

基于粒子系统的三维场地降雨实时模拟

罗维佳 都金康 谢顺平

(南京大学城市与资源学系, 南京 210093)

摘要 粒子系统是实时模拟三维地形场景中具有真实感的降雨情景的一种有效方法。本文基于粒子系统的基本原理分析并提出了适用的雨粒子系统和雨粒子的属性, 采用了把雨粒子产生区域定义为一个视图体顶部的外接长方体、用像素点和直线作为雨粒子的形状、降落过程的重力作用模拟、用粒子组实现雨粒子的连续补充和雨粒子的降落高度检测等新的方法及算法。试验结果表明, 该方法能有效地再现三维地形环境场景中的降雨景象, 在满足实时交互漫游的前提下表现出较强的真实感, 具有一定的实用价值。

关键词 粒子系统 雨粒子 三维场地 实时模拟

中图法分类号: P208 TP391 **文献标识码**: A **文章编号**: 1006-8961(2004)04-0495-06

Real-Time Simulation of Rain in 3D Terrain Scene Based on Particle Systems

LUO Wei-jia, DU Jin-kang, XIE Shun-ping

(Department of Urban and Resources Sciences, Nanjing University, Nanjing 210093)

Abstract During the course of present investigation, a method for simulating the rain in 3D terrain scene in the real time is proposed, using particle systems. The applicable attributes have been analyzed with reference to the rain particle system and the rain particles, in accordance with the basic principles of particle systems. Methodological approach includes the employment of feasible techniques or algorithms such as defining a cube outside the top of the view frustum as the generation shape of rain particles, using a point in pixels and a subsequent line as the shape of a rain particle, simulating the gravity influence in the course of rain particles' put-down, replenishing continuously rain particles using particle groups and, testing the fall heights of rain particles in the real time etc. The results of experimentation have proved that this method is more effective in the representation of rain in 3D terrain scene which also found to be realistic at the time of satisfying basic need of real-time interactive navigation.

Keywords particle systems, rain particles, 3D terrain scene, real-time simulation

1 引言

Reeves 于 1983 年提出了用粒子系统模拟模糊物体, 并用来模拟行星被撞击后产生的火焰^[1]。此后, 粒子系统受到越来越多的重视, 不少研究人员对它进行应用研究, 具有代表性的研究有: 用结构化粒子系统模拟树林和树林中草丛覆盖的地面, 并用近似概率性算法来解决关于粒子系统渲染时的明暗和可见面问题^[2]; 用固定粒子数的粒子系统结合分布

式行为模型模拟了鸟群的飞行, 每一个粒子对应一只鸟, 该粒子有方向, 考虑粒子间的相互影响, 遵从复杂的群体行为规则^[3]; 综合可变形表面模型和交互粒子模型的方法, 并给粒子加上 3 个旋转自由度, 提出的一种同时适用于可变形物体和固体的表面建模方法——方向粒子系统^[4]。国内对粒子系统的研究较少, 主要是对已有方法和理论的应用研究, 但没有较大的突破。粒子系统早期的应用主要是生成各种粒子图像, 之后再与其他景物的图像合成。后期随着计算机硬件性能的提高和软件算法的成熟,

已有不少实时生成粒子及其场景图像的实例。降雨现象是自然界中一种常见现象,生成真实感强的降雨图像在计算机艺术、广告、训练、娱乐等应用领域具有相当的价值。“雨场”可以看作是一种模糊物体,因此可以用粒子系统模拟降雨。文献[5]用粒子系统的方法仿真了降雨和下雪,其研究重点是在满足视觉效果的前提下尽可能加快显示速度,但没有考虑雨雪的周围场景,雨点和雪花粒子属性的分析较为简略,不够详尽。较为真实的场景必须包括地形,为此在详细分析雨粒子属性的基础上,提出用粒子系统的方法模拟起伏不平的山地环境中的降雨,实时生成场景图像,并进行场景交互漫游;另外,常规粒子系统方法以最大生存期为一图像周期,不同图像周期存在图像跳跃,不连续,为此,研究了用粒子组及时补充粒子,确保粒子在有生有灭的同时保持粒子总数不变,从而实现了平稳的连续降雨模拟。

2 粒子系统和雨粒子系统

粒子系统从应用的角度可分为3类^[6]:

(1)随机粒子系统,主要通过可控制的随机过程控制粒子属性的变化,可用来生成火、烟、灰尘、爆炸等场景图像;

(2)结构化粒子系统,主要用来模拟具有一定结构的物体或现象,如树、草、云、彩虹等;

(3)方向粒子系统,考虑粒子间的相互影响,粒子除了具有速度和位置等动态属性外,还必须具有方向属性,可用来模拟织物、可变形物体和刚体等。降雨现象没有一定的结构,模拟时不需考虑雨粒子之间的相互作用,宜采用随机粒子系统进行模拟。

2.1 随机粒子系统

随机粒子系统的基本思想是用大量简单的粒子组成的粒子群模拟模糊物体(图1),这些粒子具有大小、形状、颜色、质量、位置、速度以及受力等属性,随着时间的推移,粒子的形式和运动状态发生变化,各粒子状态的改变引起整个粒子群形态的变化。在这过程中,旧粒子不断死亡,新粒子不断诞生。粒子属性的变化及生死都可以通过受控的随机过程来模拟。粒子系统由许多粒子组成,并时时控制每个粒子的行为。为了计算一个运动系列的每一帧,要完成下列操作:在系统里生成新粒子;给新生粒子赋初始值;去除应消失的粒子(例如生存期满或位置超出显示范围等);修改存在粒子的属性(如位置、颜色和透

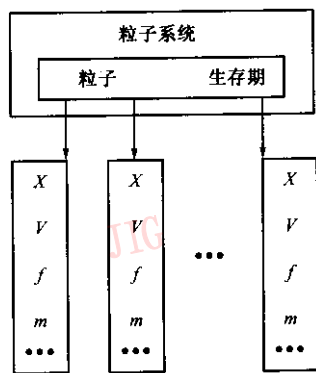


图1 粒子系统示意图

明度等);渲染存在粒子形成的图像。

2.2 雨粒子系统及雨粒子属性

雨粒子系统由大量的雨滴粒子组成,对雨粒子系统要考虑的属性有粒子数、粒子产生区域(generation shape)、粒子平均生存期,对雨滴粒子则要考虑形状、位置、速度、大小、生存期、颜色、透明度等属性。

雨粒子产生区域是一个覆盖地面场景的长方体,由于雨点自上而降落,这个区域在地面之上,与水平面平行。在粒子密度相同的情况下,较大的产生区域需要较多的粒子,因此产生区域在满足视觉效果的前提下应尽可能小,这里,仅用视图体顶部的外接长方体作为产生区域。随着视点位置、视线方向的变化,视图体在空间移动,产生区域也作相应移动。该产生区域可用一个数组来表示,每一个数组元素就表示一个雨滴粒子。雨滴粒子在水平面上的投影坐标可根据数组元素的下标换算得到,用一个属性变量来记录在竖直方向上的坐标。雨滴粒子的生存期表征了其在屏幕上停留的时间,用帧数表示。速度一定时,它决定了粒子下降的最大高度。若雨的平均下降距离为 $Height$ (为了处理方便, $Height$ 可用显示窗口高度换算得到),平均下降速度为 $Speed$,粒子的平均生存期为

$$Meanlife = Height / Speed$$

若生存期方差为 $Varlife$,根据随机粒子系统的思想,粒子的生存期由下式确定

$$Life() = Meanlife + rand() \times Varlife \quad (1)$$

其中, $rand()$ 是一个在 -1.0 到 $+1.0$ 之间的随机数。平均生存期过大或过小都会引起明显的降雨不连续。

要在给定屏幕显示区域内产生一定数量的雨滴粒子,粒子数量很关键,它决定了雨滴的密度,数目过小

或数目过大都会影响到真实感。另外,如果数目过大,则计算及绘制的时间增长,实时性受到影响。此外,该数目还与雨滴粒子的大小有关。设屏幕上单位区域内产生的粒子数目的平均值为 $Meannumber$, 方差为 $Varnumber$, 显示区域大小为 $Area$, 则初始时刻(第 1 帧)应产生的粒子数由下式给出^[1]

$$Number(1) = [Meannumber + rand() \times Varnumber] \times Area \quad (2)$$

假定降雨过程中雨的密度不发生变化,对应的要求是各帧图像上的雨粒子数大致相等,即

$$Number(i+1) \approx Number(i)$$

为了简化处理,可令

$$Number(i) = Number(1)$$

由于粒子的平均生存期为 $Meanlife$, 第 $i+1$ 帧图像上出现第 i 帧粒子的数目约为 $Number(i)(1 - 1/Meanlife)$, 则第 $i+1$ 帧图像上需新产生的粒子数为 $Number(1)/Meanlife$ 。

雨滴粒子的形状根据整个场景的复杂程度、计算机性能和实时显示的要求进行选择,如果整个场景较为简单、计算机性能较好、实时要求不高,可考虑选择真实感强的三维图元,如四面体、长方体和球体等,反之则选择二维图元甚至一维图元,如三角形、矩形、圆形或直线、点。由于地面和天空的数据量较大(分别为 512×512 、 256×256 的纹理),在场景交互漫游时需使用纹理滤镜在纹理像素之间进行线性内插以及对场景雾化等耗时计算,有一定实时要求,以最简单的像素点作为雨滴粒子的形状。考虑到雨点接近地面时速度较快,而人眼观察物体存在视觉暂留,为了产生运动模糊度的效果,可在像素点上画一适当长度的直线来进行模拟。

雨滴粒子的大小与雨的强度有关,通常呈正比关系。可根据雨的强度确定一合适的平均大小 $Meansize$ (以像素为单位),然后按照随机粒子系统的方法通过下式计算新生粒子的大小

$$Size() = Meansize + rand() \times Varsize \quad (3)$$

在显示雨滴时,以 $Size()$ 作为像素点大小和直线宽度。

雨滴粒子的颜色采用 RGBA 模型, R 、 G 、 B 分别表示颜色的红、绿、蓝 3 个组成分量, A 表示粒子的不透明度,用于表现当雨滴粒子与背景相融的效果, R 、 G 、 B 、 A 4 个分量的取值范围均为 0.0 到 1.0。雨滴主要由水组成,清水是无色透明的。由于雨滴中含有灰尘等极少量杂质,同时下雨时整个天空背景呈灰白色,雨滴粒子的颜色可设为灰白色, R 、 G 、 B

的值都可设在 0.7 或 0.8 附近;透明度较高,即表示不透明度的 A 较小,雨滴像素点部分的 A 约为 0.2,直线部分的 A 稍小,约为 0.15。

雨滴不同时刻的状态由粒子的动力学性质决定,如果不受外力作用,雨滴粒子将始终按初始速度匀速下降;当受到外力的作用时,按牛顿第二定律,粒子将产生一定的加速度。雨点在空中运动过程中受到 3 个力的作用:地球向下的引力、空气向上的阻力和旁向的风力。这里不考虑旁向风力的作用,地球引力产生一个向下的重力加速度,其值为一常量 $g \approx 9.8 \text{ m/s}^2$,雨粒子在下降过程中受空气阻挡后,其合成的加速度 a 在数值上小于 g 。雨滴的初始速度 V_0 用类似于上述确定大小的方法来确定;如果坐标原点在窗口左下角, y 轴向上,把窗口高度看作 1.0,则在 y 方向上的初始位置 P_0 根据下式确定:

$$P_0 = 1.0 - \text{Beginrand}() / \text{Meanlife} \quad (4)$$

$\text{Beginrand}()$ 为一个 0.0 到 1.0 之间的随机数。在雨滴下降过程中,按照牛顿力学定律,其速度 V 、位置 P 和加速度 a 有如下约束关系:

$$V = V_0 + \int a dt$$

$$P = P_0 + \int V dt$$

由于加速度接近一个常量,实际计算时可按等加速度运动的方法来实现离散情况下物体运动状态的简化计算,等加速度运动的计算方法如下所示:

$$V_n = V_{n-1} + a \Delta t \quad (5)$$

$$P_n = P_{n-1} + V_n \Delta t \quad (6)$$

其中, P_n 是时刻 n 的物体位置, V_n 是时刻 n 的物体速度, Δt 是离散的时间间隔,计算时取每两个连续帧图像之间的时间间隔。

所有雨滴粒子的形状、颜色和透明度都相同,并且从粒子的产生到消亡这一过程中一直不发生变化,可作为常量属性处理,每一个雨滴粒子只需记录位置、速度、大小、生存期等变量属性,因此可表示雨滴如下:

雨滴 = {位置, 速度, 大小, 生存期}

3 三维场地的降雨模拟

降雨模拟的相关场景包括起伏不平的山地和乌云笼罩的天空。山地的模拟通过对浙江涌江流域的黄土岭地区的 DEM(数字高程模型)进行三维建模、变换、设色和光照渲染等处理实现^[7];光照处理时用

较弱的光来模拟阴天,天空的形状可看作一个上下稍扁的半球面,用一个贝塞尔曲面模拟^[8]可使计算方便,曲面上贴上事先用分形方法产生的乌云的纹理图像,从而得到不包括降雨的场景图像。

雨粒子系统的实现流程如图 2 所示。

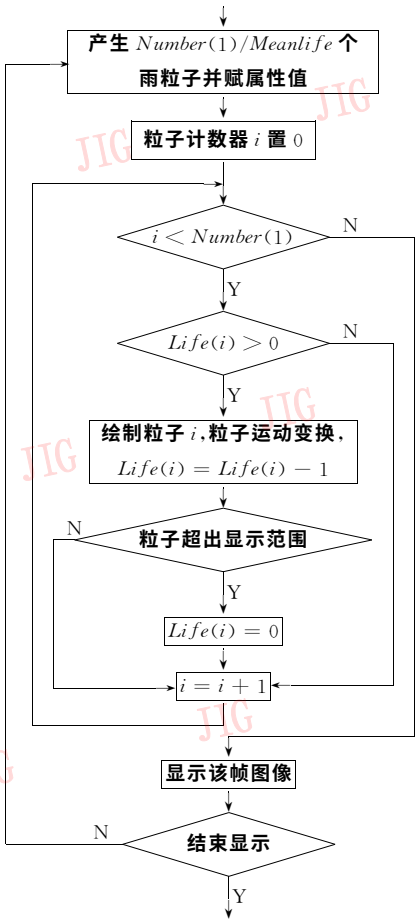


图 2 雨粒子系统实现流程

系统开始运行后,先在产生区域内生成 $Number(1)/Meanlife$ 个雨粒子,并按式(1)、式(3)、式(4)等式给这些粒子赋初始属性值,渲染所有存活粒子,得到第 1 帧图像;接下来每个存活粒子按式(5)、式(6)运动变换,生存期减 1,再检测其位置是否超出显示范围,如果条件成立,则应让该粒子消亡,把它的生存期置为 0;显示该帧图像后接着进行下一帧图像的粒子生成、渲染、变换及检测等处理。如此下去,直到接到停止显示的消息为止。

为了说明降雨过程,假定粒子的平均生存期为 $Meanlife = Height/Speed = 3$, 则每帧新产生 $Number(1)/3$ 个粒子,因此可把所有粒子分为 3 组,

对应的“雨场”也可分为上、中、下 3 层,最上层为产生区域。当渲染第 1 帧图像时,仅有第 1 组粒子被激活 ($Life(i) > 0$),这些粒子位于产生区域内,它们在 y 方向的位置 P_0 的取值范围为 $[2/3, 1.0]$ (图 3(a));渲染第 2 帧图像时,第 1 组粒子移动到中层,它们在 y 方向的位置 P 的取值范围为 $[1/3, 2/3]$,同时在产生区域中激活第 2 组粒子(图 3(b));渲染第 3 帧图像时,第 1 组粒子移动到下层,它们在 y 方向的位置 P 的取值范围为 $[0.0, 1/3]$,第 2 组粒子移动到中层,它们在 y 方向的位置 P 的取值范围为 $[1/3, 2/3]$,同时在产生区域中激活第 3 组粒子(图 3(c));渲染第 4 帧图像时,第 2、第 3 组粒子依次下移,第 1 组粒子被重新移到产生区域并赋予新的属性……,如此下去,便可得到一系列连续的降雨图像。

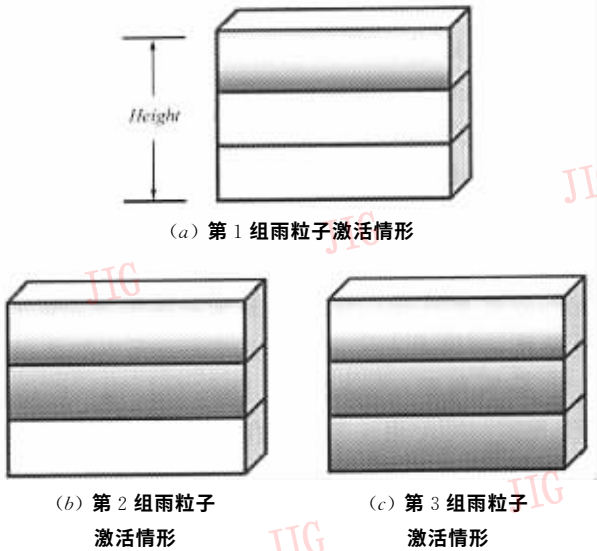


图 3 降雨过程示意图

上述移动是在平均速度的条件下进行的,而实际雨滴的下降是一种加速运动,由于各组粒子平均速度不相等(上层慢下层快),从理论上讲上层粒子图像和中层粒子图像之间、中层粒子图像和下层粒子图像之间都会出现不连续,但因为两个因素的影响使得这种断裂不会出现在计算机显示屏上,这两个因素,一是,雨滴在接近地面时速度较快,其速度的增量相对于初始速度来说较小,亦即上层粒子和中层粒子、中层粒子和下层粒子之间的速度差别较小;二是,各粒子的初始速度并不完全相同,而是在平均速度上下波动,因而上层中部分粒子的速度可能达到甚至超过中层中部分粒子的速度,中层和下层之间也是如此。

使用粒子系统模拟自然景物,通常由于粒子数巨大,计算耗时,多数情形下是先生成仅包含粒子的

图像(如在渲染存活粒子图像的时候假定粒子不与基于表面构型的图元(surface-based modeling primitives)相交^[1]),再与其他景物的图像合成,是一种滞后合成。文献[5]虽然介绍了一些加速生成图像的方法,但提供的图像上也没有包含其他景物。要即时生成可交互漫游的山地场景,因此必须实时检测雨滴的高度,不能让它们穿透地面,影响图像的真实感。由于山地起伏不平,雨滴可降落的高度各不一样,可根据雨滴的平面坐标得到对应于该平面坐标的地面高程,用雨滴的高度和该地面高程进行比较,便可判断雨滴是否到达地面。假定场景的平面坐标进行了归一化,如果把所有雨粒子看成 m 行 n 列的矩形点阵($m \times n = \text{Number}()$),则第 i 个粒子的在产生区域中的平面坐标为

$$X' = (i \% n) / n \times q$$
$$Z' = (i / n) / m \times q$$

其中, q 为缩放系数,它的作用是把产生区域缩放为视图体顶部的外接长方体。如果视线方向与水平基准方向的夹角为 θ ,视点的平面坐标为 $(X_{\text{eye}}, Z_{\text{eye}})$,则雨滴 i 在场景中的平面坐标为

$$X = X' \times \cos(\theta) + Z' \times \sin(\theta) + X_{\text{eye}}$$
$$Z = X' \times \sin(\theta) + Z' \times \cos(\theta) + Z_{\text{eye}}$$

如果地形场景为 $\text{Row} \times \text{Col}$ 的 DEM,则雨滴所对应的地面点的点号为

$$k = Z \times \text{Row} \times \text{Col} + X \times \text{Col}$$

在 DEM 中提取第 k 点的高程和雨滴的高度进行比较便可确定雨滴是否应该消亡。

4 模拟试验

在 CPU 为 P4 1.8G,内存为 256M,显存为 64M 的微机上进行,通过在 Visual C++ 6.0 中调用 OpenGL 来实现降雨过程的模拟。显示窗口大小为 600 像素 $\times$ 600 像素,使用了 1 600 个粒子。计算机仅仅计算渲染 1 600 个雨粒子的图像能达到约为 54.5f/s 的速率,仅仅计算渲染显示 512 $\times$ 512 的山地纹理(567 $\times$ 543 个高程控制点)图像能达到约为 6.4f/s 的速率,包括雨粒子在内的全部场景同时渲染的速率约为 5.8f/s(在 SGI Onyx II 图形计算机上计算渲染 2 000 个灰尘粒子的图像的速率为 4f/s 多^[6])。由于计算机软硬件性能较好,粒子数较少(Reeves 曾用了多达 750 000 个粒子渲染火焰图像^[1]),粒子形状简单,虽然不能达到 24f/s 以上的

帧速率,基本上能满足实时交互的要求。

图 4 为不包括降雨的地形场景,图 5 为平均粒子大小为 2 个像素场景图像,感觉象在下小雨;平均粒子大小增加到 3 个像素后,场景中雨的强度也在增大(图 6)。为了增加真实感,渲染图 5 及图 6 场景时打开了雾化功能。

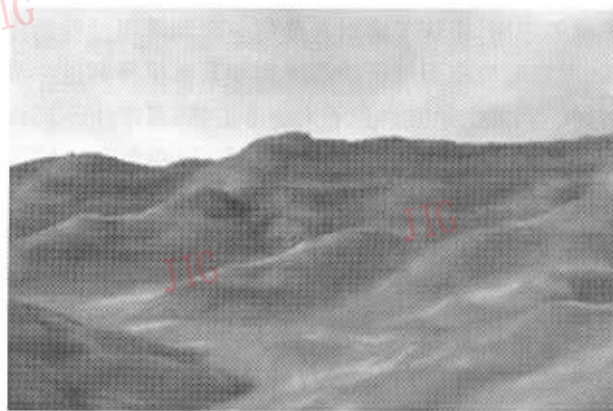


图 4 不包括降雨的地形场景



图 5 平均粒子大小为 2 个像素的降雨场景

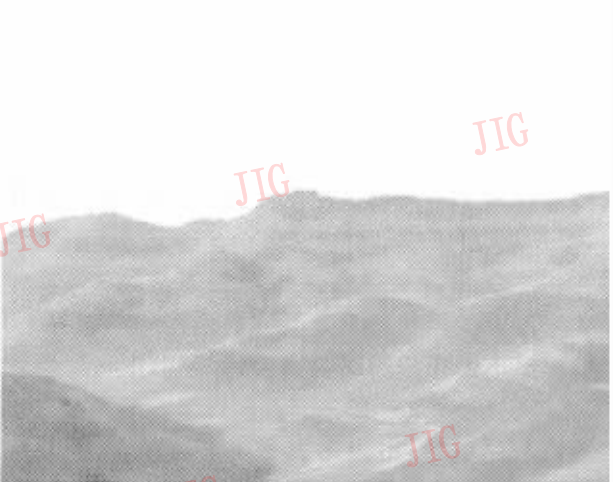


图 6 平均粒子大小为 3 个像素的降雨场景

5 结 论

本文用粒子系统模拟三维场地降雨,降雨过程连续,能进行场景交互漫游,具有一定的真实感。由于粒子的产生区域限制在视图体顶部的外接长方体内,所需的粒子数大大减少,这为实时显示提供了保证。粒子组的应用,保证了降雨密度的一致和降雨过程的连续,而实时检测雨粒子的高度则使真实感得到进一步加强。调整粒子数和粒子大小等参数,可产生不同强度的降雨图像,如果同时再调整形状、颜色和透明度等属性参数,还可用来模拟下雪场景。从试验中可看出,地形场景的渲染是导致帧速率低的主要因素,如果对地形场景数据进行适当简化,可进一步提高帧速率。本文中对雨粒子的受力情况作了简化,风力和空气阻力对雨粒子的作用有待进一步研究。

参 考 文 献

- 1 Reeves William T. Particle systems—a technique for modeling a class of fuzzy objects[J]. Computer Graphics, 1983, 17(3): 359~376.
- 2 Reeves William T, Blau Ricki. Approximate and probabilistic algorithms for shading and rendering structured particle systems[J]. Computer Graphics, 1985, 19(3): 313~322.
- 3 Reynolds Craig W. Flocks, herds, and schools: A distributed behavioral model[J]. Computer Graphics, 1987, 21(4): 25~34.
- 4 Szeliski Richard, Tonnesen David. Surface modeling with oriented particle systems[J]. Computer Graphics, 1992, 26(4): 185~194.
- 5 谢剑斌,郝建新,蔡宣平等. 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): 734~738.

- 6 Chen Jim X, Fu Xiaodong, Wegman Edward J. Real-time simulation of dust behavior generated by a fast traveling vehicle [J]. Transactions on Modeling and Computer Simulation, 1999, 9(2): 81~104.
- 7 薛安,马谡乃,李天宏. 基于 OpenGL 实现真实感地形表现的研究[J]. 中国图象图形学报, 2001, 6(8): 800~805.
- 8 米剑,刘鹏远,张锡恩等. 视景仿真中天空背景的真实感改进[J]. 系统仿真学报, 2001, 13(11): 220~223.



罗维佳 1971 年生,1995 年本科毕业于原武汉测绘科技大学摄影测量与遥感专业,现为南京大学城市与资源学系地图学与地理信息系统专业硕士生,玉溪师范学院教师。主要研究方向为地理信息系统与地学可视化等。



都金康 1964 年,副教授,1988 年于南京大学自然地理学专业硕士毕业。主要从事遥感及 GIS 在水文与水资源中的应用研究。已发表研究论文 20 余篇。



谢顺平 1957 年生,高级工程师,1981 年毕业于南京大学计算机应用专业。主要从事地理信息系统的研究与应用开发工作。已在国内学术刊物上发表研究论文 20 余篇。