

基于 GPU 的屏幕空间镜头水滴实时渲染

叶万方^{1,2}, 邵鑫¹, 黄煜³, 张建伟^{1,2}

(1. 四川大学 计算机学院, 四川 成都 610065;

2. 视觉合成图形图像技术国防重点学科实验室, 四川 成都 610065;

3. 四川川大智胜软件股份有限公司, 四川 成都 610045)

摘要:为了实时模拟真实性高的镜头水滴,提出一种在可编程图形硬件中实现的镜头水滴渲染新方法。首先三维场景被渲染到一张场景纹理,然后在 GPU 着色器中为镜头水滴产生不规则边缘接触曲线,使用该曲线快速构建水滴的曲面,最后采用曲面的表面信息并根据水滴的光学属性渲染出水滴的折射效果。采用该方法,可以实时地在屏幕上渲染出镜头水滴效果。使用 GPU 着色器进行渲染,可以在渲染出效果逼真的水滴的情况下,获得实时的帧率。采用光线折射物理方法渲染出的水滴的效果比直接使用纹理贴图方式获得的水滴的效果更逼真。

关键词:实时渲染;水滴;屏幕空间;自然特效;着色语言

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2014)05-0011-04

doi:10.3969/j.issn.1673-629X.2014.05.003

Real-time Rendering of Screen-space Water Drops for Camera Lens Based on GPU

YE Wan-fang^{1,2}, SHAO Xin¹, HUANG Yu³, ZHANG Jian-wei^{1,2}

(1. College of Computer Science, Sichuan University, Chengdu 610065, China;

2. Key Laboratory of Fundamental Science on Synthetic Vision, Sichuan University, Chengdu 610065, China;

3. Wisefsoft Co., Ltd. of Sichuan University, Chengdu 610045, China)

Abstract: One new approach was proposed to simulate realistic water drops on lens in real time using the programmable graphics hardware. First, the three-dimensional scene was rendered to a scene texture. Then the irregular edge curves of water drops on the screen were produced in the GPU shader. Surfaces of water drops were constructed quickly by using the curves. Lastly the refraction effect of water drops was rendered by using the information of the surface based on the optical properties of droplets. By using this method, can render the lens effect of water droplets on the screen with real-time. Using the GPU shader for rendering, can obtain real-time frame rate under the condition of rendering the realistic water droplets. The droplets effect rendered by light refraction physical methods is more realistic than by texture mapping methods.

Key words: real-time rendering; water drops; screen space; natural effects; shading language

0 引言

在实时应用中,比起逼真性通常更优先的考虑渲染速率。照片级的基于物理公式的自然现象模拟会牺牲大量的渲染效率。因此通常在实时应用中,在满足渲染速率的前提下应尽量去获得更加逼真的自然现象模拟。

子系统可以逼真地模拟下雪特效^[1]。使用预计算的雨滴形状和法向量,考虑光线折射、场景中光源对雨

滴的影响和视觉停留来进行雨滴的渲染,可以获得实时而又较逼真的雨滴渲染效果^[2]。雨滴滴落在镜头玻璃上的效果可以增强下雨特效的真实随机的边缘曲线,而且对场景具有折射扭曲使用不透明的漫反射材质和简单的粒的效果。基于物理的方法^[3],用于在表面模拟低运动速率的水滴,并能够产生精确的结果,但是需要高计算耗时。Kaneda 等实现了一种玻璃平面上水滴的动画^[4]。之后, Kaneda 等又将其扩展为曲面上的水滴动画^[5]。使用 SPH^[6]来模拟该特效虽然可以

收稿日期:2013-07-12

修回日期:2013-10-17

网络出版时间:2014-02-11

基金项目:国家“863”高技术发展计划项目(2012AA011804)

作者简介:叶万方(1987-),男,重庆人,硕士研究生,研究方向为计算机图形学。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140211.1619.048.html>

达到较高视觉效果,但是却很难在普通计算机硬件平台上达到实时渲染。采用变形表面模型模拟小规模的水滴^[7],涉及到平均曲率流的计算,渲染效果很好,并且可以在配置很好的大型工作站上达到实时。使用更简单的水滴光线扭曲模型^[8]来扭曲平面网格,需要将水滴的形状等信息存入纹理中,因此渲染效果受纹理分辨率的限制。将水滴的法向量和高度信息存入纹理中,通过移动纹理中水滴的量来模拟表面水滴的移动^[9]。用一张纹理存储多个水滴的高度和水滴曲面法向量信息^[10],可以简化计算量,但需要预计算高度和曲面法向量信息,而这个过程比较难准确的实现,最后的渲染效果会因为高度和法向量的精度不够高而显得效果不佳。基于物理的树枝和雨滴的交互作用可以模拟水滴在树叶表面流动和合并等现象^[11]。在重力作用下使用 Metaball 对水滴进行渲染的新技术被提出^[12]。使用类分子模型模拟光滑和粗糙水平表面上的水滴^[13],由于使用了大量的粒子间的相互作用来计算水滴外形,所以计算量较大,不适合多个水滴的实时渲染。因此 Beard 等提出了一种雨滴平衡形态的新模型^[14]。

1 水滴模型

可以通过简单的方法快速构建近似的水滴曲面信息,获得视觉上合理的水滴效果。

1.1 构建不规则接触曲线

以水滴落在垂直放置的玻璃上为例,如图 1。考虑静态的水滴,其具有不规则的接触曲线。由于其受张力的作用,所以形成的闭合不规则曲线的尺寸大小是受限的。



图 1 真实拍摄的位于垂直玻璃表面上的水滴
(相机和水滴分别在玻璃内外侧)

考虑到接触曲线的连续性和不规则性,采用具有类似性质的 Simplex 噪声^[15-16]来快速生成该曲线。该模型如图 2 所示。

在屏幕空间中,水滴在屏幕玻璃相对于观察者的另一侧表面上。建立以屏幕左下角为原点的二维坐标

系,屏幕右方和上方分别为坐标系 x 轴正方向和 y 轴正方向。屏幕上可以有多个水滴,以其中任意一个水滴为例,标记用于表示该水滴位置的中心点为 $O(x_o, y_o)$ 。在 GPU 的片元着色阶段,需要遍历屏幕的每个片元,因此标记屏幕上任意一个片元位置为 $B(x_B, y_B)$ 。

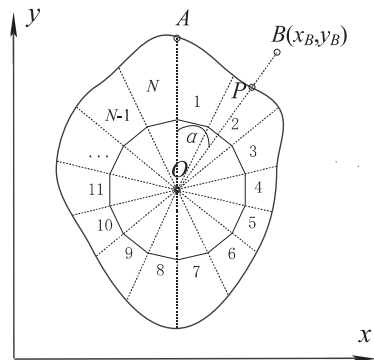


图 2 连续闭合不规则的水滴接触线模型

图 2 中 A 点所在不规则闭合曲线是待求的水滴接触曲线,为了求出该曲线,只需要判断任意的 B 点是否在该曲线之内。通过使用向量 OB 与 y 轴正方向的夹角关系,来计算向量 OB 方向上曲线的半径 OP (如图 2 所示, P 点为向量 OB 与以 O 点为标记中心的待求曲线的交点)。

标记该曲线为 $f(\varphi)$,其中 φ 对应着图 2 所示的 OB 与 y 轴正方向的夹角 α 。 α 的范围为 $[0, 2\pi)$ (按照顺时针方向增加),对应的 φ 的范围为 $[0, 1)$ 。因此,曲线 $f(\varphi)$ 表示的值是自变量 φ 表示的方向上曲线的半径长度,即图 2 中 OP 的长度。在 OB 这个向量方向上的水滴曲面上的曲线,到屏幕空间的投影表示为 OP 。 φ 的计算如公式(1)

$$\varphi = \begin{cases} \frac{y_B - y_o + 3}{4\sqrt{(x_B - x_o)^2 + (y_B - y_o)^2}}, & x_B - x_o < 0 \\ -\frac{y_B - y_o + 1}{4\sqrt{(x_B - x_o)^2 + (y_B - y_o)^2}}, & x_B - x_o \geq 0 \end{cases} \quad (1)$$

其中, x_o, y_o 为中心点 $O(x_o, y_o)$ 的片元位置坐标值; x_B, y_B 表示遍历到任一片元的位置 $B(x_B, y_B)$ 。

使用下面的公式来计算 $f(\varphi)$

$$f(\varphi) = S(\varphi) * \varepsilon + r \quad (2)$$

其中, $S(\varphi)$ 表示一维 Simplex 噪声函数; ε 表示一个可以调节的缩放因子常量; r 表示标准圆的半径,是可以调节的待定常量。

公式(2)表示在标准圆半径上加上一个连续随机变化的增量修正值,最后得到不规则的边缘曲线函数值。

该实现过程利用了 Simplex 噪声的连续性和随机

性。Simplex 噪声是在 Perlin 噪声^[17]上的改进,有效地降低了插值次数,使用 $6t^5 - 15t^4 + 10t^3$ 作为插值函数,杜绝了 Perlin 噪声中的线性部分。所求的 $f(\varphi)$ 具有至少二阶导数连续性。

1.2 水滴曲面构造

在得到接触曲线的情况下,如图 2 中的 A 点所在的闭合曲线,可以通过中心点 O 与接触曲线上的一点 P 来计算该水滴曲面上对屏幕的投影为 OP 的曲线。该曲线计算过程如图 3 所示。

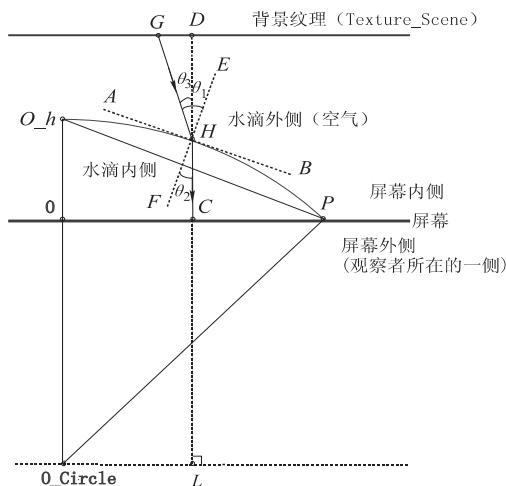


图3 光线折射计算

图3中的H点所在的弧线表示近似的水滴曲面上的曲线。在知道OP和水滴中心高度(是用户定义的常量,即O点到O_h点的距离h)的情况下,可以近似得到一个圆弧 Arc(O_h, P)。如图3,水平的O点和P点所在的粗线表示屏幕,该线上侧表示屏幕内侧,是屏幕上水滴所在的一侧;该线下侧表示屏幕外侧,是虚拟场景中相机所在的一侧。延长由O_h和O构成的线段到O_Circle,使线段 Line(O_h, O_Circle)与线段 Line(P, O_Circle)相等,即点O_h和点P在以虚拟点O_Circle为圆心,线段 Line(O_h, O_Circle)的长度为半径的圆弧上。并且该圆弧在点O_h的切线与屏幕线平行。这保证了O_h是圆弧上的最高点,并且在水滴曲面经线方向上的曲线有一阶导数连续性。

图3表示在水滴未受重力的情况下,对水滴曲面的近似模拟。图中H是圆弧 Arc(O_h, P)上一点,AB过H与该弧线相切,CD过H垂直于OP和G与D点所在的背景纹理线;EF过H并垂直于AB;θ₁为GH和EH的夹角,即入射角;θ₂为HF和HC的夹角,即出射角;θ₃为HG与HD的夹角。实际中屏幕应该是垂直的。如果考虑重力,设重力在图中的左方向。则曲线将按照一定方式向左偏移。在实际中,只需要将水滴中心向重力方向移动适当位置即可。为了实现水滴的各向异性,可以简单地将水滴在x轴方向上适当压缩,在垂直方向上适当拉伸。

1.3 光线折射渲染计算

在进行水滴的光线折射渲染时,需要得到屏幕上各个像素的颜色值。如图3所示,为了得到屏幕上像素所在位置点C处的颜色值,如果C点不在水滴内侧,那么C点的值直接为背景纹理(即将虚拟场景渲染到纹理所获得的场景纹理)中对应的D点的颜色值。图3中,圆弧 Arc(O_h, P)表示的是水滴隐式曲面上的一条弧线,该弧线下方到屏幕线OP的区域是水滴内侧,该弧线上方到背景纹理的区域是水滴外侧(空气),根据光线折射的原理,C点的颜色值应为背景纹理上G点颜色值。其中满足:

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{n_2}{n_1} \quad (3)$$

其中,θ₁和θ₂分别为光线从空气射入水滴的入射角和折射角;n₁和n₂分别为空气和水的折射率。

为了得到G点处的颜色值,需要计算G点相对于D点的偏移量GD。

对于镜头玻璃上的水滴的情况,折射将会会聚到镜头感光元件所在位置点。这里为简化计算,使透过水滴的折射光线垂直于屏幕。

在实际计算中,OC长度是已知的,CD的长度是用户设定的常量,设线段 Line(O, O_h)的长度为h,h是可以调节的常量。因此线段 Line(O_Circle, P)的长度,即 Arc(O_h, P)的半径R为

$$R = \frac{h^2 + \tau^2}{2h} \quad (4)$$

其中,τ表示OP的长度,要求τ不小于h。

经推导,GD的计算如公式(5):

$$\delta = (\gamma - (\sqrt{R^2 - \kappa^2} - (R - h))) * \tan(\theta_3) \quad (5)$$

$$\tan(\theta_2) = \frac{\kappa}{\sqrt{R^2 - \kappa^2}} \quad (6)$$

其中,δ、γ、κ分别表示GD、CD和OC的长度。

由公式(3)和公式(6)可求得tan(θ₃),其中θ₃=θ₁-θ₂,如图3所示。使用偏移值GD和O点、C点在屏幕空间的位置,可以在背景纹理中采样取得G点的颜色值,该颜色值便是从G点开始的光线经折射后到达屏幕中C点的颜色值。

2 在 GPU 中的渲染流程

渲染流程如图4所示。首先,整个3D场景通过GPU的着色语言被渲染到场景纹理 Texture_Scene,将 Texture_Scene 传入屏幕水滴渲染遍 Pass_WaterDropRender。并还向 Pass_WaterDropRender 中传入了一张随机数噪声纹理 Texture_RandomNoise 和一张 Simplex 噪声纹理 Texture_SimplexNoise(预先将S(φ)的函数值存入二维纹理中,纹理的每一行对应一个一维 Sim-

plex 噪声函数)。在水滴渲染遍 Pass_WaterDropRender 中使用 Texture_RandomNoise 计算水滴在屏幕上的随机位置,使用 Texture_SimplexNoise 生成每个水滴的不

规则接触曲线,再进行水滴曲面近似计算,最后进行光线折射渲染计算,得到最终屏幕水滴渲染效果。另外,需要控制水滴的最大显示数量和水滴的产生与消退。

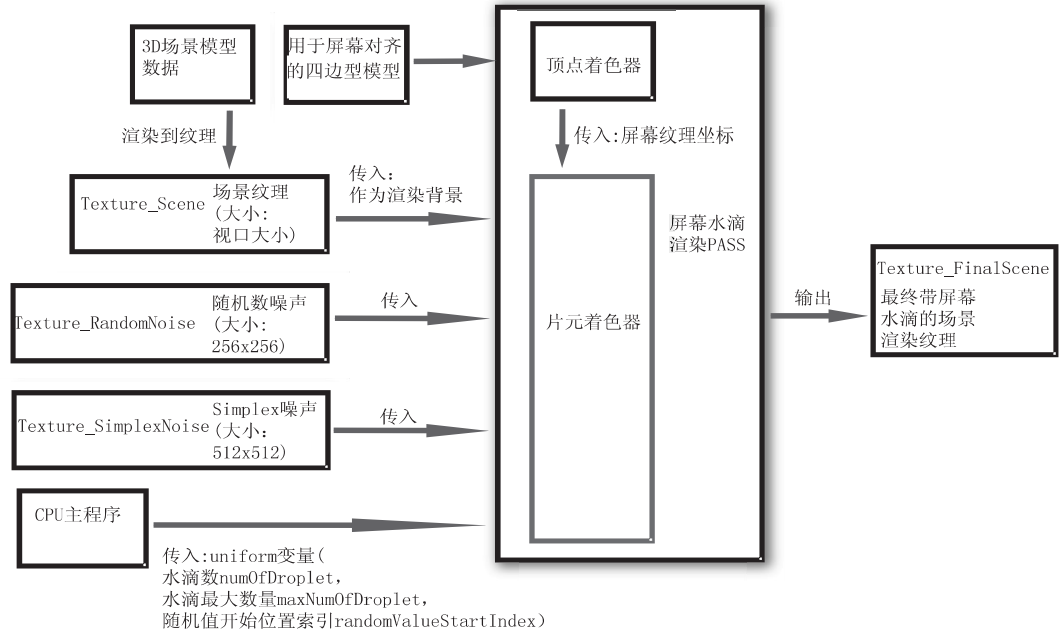


图 4 屏幕空间水滴渲染流程

3 结果和分析

整个渲染过程是在 Windows 7 系统中的 Visual Studio 2008 平台上使用 C++、OpenGL 和 GLSL 编程实现。硬件配置环境为: 显卡 (NVIDIA GeForce GTX 470), CPU (Intel® Core i7 930 @ 2.80 GHz), 内存 (6 144 MB RAM)。

从图 5 中的帧率与水滴数的变化关系曲线可以看出,在 0 ~ 20 个水滴数区间,帧率变化缓慢;在 20 ~ 100 个水滴数区间,帧率近似于线性下降并逐渐平缓。帧率线性下降的原因是在片元着色中使用了一个用于判断当前片元在哪个水滴内的循环,该循环遍历了每个水滴。

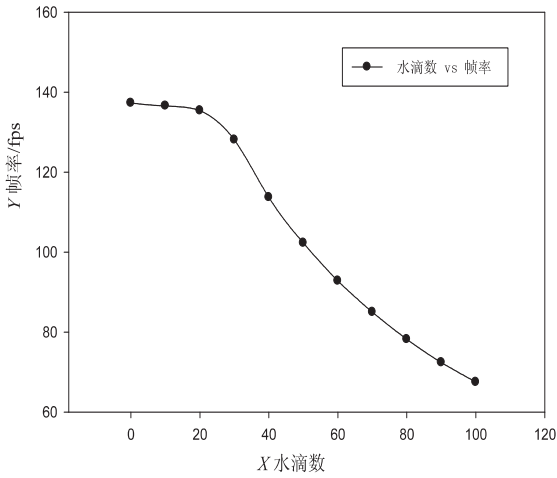


图 5 水滴数与帧率的关系图

最终的屏幕空间水滴特效如图 6 所示。



(a)单个水滴效果 (b)水滴数为 100

图 6 屏幕空间镜头水滴渲染效果

4 结束语

文中提出了一种渲染屏幕空间镜头水滴特效的新方法。该方法快速生成水滴不规则接触曲线,简化了水滴曲面计算,并能实时渲染出视觉效果很好的屏幕空间镜头水滴特效。

期望在将来的工作中针对片元着色阶段的循环进行优化,以此进一步提高现有算法的效率。并为水滴添加向下流动的效果,增加镜头水滴特效的真实性。

参考文献:

[1] Langer M S, Zhang L, Klein A W, et al. A spectral-particle hybrid method for rendering falling snow[C]//Proc of the fifteenth Eurographics conference on rendering techniques. Aire-la-Ville, Switzerland: Eurographics Association, 2004: 217-226.

[2] Rousseau P, Jolivet V, Ghazanfarpour D. Realistic real-time

确、颜色变化剧烈的图像分割效果较差。同时,由于图像分割本身也存在相当大的复杂性,想要获得不同类型图像较好的分割效果还非常困难,相关理论和方法还需要进一步的研究。

参考文献:

[1] 黄长专,王 彪,杨 忠. 图像分割方法研究[J]. 计算机技术与发展,2009,19(6):76-79.

[2] 范静辉,吴建华,刘 晔. 基于矢量量化和区域生长的彩色图像分割新算法[J]. 中国图象图形学报,2005,10(9):1079-1081.

[3] Deng Yining, Manjunath B S. Unsupervised segmentation of color-texture regions in images and video[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2001,23(8):800-810.

[4] Shi J, Malik J. Normalized cuts and image segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2000,22(8):888-905.

[5] 孙卫芳. 基于JSEG的视频/图像分割技术研究[D]. 郑州: 郑州大学,2010.

[6] Zheng Yuanjie, Yang Jie, Zhou Yue. Unsupervised segmentation on image with JSEG using soft class map[J]. Lecture Notes in Computer Science,2004,3177:197-202.

(上接第14页)

rain rendering[J]. Computers & Graphics,2006,30(4):507-518.

[3] Sato T, Dobashi Y, Yamamoto T. A method for real-time rendering of water droplets taking into account interactive depth of field effects[J]. Entertainment Computing,2003,112:125-132.

[4] Kaneda K, Kagawa T, Yamashita H. Animation of water droplets on a glass plate[C]//Proceedings of computer animation. [s. l.]:[s. n.],1993:177-189.

[5] Kaneda K, Ikeda S, Yamashita H. Animation of water droplets moving down a surface[J]. Journal of Visualization and Computer Animation,1999,10(1):15-26.

[6] Gingold R A, Monaghan J J. Smoothed particle hydrodynamics - theory and application to non-spherical stars[J]. Monthly Notices of the Royal Astronomical Society,1977,181:375-389.

[7] Zhang Yizhong, Wang Huamin, Wang Shuai, et al. A deformable surface model for real-time water drop animation[J]. IEEE Transactions on Visualization and Computer Graphics,2012,18(8):1281-1289.

[8] Yang Yonggao, Zhu Changqian, Zhang Hua. Real-time simulation: water droplets on glass windows[J]. Computing in Science & Engineering,2004,6(4):69-73.

[9] Stuppacher I, Supan P. Rendering of water drops in real-time

[7] Chen H C, Chien W J, Wang S J. Contrast-based color image segmentation[J]. IEEE Signal Processing Letters,2004,11(7):641-644.

[8] Wang Yonggang, Yang Jie, Chang Yuchou. Color-texture image segmentation by integrating directional operators into JSEG method[J]. Pattern Recognition Letters,2006,27(16):1983-1990.

[9] Kibria A F M G, Islam M M. Color image segmentation using visible color difference and Canny edge detector[C]//Proc of 15th international conference on computer and information technology. Chittagong:[s. n.],2012:138-143.

[10] Komati K S, Salles E O T, Sarcinelli-Filho M. Unsupervised color image segmentation based on local fractal descriptor and J-images[C]//Proc of IEEE international conference on industrial technology. Vi a del Mar:[s. n.],2010:303-308.

[11] 刘建龙. 基于图论的图像分割算法研究[D]. 哈尔滨: 哈尔滨工业大学,2006.

[12] 林开颜, 吴军辉, 陈立鸿. 彩色图像分割方法综述[J]. 中国图象图形学报,2005,10(1):1-10.

[13] 周爱武, 于亚飞. K-Means 聚类算法的研究[J]. 计算机技术与发展,2011,21(2):62-65.

[14] 郑洪英. 数据挖掘聚类算法的分析和应用研究[D]. 重庆: 重庆大学,2002.

[C]//Proc of Central European seminar on computer graphics for students. [s. l.]:[s. n.],2007.

[10] Fernando R, Pharr M. GPU gems: programming techniques, tips, and tricks for real-time graphics[M]. [s. l.]: Addison-Wesley,2004.

[11] 杨 猛, 吴恩华. 一种基于物理的雨打树枝动态仿真方法[J]. 软件学报,2011,22(8):1934-1937.

[12] Yu Y J, Jung H Y, Cho H G. A new rendering technique for water droplet using metaball in the gravitation force[C]//Proc of 6th international conference in Central Europe on computer graphics and visualization. [s. l.]:[s. n.],1998.

[13] Kulsri S, Jaroensutasinee M, Jaroensutasinee K. Simulation of water droplet on horizontally smooth and rough surfaces using quasi-molecular modelling[J]. International Journal of Applied Science, Engineering and Technology,2006,2(3):136-140.

[14] Beard K V, Chuang C. A new model for the equilibrium shape of raindrops[J]. Journal of Atmospheric Sciences,1987,44(11):1509-1524.

[15] Perlin K. Improving noise[J]. ACM Transactions on Graphics,2002,21(3):681-682.

[16] Gustavson S. Simplex noise demystified[D]. Sweden: Linköping University,2005.

[17] Perlin K. An image synthesizer[J]. ACM SIGGRAPH Computer Graphics,1985,19(3):287-296.

基于GPU的屏幕空间镜头水滴实时渲染


作者：

[叶万方](#)，[邵鑫](#)，[黄煜](#)，[张建伟](#)，[YE Wan-fang](#)，[SHAO Xin](#)，[HUANG Yu](#)，[ZHANG Jian-wei](#)

作者单位：

[叶万方, 张建伟, YE Wan-fang, ZHANG Jian-wei\(四川大学 计算机学院, 四川 成都 610065; 视觉合成图形图像技术国防重点学科实验室, 四川 成都 610065\)](#)，[邵鑫, SHAO Xin\(四川大学 计算机学院, 四川 成都, 610065\)](#)，[黄煜, HUANG Yu\(四川川大智胜软件股份有限公司, 四川 成都, 610045\)](#)

刊名：

[计算机技术与发展](#)

英文刊名：

[Computer Technology and Development](#)

年，卷(期)：

2014(5)

本文链接：http://d.wanfangdata.com.cn/Periodical_wjfz201405003.aspx