

一种基于 GPU 的方差阴影映射改进算法

吴 跃,马小虎

(苏州大学 计算机科学与技术学院,江苏 苏州 215006)

摘 要: 提出了一种改进方差阴影映射的方法.该方法在传统的方差阴影映射中引入透视阴影的概念,在后透视空间中生成深度纹理,以牺牲对远处景物的采样率为代价,获得较高的对近处景物的采样率,减少了因采样率不足而导致的像素与纹元之间的误匹配.实验结果表明,该方法减小了光溢出现象的发生,提高了方差阴影的生成质量.

关键词: 阴影映射;方差阴影映射;透视阴影映射;GPU;OpenGL

中图分类号: TP391

文献标识码: A

文章编号: 1000-7180(2010)11-0071-03

An Improved Algorithm of Variance Shadow Maps Based on GPU

WU Yue, MA Xiao-hu

(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)

Abstract: A new method to improve variance shadow maps is proposed. In this method, the concept of perspective shadow maps is introduced into the traditional variance shadow maps. By rendering the depth texture in the post-perspective space, high resolution is provided for near objects while low resolution is provided for far objects. The miss-match between pixels and texels caused by the lack of sampling is reduced. Experiments demonstrate that our algorithm reduce the light bleeding and improve the quality of shadows.

Key words: shadow maps; variance shadow maps; perspective shadow maps; GPU; OpenGL

1 引言

阴影生成技术是虚拟场景中增强真实感的重要技术,目前已得到了广泛的应用^[1].阴影映射技术是现今阴影生成的主要技术之一.它的优点是容易被图形硬件实现,算法的效率比较高,而且与场景的复杂度无关.与以往的阴影映射方法^[2]不同,方差阴影映射通过存储深度的数据信息而不是简单的深度值,使得图形硬件内建的纹理过滤操作同样适用于深度纹理,从而在较小的代价下提升了阴影的效果.但是场景生成的深度纹理始终是阴影生成中的稀缺资源,如何在新方法的基础上合理地应用纹理仍然是值得研究的问题.这里,我们引入透视阴影映射^[3,4]的处理方法,通过在生成深度纹理时做适当的处理,使距离观察者较近的区域获得更多的资源,减少了光溢出现象的发生,进一步提升了方差阴影的质量.

2 相关工作

文献[5]提出方差阴影映射(Variance shadow maps, VSM),VSM在第一次绘制时,不是记录各个点的深度值,而是记录深度的分布信息,并最终通过对这个信息的处理,生成阴影^[6].因为记录的不是离散深度值,在生成阴影的时候就可以对深度纹理进行各向异性过滤、三线性差值等过滤操作,从而提升阴影的生成质量.但是这种算法需要较高的数据精度,纹理的深度至少为16位才能取得较为真实的阴影,同时还可能出现光溢出现象.为此,William和Andrew在2008年提出了Layered variance shadow maps,根据像素的深度值划分多个层次,以此限制不同层之间的光溢出现象,同时也因为单个层次内深度域的减小,降低了对精度的要求.

3 算法描述

VSM 算法是一种极为优秀的阴影算法,结合硬件技术的发展,大大地拓展了阴影生成技术.但是其本身所具有的问题也是我们不能忽视的.当数值稳定的问题渐渐被硬件技术的发展所克服的时候,如何更好地减少光溢出现象就成为当前的主要问题.为此我们将目光投向了阴影映射算法中业已成熟的翘曲类算法.

光溢出现象是 VSM 算法最大的缺陷,而且是无法避免的^[7].因此我们改进的目标应当是在合理的代价下,最大地减少光溢出现象.

翘曲与划分是阴影映射算法主要的改进手段,其中的诸如 PSM 算法、TSM 算法,都是较为成熟的优秀算法.就其根本,它们根据人类视觉的特性将整个场景划分为不同的区域,并赋予不同的采样率,最终实现更符合人类视觉的渲染.所以从某种程度上说,它们都是以提高阴影纹理的有效利用率为目的. VSM 算法的核心是其阴影计算模式的改变,通过模式的更改而实现了对硬件纹理过滤的支持,与 PSM 算法处于两个不同的方向上,这就使我们的改进成为可能.

我们将 PSM 算法的概念引入 VSM 算法,提出了基于这两种算法的混合算法.其核心思想就是将 VSM 算法中深度纹理的绘制移入后透视空间,利用 PSM 算法的原理,由提高 VSM 算法中的阴影纹理有效利用率,来提高阴影的生成质量,减少光溢出.

光溢出现象的出现区域是固定的,即其只可能出现在被重复遮挡的物体边界处,如图 1 所示,当时,切比雪夫不等式的计算结果将出现失误,最下层接受物体上位于虚线内的部分本应当被遮挡,而在实际绘制中却被照亮.这种情况将由于纹理映射的走样而被加剧.由于纹理大小的限制,场景中的多个像素往往对应于纹理中的一个纹元,这使得许多并非处于重复遮挡物体边界处的点被误判,而扩大了光溢出的范围.而由 PSM 算法生成的深度阴影,在距离观察点近的区域有着相当高的采样率,也就是说,那些在采用一般深度纹理时,受到精确度限制被判定为边界的点,此时不再被判定为边界而得到正确的阴影,从而减小了光溢出的区域.

4 算法实现

改进算法仍然需要两次绘制,在第一次绘制时,先完成相应的转换矩阵的计算,然后在后透视空间

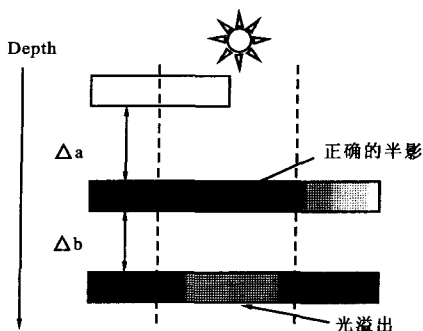


图 1 光溢出示意图

中绘制整个场景,并将相应的深度信息记录进纹理,我们选用了拥有两通道的颜色纹理来存储相应的数据.在纹理生成后,进行相应的模糊和纹理过滤操作,然后进行第二次绘制,完成 VSM 算法所需的阴影遮挡计算,并最终完成整个场景的绘制.这里,我们设计了两个着色器分别完成两次绘制.在第一个着色器中应用帧缓冲对象技术,直接将场景的深度信息写入帧缓冲,生成深度纹理.在第二个着色器中,完成 VSM 算法的计算过程,实现场景的最终效果.

由于涉及了两种阴影生成算法,所以对矩阵的转换和控制就比较复杂.在实现时,我们首先生成并记录了以光源位置为观察点的模型矩阵 $g_mModelView$ 和投影矩阵 $g_mProjection$.然后采用已观察者位置为观察点的模型矩阵和透视矩阵,激活深度纹理生成着色器,将 $g_mModelView$ 、 $g_mProjection$ 传入其中,计算出后透视空间的转换矩阵,在帧缓冲中采用计算得出的转换矩阵开始第一次场景绘制.绘制完成后,仍然采用已观察者位置为观察点的模型矩阵和透视矩阵并计算由观察者空间向场景空间的转换矩阵 $g_mCameraInverse$.应用 $g_mModelView$ 、 $g_mProjection$ 、 $g_mCameraInverse$ 和偏转矩阵(将数值范围 $[-1, 1]$ 转换至 $[0, 1]$)计算出纹理矩阵在第二次绘制中供阴影生成着色器使用.

需要注意的是深度纹理的内部格式一定要与分配的缓冲区的数据类型保持一致,当纹理的类型为 GL_FLOAT 时,纹理的内部格式应当为 $GL_RGBA32F_ARB$ 或者 $GL_RGBA16F_ARB$.

帧缓冲对象技术的应用需要得到 $EXT_framebuffer_object$ 扩展和图形显卡的支持.虽然这是一个通用的扩展,但是由于 OpenGL 的具体实现是由各硬件厂商自己完成的,所以在不同厂家的显卡上具体的命令有所不同,需要加以注意.

5 实验结果与分析

我们实现了基于 OpenGL2.0 + GLSL 的改进 VSM 算法,进行实验的计算机配置为 Intel Pentium4 3.0GHz CPU,图形显示卡是 NVIDIA 8600GT,显存为 256MB.由实验结果可以看出经过改进后的算法明显减小了光溢出现象,同时得益于采样率的提高,阴影更加平滑、清晰.

我们记录了 VSM 算法和改进的 VSM 算法在不同场景复杂度以及不同纹理大小下的帧数数据,可以看出,在复杂场景中,改进的 VSM 算法稍稍占有速度上的优势,两者并无明显差距,说明算法的改进并没有增加程序的计算量.同时我们也可以看到,随着纹理分辨率的增加,程序的效率逐渐降低,表示出纹理的内存消耗仍然可观.此外,纹理单通道的存储精度极大影响了程序的效率.

由实验结果看出,在简单场景中,改进算法的效率不如传统算法,这与在复杂场景中的情况恰恰相反.这里我们给出的解释是当场景较为简单时,应用透视阴影映射技术生成的深度纹理记录了更多的阴影区域的信息,增加了阴影生成时的计算量,而在复杂的场景中则不会出现这样的差别.具体的说,也就是在场景较为简单的情况下,只有近处的场景,而没有远处的场景,在不需要绘制远处阴影的前提下,改进算法用较高分辨率得到近处阴影的计算显然比传统算法只计算了正常分辨率下阴影的计算量要大;而在远处场景的阴影也需要绘制时,改进算法由于较低的远处场景分辨率只需要较小的阴影计算量,平衡了近处的代价,甚至稍稍优于传统算法的效率.此外,与复杂场景的情况相同,采用 16 位通道存储时算法的速度较快,从这我们也可以看出 VSM 算法对于硬件的依赖.

另外一个值得关注的地方就是在简单场景中,随着纹理分辨率的提高,程序的运行效率不断提高,有悖于其内存的消耗趋势,其原因应当也是简单场景这个前提.在这样的前提下,场景本身所需求的内存资源就会比较少,内存的压力也就不会如复杂场景中那样大.同时在简单场景中绘制阴影时,虽然大范围的空间中只有少数的物体,然而由于纹理分辨率较低,多数像素将会被误判为阴影而进行相应的

阴影计算,这种情况随着纹理分辨率的提高逐渐减少,使得纹理分辨率的提升没有因为内存的消耗增加而降低程序的效率,反而因为减少了总的计算量而提升了效率.

总的说来,VSM 改进算法在较明显地改善传统 VSM 算法阴影生成质量的同时,并没有显著地增加计算量,在某些情况下程序的运行速度甚至稍稍优于传统 VSM 算法,达到了我们的设计目标.

6 结束语

VSM 方法以其独特的遮挡判定使深度纹理滤波成为现实,已经成为阴影生成领域最主要的方法之一.为了进一步提升阴影的质量,我们引用了 PSM 方法的原理,减少了光溢出现象的发生,使得阴影边界更加平滑.接下来,我们将继续研究、改进 VSM 方法,以期得到更好的阴影.

参考文献:

- [1] 赵红漫,白振兴,王宇宙,等.一个地形可视化系统的建立与阴影分析计算[J].微电子学与计算机,2004,21(12):184-186.
- [2] Williams, L. Casting curved shadows on curved surfaces [J]. SIGGRAPH, 1978(12):270-274.
- [3] Stamminger M, Drettakis G. Perspective shadow maps [J]. ACM Transactions on Graphics, 2002(21):557-562.
- [4] Marc Stamminger. 游戏编程精粹 4[M].北京:人民邮电出版社,2005:331-340.
- [5] Simon Kozlov. GPU gems[M]. Boston:Addison Wesley, 2004:217-244.
- [6] Donnelly, William, Andrew Lauritzen. Variance Shadow Maps[C]// Proceedings of the Symposium on Interactive 3D Graphics and Games 2006. Redwood City:The ACM Siggraph, 2006:161-165.
- [7] Andrew Lauritzen. Summed-area variance shadow maps [EB/OL]. [2009-07-10]. <http://developer.nvidia.com/GPUGems3/gpugems3-ch08.html>

作者简介:

吴跃 男,(1983-),硕士研究生.研究方向为虚拟现实技术.

马小虎 男,(1964-),教授,硕士研究生导师.研究方向为计算机图形、图像处理技术等.