

# 一种改进的实时软阴影算法的设计与实现

张方彦<sup>1</sup>, 杨 猛<sup>2</sup>, 刘金刚<sup>1,2</sup>

(1. 首都师范大学,北京,100048;2. 中国科学院计算技术研究所,北京,100080)

**摘要** 在解决实时渲染时的阴影边界走样时,常用的(Percentage Closer Filter, PCF)方法不仅效率低下,而且边界处的平滑效果也很有限。改进了 PCF 算法,使算法的执行效率更高,且有更好的反走样效果。首先分别以相机和光源为基准来构造裁剪坐标系,通过 2 次绘制来锁定产生阴影的像素区域,然后用改进的高精度滤波模板对阴影进行适应性反走样处理,使平滑操作集中在对视觉效果有重要影响的边界区域上,有选择地优化了数据,从而大幅提高了全局计算效率。实验结果证明,文中算法比常规的 PCF 算法在执行速度上有明显提高,效果上也得到一定程度的改善,达到了效率与效果的优化折中。

**关键词** 阴影映射;软阴影;实时渲染

**DOI** 10.3969/j.issn.1009-3516.2012.02.015

**中图分类号** TP391 **文献标识码** A **文章编号** 1009-3516(2012)02-0070-04

随着计算机计算能力的逐步加强,图形加速硬件的不断进步,虚拟现实技术的应用需求日益强烈。因此,快速地实时生成具有较好真实感的图像,阴影的构建是一个非常重要的方面。阴影映射算法是实时阴影生成的一个重要方法。该算法简单易行,而且有良好的并行性,可结合图形硬件来实现加速计算,因此具有很高的绘制效率。该算法的种种优点已经使它已经在三维游戏、虚拟现实环境中备受青睐,但是它也有自身的局限性和有待改进之处,比如边界走样和量化误差等明显的缺陷。为了弥补处理边界反走样 PCF 算法的缺陷,本文在其基础上设计出一种基于阴影边界的适应性 PCF 算法。

阴影映射(Shadow Map)算法最早是由 William<sup>[1]</sup>提出的一种用深度缓存生成场景中物体阴影的方法。该算法存在着一些缺点:①走样问题。由于在深度比较过程中涉及到对阴影图的采样,在阴影图的分辨率低于采样需求的情况下就会产生走样,其典型的表现则为锯齿状阴影边界。为了解决阴影图的走样问题,提出了很多变种的阴影图算法。②处理全向光或半球形光的问题。在这些情况下,光源的视域非常大,一般一个简单的阴影图难以保存所有的深度值。

为了改进以上的缺点,后来的研究主要分成 2 个方向:一是记录区域的深度信息以取代单纯的记录逐像素深度。例如 2000 年 Lokovic<sup>[2]</sup>和 Veach 提出的 Deep Shadow Maps 算法和 2001 年 Kim and Neumann<sup>[3]</sup>提出的 Opacity Shadow Maps 算法等;二是使距离观察者近的区域能够获得更多的采样,对生成阴影深度纹理的方式进行改进。例如 2002 年 Stamminger<sup>[4]</sup>和 Drettakis 提出的 Perspective Shadow Maps 算法和 Wimmer<sup>[5]</sup>等 2004 年提出的 Light Space Perspective Shadow Maps 算法。

## 1 PCF 反走样算法

PCF 方法是 Reeves<sup>[6]</sup>等人于 1987 年提出的用来针对阴影走样问题的算法,它的优点就是良好的适用性。由于其功能相对独立,因此可以非常方便地融入绝大多数现有的 SM 算法框架中。该算法的核心思想

\* 收稿日期:2011-10-20

基金项目:国家“863”计划资助项目(2001AA114201)

作者简介:张方彦(1985-),男,河南周口人,硕士生,主要从事计算机图形学研究。

E-mail:zhangfangyan@gmail.com

与传统 SM 方法的不同之处在于在第 2 遍绘制时将当前像素与邻近采样区域内的多个像素进行深度比较并进行滤波操作,用一个介于 $[0,1]$ 之间称为遮挡平均数的值为基础来能产生平滑过渡的像素灰度值来模糊锯齿,从而达到改善阴影边界视觉效果的目的。这种办法也称软阴影方法,是应对锯齿化走样的有效方案。

反走样效果与 PCF 采样点的多少有着直接的关系。一般来说,采样点越多,反走样的效果就越好,但计算消耗就越大。然而,在 PCF 算法的实际执行中,真正起到模糊锯齿作用的采样点都位于阴影边界附近,而在阴影内部和阴影外部进行的采样对视觉效果的影响极其微弱。也就是说,在非关键区域应用 PCF 算法是冗余的,会造成不必要的计算资源的损耗,从而影响算法的执行效率,因此阴影采用适应性 PCF 算法才是更合理的策略。

此外,经典 PCF 中,滤波模板是简单的均值操作,即目标像素的深度值等于其模板中节点(周围的像素)的平均深度值。这种均值操作的数值精度较低,它过分地抹平了像素深度值之间的差异,易造成关键信息的丢失。事实上邻近像素点的深度值对当前像素的贡献应随着距离的增加而衰减,即每个节点都有不同的权重,距离越近的权重越大。

## 2 改进的 PCF 算法

### 2.1 基本思想

为了改善 PCF 算法的缺陷并更好地解决阴影映射算法的走样问题,本文提出一种适应性的 PCF 算法,这与 Randima Fernando<sup>[7]</sup>等人提出的适应性 SM 算法明显不同。该算法让抗锯齿的过程只在阴影边界附近的区域进行,并且使用精度更高的滤波模板,而对于阴影中的其他区域不再执行反锯齿滤波。在本文的算法中,“附近区域”被定义为距离图元边界 2 层像素宽度的区域。

由于图形的边界比其整体区域在几何上低了一维,所以适应性滤波需要涉及的像素数量也比该图形的像素数量低了一个数量级,这是本算法可以解决 PCF 低效问题的主要原因。

此外,本文使用了精度更高的滤波模板:对于当前像素的深度值,不再简单地对当前像素的周围采样点的深度取平均值,而是对其周围的采样点赋予一定梯度的权值。这样可以获得更连续的场,得到更好的反走样效果。

### 2.2 算法描述

改进型 PCF 算法就是通过使用精度更高的滤波模板对阴影进行适应性反走样处理,重点体现对视觉效果有重要影响的区域的信息。如图 1 所示,每个方格代表阴影图的一个像素。其中,由三角▲形围成的区域表示阴影区域,改进型 PCF 算法只对阴影边界附近区域做抗锯齿处理(如图 1 中由▲和●并起来的区域),而阴影内部和外部不做任何处理。在边界附近区域,采用改进型的 PCF 算法对此区域的阴影锯齿边缘做平滑处理。对当前像素进行滤波时可以取其周围 24 个采样点,这 24 个像素点的权值设置见图 2,正中心位置的方格代表当前像素,全部  $5 \times 5$  的方格代表着当前像素的滤波模板节点。所以,目标像素的阴影深度值就由此 25 个像素的加权累积得到。

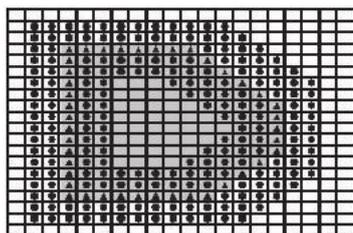


图 1 改进型边界 PCF 算法描述图

Fig. 1 Boundary descriptive graph of improved PCF algorithm

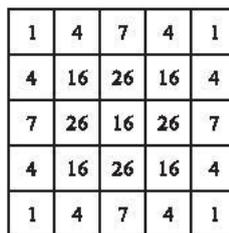


图 2 权值分布图

Fig. 2 Weight distribution diagram

### 2.3 算法实现

改进型 PCF 算法的实现可以分为适应性反走样处理的实现和非均匀滤波模板的实现。在实现适应性反走样处理时,本文利用了 SM 算法第 2 个绘制过程时阴影深度值的比较结果。由于在第 2 个绘制过程中像素深度比较的结果非 0 即 1,其代表着当前像素是处在光照中或者处在阴影中。所以,在 Fragment Shader 中可以对当前像素周围的左上最远处、右上最远处、左下最远处和右下最远处 4 个采样点分别做比较判

断操作(图 2 中权值为 1 的 4 个采样点),以确定该像素点是否处在阴影边界的附近区域。如果当前像素点处在阴影边界的附近区域,此像素的阴影深度系数将使用上述的滤波模板来更新。反之,如果当前像素点处在阴影边界的内部和外部,则此像素的阴影深度系数将使用 SM 算法中第 2 个绘制过程的结果。该过程伪代码如下:

```
If(离当前像素最远的 4 个采样点的阴影深度值都等于 0)
return 0; //该像素点处在阴影的内部,未处在阴影的边界区域,不做任何处理
else If(离当前像素最远的 4 个采样点的阴影深度值都等于 1)
return 1; //该像素点处在阴影的外部,未处在阴影的边界区域,不做任何处理
else 做改进型 PCF 反锯齿处理;
该过程流程图见图 3。
```

当然,如果想生成软阴影效果(见文献[8-9]),可以对以上程序进行适当改进,对阴影深度系数不再赋值为 0。在本文的试验中,阴影边界内部区域像素点的深度系数被赋值为 0.5。

当采用改进型 PCF 处理锯齿时,首先要获得当前像素的周围各个采样点的深度系数,然后再由这些深度系数值做加权计算。其伪代码如下:

```
weight[25]; //采样点的权值数组
sum ← 0; temp ← 0;
simple[25]; //采样点的阴影深度系数数组
for i ← 1 to 25
temp ← temp + simple[i] * weight[i]; sum ← sum + weight[i];
depth ← temp / sum;
```

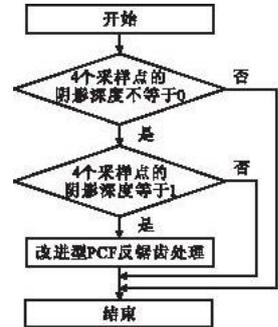


图 3 改进型 PCF 算法流程图  
Fig. 3 Flow chart of improved PCF algorithm

### 3 实验结果及分析

本算法的实现平台为 DirectX 9.0 VS2008,实验机器配置为:CPU intel Core 2 T6400 2.1 GHz,内存 2 GBytes,显卡 NVIDIA(英伟达) GeForce 9300,操作系统 Windows XP。实验场景包含 2 个物体和 1 个光源,两物体为一个正四边型的基面和一个茶壶。



图 4 普通的硬阴影

Fig. 4 Ordinary hard shadow

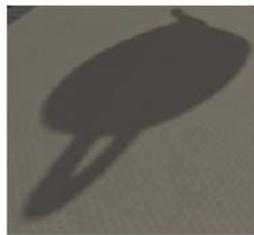


图 5 PCF 算法处理

Fig. 5 Conventional PCF algorithm



图 6 改进型 PCF 算法处理

Fig. 6 Improved PCF algorithm

该实验使用  $512 \times 512$  分辨率的阴影映射图。图 4 为普通硬阴影算法的结果,图 5 为传统的 PCF 边缘锯齿平滑处理后的效果,图 6 为本文算法的效果。可以看到,图 5 做了 PCF 处理,锯齿现象明显有所改善。图 6 通过改进型的 PCF 算法处理,比之经典 PCF 的效果,图 6 的阴影边缘区域颜色的过渡更加自然;最重要的是,本文算法在全局的执行效率上比传统的 PCF 算法有了明显提高。以上 3 种算法的 FPS(帧/s),见表 1。

表 1 3 种阴影的 FPS(帧/s)

Tab. 1 Three sets of shadows FPS (F/s)

	普通硬阴影	PCF 反走样	改进型 PCF 反走样
FPS	59.6	52.3	58.4
空间复杂度	0(1)	0(1)0(1)	

## 4 结束语

本文详细论述了 PCF 阴影映射方法,在此基础上对传统的 PCF 阴影映射方法进行了改进,形成了对阴影进行适应性反走样处理的改进型 PCF 算法,通过适应性策略和新的权值配置,在 GPU 上进行了实现。实验表明改进后,阴影的生成效果有一定程度的改善,阴影生成效率得到了明显提高。

### 参考文献(References):

- [1] Williams L. Casting curved shadows on curved surfaces [J]. Computers and graphics, 1978, 12(3): 270 - 274.
- [2] Lokovic T, Veach E. Deep shadow maps [C]//Proceedings of the 27th annual conference on computer graphics and interactive techniques. New York: ACM press, 2000: 385 - 392.
- [3] Kim T Y, Neumann U. Opacity shadow maps [C]//Proceedings of the 12th Eurographics workshop on rendering techniques. London, UK: Springer - verlag, 2001: 177 - 182.
- [4] Stamminger M, Drettakis G. Perspective shadow maps [J]. ACM transactions on graphics, 2002, 21(3): 557 - 562.
- [5] Wimmer M, Scherzer D, Purgathofer W. Light space perspective shadow maps [EB/OL]. <http://www.cg.tuwen.ac.at/research/vr/lispsm>.
- [6] Reeves T, Salesin D H, Cook R L. Rendering antialiased shadows with depth maps [C]//Proceedings of SIGGRAPH'87. New York: ACM press, 1987: 283 - 291.
- [7] Fernando Randima, Fernandez Sebastian, Bala Kavita, et al. Adaptive shadow maps [C]//Computer graphics proceedings, annual conference series. Los Angeles, California: ACM, 2001: 387 - 390.
- [8] 曾晓一, 何援军. 软阴影算法及实现 [J]. 工程图学学报, 2010, 31(4): 1 - 5.  
ZENG Xiaoyi, HE Yuanjun. Algorithm and implementation of soft shadow [J]. Journal of engineering graphics, 2010, 31(4): 1 - 5. (in Chinese)
- [9] 吕伟伟, 孟维亮, 薛盖超, 等. 基于 GPU 的近似软影实时绘制 [J]. 计算机辅助设计与图形学学报, 2009, 21(3): 3 - 9.  
LÜ Weiwei, MENG Weiliang, XUE Gaichao, et al. Real - time approximate soft shadow rendering on GPU [J]. Journal of computer - aided design & computer graphics, 2009, 21(3): 3 - 9. (in Chinese)

(编辑: 徐楠楠)

## Design and Implementation of An Algorithm of Real - time Shadow Rendering

ZHANG Fang - yan<sup>1</sup>, YANG Meng<sup>2</sup>, LIU Jin - gang<sup>1, 2</sup>

(1. Capital Normal University, Beijing 100048, China; 2. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China)

**Abstract:** In realistic graphics, how to reflect realistic shadow effect is an important aspect to enhance the sense of reality of the model. The shadow should meet the real physical situation as far as possible, meanwhile, a smooth edge is necessary. However, rectangular array of pixels will easily cause jagged aliasing phenomenon at the edges. How to deal with the aliasing problems effectively has been a hot topic in the field of graphics. When solving aliasing of the shadow boundary in real - time rendering, the conventional Percentage Closer Filter (PCF) is not only inefficient, but also the smoothing effect of the boundary is limited. This paper improves the PCF algorithm, the improvement makes the algorithm more efficient in implementation. This new algorithm is based on conventional Shadow Mapping and uses higher accuracy of the filter template to deal with aliasing adaptively, especially to the area which has impact on visual effects, thus increasing the overall computational efficiency substantially. The experiment results show that this algorithm is obviously improved in efficiency compared with the conventional PCF.

**Key words:** shadow mapping; soft shadow map; real - time rendering