

大规模音乐喷泉的实时模拟

王槐欣 万华根 肖书星 金小刚 彭群生

(浙江大学 CAD&CG 国家重点实验室, 杭州 310027)

摘要 音乐喷泉被广泛应用于城市和景区的美化, 如何对其进行模拟被越来越多的研究人员关注。提出一种大规模音乐喷泉的实时模拟方法, 借助音乐数据建立音乐喷泉的粒子运动模型, 通过充分挖掘图形处理器强大的并行处理能力, 突破了海量粒子系统实时运动计算及实时绘制的瓶颈, 并且针对喷泉水滴的特点, 改进了绘制方法, 成功地实现了对大规模音乐喷泉的实时模拟。

关键词 音乐 喷泉 粒子系统 图形处理器 实时绘制

中图法分类号: TP391.9 **文献标志码:** A **文章编号:** 1006-8961(2010)03-524-06

Real-time Simulation of Large-scale Music Fountain

WANG Huaixin, WAN Huagen, XIAO Shuxing, JIN Xiaogang, PENG Qunsheng

(State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou 310027)

Abstract It's a growing concern to many researchers for real-time simulation of music fountains since they are fascinating decorations in cities and scenic spots which combines modern technology, music and waterscape, etc. In this paper, a new approach for real-time simulation of large scale music fountain is proposed which incorporates fluid dynamic, particle system, music synchronization, and general purpose GPU computation. The fountain particles are dynamically synchronized with music data and simple dynamic equations of particles are used to simulate realistic fountain behavior. The real-time update of particle attributes such as position, speed and the real-time rendering of particles are achieved through the availability of high parallel processing power of modern GPU. A new particle rendering method based on adaptive sampling in image space is also proposed for particle realism. The proposed algorithm has been verified and can be easily integrated into interactive applications for real-time fountain simulation.

Keywords music fountain, particle system, GPU, real-time rendering

0 引言

视景模拟 (visual simulation) 是计算机仿真技术的一个重要分支, 已经在驾驶模拟、古迹再现、城市规划、军事模拟以及其他领域得到了广泛应用。人工喷泉是一种将水或其他液体经过一定压力通过喷

头喷洒出来而形成的具有特定形态的组合体, 现在已经逐步发展为几大类: 音乐喷泉、激光水幕电影、趣味喷泉等。喷泉效果可以增加 3 维场景的真实感, 因此对于喷泉的模拟一直被很多研究人员关注。

类似喷泉这样的物体如火、烟、雾等, 形态千变万化, 具有不规则的几何外形和内在的不确定性, 虽然目前在这一领域已经有较多的算法和理论, 但总

基金项目: 国家自然科学基金项目 (60533080); 国家高技术研究发展计划 (863) 项目 (2006AA01Z314); 国家科技支撑计划课题资助项目 (2007BAH11B03)

收稿日期: 2008-09-22 **改回日期:** 2009-02-24

第一作者简介: 王槐欣 (1984—), 男, 浙江大学计算机应用技术专业硕士研究生。主要研究方向为计算机图形学和人机交互。

Email wanghuaixin@cad.zju.edu.cn

通讯作者: 万华根, Email hgwan@cad.zju.edu.cn

体来说, 对其模拟仍充满挑战。Reeves 在 1983 年首次提出粒子系统的概念^[1], 在此之前虽然有比如动态纹理、元胞自动机等诸多方法, 但粒子系统被证明是最成功的方法, 在模拟各种不规则变化的物体和现象中得到广泛的应用。由于 CPU 运算能力和内存带宽的限制, 传统的粒子系统能处理的粒子数量较少, 在满足实时模拟的条件下, 粒子数量最多只能达到几万。尽管利用优化算法(如 SLOD(simulation levels of detail)技术)可以补偿一部分 CPU 运算能力的不足^[2], 但效果并不理想。

高性能图形处理器(GPU)性能的不断发 展, 使得 GPU 在浮点运算能力上大大优于 CPU, 使其在很多领域有着广泛的应用, 如视频编解码中的运动补偿^[3]和 DCT/IDCT^[4]计算、加速光线跟踪^[5]、体绘制^[6]和实时阴影计算^[7]等。在 GPU 应用于流体模拟方面, L 等人用 LBM(lattice boltzmann method)来模拟流动效果^[8]; Krüger 等人利用 GPU 在基本代数运算的基础上, 实现了共轭梯度法和高斯-赛德迭代法^[9], 从而更进一步地求解 2 维波动方程和不可压 NSEs(navier-stokes equations); Batty 等人利用 GPU 采用共轭梯度法来模拟流体运动^[10]。在 GPU 应用于粒子系统方面, Latta 实现了并行处理大规模粒子系统的方法^[11]; Krüger 等人用 GPU 来实现火焰、流体等的建模和绘制^[12]; Sato 等人使用 GPU 实时地绘制水滴从玻璃窗上滑落^[13]; K ipfer 等人利用 GPU 实时地绘制大量粒子并检测其碰撞^[14]。

音乐喷泉作为一种视觉和听觉相结合的艺术景观形式, 正越来越被大众所喜爱。现代音乐喷泉少则一二十个喷口, 多则成百上千, 甚至数万个喷口, 气势宏伟, 规模庞大, 已经成为很多旅游城市和景区一道亮丽的风景线, 但这大大增加了对其视觉模拟的难度。音乐喷泉的粒子数量可以达到百万级别, 传统的粒子系统方法几乎不可能实现对如此海量粒子的实时运动计算与绘制。

本文提出一种基于图形硬件加速的大规模音乐喷泉实时模拟方法, 通过分析和提取音乐数据对喷泉进行粒子运动建模, 利用高性能图形处理器强大的并行处理能力, 使得能够对海量粒子进行实时运动计算与绘制。实验结果表明, 该模拟方法行而有效, 既保证了对音乐喷泉特征的描述, 又保证了绘制速度, 在绘制效果和效率间取得了较好的平衡。

1 音乐喷泉的粒子运动建模

1.1 音乐喷泉模拟流程

粒子系统被广泛地应用于对流体的建模, 并被证明是模拟不定形物体行而有效的方法, 其发展至今已经相对成熟, 主要有两种: 状态无关粒子系统和状态相关粒子系统^[11]。本文采用状态无关粒子系统。根据粒子系统建模方法, 主要的喷泉粒子属性均可采用 3 维向量表示, 例如: 粒子的当前位置 (x, y, z) 、当前速度 (v_x, v_y, v_z) 等, 运用动力学方程可计算喷泉运动粒子的属性变化^[15-16]。音乐喷泉模拟流程如图 1 所示。

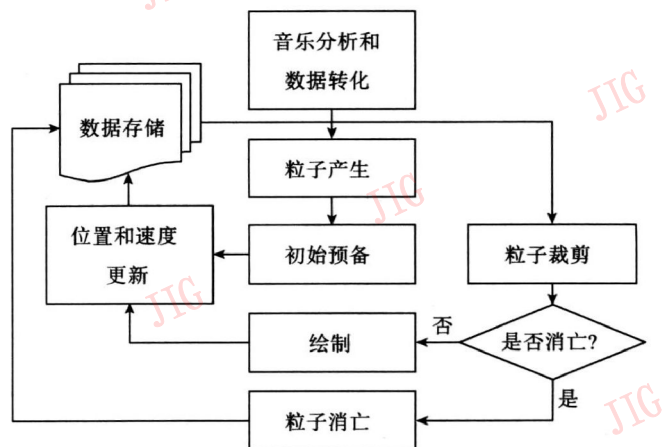


图 1 音乐喷泉仿真流程

Fig. 1 Music fountain simulation process

1.2 音乐数据分析和提取

音乐的情感因素只能被人类有意识地接受, 而机械无法感知, 人工智能领域的情感化计算也在起步阶段, 对于一般的音乐喷泉系统, 其表演程序的设计非常费时费力, 也正是由于这个原因, 即便是美国 Las Vegas 这样著名的音乐喷泉也只能表演几首曲目。通过人工的方法提取音乐的音高、情感等特征工作量较大, 也有人试图用专家系统^[17]来代替人的这一工作, 但因为必须要建立一个强大的音乐情感库, 而这一工作的难度不亚于人工提取, 效果不是很理想。因此, 本文通过控制喷泉水流高度和音乐音高同步, 从而简化了音乐喷泉的控制机制。相对于未简化前, 本文的方法无需任何人为的参与, 用一段普通音乐作为输入就可以得到较好的结果。

当前流行的音乐格式有很多种, 其储存格式、采样率、声道等各不相同。如图 2 所示是一段 WAV

(wavform)格式音乐文件的音高数据图,沿着时间方向进行采样,可以得到相对应的音高数值。对于音乐喷泉的控制,需要分析和提取的信息有:采样率、声道数、样本数及音高数据,可以根据相应的音乐存储格式,读取相应的数据。以时间作为度量的标尺,根据不断变化的音高数值,来控制同样不断产生的粒子初始垂直速度,从而达到喷泉的运动和音乐同步变化。

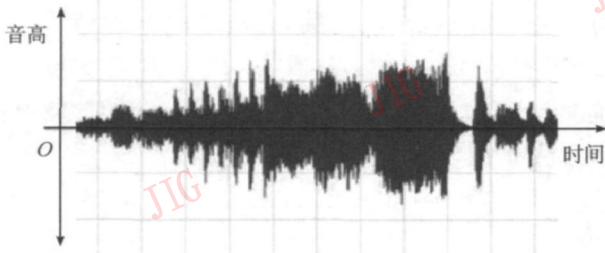


图 2 音乐音阶数据曲线
Fig. 2 Music scale data curve

1.3 音乐喷泉水流控制

一般地, Navier-Stokes 方程可以很好地描述流体的运动^[18],基于喷泉的形态和粒子特点,可以使用其中的非压缩牛顿流体运动的完整数学方程组来描述,该方程组非常复杂,虽然理论上可以精确求解,但由于其中对流加速项的非线性属性,现实中求解变得十分困难。幸运的是,对于均匀圆管中的平稳流体,往往可忽略该对流加速项,从而使得方程组可精确求解, Hagen-Poiseuille 流就是一个非常著名的例子。此时可以得到^[15-16]:

$$v_y = V_{max} \left[1 - \left(\frac{r}{R} \right)^2 \right] \quad (1)$$

式中, V_{max} 是基于对喷泉的高度控制而得到的出水速度的最大值,使得喷泉的高度不会过高,影响到观赏性, R 是喷泉出水口圆管的半径, r 是该粒子所在位置到圆管中轴的距离。式 (1) 给出了在初始预备阶段,圆管中的流体轴向速度的分布。在 t 时刻,喷泉的高度表示为

$$H = H_{min} + (H_{max} - H_{min}) \frac{VOL - VOL_{min}}{VOL_{max} - VOL_{min}} \quad (2)$$

式中, VOL_{max} , VOL_{min} 表示音乐数据的最大最小值, VOL 是当前采样值, H_{max} , H_{min} 是指定的最大最小喷泉高度,根据粒子运动方程,可得出: $V_{max} =$

$$\sqrt{2gH_{max}}, v_y = \sqrt{2gH}, t' = \sqrt{\frac{2H}{g}}, \text{继而}$$

$$r = R \sqrt{1 - \frac{v_y}{V_{max}}} \quad (3)$$

表明粒子要在 t 时刻到达高度 H , 必须分布在管内半径为 r 的圆周上,且在 $t - t'$ 时刻从圆管中以垂直初速度 v_y 喷出。

2 硬件加速的喷泉粒子运动计算

由于必须对每一个粒子进行动力学计算过程,一般商用 CPU 的运算性能不可能实现对大规模喷泉粒子系统的实时绘制。高性能图形处理器的出现,使理论上可以并行加速的计算过程能够快速完成。

图形处理器离屏绘制技术允许把离屏缓冲区作为绘制运算的目标,这一技术使得绘制的结果保存在纹理中,也就是某些离屏缓冲区。在一张纹理中,数据是可以索引的,可以将其想象成数组,这样就可以将计算的结果和所需的参数都保存在图形处理器的缓冲中,使得 CPU 和 GPU 协同工作,以便利用图形处理器强大的并行处理能力提高各种计算的速度。

对于粒子的位置属性而言,可使用一张浮点纹理存储所有粒子的位置,该纹理的颜色分量分别保存 x, y, z 坐标。由于一张纹理可以看作 2 维数组,其中的纹理坐标就可以代表数组索引。同样地,对于粒子的速度等其他属性,也做上述处理,如图 3 所示。

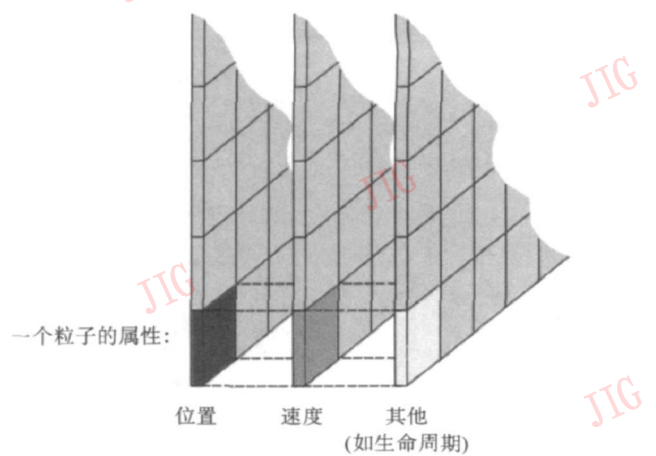


图 3 粒子属性储存

Fig. 3 Storage of particle attributes

其他粒子属性(如生命周期、粒子类型等),可以储存为静态纹理,而对于粒子速度和位置属性,由

于要对其做更新操作, 储存这两种数据的纹理必须同时被定义为绘制目标, 以确保能保存动态计算后的数据。使用离屏绘制技术, 通过绘制一个全屏的方形并绑定相应纹理, 图形处理器会为绘制目标上每个像素调用一次片段着色器。计算粒子新位置(或速度)时, 片段着色器需要得到上一次计算后粒子的位置, 该位置也必须从纹理中读取, 由于不能同时读取和绘制同一张纹理, 因此可以使用双缓存技术——创建两张纹理分别存储上帧数据和本帧计算后的新数据。

当使用了双缓存存储数据时, 必须分清哪张纹理用于读入上一次计算得到的数据, 哪张纹理用于储存本次计算的结果。假设两张位置和速度纹理分别标记为 1 号和 0 号, 则它们就被分成两组, 分别拥有一张位置纹理和一张速度纹理, 这样当第 1 组纹理用于输入的时候, 第 2 组纹理就用于输出, 下一个更新时刻则将第 2 组纹理用于输入, 第 1 组纹理用于输出。如图 4 所示。

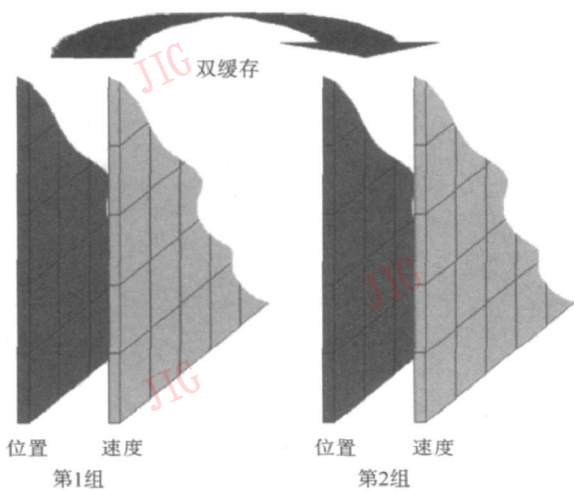


图 4 双缓存技术

Fig. 4 Double cache

3 音乐喷泉的实时绘制

对于音乐喷泉的模拟来说, 最重要的绘制过程就是对粒子的绘制, 比较传统的方法有: 点、线、billboard 等等。对于不同的绘制方法而言, 都具有自己的优点和缺点, 根据不同的使用场景选择运用不同的绘制方法。与点、线相比, billboard 的方式具有很多优势, 该方法在屏幕上绘制一个面向观察方向的方形, 并贴上纹理来代表一个粒子, 这样的粒子具有

比点和线模式更强的真实感, 被广泛应用于粒子的绘制, 但由于音乐喷泉的绘制对光照效果有较高要求, billboard 的方法效果并不很理想。

本文采用圆球作为喷泉粒子的绘制对象, 但并不直接绘制圆球的 3 维几何。为此, 本文提出 billboard 的改进方法, 设计了顶点着色器和片段着色器, 只使用一个方形在像素级别的精确度上来快速模拟圆球的绘制效果, 算法思想如下:

如图 5(a) 所示, 和一般的 billboard 方法相似, 所绘制的几何对象是一个方形, 方形的 4 个角分别为 V_0, V_1, V_2, V_3 , 为了转化为半径为 R 的球, 首先, 需进行裁剪处理, 即判断一个像素 V 是否在要得到的球内部, 这可通过计算这个像素到方形中心的距离并与球的半径做比较得到。如图 5(b) 所示, 经裁剪处理后结果为一个圆。接着, 需将圆内的点映射为球面上的点, 即于中心距离为 d 的点 P 要最终移动到 P' 位置, 最终结果如图 5(c) 所示。

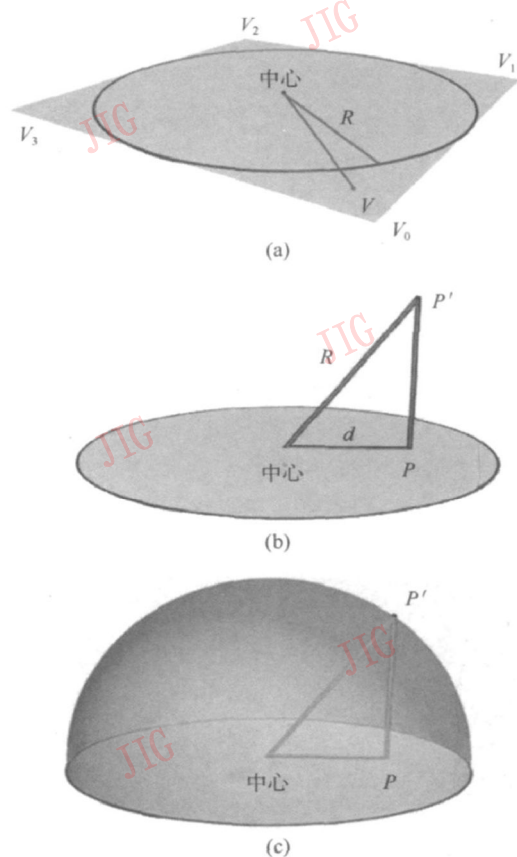


图 5 粒子绘制

Fig. 5 Particle rendering

由于 GPU 要对输入物体上的每个面片进行处理, 如果直接绘制一个圆球, GPU 不但要处理能见的部分的面片, 还要处理不可见的半球区域。例如要

绘制一个在屏幕上相对半径为 10 个像素的圆球, GPU 处理绘制这样一个球要进行至少 628 次像素数据输入。对于本文的绘制方法, GPU 只需要处理可见半球上的面片, 从而简化了绘制过程, 加快了绘制速度。于前例比较, 此时 GPU 只需要进行至多 400 次像素数据输入。又由于采用了 billboard 的方法, 所以该绘制半球总是面向可视方向。总体上, 本文的方法在不降低绘制效果的同时, 使绘制效率大大提升。

4 实验结果

为验证本文所述的音乐喷泉的模拟方法, 设计一个大规模的音乐喷泉系统, 其草图如图 6 所示。该音乐喷泉共有 61 个喷口, 3 个圆心水量要比其他喷口多, 中心处喷泉最高, 因此喷泉高度有 3 个层次, 中心为第 1 级别, 两旁的圆心为第 2 级别, 其他属于第 3 级别。

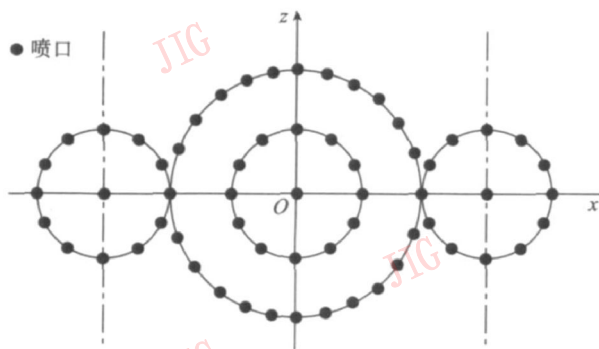


图 6 音乐喷泉设计草图
Fig. 6 Music fountain design

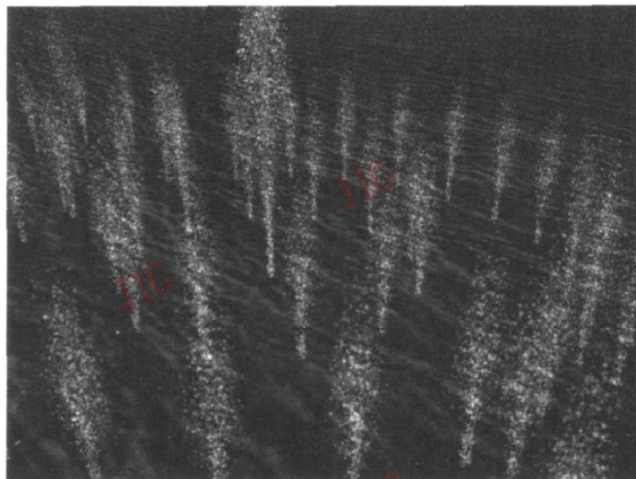


图 8 近景效果
Fig. 8 Close-range effect

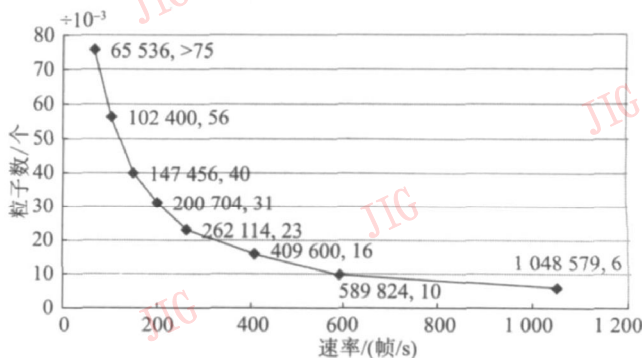


图 9 绘制速率
Fig. 9 Rendering rate

根据设计实验, 效果如图 7、8 所示。在普通的 PC 机 (Intel Core2 2.0 G 处理器; NV ID IA 8800GTS 320 MB 显卡; 2 GB 内存) 上进行绘制速率实验, 结果如图 9 所示。

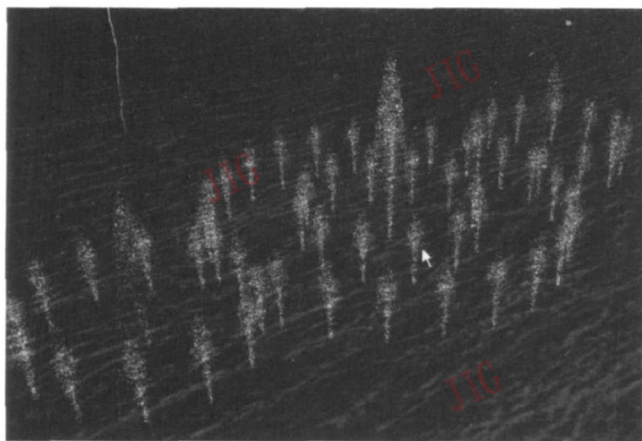


图 7 总体效果
Fig. 7 Overall effect

5 结 论

基于高性能图形处理器的并行处理能力, 给出了一种大规模音乐喷泉的实时模拟算法, 使得对音乐喷泉模拟突破了粒子数量的限制, 海量粒子的运动计算和绘制过程能够达到实时; 提出了简化的音乐喷泉运动控制机制以取代复杂的音乐喷泉情感化计算; 通过对 billboard 绘制方法进行有效改进, 既保证了对喷泉粒子的特征描述, 又保证了绘制速度, 在绘制效果和效率间取得了较好的平衡。今后将进一步研究: 基于对粒子深度的排序和水滴折射、反射计算, 达到更好的水滴的透明效果和光照效果模拟, 使得音乐喷泉拥有现实中光影交融的视觉效果; 建立大规模音乐喷泉粒子运动的情感计算模型, 实现基

于对音乐真正理解的音乐喷泉控制机制。使得喷泉的形态丰富多彩, 对音乐有更强的表现力, 给人赏心悦目的感觉。

参考文献 (References)

- [1] Reeves W T. Particle systems-A technique for modeling a class of fuzzy objects[J]. Computer Graphics 1983, 17(3): 359-376.
- [2] O'Brien D, Fisher S, Lin M C. Automatic simplification of particle system dynamics[J]. IEEE Computer Animation, 2001, 20(3): 210-218.
- [3] Shen G B, Gao G P, Li S P, et al. Accelerate video decoding with generic GPU[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2005, 15(5): 685-693.
- [4] Fang B, Shen G B, Li S P, et al. Techniques for Efficient DCT / IDCT implementation on generic GPU circuits and systems[C] // Proceedings of IEEE International Symposium on Circuits and Systems. Kobe: IEEE Press, 2005, 2: 1126-1129.
- [5] Schmittler J, Wald I, Slusallek P. SaarCOR-A hardware architecture for ray tracing [C] // Proceedings of ACM SIGGRAPH / EUROGRAPHICS Conference on Graphics Hardware. Saarbrücken: ACM Press, 2002, 27-36.
- [6] Guthe S, Roettger S, Schieber A, et al. High-quality unstructured volume rendering on the PC platform [C] // Proceedings of ACM SIGGRAPH / EUROGRAPHICS Conference on Graphics Hardware. Saarbrücken, Germany: ACM Press, 2002, 119-125.
- [7] Chan E, Ng R, Sen P, et al. Efficient partitioning of fragment shaders for multipass rendering on programmable graphics hardware [C] // Proceedings of ACM SIGGRAPH / EUROGRAPHICS Conference on Graphics Hardware. Saarbrücken, Germany: ACM Press, 2002, 69-78.
- [8] Li W, Wei X, Kaufman A. Implementing lattice boltzmann computation on graphics hardware[J]. The Visual Computer, 2003, 19(7-8): 444-456.
- [9] Krüger J, Westermann R. Linear algebra operators for GPU implementation of numerical algorithms[J]. ACM Transactions on Graphics, 2003, 22(3): 908-916.
- [10] Batty C, Wiebe M, Houston B. High Performance Production-quality Fluid Simulation via NVIDIA's Quadro FX[EB/OL]. (2003-07-10) [2008-09-20]. http://film.nvidia.com/docs/CP/4449/frantic_GPU_acceleration_of_fluids.pdf.
- [11] Latta L. Building a Million Particle System [OL]. [2008-09-20]. (2004-07-28). http://www.gam-asutra.com/features/20040728/latta_01.shtml.
- [12] Krüger J, Westermann R. GPU simulation and rendering of volumetric effects for computer games and virtual environments [J]. Computer Graphics Forum, 2005, 24(3): 685-693.
- [13] Sato T, Dobashi Y, Yamamoto T. A method for real-time rendering of water droplets taking into account interactive depth of field effects[C] // Proceedings of the 1st International Workshop on Entertainment Computer. Makuhari: IEEE, 2002, 110-117.
- [14] Kipfer P, Segal M, Westermann R. UberFlow: A GPU-based particle engine [C] // Proceedings of ACM SIGGRAPH. Los Angeles USA: ACM Press, 2004, 115-122.
- [15] Wan Huagen, Jin Xiaogang, Peng Qunsheng. Physically based real-time fountain simulation[J]. Chinese Journal of Computers, 1998, 21(9): 774-779. [万华根, 金小刚, 彭群生. 基于物理模型的实时喷泉水流运动模拟 [J]. 计算机学报, 1998, 21(9): 774-779.]
- [16] Wan H G, Cao Y J, Han X X, et al. Physically-based real-time music fountain simulation [C] // Proceedings of Entertainment Lecture Notes in Computer Science. Hangzhou: Springer, 2006, 1058-1061.
- [17] Liu Dan, Zhang Nairao, Zhu Hancheng. Computer-aided music fountain design system based on identifying characteristics of the music fountain[J]. Application of Electronic Technique, 2003, 29(3): 12-15. [刘丹, 张乃尧, 朱汉城. 基于音乐特征识别的音乐喷泉计算机辅助设计系统 [J]. 电子技术应用, 2003, 29(3): 12-15.]
- [18] Anderson D A, Tannehill J C, Pletcher R H. Computational Fluid Mechanics and Heat Transfer[M]. New York NY, USA: Hemisphere Publishing Corporation, 1984.