一种基于 GPU 的柔和阴影实现方法研究

付燕丽,刘 循
(四川大学 计算机学院,四川 成都 610064)

摘 要:阴影是虚拟环境中真实感的重要特征。由于并行分割的阴影贴图算法的通用性和效率,它对大规模复杂的虚拟环境中的实时阴影渲染起了重要作用。但是对于塔台仿真系统这样的大规模动态场景,锯齿现象依然存在,利用差值阴影贴图等方法,对其效果进行了改进,并利用 GPU 强大的浮点数运算能力和并行处理能力加速了阴影的实现,有效地保证了阴影的真实性和实时性。实验结果表明生成的阴影边缘柔和,抗锯齿效果好,有效地解决了偏离和光渗问题,达到了实时仿真的要求。

关键词:阴影图;反走样;图形处理单元;大规模场景;并行分割的阴影贴图;差值阴影贴图
中图分类号:TP391.41 **文献标识码**:A **文章编号**:1673-629X(2013)02-0052-05
doi:10.3969/j.jssn.1673-629X.2013.02.013

Research on Realizing Method of Soft Shadow Based on GPU

FU Yan-li,LIU Xun
(College of Computer,Sichuan University,Chengdu 610064,China)

Abstract:The shadow is the important characteristics of reality in the virtual environment. Because of the universal and efficiency,parallel split shadow maps algorithm plays an important role to the real-time shadow rendering in the large scale and complicated virtual environment. But to tower simulation system which is large-scale dynamic scene,alias is still happened,using such as variance shadow maps to improve its effect,and using the powerful floating-point operation ability and parallel processing ability of GPU to accelerate the realization of shadow,so it can effectively guarantee the authenticity and real-time of the shadow. The experimental results show that the shadow generated has soft edge,well anti-aliasing effect,and solving the problem of shadow biasing and light bleeding,achieving the requirements of real-time simulation.

Key words:shadow mapping;anti-aliasing;GPU;large scale environment;PSSMs;VSM

0 引 言

在虚拟环境中,阴影的主要作用表现在它保证了虚拟环境的真实感,反映了物体表面的几何特征,还提供了人们判断物体空间位置的重要视觉线索。近年来随着计算机图形硬件的发展,特别是 GPU(图形处理单元)的出现,使各种阴影的改进算法层出不穷。目前流行的阴影算法主要分为全局光照算法、基于物体的方法和基于图像的方法,下面将就这些算法的特点做概要的分析[1]。

全局光照模型相对于局部光照模型而言,它最显著的特征就是不仅要考虑直接来自于光源的光照,还需考虑整个场景对当前着色点的影响。这些影响包括了反射、透明物体的折射、半透明物体的表面散射及物体与光源间有遮挡物时产生的阴影等效果。全局光照主要的算法有光线跟踪算法(RayTracing)与辐射度算法(Radiosity),其后又出现了光子影射算法(Photo Mapping)。在 2009 年 8 月的 SIGGRAPH 会议上,Crytek 公司发布了一种名为 light propagation volumes 的近似计算全局光照的算法,该算法能够在硬件配置较低的平台上实时模拟出漫反射光对其他物体的作用,它联合阴影贴图算法和 SSAO(Screen-Space Ambient Occlusion,屏幕空间环境光遮蔽)等算法能够获得非常逼真的视觉效果。

基于物体的算法以 Crow 的阴影体算法为代表。阴影体算法的绘制效率不独立于场景中模型的几何复杂度,在阴影测试过程中需要绘制大量三角面,从而占用了大量的像素填充率,而且因为对场景中模型的数据构成有着严格的要求,导致了适用范围的受限。尽管随着鲁棒算法的发展和更好的图形硬件的支持,有了一定的实用[2~4],但对复杂的场景应用的实时性和

到稿日期:2012-05-16;**修回日期**:2012-08-20
基金项目:国家重点基础研究发展计划(2009CB320803)
作者简介:付燕丽(1987-),女,硕士研究生,主要研究领域为计算机图形图像;刘 循,博士,副教授,硕士研究生导师,主要研究领域为计算机网络、计算机图像处理。

实用性还有距离。

基于图像的算法中最具代表性的就是文中将要阐述的由 Williams 在 1978 年提出的阴影贴图算法[5]。阴影贴图算法是基于图像的,该算法计算效率高、对虚拟场景的适应性好,但同时存在着走样、柔和阴影实现困难等问题,克服这些缺点就成为了阴影绘制的一个重要研究内容。目前,人们关于阴影贴图算法的研究方向主要有两方面,首先是硬阴影计算,目标是实时绘制出理想光源下边界锐利的硬阴影,主要包括阴影反走样技术、全方向光源下的阴影绘制技术等。另一方面是柔和阴影计算,目标是模拟扩展光源下的阴影效果,关键是平衡效果和效率。由于硬阴影具有二值性,生成的效果不符合自然界中的实际阴影效果,而柔和阴影在半影区域中有明暗过渡地带,产生的阴影非常细腻,体现了环境遮挡作用下物体的关系,更适用于大规模场景中的阴影仿真。

1　基本阴影图算法及其优缺点

相比于其它阴影算法,阴影图算法具有以下优势:

1)独立于场景复杂度。阴影图算法是一种基于图像的技术,本质上是一张二维纹理存储了场景的深度信息,它的复杂度仅取决于光源数目和阴影图的分辨率。

2)适合用 GPU 硬件实现。GPU 是一种专门用于处理图形的硬件,它具有强大的浮点运算能力和并行处理能力,特别适用于处理大数据量的运算。目前的 GPU 提供了高度的可编程性,所以一般的阴影图算法可以移植到 GPU 上,从而提高渲染速率。

3)实现简单,适用性好。阴影图算法不需要预计算,不限制模型图元模式,对软硬件设备没有特别要求。

同时,阴影图算法也存在着缺点,最典型的就是走样问题。深度比较时对阴影图进行了采样,如果阴影图的分辨率低于采样需求,就会产生走样,其典型的表现为锯齿状的阴影边界。还有一种是受各类数值精度影响而产生的自阴影走样,包括缓存本身的精度和各种转换的数值精度。光源采样和屏幕采样的偏差也会产生这种问题,其典型表现为阴影中一些斑驳色块,即光渗现象。针对这些走样问题提出了很多改进的阴影图算法,参见文献[1,6]。

文中重点讨论大型场景中针对太阳光实现的阴影。由于太阳光的范围照射非常大,一般情况下,一张简单的阴影图无法保存所有的深度值,但是通过一些改进方法可以即保证阴影图算法的优点同时解决这个问题,这使得阴影图算法成为事实上的标准算法。文中重点讨论大型场景中针对太阳光实现的阴影。

2　阴影图算法实现

根据走样的原理和理论分析详见文献[7~9],在方向光源的作用下,如图 1 所示:

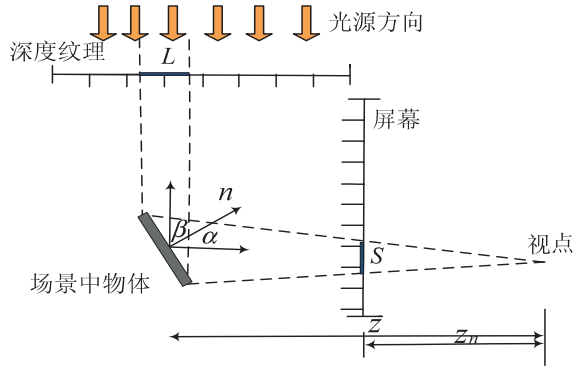

图 1　方向光源下的阴影走样分析

对于小的平面,通过计算得到屏幕对深度纹理的采样精度是 $w=\frac{S}{L}\approx\frac{t_l}{t_s}\frac{z_n}{z}\frac{\cos\alpha}{\cos\beta}$,其中 t_s 和 t_l 为屏幕和深度纹理在图中所示方向上的尺寸,α 为视线与物体表面向量 n 的夹角,β 为光源方向与 n 的夹角。从图中和公式可以看出当视锥参数和光源参数固定后,场景中的采样精度只与 Z,α,β 有关,而 Z 与透视走样有关,α,β 与投影走样有关。投影走样取决于局部的几何细节,在大型场景中不易消除,而透视走样产生于透视缩短效应,不依赖于场景,可以通过扭曲投影平面来减小。所以对透视走样的改进成为热点,而它又是阴影贴图算法中最难克服的难点之一,在最近几年的技术文章里,有许多关于透视走样的解决办法,其中 PPSMs(Parallel split shadow maps)[10]可以说是其中的佼佼者,被广泛用于现代的游戏和各种应用中,实现大型场景中的实时光影效果。

在 PPSMs 中,视锥体被平行于投影平面的多个裁剪平面(设为 m)分割成多个深度层,并且每层会被一个独立的阴影贴图所渲染。具体步骤是:

1)沿着视线方向使用平行于投影平面的分割平面将相机视锥体分割为 m 个部分;

2)对每个分割出来的相机视锥体设置一个光源视锥体,计算光源视锥体的观察投影变换矩阵;

3)为每个分割部分绘制阴影贴图;

4)绘制整个场景的阴影。

在第一步中,分割视锥体时需要计算分割位置,可以根据具体的场景调整分割位置,也可以使用 PPSMs 的实用分割方案(practical split scheme)。它是对数分割方案和一致分割方案的组合,可以通过调整权重以使整个深度范围产生折中的采样密度。

在第二步中,文献[11]展示了生成光源视锥体的两种方法:独立于场景的方法和依赖于场景的方法,前者简单的把光源视锥体绑定到分割视锥体中,不考虑

具体的场景信息计算简单,而后者建立的光源视锥体只包含了可能会把阴影投射到分割视锥体上的物体,因此是对纹理分辨率的最优应用。在文中,考虑到大规模场景的复杂性,采用独立于场景的方法来产生光源视锥体。

在第三步中,针对不同的硬件,可以有不同的实现方法参见文献[12],文中利用几何着色器复制法实现了 DirectX10 级别的硬件加速。渲染流水线如图 2 所示。

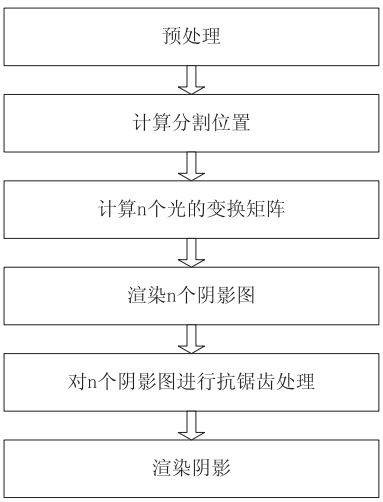

图 2 渲染流水线

图 2 中,因为在应用 PPSMs 后阴影部分依然有很明显的锯齿现象,为了进一步提高阴影质量,可以在阴影图生成后利用各种优化方法对阴影图进行抗锯齿处理。具体的优化方法有过滤技术,其中 PCF[13] 通过在一个像素上执行多次深度测试以达到抗锯齿效果,而 VSMs 用于在任意的过滤区域内估计物体被遮挡的概率;打包纹理,由于 GPU 中一次加载的可用的纹理贴图有限,可以把多个纹理放到一张单独的纹理中;应用不同的扭曲算法,即在第二步中,针对不同的扭曲算法(如 PSMs,LiSPSMs,TSMs)计算得到光源变换矩阵;改变纹理分辨率,为了减少纹理的开销,可以在不同的分割层中使用不同大小的纹理贴图;线性深度分布[14],对于点光源,用线性的深度代替标准投影得到的深度,以增强深度的精确值。这里主要介绍使用过滤技术方差阴影图 VSMs[15] 对其进行优化。它是计算没有锯齿现象的柔和阴影的理想算法。

VSM(Variance Shadow Maps,方差阴影图),运用了切尔雪夫(Chebyshev)不等式,通过计算阴影图中每个像素的深度值的期望和方差来计算像素颜色值。VSM 本质上对阴影图进行了滤波,它保证了直接对阴影图滤波就可以达到反走样的目的。

VSM 中的阴影图里可以使用两个颜色通道,一个保存深度值,另一个保存深度值的平方。这里的两个值可以对阴影图进行过滤后得到。文中实现是用两个 Pass,一个做横向过滤,一个做纵向过滤。比简单的箱型滤波器更有效率。

由以下公式可以得到深度值的期望和方差。

$$M_1 = E(x) = \int_{-\infty}^{\infty} xp(x)\,\mathrm{d}x$$

$$M_2 = E(x^2) = \int_{-\infty}^{\infty} x^2 p(x)\,\mathrm{d}x$$

$$\mu = E(x) = M_1$$

$$\sigma^2 = E(x^2) - E(x)^2 = M_2 - M_1^2$$

根据以上公式,就能够利用切尔雪夫不等式:

$$P(x \geqslant t) \leqslant p_{\max}(t) = \frac{\sigma^2}{\sigma^2 + (t-\varphi)^2}$$

切尔雪夫不等式的 one-tailed 形式只对 $t > \varphi$ 有效,如果 $t \leqslant \varphi$,$p_{\max} = 1$,表面被完全照亮。

只对深度值大于期望的像素应用切尔雪夫不等式,求每个像素大于阴影图中值(深度 t)的最大概率。

方差阴影贴图也继承了标准阴影贴图的缺点,如偏离、数值精度问题和光渗色现象。当多边形的深度跨越范围较大时(可能由于多边形几乎平行于光源方向),阴影贴图算法很容易产生偏离问题。VSM 用二次矩存储了每个像素的深度范围,把阴影贴图中的一个纹理像素当做一个局部的平面分布,而不是把整个阴影贴图的全部像素的深度视为 φ,可以表示为

$$f(x,y) = \varphi + x\frac{\partial f}{\partial x} + y\frac{\partial f}{\partial y}$$

利用期望的线性操作 $E(x) = E(y) = E(xy) = 0$,计算得到

$$M_2 = E(f^2) = E((\varphi + x\frac{\partial f}{\partial x} + y\frac{\partial f}{\partial y})^2)$$

$$= E(\varphi^2) + E(x^2)\left[\frac{\partial f}{\partial x}\right]^2 + E(y^2)\left[\frac{\partial f}{\partial y}\right]^2$$

现在认为像素具有对称高斯分布,方差为半像素 ∂,有 $E(\varphi^2) = \varphi^2$,$E(x^2) = E(y^2) = \sigma^2 = (0.5)^2 = 0.25$

因此 $M_2 = \varphi^2 + 0.25\left(\left[\frac{\partial f}{\partial x}\right]^2 + \left[\frac{\partial f}{\partial y}\right]^2\right)$

当偏微分都为 0 时,公式简化为 $M_2 = \varphi^2$(即当表面与光投影平面平行)。

计算中可能出现数值精度问题,可以通过在应用切尔雪夫公式之前把方差的最小值设为一个非常小的值来处理。而且这个值与具体的场景里的几何体无关,只需设置一次。

阴影贴图的光渗色现象,如图 3 所示,当 $\frac{\Delta a}{\Delta b}$ 变大时,就会出现光渗现象。

针对光渗现象,在大场景中主要关注得到较高的性能,可以接受某些不常出现的物理误差,所以可以使用近似算法。通过观察,如果深度为 t 的一个表面被某些平均深度为 μ 的过滤区域遮挡,那么 $t > \mu$,根据

Chebyshev 不等式, $p_{\max}(t)=\frac{\sigma^2}{\sigma^2+(t-\varphi)^2}<1$,即完全被遮挡的区域表面上错误的半影绝对不会达到完全的亮度。可以通过修改 $p_{\max}(t)$ 来移除这些区域。再重新调节其他的值,把它们映射到(0,1)区间。移除的区域可以根据具体的场景来调节,一般情况下,光渗色越严重,就要求更高的值,但这会减少阴影的细节。

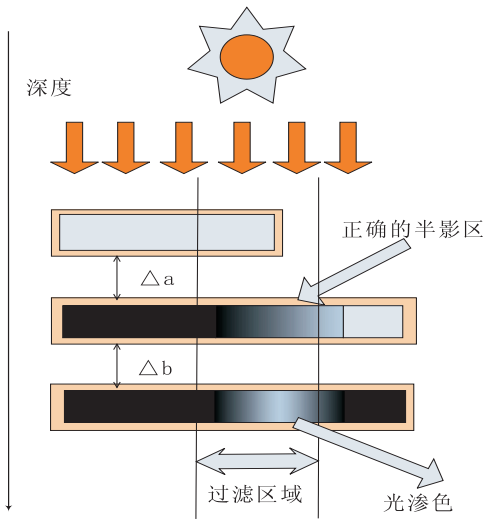

图 3　光渗色现象

经过上述变换之后,可以看到层与层之间的过渡很明显,包括明暗变化和阴影范围变化,需要进一步进行改进。据观察,离视点越远,阴影越淡,阴影范围越大,由于投影扭曲的原因,这是不可避免的,可以修正阴影的百分比,把 $P(x\geqslant t)\leqslant p_{\max}(t)=\frac{\sigma^2}{\sigma^2+(t-\varphi)^2}$ 改为 $P(x\geqslant t)\leqslant p_{\max}(t)=\frac{\sigma^2}{\sigma^2+C(t-\varphi)^2}$,为每一个层设置变量 C 的值可以使各层的阴影过渡平滑达到实用效果。

3　综合讨论

如图 4 所示,其中四层阴影分别减弱了 r、g、b 和 rg 颜色通道的分量,体现了阴影分层。阴影部分不同的颜色体现了 VSMs 的计算结果,不同的范围用不同的颜色值表示。

图 4　塔台中阴影分层和 VSMs 后效果

图 5 中可以看出视点远离阴影接收面的效果。

图 5　塔台中远景效果

图 6 中靠近阴影接受面时的第一层阴影边缘柔和,边缘锯齿几乎没有。

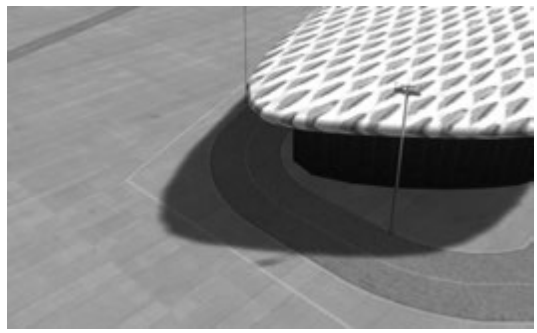

图 6　塔台中近景效果

实验结果在 GeForce GTX470 图形显示卡下获得的,分辨率为 1600 * 1200,使用了 1024 * 1024 的阴影贴图,4 层 PPSMs,VSM 横向滤波核和纵向滤波核为 4。

4　结束语

文中介绍了塔台模拟系统中的阴影生成算法。对阴影图算法的进一步研究可以从以下几个方面进行:

1. 随着 GPU 技术的发展,一些新的特性出现,阴影图算法可以以此提高自己的效率和效果。

2. 复杂光源的处理。复杂的光源包括任意形状的面光源、体光源、多面光源和多体光源,以及复杂的环境光等。这些光源类型虽然生成了复杂的软阴影,从而增强场景真实感,但是却增加了生成阴影的难度。目前,大多数阴影图算法在处理这类光源时有一定的局限性,如何即降低存储消耗,又提高绘制效率是进一步研究的重点之一。

3. 处理复杂场景和动态场景[16]。复杂的场景,如水下环境、浓雾环境、雨景、雪景,这类场景下的阴影涉及到一些物理模拟过程,生成高真实感的阴影比较复杂。动态场景的阴影实时绘制是该领域另外一个研究热点,需要考虑真实感和实时性的平衡。

4. 光源方向与接收平面夹角小时阴影拉得太长,与实际效果相差远,且阴影图有较大的冗余。如何提

高效率,减小冗余也是研究重点[17]。

参考文献:

[1] 张威巍. 虚拟环境中阴影的实时绘制算法研究[D]. 郑州:解放军信息工程大学,2009.

[2] Cass E, Kilgard M. Practical and Robust Stenciled Shadow Volumes for Hardware-accelerated Rendering[R]. [s. l.]: NVIDIA Corporation,2002.

[3] McGuire M. Efficient Shadow Volume Rendering[M]. [s. l.]:GPU Gems,2004:137-166.

[4] McGuire M, Hughes J F, Egan K, et al. Fast, Practical and Robust Shadows[R]. [s. l.]:Brown Univ. ,2003.

[5] Williams L. Casting Curved Shadows on Curved Surfaces [J]. Computer Graphics,1987,12(3):270-274.

[6] 过 洁,许 晓,潘金贵. 基于阴影图的阴影生成算法研究现状[J]. 计算机辅助设计与图形学学报,2010(4):579-591.

[7] Stamminger M, Drettakis G. Perspective Shadow Maps[J]. ACM Transactions on Graphics,2002,21(3):557-562.

[8] Wimmer M, Scherzer D, Purgathofer W. Light Space Perspective Shadow Maps[C]//Proceedings of the Eurographics Symposium on Rendering. [s. l.]:[s. n.],2004:143-152.

[9] Lloyd B, Tuft D, Yoon Sung-Eui, et al. Warping and Partitioning for Low Error Shadow Maps[C]//Proceedings of the Eurographics Symposium on Rendering. [s. l.]:[s. n.], 2006:215-226.

[10] Zhang Fan, Sun Hanqiu, Xu Leilei, et al. Parallel-Split Shadow Maps for Large-Scale Virtual Environments[C]//Proceedings of ACM International Conference on Virtual Reality Continuum and Its Applications. [s. l.]:[s. n.],2006:311-318.

[11] Zhang Fan, Sun Hanqiu, Xu Leilei, et al. Hardware-accelerated Parallel-split Shadow Maps[J]. International Journal of Image and Graphics,2008,8(2):223-241.

[12] Nguyen H. GPU 精粹 3[M]. 北京:清华大学出版社,2010:164-177.

[13] Reeves W, Salesin D, Cook R. Rendering Antialiased Shadows with Depth Maps[J]. Computer Graphics,1987,21(3):283-291.

[14] Brabec S, Annen T, Seidel H P. Practical Shadow Mapping [J]. Journal of Graphical Tools,2002,7(4):9-18.

[15] Donnelly W, Lauritzen A. Variance Shadow Maps[C]//Proceedings of the Symposium on Interactive 3D Graphics and Games. [s. l.]:[s. n.],2006:161-165.

[16] Annen T, Zhao D, Mertens T, et al. Real-time all-frequency shadows in dynamic scenes [J]. ACM Transactions on Graphics,2008,27(3):341-348.

[17] Liang X H, Ma S, Cen L X, et al. Light Space Cascaded Shadow Maps Algorithm for Real Time Rendering[J]. Journal of computer science and technology,2011,26(1):176-186.

(上接第 51 页)

[2] 邓璐娟,卢化琦,孙义坤,等. 改进的粒子群算法在测试数据生成中的应用[J]. 计算机技术与发展,2010,20(7):216-218.

[3] 宋晓琳,李 红,郭孔辉. 基于改进的粒子群优化算法的轮胎参数辨识[J]. 科技导报,2011,29(9):53-56.

[4] Yang Kai, Zhao Zhiqin, Nie Zaiping. Optimization of Unequally Spaced Antenna Arrays Using Fuzzy Discrete Partical Swarm Algorithm[J]. Journal of University of Electronic Science and Technology of China,2012,41(1):1-6.

[5] Wen Guangrui, Zhang Xinning, Zhao Ming. Application of Partical Swarm Optimization Algorithm in Field Holo-balancing[M]. Berlin Heidelberg:Springer-Verlag,2010.

[6] 王 丽,王晓凯. 一种非线性改变惯性权重的粒子群算法[J]. 计算机工程与应用,2007,43(4):47-48.

[7] 刘长良,高亚龙. 带压缩因子的粒子群算法在汽包压力控制系统中的应用[J]. 计算机系统应用,2012,21(1):13-16.

[8] 贾冀婷. 基于粒子群算法的测试用例自动生成方法研究[J]. 计算机技术与发展,2010,20(9):24-27.

[9] 苏守宝,汪继文,方 杰. 粒子群优化技术的研究与应用进展[J]. 计算机技术与发展,2007,17(5):249-252.

[10] 黄席樾,向长城,殷礼胜. 现代智能算法理论及应用[M]. 北京:科学出版社,2009:180-247.

[11] 张国印,程慧杰,刘咏梅,等. 一种新算法在基因表达谱聚类中的应用[J]. 计算机工程与应用,2009,45(36):216-218.

[12] 贾瑞玉,黄义堂,邢 猛. 一种动态改变权值的简化粒子群算法[J]. 计算机技术与发展,2009,19(2):137-139.

[13] 田 野. 粒子群优化算法及其应用研究[D]. 长春:吉林大学,2010.

[14] 王伯成,施锦丹,王 凯. 粒子群优化算法的研究现状与发展概述[J]. 电视技术,2008,48(5):7-11.

[15] 朱玉平. 一种改进粒子群优化算法[J]. 计算机技术与发展,2008,18(11):106-108.

[16] 王俊伟,汪定伟. 粒子群算法中惯性权重的实验与分析[J]. 系统工程学报,2005,20(2):194-198.

[17] 徐青鹤. 改进粒子群算法及其应用研究[D]. 杭州:杭州电子科技大学,2009.

[18] Trelea I. The partical swarm optimization algorithm:Convergence analysis and parameter selection[J]. Information Processing Letters,2003,85(6):317-325.

[19] Eberhart R, Shi Y. Comparing Inertia Weights and Constriction Factors in Partical Swarm Optimization [C]//IEEE Congress on Evolutionary Computation. Piscaway: IEEE Service Center,2000:84-88.

一种基于GPU的柔和阴影实现方法研究

作者： 付燕丽， 刘循
作者单位： 四川大学 计算机学院,四川 成都 610064
刊名： 计算机技术与发展
英文刊名： Computer Technology and Development
年，卷(期)： 2013(2)