

文章编号:1001-9081(2008)04-1070-04

# 基于 Direct3D 与粒子系统实现喷泉效果

谭同德,常志伟,赵红领

(郑州大学信息工程学院 虚拟现实与地理信息系统实验室,郑州 450001)

(ttde@zzu.edu.cn)

**摘要:**粒子系统是实现喷泉效果的有效方法,运用 Direct3D 中的点精灵作为粒子图元渲染喷泉粒子具有一定的灵活性。首先介绍 3D 图形函数库 Direct3D 及粒子系统的基本原理,然后运用物理动力学分析现实生活中的喷泉运动,构建出喷泉粒子系统模型并实现喷泉的三维效果。提出伪粒子黏度的方法,使多个粒子“黏合”形成不同大小的水珠,增强了喷泉的真实效果。

**关键词:**Direct3D;粒子系统;喷泉;伪粒子黏度

**中图分类号:**TP391.9 **文献标志码:**A

## Three-dimension fountain effect of implementation based on Direct3D and particle system

TAN Tong-de, CHANG Zhi-wei, ZHAO Hong-ling

(Laboratory of Virtual Reality and Geography Information System, School of Information Engineering, Zhengzhou University, Zhengzhou Henan 450001, China)

**Abstract:** Particle system is an efficient way to implement fountain effect. Making use of the point-sprite of Direct3D by way of particle primitive to render fountain particle system possesses definite agility. First of all, 3D graphics function library Direct3D was introduced and the basic principles of particle system were described. Secondly the fountain movement in the true life was analyzed by physics dynamics, then fountain particle system model was constructed and 3D effect of fountain was implemented. The way of fake particle viscosity was proposed, so lots of particles could be agglutinated to form different big or small water beads. This way enhances reality effect of fountain.

**Key words:** Direct3D; particle system; fountain; fake particle viscosity

### 0 引言

瀑布、火焰、流水等自然景物具有不规则性、动态性和随机性,在虚拟场景中的模拟十分复杂困难,自 1983 年 William T. Reeves 提出用粒子系统模拟不规则模糊物体的方法<sup>[1]</sup>以来,复杂自然景物的模拟得以在虚拟场景中一展光彩。W. T. Reeves 模拟了行星被撞击后所产生的爆炸及火焰效果;Fournier A 等在研究海浪模型中利用粒子系统模拟了浪花;Karl Sims 研究了粒子动画及绘制算法<sup>[2]</sup>。国内对粒子系统的研究也在深入,如谢剑斌、郝建新等运用粒子系统模拟了雨点和雪花的降落<sup>[3]</sup>;彭群生和管宇实现了瀑布的模拟<sup>[4]</sup>;徐利明与姜显明实现了雨、雪的效果<sup>[5]</sup>。此外一些研究人员利用粒子系统实现了火焰、烟、云、导弹尾气等复杂景物的效果。

马骏和朱衡君采用矩形作为图元以粒子系统实现了喷泉的模拟<sup>[6]</sup>;万华根等采用圆球作为粒子图元模拟了喷泉水流效果<sup>[7]</sup>。本文基于粒子系统的原理,采用 Direct3D 的点精灵作为图元,分析并构建喷泉粒子系统模型,提出用伪粒子黏度方法将多个粒子“粘合”在一起构成不同大小的水珠,来增强喷泉的真实感。

### 1 Direct3D 简介

Direct3D 是 Microsoft 的 DirectX 软件开发包的组件<sup>[8]</sup>,是微软公司自行开发的 3D 图形函数库,目前已经发展到第 10

个版本(D3D10)。Direct3D 的任务是向具有设备无关性的三维视频显示设备提供设备相关的访问。

通常应用程序是通过调用图形设备接口(Graphics Device Interface, GDI)来实现的。由于 GDI 的技术局限,使得应用程序不能高效地访问或操作硬件资源。OpenGL 和 Direct3D 都提供了专业的图形硬件访问接口,OpenGL 主要通过直接访问设备驱动接口(Device Drive Interface, DDI)来实现,但 DDI 的通用性降低了对图形硬件的访问效率。Direct3D 则提供了访问图形设备的立即模式,该模式通过硬件抽象层(Hardware Abstraction Layer, HAL)获得了更高效率。HAL 还允许图形硬件在渲染、光栅化等方面保留自己独特的性能,从而获得更优化的显示效果。

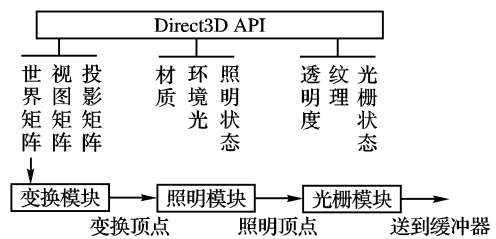


图 1 Direct3D 渲染引擎模块

与 OpenGL 相比,Direct3D 在速度方面占有一定的优势,渲染成像质量也进一步提高。Direct3D 的核心体系结构是一个三维绘制引擎,由 3 个单独的模块组成,即变换模块、照明

收稿日期:2007-10-31;修回日期:2007-12-29。 基金项目:国家 973 规划项目(2002CB312103)。

作者简介:谭同德(1950-),男,山东莱阳人,教授,博士,主要研究方向:计算机辅助设计、图形图像处理、虚拟现实;常志伟(1980-),男,河南沈丘人,硕士,主要研究方向:虚拟现实;赵红领(1978-),男,河南永城人,硕士,主要研究方向:虚拟现实、数据库。

模块和光栅化模块,这三个模块一起构成了 Direct3D 的渲染流水线,如图1。

Direct3D 的一个特点是提供了点精灵,点精灵是 DirectX8 版本以来所具有的新特性,是由其自身编程接口支持的。点精灵可以是一个点、一个像素,也可以是很小的正方形平面采用 billboard 技术得到的效果。可以对点精灵贴图,其纹理坐标是左上角为(0.0f,0.0f),右下角为(1.0f,1.0f),不需要编程者自己指定,因此渲染时可直接设置纹理,不需要指定纹理坐标。除纹理外,还能灵活设置点精灵的大小、颜色、透明度等属性。一般情况下点精灵需与粒子系统相结合,它作为粒子图元能较好地实现雨、雪、水珠、爆炸等特殊效果。本文中运用点精灵作为喷泉水粒子图元来实现水珠的渲染,取得了较好的效果。

## 2 粒子系统基本原理

粒子系统是 Reeves 在 1983 年提出的,迄今为止,被认为是模拟不规则模糊物体最成功的一种图形生成算法。粒子系统基本原理是采用大量的、具有一定生命的微小粒子图元作为基本元素,表示不规则的模糊物体。在粒子系统中,粒子包含一定的属性如大小、颜色、速度、位置等,要经历产生、运动、消亡的过程,粒子图元可以是许多简单的形状,如小球<sup>[7]</sup>、立方体、正四面体<sup>[9]</sup>等。

通常情况下,运用粒子系统描述模糊物体要经过下面的步骤:

- 1) 生成新的粒子加入到粒子系统中,为新生粒子各属性设置初始值。
- 2) 粒子在系统中不断运动,同时不断更新其属性值。
- 3) 删除系统中已消亡的粒子。
- 4) 将未消亡的粒子绘制并显示成图像。

## 3 喷泉粒子系统模型的实现

用粒子系统模拟不规则模糊物体的运动与物理动力学是分不开的,运用物理动力学分析物体的运动及特性,依此构造出数学模型,以实现更为逼真的虚拟效果。

### 3.1 分析喷泉粒子系统模型

在现实生活中所看到的喷泉由喷头(发射器)发出水体,在一定水压下向一定方向上发出水柱,而后下落散开,形成不同形状的喷泉。喷泉粒子系统中喷泉体是由一个或多个喷泉柱组成,每个喷泉柱由许多的水珠粒子组成。由喷泉的物理动力学原理可知,无其他外力(除重力外)作用下水珠运动轨迹近似于抛物线,依据牛顿第二定律可知水珠粒子的运动速度  $V$  与位置  $P$  为:

$$V = V_0 + \int_0^t a \cdot dt \quad (1)$$

$$P = P_0 + \int_0^t V \cdot dt \quad (2)$$

其中  $V$  为水珠粒子的当前速度,  $V_0$  为水珠粒子的初始速度,  $a$  为水珠粒子的加速度,  $t$  为时间的变化,  $P$  为当前位置,  $P_0$  为初始位置。一般情况下,喷泉的初始速度大小及方向由水压和喷头的形状决定,从而决定喷泉的形状。

从上述分析知,在计算机中模拟喷泉需要以下步骤:

- 1) 分析喷泉粒子的静态属性和动态特性。
- 2) 生成新的粒子加入到喷泉粒子系统中,并初始化粒子属性。

3) 根据喷泉粒子的运动规律,更新系统中粒子相应的属性值。

4) 由粒子的生命值判断粒子是否死亡,若是则删除喷泉粒子系统中已死亡的粒子。

5) 渲染并显示没有死亡的粒子。

喷泉的形成是由步骤2)~5)不断循环运动的结果。渲染喷泉的步骤如图2所示。

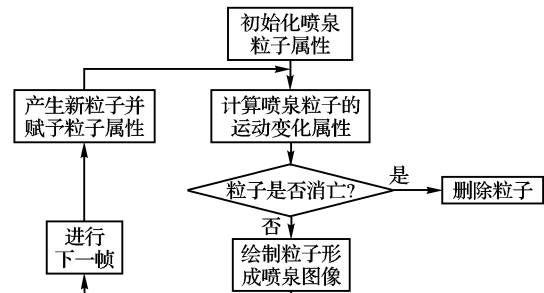


图2 渲染喷泉粒子的步骤

### 3.2 喷泉粒子的属性分析

在虚拟场景中模拟喷泉的效果,需分析喷泉的各个属性。在本文喷泉粒子系统中粒子的属性主要有初始速度  $m\_vVel0$ , 初始位置  $m\_vPos0$ , 当前速度  $m\_vVel$ , 当前位置  $m\_vPos$ , 粒子的颜色  $m\_clrDiffuse$ , 粒子的生命周期  $m\_fTime$ , 判断水珠粒子是否会溅起水花的布尔值  $m\_bSpark$  等。喷泉粒子结构体如下:

```

struct PARTICLE //粒子结构体
{
    BOOL m_bSpark; //布尔值标识粒子是否产生水花
    D3DXVECTOR3 m_vPos; //当前位置
    D3DXVECTOR3 m_vVel; //当前速度
    D3DXVECTOR3 m_vPos0; //初始化位置
    D3DXVECTOR3 m_vVel0; //初始化速度
    FLOAT m_fTime; //生命周期
    D3DXCOLOR m_clrDiffuse; //初始化时粒子的颜色
    PARTICLE * m_pNext; //指向下一个粒子指针
};
  
```

### 3.3 喷泉粒子的产生

喷泉系统中产生新粒子需要初始化其属性如下:

初始速度  $m\_vVel0 = V + \mu_1 \cdot rand()$

初始位置  $m\_vPos0.x = P_0.x + \mu_2 \cdot rand()$

$m\_vPos0.z = P_0.z + \mu_2 \cdot rand()$

$m\_vPos0.y = P_0.y$

粒子颜色  $m\_clrDiffuse = D3DXCOLOR(0.05, 0.150, 0.30, 0.50)$ ,

粒子生命  $m\_fTime = life + \mu_3 \cdot rand()$

$\mu_1, \mu_2, \mu_3$  为常量,  $rand()$  为一个随机函数,  $V$  为平均速度,  $life$  为平均寿命值。

喷泉粒子类中设置一个参数控制水珠粒子最大数  $m\_dwParticlesLim$ , 粒子系统中的水珠粒子总数不大于  $m\_dwParticlesLim$ , 此外设置参数  $m\_dwNumParticlesToEmit$  表示每一帧粒子发射的数目,它直接影响着喷泉水体的密度。

### 3.4 喷泉粒子的运动

水珠粒子在三维空间中运动,以 Direct3D 中的左手坐标系为基准,平行于 XOZ 平面的方向受风力影响, Y 轴方向受重力的影响。粒子的速度与位置随着时间的变化而变化,如上述分析,在计算机中按帧绘制图像:

$$V(f_i) = V(f_{i-1}) + a \cdot \Delta t \quad (3)$$

$$P(f_i) = P(f_{i-1}) + V(f_i) \cdot \Delta t \quad (4)$$

则喷泉粒子第  $i$  帧速度为:

$$\begin{aligned} V(f_i).x &= V(f_{i-1}).x + windV \cdot \cos(\theta) \\ V(f_i).z &= V(f_{i-1}).z + windV \cdot \sin(\theta) \end{aligned} \quad (5)$$

$$V(f_i).y = V(f_{i-1}).y - g \cdot \Delta t$$

其中  $windV$  为风的速度,  $\theta$  为风的方向。

喷泉粒子第  $i$  帧位置为:

$$\begin{aligned} P(f_i).x &= P(f_{i-1}).x + V(f_i).x \cdot \Delta t \\ P(f_i).z &= P(f_{i-1}).z + V(f_i).z \cdot \Delta t \\ P(f_i).y &= P(f_{i-1}).y + V(f_i).y \cdot \Delta t \end{aligned} \quad (6)$$

### 3.5 喷泉粒子的消亡

粒子不能无限制地运动下去,不然系统中将不会有新粒子产生,粒子的生存由粒子的生命值所决定。有两种情况粒子消亡:

1) 水珠粒子的生命值  $m\_fTime \leq 0.0$  时,粒子的生命周期结束,粒子消亡;

2) 设置一个与 Y 轴正交的水面,当水珠粒子碰到水面时,粒子的生命值  $m\_fTime = 0.0$ ,粒子消亡。

### 3.6 喷泉粒子黏度

黏度是流体的一种物理属性,用以度量流体黏性大小的物理量,又称为黏性系数、剪切黏度或动力黏度<sup>[10-12]</sup>,记为  $\mu$ 。

对于牛顿液体,由牛顿内摩擦定律可知黏度为  $\mu = \frac{\tau_{yx}}{d_{u_x}/d_y}$ ,其中  $\mu$  为流体黏度,  $\tau_{yx}$  为剪切应力;  $u_x$  为速度分量;  $d_{u_x}/d_y$  为剪切应变率;  $x, y$  为坐标轴。不同流体的黏度数值不同,同种流体的黏度与压强几乎无关;而与温度显著有关,随着温度的升高而减小。

由于粘滞性的存在,流动的液体内部存在着摩擦阻力(即粘滞力)。

水的黏度是水体在移动时,水体分子之间产生的内摩擦力的量度。水体黏度与水的特性有关,它取决于水的温度、表面状况和密度,而主要受温度的影响,随着温度的升高而减小。水的黏度系数是由实验测定的(如用黏度计测量),在一定温度下,水的黏度值是一个常量,表 1 是由实验测定的在不同温度下水的黏度值。

表 1 水在不同温度下的黏度值

温度 $T/^\circ\text{C}$	黏度 $\mu/\text{厘泊}$	温度 $T/^\circ\text{C}$	黏度 $\mu/\text{厘泊}$
0	1.7921	18.0	1.0559
1.0	1.7313	19.0	1.0299
5.0	1.5188	20.0	1.0050
10.0	1.3077	20.2	1.0000
15.0	1.1404	21.0	0.9810
16.0	1.1111	25.0	0.8937
17.0	1.0828	30.0	0.8007

在虚拟场景中,微面积  $S_i$  下的两层粒子流体(如图 3 所示),层与层之间发生相对运动,速度快的粒子流体层对相邻速度较慢的粒子流体层产生一个拖动其向运动方向前进的力,而同时运动较慢的粒子流体层对速度快的粒子流体层也作用一个大小相等、方向相反的力,从而阻碍较快的粒子流体层向前运动。这种运动着的相邻两粒子流体层间的相互作用力,称为流体的内摩擦力(即粘滞力)。

在虚拟场景中很难表示出水体内部的粘滞力,由于粘滞

力的存在,才具有不同大小的水珠粒子。一般的喷泉模拟是通过均匀分布函数使粒子分布在三维空间中,所看到的只是单个的粒子。为了更好地模拟水珠粒子,得到更好的喷泉效果,在此提出伪粒子黏度的方法。与流体中的粘滞力的原理相同,速度快的粒子流体层对速度较慢的粒子流体层产生一个向前运动的力,运动较慢的粒子流体层对速度较快的粒子流体层作用一个大小相等、方向相反的力,阻碍较快的粒子流体层向前运动。这种相互作用的力,称为粒子粘滞力。粒子移动时,粒子之间产生的粘滞力的衡量尺度称为粒子黏度,记为  $\mu$ 。

喷泉的水珠形状具有不规则性,当多个水珠粘在一起形成一个小小水柱下落时,人的视觉所看到的几乎是断断续续的柱体形状。因此在虚拟场景中可将喷泉水体模拟成柱状的粒子体,这样在微面积下所形成的水珠粒子具有一定的真实性。依据牛顿黏性定律,微面积所产生的摩擦力为:

$$F_i = \mu \cdot V_i \cdot S_i \quad (5)$$

式(5)中:  $F_i$  为流体微元在水流变化中平行于摩擦面的内摩擦力(即粘滞力);  $V_i$  为在每一瞬时变化的速率;  $S_i$  为摩擦微面积,  $S_i = S + k \cdot rand()$ ,其中  $S$  为粒子系统中层与层之间的平均面积,  $k$  为一常数,  $rand()$  为随机函数;  $\mu$  为流层间摩擦系数,即流体的黏度值。

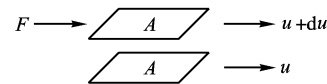


图 3 粒子流体在微面积下两层之间的流动

由水珠粒子黏度与水珠粒子某时刻的速率及微摩擦面积可计算出水体的粘滞力;再根据水珠粒子粘滞力的大小,使粒子图元组成不同大小的水珠,即多个粒子“粘”在一起形成不同大小的水珠(图 4 所示);当两滴水珠之间的粘滞力大时,它们联在一起,形成一个大水珠,将此方法称为伪粒子黏度方法。

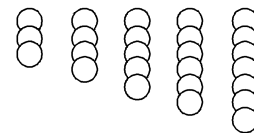


图 4 由多个粒子构成的水珠(图中圆环表示粒子图元)

由于下落的水体呈现柱状,在不影响整体真实性效果情况下,设水体的摩擦面与竖直面平行,且微面积为  $S_i$ ;水的黏度是由实验测定的,在温度为  $20.2^\circ\text{C}$  时水的黏度为  $1.0$  厘泊,则虚拟场景中以水在  $20.2^\circ\text{C}$  温度下的黏度值来定为粒子黏度系数  $m\_viscosity = \mu = 1.0$  厘泊;粒子速率为  $m\_vVel$ ,则由式(5)得出粒子流体内部的粒子粘力  $F = m\_viscosity \times m\_vVel \times S_i$ 。由伪粒子黏度方法知,根据粒子流体内摩擦力的大小可将粒子图元组成不同大小的水珠,这样在虚拟场景中,人的视觉看到的是一些连续的水珠在下落,从而可看到很逼真的喷泉。

实验证明这种方法能较好地渲染出喷泉的效果。

### 3.7 喷泉的渲染

粒子的渲染:粒子经过产生、运动、消亡后,需要对未消亡的粒子进行渲染。渲染过程中需要对粒子作如下处理:

1) 纹理处理,设置粒子的纹理:

SetTexture( 0, m\_pParticleTexture );

2) 透明度处理,运用函数 SetRenderState 进行设置,即:

SetRenderState( D3DRS\_ALPHABLENDENABLE, TRUE );

3) 灯光处理,对喷泉粒子系统场景设置环境光:

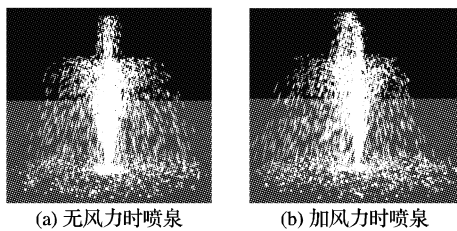
```
SetRenderState( D3DRS_AMBIENT, 0x00808080 );
```

此外可以对喷泉体设置聚光灯,并设置聚光灯的属性:

```
D3DLIGHT9 light ; //定义灯光
light.Type = D3DLIGHT_SPOT ; //指定灯光的类型为聚光灯
经过以上的处理可渲染喷泉画面。
```

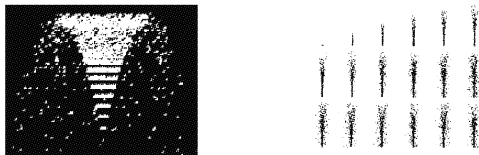
#### 4 实验结果与比较分析

此实验是在郑州大学虚拟现实与地理信息系统研究室 PC 机上实现的,处理器为 Pentium 4,主频 3.0 GHz,显卡为 GeForce6600GT,内存为 512 MB。当粒子数为 30 720,分辨率为 800×600 时,帧率为 113.45 fps。图 5 分别为无风力时及有风力时某一帧的效果。当帧率为 113.45 fps 时,可以看到流畅的、逼真的画面,具有一定的实时性。在虚拟场景中漫游时,从不同的角度观看场景时,可看到立体的喷泉,实现了喷泉的三维效果。



(a) 无风力时喷泉 (b) 加风力时喷泉

图5 本文喷泉系统画面



(a) 马骏和朱衡君实现的纹理喷泉 (b) 万华根模拟的喷泉形成过程

图6 其他文献喷泉系统画面

图 6(a) 为马骏和朱衡君实现的喷泉模拟,其方法是假定喷泉由均匀分布的水柱组成,水柱的运动轨迹是抛物线,用矢量控制法以及恰当的固定不变的调控参数控制水柱的路径。这种用固定不变的调控参数控制水柱路径的方法缺乏灵活性,形成的喷泉形状像一条抛物线旋转一周形成的空间物体(见图 6(a) 的形状),不便实现不同类型不同形态的喷泉。图 6(b) 为万华根等模拟喷泉的过程,其方法是利用圆球相互靠近时产生变形,当进一步靠近时融合在一起的图形特性,来表现水滴相互靠近形成水团的现象,因而采用圆球作为粒子图元,运用光线跟踪算法表现水珠的颜色亮度。小球由多个面形成,绘制许多的小球增加了绘制面的个数,采用光线跟踪算法绘制计算量大,因此实现较为复杂。

图 5 为本文方法的实验结果,可见实现的喷泉效果更加逼真,具有较强的真实感。与上述方法相比,伪粒子黏度方法可实现不同类型不同状态的喷泉,具有一定的灵活性;运用物理动力学分析喷泉的运动,运用 Direct3D 中的点精灵作为粒子图元较为简单。

从下面几方面加强了喷泉的真实感:

1) 运用纹理映射技术与粒子系统相结合,在模拟喷泉时加上了适当的纹理贴图提高了图像的真实感。

2) 考虑到风的影响,对不同的风力进行了模拟,使得喷泉因受外界力的影响而显示不同的效果,这体现了喷泉的动态性。

3) 应用伪粒子黏度法,形成由许多的水珠粒子组成的喷泉体,再利用 Direct3D 中提供的光照技术对喷泉场景设置光源,通过光照技术能看到粒子的透明状况。

#### 5 结语

本文运用微软公司的 3D 图形函数库 Direct3D 与粒子系统结合,模拟了喷泉体的不规则性、动态性和随机性,从流体动力学角度,分析流体的黏度,通过牛顿黏性定律得出流体粒子之间的内摩擦力(粘滞力),从而提出了伪粒子黏度的方法,并运用伪粒子黏度将多个粒子粘在一起形成水珠,实现了喷泉的三维效果,具有较好的真实感和实时性。

运用粒子系统与伪粒子黏度方法对喷泉模拟,具有一定真实性,但还需进一步完善。在今后工作中需要对喷泉粒子进行雾化处理、碰撞检测处理、水珠落到水面产生水波的处理等。

#### 参考文献:

- [1] REEVES W T. Particle system - A technique for modeling a class of fuzzy objects[J]. Computer Graphics, 1983, 17(3): 359 - 376.
- [2] SMIS K. Particle animation and rendering using data parallel computation[C]// Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM, 1990: 405 - 413.
- [3] 谢剑斌,郝建新. 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, 9(4): 734 - 738.
- [4] 彭群生,管宇. 基于粒子系统的实时瀑布模拟[J]. 系统仿真学报, 2004, 16(11): 2471 - 2474.
- [5] 徐利明,姜昱明. 基于粒子系统与 OpenGL 的实时雨雪模拟[J]. 计算机仿真, 2005, 22(7): 242 - 245.
- [6] 马骏,朱衡君. 基于动态纹理和粒子系统的喷泉模拟[J]. 北京交通大学学报: 自然科学版, 2005, 29(1): 90 - 94.
- [7] 万华根,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.
- [8] Microsoft DirectX 9.0 SDK Documents[EB/OL]. [2007 - 10 - 01]. <http://www.microsoft.com>.
- [9] 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.
- [10] 张晖,张秉坚,梁世强,等. 微孔中简单流体粘度的分子动力学模拟及关联模型[J]. 物理化学报, 2003, 19(4): 352 - 355.
- [11] 杨海明,张胡铭,徐小云,等. 毛细管表面疏水改性对粘度测量的影响[J]. 化学通报, 2003, 66(w053): 1 - 5.
- [12] 王文,刘志刚,陈钟顺. 流体粘度的计算机模拟[J]. 计算物理, 1998, 15(6), 687 - 691.
- [7] Symbian OS - the open mobile operating system[EB/OL]. [2007 - 10 - 01]. <http://www.symbian.com>.
- [8] SALES J. Symbian OS Internals: Real Time Kernel Programming [M]. New York: Wiley & Sons, 2005.
- [9] JOHNSON R. Frameworks = Patterns + Components [J]. Communications of the ACM, 1997, 40(10): 39 - 42.
- [10] RFC 2246, The TLS Protocol Version 1.0[S/OL]. [2007 - 10 - 01]. <http://www.ietf.org/rfc/rfc2246.txt>.
- [11] GAMMA E, HELM R, JOHNSON R, et al. Design patterns: Elements of reusable object-oriented software [M]. Boston: Addison-Wesley, 1995.
- [12] SCHMIDT D C, STAL M, ROHNERT H, et al. Pattern-Oriented Software Architecture: Patterns for Concurrent and Networked Objects[M]. New York: Wiley & Sons, 2000.

(上接第 1048 页)