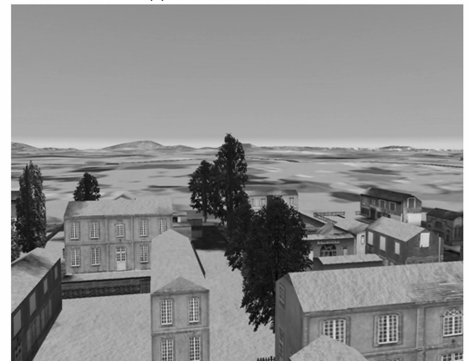
(a) $A_S=50\%$ 时的场景



(b) $A_S=50\%$, $M_S=50\%$ 时的场景



(c) 风的作用效果



(d) $A_S=50\%$ 时的沙尘场景

图3 不同条件下雪景仿真渲染效果图

$$a = a \times (1.0 - R_d \times M_M) \quad (18)$$

渲染结果的 Alpha 分量为

$$C_{out} \cdot a = 1.0 - a \quad (19)$$

当风经过地形表面时,地面上的积雪会随着风的运动而运动,为了模拟这种效果,本文采用动画纹理,较好的模拟了雪在地面的运动。

6 结果与分析

本文对提出的方法进行了仿真实验.实验中采用的是普通 PC 计算机,其具体配置为:CPU 为 Pentium D,主频 3.0GHz,内存 1GB,显卡为 NVIDIA GTX260+,分辨率为 1280 × 1024.当雪花粒子数为 10000 时,采用本文的方法平均帧率可以达到 500fps 以上.分析表明,本文给出的雪景渲染方法能够取得较高的渲染帧率主要得益于本文建立的风场模型以及基于 GPU 的粒子系统,从而大大降低了 CPU 的工作负载,保证其有足够的资源完成场景管理、任务管理等功能。

在某直升机模拟器中,本文的方法成功地实现了大规模雪景的实时渲染.在包括全球数据库、多种三维实体及多种特殊效果等的复杂场景下,在上述实验用 PC 上,雪景的渲染帧率能够在 1400 × 1050 的较高分辨率下稳定保持为 60fps,部分场景的渲染效果如图 3 所示.其中,图 3(a)为 $A_s = 50\%$ 时的场景;图 3(b)为 $A_s = 50\%$, $M_s = 50\%$ 时的场景,可见雪边缘处雨暗效果;图 3(c)考虑风作用时的效果;图 3(d)为 $A_s = 50\%$ 时的沙尘场景.上述数据库中,对于雪景可以覆盖的区域进行了标记,如图 3 中雪或沙尘并没有出现在墙壁上。

7 结论与将来的工作

本文对大规模真实感雪景的实时渲染进行了研究,所做的主要工作包括:(1)建立了基于物理特性的雪花模型及其运动模型,提高了降雪模拟的真实感;(2)提出了一种简化的风场模型,将风速划分为平均风速和随机风速,并将采用 Perlin 噪声预先计算的随机风速结果存储在 3D 纹理中;建立了基于 GPU 的粒子系统,并对上述 3D 纹理加以应用,大大提高了降雪过程的渲染速度;(3)建立了雪的累积与融化模型,通过 2D 纹理实现了雪的累积与融化,并通过动画纹理实现了地面积雪跟随风的运动,取得了比较好的视觉效果.实验结果表明,本文提出的研究方法能够对复杂场景下的大规模雪景进行实时渲染,并获得了较好的真实感。

本文在进行雪的累积与融化的模拟时,仅仅采用 2D 纹理实现了其视觉效果,没有对因积雪或融化时引起的地面高度变化进行模拟;本文建立的粒子系统没有考虑不同雪花的外观差异,也没有对雪花下落过程中的旋转进行仿真,这些都是在后续的研究工作中需

要完善的。

参考文献

- [1] 桑梓勤,丁明跃,张天序.雨雾天气下的户外场景成像[J].电子学报,2000,28(3):131-133.
SANG Zi-qin, DING Ming-yue, ZHANG Tian-xu. Imaging for outdoor scene in rain and fog[J]. Acta Electronica Sinica, 2000, 28(3): 131-133. (in Chinese)
- [2] 吴春明,钱徽,朱森良.一个绘制大气介质效果的软件框架[J].电子学报,2004,32(5):735-739.
WU Chun-ming, QIAN Hui, ZHU Miao-liang. A software framework for rendering atmospheric effect[J]. Acta Electronica Sinica, 2004, 32(5): 735-739. (in Chinese)
- [3] XIE Wen-jie, XU De, TANG Ying-jun, LIU Shuo-yan, FENG Song-he. Mutual information based codebooks construction for natural scene categorization[J]. Chinese Journal of Electronics, 2011, 20(3): 419-424.
- [4] Reeves W T. Particle systems-A technique for modeling a class of fuzzy objects[A]. Proceedings of the 10th Annual Conference on Computer Graphics and Interactive Techniques[C]. New York: ACM Press, 1983. 359-375.
- [5] Sims K. Particle animation and rendering using data parallel computation[A]. Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques[C]. New York: ACM Press, 1990. 405-413.
- [6] Wei X, Zhao Y, Fan Z, Li W, Stover S Y, Kaufman A. Blowing in the wind[A]. Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation[C]. New York: ACM Press, 2003. 75-85.
- [7] Drone H. Real-time particle systems on the GPU in dynamic environments[A]. ACM SIGGRAPH 2007 Courses[C]. New York: ACM Press, 2007. 80-96.
- [8] Langer M S, Zhang Q. Rendering falling snow using an inverse Fourier transform[A]. ACM SIGGRAPH 2003 Technical Sketches Program, Computer Graphics Proceedings, Annual Conference Series[C]. New York: ACM Press, 2003. 58-64.
- [9] Langer M S, Zhang L, Klein A W, Bhatia A, Pereira J, Rekhil D. A spectral-particle hybrid method for rendering falling snow[A]. Rendering Techniques 2004 Eurographics Symposium on Rendering[C]. Norrköping, Sweden: Eurographics Association, 2004. 217-226.
- [10] Wang N, Wade B. Rendering falling rain and snow[A]. ACM SIGGRAPH 2004 Sketches[C]. New York: ACM Press, 2004. 14-es.
- [11] Fearing P. Computer modeling of fallen snow[A]. Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques[C]. New York: ACM Press, 2000. 37-46.
- [12] Haglund H, Andersson M, Hast A. Snow accumulation in real-

- time[A]. Proceedings of SIGRAD 2002. Special Theme-Special Effects and Rendering [C]. Norrköping, Sweden: Linköping University Electronic Press, 2002. 11 – 15.
- [13] Foldes D, Beneš B. Occlusion-based snow accumulation simulation[A]. Proceedings of the 4th Workshop on Virtual Reality Interactions and Physical Simulation [C]. Norrköping, Sweden: Eurographics Association, 2007. 35 – 41.
- [14] Feldman B E, O' Brien J F. Modeling the accumulation of wind-driven snow[A]. Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques [C]. New York: ACM Press, 2002. 218 – 218.
- [15] Wang C, Wang Z, Xia T, Peng Q. Real-time snowing simulation[J]. The Visual Computer, 2006, 22(5): 315 – 323.
- [16] Saltvik I, Elster A C, Nagel H R. Parallel methods for real-time visualization of snow[A]. Proceedings of the 8th International Conference on Applied Parallel Computing: State of the Art in Scientific Computing[C]. Heidelberg, Berlin: Springer-Verlag, 2007. 218 – 227.
- [17] Nishita T, Iwasaki H, Dobashi Y, Nakamae E. A modeling and rendering method for snow by using metaballs[J]. Computer Graphics Forum, 1997, 16(3): 357 – 364.
- [18] Premoze S, Thompson W B, Shirley P. Geospecific rendering of Alpine terrain[A]. Proceedings of 10th Eurographics Rendering Workshop [C]. Heidelberg, Berlin: Springer-Verlag, 1999. 107 – 118.
- [19] 陈彦云, 孙汉秋, 郭百宁, 吴恩华. 自然雪景的构造和绘制[J]. 计算机辅助设计与图形学学报, 2003, 15(1): 129 – 129.
Chen Yan-yun, Sun Han-qiu, Guo Bai-ning, Wu En-hua. Modeling and rendering of snowy natural scenery[J]. Journal of Computer-Aided Design & Computer Graphics, 2003, 15(1): 129 – 129. (in Chinese)
- [20] Ohlsson H, Seipel S. Real-time rendering of accumulated snow [A]. Proceedings of SIGRAD 2004. Special Theme-Environmental visualization[C]. Norrköping, Sweden: Linköping University Electronic Press, 2004. 25 – 32.
- [21] Sumner R W, O' Brien J F, Hodgins J K. Animating sand, mud, and snow[J]. Computer Graphics Forum, 1999, 18(1): 17 – 26.
- [22] Festenberg N, Gumhold S. A geometric algorithm for snow distribution in virtual scenes[A]. Proceedings of Eurographics Workshop on Natural Phenomena 2009[C]. Norrköping, Sweden: Eurographics association, 2009. 15 – 25.
- [23] Moeslund T B, Madsen C B, Aagaard M, Lerche D. Modeling falling and accumulating snow[A]. Proceedings of Second International Conference on Vision, Video and Graphics [C]. Norrköping, Sweden: Eurographics Association, 2005. 61 – 68.
- [24] Muraoka K, Chiba N. Visual simulation of snowfall, snow cover and snowmelt[A]. Proceedings of the 7th International Conference on Parallel and Distributed Systems: Workshops 2000 [C]. Washington, DC, USA: IEEE Computer Society, 2000. 187 – 194.
- [25] Maréchal N, Guérin E, Galin E, Mérillou S, Mérillou N. Heat transfer simulation for modeling realistic winter sceneries[J]. Computer Graphics Forum, 2010, 29(2): 449 – 458.
- [26] Aagaard M, Lerche D. Realistic Modelling of Falling and Accumulating Snow[D]. Denmark: Aalborg University, 2004.
- [27] LIU Li, YANG Guang-wen. A highly efficient GPU-CPU hybrid parallel implementation of sparse LU factorization [J]. Chinese Journal of Electronics, 2012, 21(1): 7 – 12.
- [28] LIU Bu-quan, YAO Yi-ping, WANG Huai-min. On the technology of high-performance parallel simulation [J]. Chinese Journal of Electronics, 2012, 21(1): 1 – 6.
- [29] Perlin K. An image synthesizer[A]. Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques [C]. New York: ACM Press, 1985. 287 – 296.

作者简介



王 纲 男, 1967 年 11 月生于山东牟平. 博士研究生, 高工, 主要研究方向为可视化仿真和视觉计算.

E-mail: gmkwang@gmail.com



季振洲 男, 1965 年 12 月生于黑龙江哈尔滨. 博士, 教授, 博导, 主要研究方向为计算机体系结构与高性能并行计算等.

E-mail: jizhenzhou@hit.edu.cn



张泽旭 1971 年 3 月生于吉林省吉林市. 博士, 教授, 博导, 主要研究方向为自主导航与控制 and 仿真模拟器等.

E-mail: zexuzhang@hit.edu.cn