

Journal of Image
and Graphics

中国图象图形学报



ISSN1006-8961
CN11-3758/TB

2013 **3**
Vol.18 No.

中国科学院遥感应用研究所
中国图象图形学学会主办
北京应用物理与计算数学研究所

中国图象图形学报

Zhongguo Tuxiang Tuxing Xuebao

2013 年 3 月 第 18 卷 第 3 期(总第 203 期)

目 次

图像处理和编码

沿边局部灰度差分椒盐噪声滤波 孔勇奇,潘志庚(249)

图像分析和识别

核稀疏保留投影及在步态识别中的应用 王科俊,阎涛,吕卓纹,唐墨(257)

局部 Gist 特征匹配核的场景分类 杨昭,高隼,谢昭,吴克伟(264)

图像理解和计算机视觉

条件随机场模型的场景描述 赵龙,郭立,谢锦生(271)

基于 L_1 与空间点分类的 3 维场景重建 李静,杨宜民,付根平,陈建平(277)

稀疏表示的 Lucas-Kanade 目标跟踪 徐如意,陈靛影(283)

医学图像处理

管状特性和主动轮廓的 3 维血管自动提取 尚岩峰,汪辉,汪宁,陈杰,姚丽萍,孙琨(290)

“第 17 届 CAD&CG 学术会议”专栏

多步攻击的规则树检测及可视化 胡亮,赵剑明,解男男,努尔布力(299)

显著性边缘引导下的图像抽象化 王勋,李红,刘春晓,严彩萍(305)

稀疏表达的运动数据压缩 齐天,肖俊,庄越挺(311)

基于语义的用户关注度计算 陈蕾英,陈佳舟,潘斌,彭群生(318)

九宫格空间框架的图书图像检索 李宗民,唐志辉(325)

光学相干层析视网膜体数据的 3 维分割 樊鲁杰,孙延奎,张田,田小林(330)

混合软影绘制算法 刘浏,周炜,李华(336)

基于场的人群运动仿真 赵欣欣,张勇,孔德慧,尹宝才(344)

利用大位移视图的自动可信图像修补 刘春晓,金剑秋,彭群生(351)

第八届国际数字地球会议征稿通知 (358)

Journal of Image and Graphics

(Monthly, Started in 1996)

Vol. 18 No. 3 March 2013

Contents

Image Processing and Coding

Salt and pepper noise filtering algorithm based on local border gray-scale differences Kong Yongqi, Pan Zhigeng(249)

Image Analysis and Recognition

Kernel sparsity preserving projections and its application to gait recognition Wang Kejun, Yan Tao, Lv Zhuowen, Tang Mo(257)

Scene categorization of local Gist feature match kernel Yang Zhao, Gao Jun, Xie Zhao, Wu Kewei(264)

Image Understanding and Computer Vision

Scene description based on the conditional random fields model Zhao Long, Guo Li, Xie Jinsheng(271)

Three dimensional scene reconstruction based on L_1 and scene point classification
..... Li Jing, Yang Yimin, Fu Genping Chen Jianping(277)

Lucas-Kanade tracking based on sparse representation Xu Ruyi, Chen Jingying(283)

Medical Image Processing

Three dimensional vessel extraction model based on tubular characters and active contour model
..... Shang Yanfeng, Wang Hui, Wang Ning, Chen Jie, Yao Liping, Sun Kun(290)

Issum of the CAD&CG'2012

Multi-step attacks detected by rules tree and visualization Hu Liang, Zhao Jianming, Xie Nannan, Nurbol(299)

Salient edge guided image abstraction optimization Wang Xun, Li Hong, Liu Chunxiao, Yan Caiping(305)

Motion data compression using sparse representation Qi Tian, Xiao Jun, Zhuang Yueting(311)

Semantic-based user interests computation Chen Leiying, Chen Jiazhou, Pan Bin, Peng Qunsheng(318)

Books image retrieval based on grid space frame Li Zongmin, Tang Zhihui(325)

Three dimensional segmentation to detect retinal boundary surfaces from OCT volume data
..... Fan Lujie, Sun Yankui, Zhang Tian, Tian Xiaolin(330)

Hybrid soft shadow rendering method Liu Liu, Zhou Wei, Li Hua(336)

Field-based crowd simulation Zhao Xinxin, Zhang Yong, Kong Dehui, Yin Baocai(344)

Automatic and accurate image completion from a large displacement view Liu Chunxiao, Jin Jianqiu, Peng Qunsheng(351)

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2013)03-0336-08
论文引用格式: 刘浏,周炜,李华. 混合软影绘制算法[J]. 中国图象图形学报,2013,18(3):336-343.

混合软影绘制算法

刘浏^{1,2}, 周炜¹, 李华¹

1. 中国科学院计算技术研究所, 北京 100190; 2. 中国科学院研究生院, 北京 100049

摘要: 实时软影映射算法能够使用一张阴影映射图绘制出复杂动态场景的真实的软影效果。结合物体空间和图像空间的算法,提出一种在物体空间中提取面光源轮廓边的方法;然后利用面光源轮廓边的属性,提出一个在图像空间中确定轮廓边像素走向的方法;最后反投影拟合的轮廓边计算最终的几何可见性。实验结果表明,本文算法减少了提取轮廓边的消耗,提高了绘制效率;增加了拟合轮廓边的精度,提高了绘制效果。

关键词: 实时软影;面光源;物体空间;轮廓边提取;反投影

Hybrid soft shadow rendering method

Liu Liu^{1,2}, Zhou Wei¹, Li Hua¹

1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
2. Graduate School of the Chinese Academy of Sciences, Beijing 100049, China

Abstract: Real-time soft shadow algorithms can render convincing soft shadows on complex and dynamic scenes with a single shadow map. In this paper, we propose a hybrid rendering method, which extracts edges for area light in object space and determines the edge pixel's direction in image space. Then, we use the Hierarchical Edge Map to compute the visibility. The results show, the cost of silhouette extracting is reduced and the rendering performance is improved. Furthermore, the accuracy of the contour and the shadow quality are both enhanced.

Key words: real-time soft shadow; area light; object space; edge extraction; backprojection

0 引言

阴影是重要的真实感效果,它不但能增强所绘制图片的真实感,还有助于观察者理解物体之间的相对位置关系。根据对光源建模的不同,所绘制的阴影主要分为硬影和软影。在过去的十几年中,众多研究者都关注于软影的绘制,并提出了许多重要的算法,但是在实时条件下绘制软影仍然是一个挑战。

近年来,许多研究人员提出很多近似实时软影算法。Guennebaud 等人^[1-2]提出了使用离散阴影贴

图表示连续场景,并通过反投影的方法计算光源可见性的软影绘制算法(SSM),这个算法近似模拟了真实的物理过程,在实时的条件下获得了高质量的软影效果。然而,该算法的效率随着光源面积的增加而显著降低。周炜等人^[3]分析了 Guennebaud 的算法,提出了一种预提取轮廓边的方法(PSSM),提高了绘制效率。文献^[4]中提出了一种在图像空间中拟合轮廓边的方法,进一步提高了绘制效率。

在当前 GPU 的体系结构中,纹理采样相当耗时。当前软影映射算法效率的瓶颈在于过多的纹理采样,而最近的一些软影映射算法也致力于减少纹理采样。基于这个思想,提出了一种物体空间和图

收稿日期:2012-08-20;修回日期:2012-10-18

基金项目:国家自然科学基金项目(60873164);国家高技术研究发展计划(863)基金项目(2007AA01Z317)

第一作者简介:刘浏(1982—),男,中国科学院计算技术研究所博士研究生,主要研究方向为计算机图形学、虚拟现实。E-mail: liuliu@

nic. ac. cn

像空间相结合的软影算法。算法在 GPU 中实现了基于物体空间提取面光源有向轮廓边的方法,提出了一种在图像空间确定轮廓边走向的方法,最后利用文献[4]可见性算法计算最终的阴影值。实验结果表明,本文算法减少了提取轮廓边所需要纹理采样的数目,加快了绘制的速度;增加了拟合轮廓边的精确度,提高了绘制效果。

1 相关工作

简单回顾一下近期实时软影渲染领域的工作,更完整的论述可以参考最近的阴影算法综述^[5]。

阴影体算法^[6]和阴影映射算法^[7]是阴影实时绘制中最为常用的两种算法。Akenine-Möller 等人^[8]基于阴影体算法提出了软影体算法。此算法的主要思想在于光源轮廓边包含了用来生成阴影的所有信息。算法通过为每条轮廓边构造半影楔形(penumbra wedge)来包裹半影区域,进而通过简单的插值计算出半影楔形内部像素的可见性。Assarsson 等人^[9-10]简化了半影楔形的构造方法,同时他们还使用反投影轮廓边的方法来计算可见性,提高了绘制阴影的质量。半影楔形算法仅仅通过相加来叠加阴影,这导致了过阴影问题。Forest 等人^[11]提出了一种半影楔形的混合算法,减少了过阴影问题。由于需要使用轮廓边和光栅化操作,半影楔形的算法和经典的阴影体算法一样,受到了填充率限制。

Chan 等人^[12]和 Wyman 等人^[13]扩展了阴影映射算法,他们分别提出了使用半影映射图来生成软影。半影映射图并不需要光栅化半影楔形,这样就移除了填充率的限制,但是这个算法依然需要使用 CPU 来提出轮廓边。吕伟伟等人^[14]使用几何着色器来生成半影图,把所有的生成过程移动到了 GPU 里面。在算法实现中,他们使用了几何着色器(geometry shader)^[15]来提取点光源的轮廓边,进一步提高算法效率。孙明彦等人^[16]则利用深度剥离技术,并同时生成内外半影,提高了软影质量。这一类算法的绘制效率虽然很高,但是由于算法原理上并不是基于物理模型,因此所绘制的阴影与真实效果相差较大。

Atty 等人^[17]和 Guennebaud 等人^[1]分别对阴影映射算法进行了扩展,提出了完整的软阴影映射算法,这个算法近似模拟了物理模型,通过反投影离散的阴影映射图中的像素到光源平面来计算可见性。但 Atty 等人的算法需要区分阴影的投射者和接受

者,限制了算法的通用性。Guennebaud 等人^[1]还通过应用延长像素的方法解决了离散采样引起的漏光问题,但是他们的算法并没有解决因为反投影重叠而引起的过阴影问题。Schwarz 等人^[18]使用位掩码表表示光源,解决了反投影重叠造成的过阴影问题。Guennebaud 等人^[2]提出了一种新的反投影方法,他们不反投影阴影像素而是反投影轮廓边,并使用径向积分^[9]计算可见性,同时解决了漏光问题和过阴影问题。Yang 等人^[19]和周炜等人^[3]分别利用像素的空间连续性和阴影映射图中轮廓边的稀疏性进一步加快了软影映射算法的绘制效率。文献[4]发现在预先提取的轮廓边像素中,70%以上都是在同一条直线上,把这些轮廓边像素拟合成了直线,并把拟合的信息保存到一个层次化边界信息图(HEM)^[4]中,最后,在反投影计算可见性的时候,仅需要反投影拟合的直线,减少了纹理采样数目,进一步提高了绘制效率。

从总体上看,本文软影绘制算法和文献[4]算法类似。主要的不同之处在于:本文的软影绘制算法结合了物体空间和图像空间,提出了一种基于物体空间提取面光源有向轮廓边的方法,进而在图像空间确定了轮廓边的走向。本文算法仅仅在物体空间提取轮廓边,而没有生成阴影体,避免了填充率的问题;同时遵循使用逻辑运算代替纹理采样的原则,避免做过多的纹理采样,提高了绘制低复杂度场景的效率。与图像空间的算法相比,本文算法在物体空间提取轮廓边的精度更高,因此在一定程度上提升了绘制阴影的质量。

2 算法概述

经典的软影映射算法是通过反投影轮廓边来计算阴影,因此,对于目标像素而言,只有轮廓边才是有效信息,其余均是冗余信息。如图1所示,整张图表示光源中心活动的场景的深度图,场景由平面1和平面2组成,其中平面1距离光源较近,平面2则相对较远。图1中 P 点表示着色点, M 表示在阴影映射中 P 点对应的搜索区域, M 中仅有边界直线“Line”能够影响 P 点的阴影计算。Guennebaud 等人算法^[2]在计算 P 点阴影值的时候需要迭代 M 中所有的像素点。PSSM 算法^[3]和 HEM 算法在预提取轮廓边的时候也需要迭代 M 中所有的像素点。如果直接从物体空间来提取轮廓边,那就不需要多

余的纹理采样,而全部使用高效的浮点运算来实现,因此可以提高绘制效率。

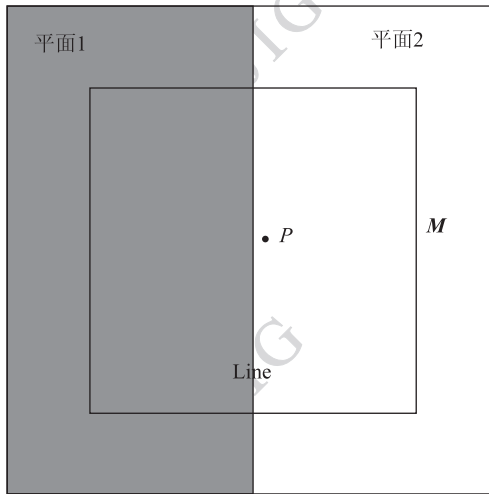


图 1 算法原理示意图

Fig. 1 Illustration for the algorithm idea

本文算法的主要流程如下:

- 1) 从光源中心绘制一张标准阴影映射图;
- 2) 在物体空间提取轮廓边,把轮廓边投影到平面内,并利用光源中心的深度图去除不可见的轮廓边;
- 3) 根据在物体空间提取轮廓边的斜率,确定轮廓边的走向;
- 4) 在固定的块中拟合轮廓边直线,并把结果保存在 HEM 中;
- 5) 根据光源大小确定搜索区域,迭代搜索区域内的所有 HEM 块,通过反投影块中的轮廓边计算光源的可见性;
- 6) 使用可见性因子绘制最终结果。

从总体上看,本文算法类似于 HEM 算法^[4],主要区别在于步骤 2)3)。本文实现了一种在物体空间中高效地准确地提取轮廓边的方法。在物体空间中提取轮廓边后,能够获得轮廓边在图像投影的斜率。本文算法利用这个斜率确定了轮廓边的走向。步骤 4)中,重用 HEM 的思想——在图像空间内拟合固定大小的块中的轮廓边像素。

3 轮廓边

3.1 提取轮廓边

对于点光源,检测轮廓边^[14,20]的方法非常简单。两个三角面片相邻,若这两个三角面片中一个面向光源,另一个背向光源,则公共边是轮廓边。

对于面光源,检测轮廓边的方法也比较直观,Laine 等人^[21]提出了一种检测潜在轮廓边方法,如图 2 所示。两个相邻的三角面片把空间分成 4 个部分,如果光源完全在第 1 或者第 3 部分内,则公共边肯定不是轮廓边,否则公共边就可能是轮廓边。

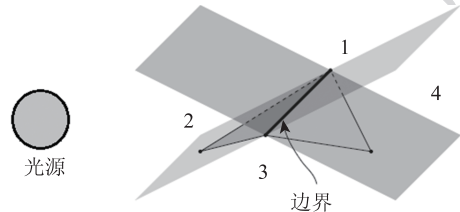
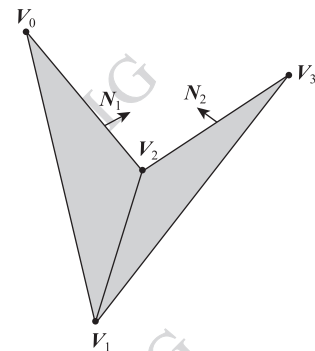


图 2 检测面光源的轮廓边的示意图

Fig. 2 Illustration for detecting silhouettes of area light

以上判断方法都只能确定两相邻面片的公共边是否可能是轮廓边。真实的模型中的面片是有朝向的,即面片有法向量。如果两个相邻面片的法向量相对如图 3 所示,或者法向量平行,那么公共边则是模型的一条内部边,而不是光源的轮廓边。使用判断内部边界算子 $E(V_1, V_2)$ 就能检测出这种情况,进而在检测轮廓边的时候抛弃这些公共边。同时,一些正确的面光源轮廓边还可能被模型本身完全或者部分遮挡,因此需要尽可能地去掉这些被遮挡的轮廓边。这里使用近似算法:把这些面光源轮廓边投影到光源中心的深度图上,去除那些没有通过深度测试的轮廓边像素点。近似算法的好处是:能够在 GPU 中简单高效的实现,与现有的软影绘制算法无缝结合,同时能够保留轮廓边上没有被模型完全遮挡的部分;缺点是:所得到的轮廓边是不完整的,对于光源中心不可见而对于光源其他位置可见的轮廓边被舍弃了,此缺点可能会使得所绘制的阴影在某些细微处有偏差,但是不会影响整体效果。



$$E(V_1, V_2) = (N_1 \cdot (V_0 - V_3) \leq 0)$$

图 3 模型的内部边

Fig. 3 Internal silhouette

在物体空间内检测面光源轮廓边,需要迭代模型的每两个相邻的三角面片。在 CPU 中执行这样的计算是相当耗时的。然而,现代的 GPU 都支持几何着色器,它可以在 GPU 中并行地处理各个三角面片。利用这个特性,能够在几乎不影响绘制效率的情况下,完整地提取所有的面光源轮廓边(图 4(b))。几何着色器算法步骤如下:

- 1) 如果 $E(V_1 V_2)$ 返回为 True, 表示边界为内部边界, 不可能为轮廓边;
- 2) 如果给定边界与光源相交, 那么此边界为轮廓边;
- 3) 如果给定边界的两相邻三角形所在平面与光源相交, 那么此边界为轮廓边;
- 4) 给定边界的两相邻三角形所在平面把空间分成 4 个部分(图 2), 如果光源整体位于第 2

或者第 4 部分, 那么边界为轮廓边, 否则边界不是轮廓边。

在物体空间检测完轮廓边后, 需要把这些轮廓边缓存以供以后的计算使用。基于 GPU 的特性, 本文算法是把轮廓边投影到图片上, 使用纹理缓存轮廓边。因此需要对轮廓边进行光栅化, 这样不可避免地会引入走样误差, 同时也可能引起轮廓边像素重叠的问题。因此, 本文算法在实现的时候, 对于那些走样较严重的场景使用了多渲染目标保存像素所在轮廓边的信息, 降低了走样误差对于绘制结果的影响。还利用了光源中心的深度图, 剔除了那些不可见的轮廓边像素(图 4(c)), 同时也解决了轮廓边像素重叠的问题。因为在绘制深度图和投影轮廓边的时候, 使用光栅化的配置是不相同的, 因此去除不可见的轮廓边还是需要使用深度偏移。

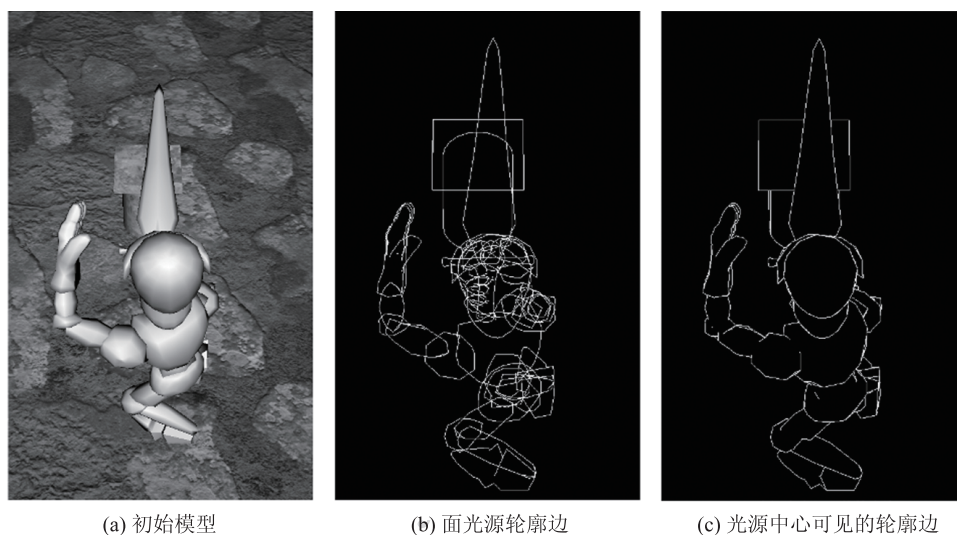


图 4 面光源轮廓边提取算法的实验结果

Fig. 4 The extracted silhouettes of our algorithm

3.2 确定轮廓边走向

在获得光源中心可见的轮廓边像素(图 4(c))之后, 还需要确定轮廓边的走向——对于某个轮廓边像素, 找到它前一个轮廓边像素和后一个轮廓边像素的位置。对于轮廓边走向的确定, 本文算法遵循以下原则: 轮廓的方向应使得轮廓边左侧像素与轮廓边在同一面片上, 右侧的像素则不能在同一面片上。

在文献[4]算法的框架中, 确定轮廓边的走向是在图像空间中执行的。对于每个轮廓边像素(图 5 中像素 0), 需要对它周围的 8 个环绕像素点进行采样(图 5 中像素 1—8), 然后通过比较深度值, 来

确定前一个轮廓边像素和后一个轮廓边像素的位置。

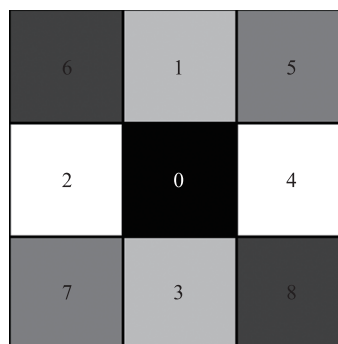


图 5 边界点及其 8 个相邻点

Fig. 5 Edge pixel and its 8 neighbors

本文算法是在物体空间中先检测出轮廓边,然后再把这些轮廓边投影到图像空间。在物体空间检测到轮廓边后,就能获得轮廓边的一些物理属性——斜率、方向等,进而只要把这些数据保存到轮廓边像素中,就能加快确定轮廓边走向的速度。

若像素 0 为边界点像素,已知它的斜率为 k ,那么根据 GPU 对直线的光栅化规则 (rasterization rules)^[15],可以推算出它相邻轮廓边像素的大概位置。例如,如果轮廓边斜率为 0,那么相邻的轮廓边像素为 2 和 4;如果斜率为 0.5,那么 2 和 4、5 和 7 可能是相邻的轮廓边像素。那么仅需要采样这些大概位置上的像素,就能验证得到准确的结果。这样,就能把采样次数减少为 2~4 次,减少为原始算法的 1/4~1/2。

利用斜率能够高效地找到相邻的轮廓边像素,但是无法确定轮廓边的走向。在真实的模型中,面片是有朝向的,而面片朝向是由组成面片的 3 个顶点的环绕顺序决定的。同时,它们也确定了面片边界的方向。图 6 表示光源空间中的一个面片,有 3 个顶点 A 、 B 和 C ,其中 AB 边是轮廓边。如果顶点的环绕顺序为 $(A、B、C)$,那么法向量面向光源,边界方向为 AB ;若顶点的环绕顺序为 $(C、B、A)$,那么法向量背向光源,边界方向为 BA 。边界方向 AB 是满足确定轮廓边走向的原则的,称为正向;而边界方向 BA 则刚好相反,称为负向。因此,在轮廓边像素中增加一个标识——正向为正、负向为负,就能很容易地找到下一个像素,进而方便了轮廓边直线的拟合,以及最终的阴影值的计算。

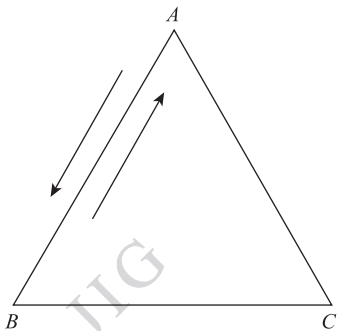


图 6 轮廓边方向

Fig. 6 The direction of a silhouette

4 计算可见性

本文算法使用了文献[4]中拟合轮廓边像素、构建层次化的 HEM 算法。基本思想是:在有向的

轮廓边像素图的固定块中拟合直线,并利用层次化的贴图来保存拟合的直线信息,最后在计算可见性的时候,只要反投影那些拟合的直线,就能极大地减少纹理采样数目,从而提高绘制效率。

图 7 展示了 HEM 的空间结构。图 7 中 HEM 共分 3 层,每一层所表示的固定块大小分别为 4×4 、 8×8 、 16×16 。在 HEM 中,共有 4 中类型的块:空块(白色)、直线块(浅灰色)、简单块(深灰色)、复杂块(黑色)。这些块中大多数为空块,约为 90% (4×4 块)^[4];而在非空块中,直线块占大多数(70% 以上)^[4]。因此在计算可见性的时候,跳过这些占大部分的空块,直接反投影直线块,使用径向积分获得最终结果。

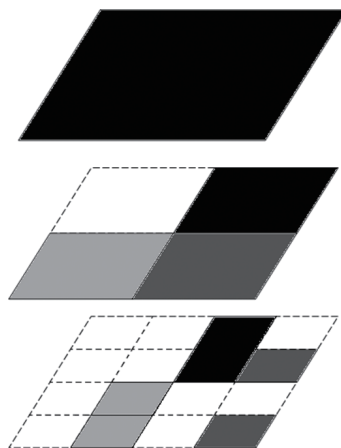


图 7 HEM 示意图

Fig. 7 Illustration for HEM

5 实验结果与分析

使用 Microsoft Direct3D10 和 HLSL SM4.0 实现本文算法。实验结果在一台酷睿 2 双核 2.66 GHz CPU, GeForce GTX 285 GPU 的 PC 机上产生的。其中,深度贴图的采样分辨率为 1024×1024 ,渲染窗口的分辨率为 1024×1024 ,默认光源为圆形光源,半径 r 为 1.0。使用 4 个模型进行实验:骑士 (Knight),植物 (Plant),树木 (Tree) 和龙 (Dragon)。它们的面片数分别为 6 000、67 000、110 000 和 871 000。图 8 显示这个 4 个模型分别使用本文算法绘制得到的阴影效果。图 9 显示了骑士模型在不同的光源半径下所提取轮廓边的效果。

表 1 和表 2 显示了分别使用本文轮廓边提取算法和 ALE^[3] 算子对这 4 个模型进行测试的结果。表 1 中 OSE 表示物体空间轮廓边提取算法;OSE +

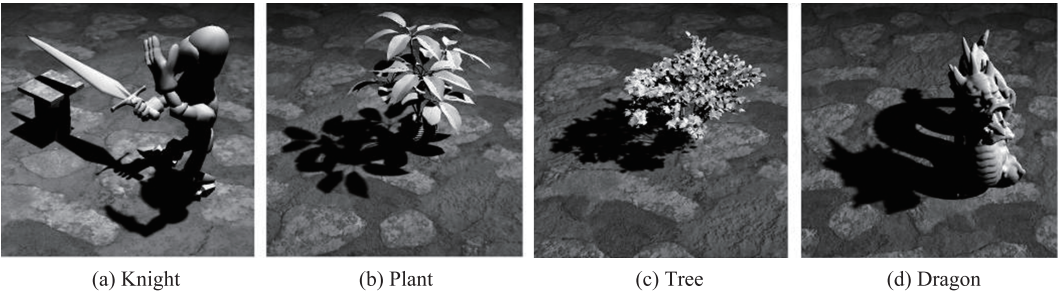


图8 4个模型使用本文阴影算法的绘制结果
Fig. 8 Rendering results of 4 models

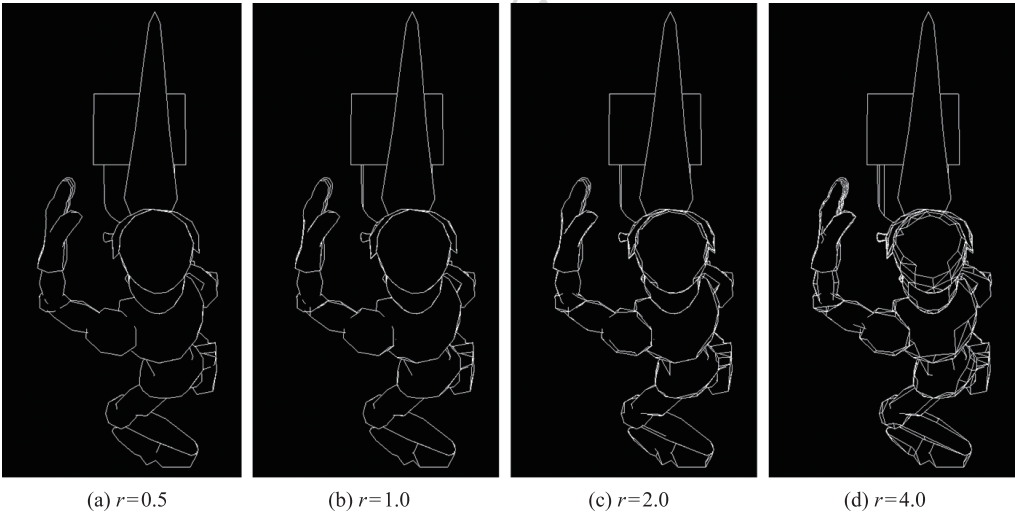


图9 本文算法在不同光源半径下提取轮廓边的效果
Fig. 9 The silhouettes extracting in different light radius

D 表示物体空间轮廓边提取算法加上本文轮廓边像素走向确定算法;D-Cost 表示确定走向的消耗。表 2 中 ALE + N 表示 ALE 算子加上 HEM 中轮廓边像素走向确定算法。实验结果表明,在模型面片数相对较少的情况下,本文轮廓边提取算法的效率大大优于 ALE 算子。对于龙模型,由于它的面片数巨大, GPU 的顶点处理器数目不能满足如此巨大的面片数,因此轮廓边检测的效率显著降低。而本文轮廓边检测算法必须对模型绘制两次(第 1 次绘制深度图,第 2 次在物体空间中检测轮廓边),这也一定程度上加重了面片数目对于帧率的影响。另外,本文轮廓边像素走向确定算法对于绘制效率的影响也相对较小。

表 3 中显示分别使用 GBP07^[2]、PSSM^[3]、HEM8^[4] 和本文算法对这 4 个模型进行软影绘制的统计结果。可以看到,对于面片数较少的模型,本文算法在绘制效率上有了一定程度的提升。而对于面片数很大的模型,本文算法也不会显著降低整体的绘制效率。

表 1 本文轮廓边提取算法效率统计

Table 1 Performances of our silhouette extracting algorithm

模型	OSE		OSE + D		D-Cost
	<i>v</i> /(帧/s)	<i>t</i> /ms	<i>v</i> /(帧/s)	<i>t</i> /ms	<i>t</i> /ms
Knight	449.33	2.23	442.14	2.27	0.04
Plant	365.92	2.73	342.77	2.92	0.19
Tree	285.02	3.51	257.01	3.89	0.38
Dragon	45.30	22.08	44.13	22.66	0.58

表 2 ALE 算子提取轮廓边效率统计

Table 2 Performances of the ALE operator

模型	ALE		ALE + D		D-Cost
	<i>v</i> /(帧/s)	<i>t</i> /ms	<i>v</i> /(帧/s)	<i>t</i> /ms	<i>t</i> /ms
Knight	240.36	4.16	209.50	4.77	0.61
Plant	224.01	4.46	192.33	5.20	0.74
Tree	205.99	4.85	180.35	5.54	0.69
Dragon	100.64	9.94	94.35	10.59	0.64

表 3 4 种软影绘制算法的统计数据

Table 3 Performances of 4 soft shadow rendering algorithms

模型	GBP07	PSSM	HEM8	本文算法 /(帧/s)
Knight	19.36	45.80	65.89	71.30
Plant	6.55	18.39	22.61	24.73
Tree	5.74	17.95	22.04	23.17
Dragon	3.5	17.5	18.53	15.63

相比于基于深度图的 ALE 算子,本文算法是基于物体空间的,不会由于采样而丢失部分信息,因此提取的轮廓边信息更加精确(图 10)。

图 10 中三角形为一个三角面片,图中每个网格表示一个像素,乘号表示像素中心,深色填充的网格表示图像空间中的轮廓边像素。可见使用本文算法得到的轮廓边像素更加精确,轮廓边信息更加精确,

绘制阴影的质量也有提高(图 11)。图 11(a)为使用 ALE 算子提取轮廓边,边界信息不足导致所绘制阴影呈条纹状,如图 11(b)所示;图 11(d)使用本文算法提取的轮廓边,边界信息完整,所绘制阴影较柔和,如图 11(c)所示。

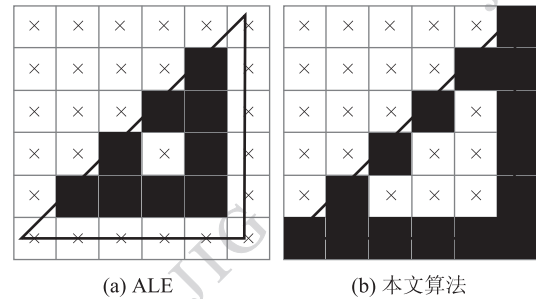


图 10 ALE 算子和本文算法所提取轮廓边的对比

Fig. 10 Differences between our silhouette extracting algorithm and ALE

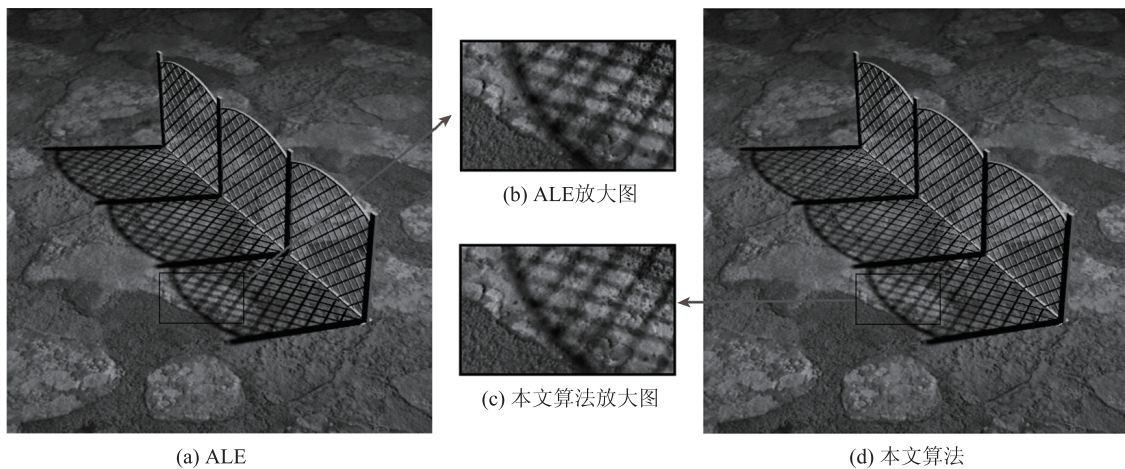


图 11 使用 ALE 算子和本文算法提取轮廓边后绘制阴影质量的对比

Fig. 11 Shadow quality comparison between our algorithm and ALE

6 结 论

真实感绘制中软影绘制算法的效率一直是众多学者关注的问题。基于使用逻辑运算代替纹理采样的原则,提出了一种混合物体空间和图像空间的快速软影算法。该算法在物体空间内提取面光源的轮廓边,然后把这些轮廓边投影到图像空间,进而基于 HEM 的思想,实现了一种新型的快速软影绘制算法。

本文算法仅是一种近似的软影算法,受到了 HEM 缺点的制约,同时也无法解决困扰软影映射算

法的单采样问题。今后,将尝试寻找一个自适应的方法来克服 HEM 的缺点。同时,也将尝试使用多张阴影映射图来解决单采样的问题。

参考文献 (References)

- [1] Guennebaud G, Barthe L, Paulin M. Real-time soft shadow mapping by backprojection [C]//The 17th Eurographics Symposium on Rendering. Aire-la-Ville: Eurographics Association Press, 2006: 227-234. [DOI: 10.2312/EGWR/EGSR06/227-234]
- [2] Guennebaud G, Barthe L, Paulin M. High-quality adaptive soft shadow mapping [J]. Computer Graphics Forum, 2007, 26(3):

- 525-533. [DOI: 10.1111/j.1467-8659.2007.01075.x]
- [3] Zhou W, Liu L, Li H. Fast soft shadow mapping by backprojection[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2011, 23(8): 1285-1293. [周炜, 刘浏, 李华. 快速反投影软影绘制算法[J]. *计算机辅助设计与图形学学报*, 2011, 23(8): 1285-1293.]
- [4] Liu L, Zhou W, Li H. Soft shadow mapping by line fitting[J]. *Journal of Computer-Aided Design & Computer Graphics*, 2012, 24(12): 1533-1541. [刘浏, 周炜, 李华. 基于轮廓边拟合的软影映射算法[J]. *计算机辅助设计与图形学学报*, 2012, 24(12): 1533-1541.]
- [5] Hasenfratz J, Lapierre M, Holzschuch N, et al. A survey of real-time soft shadows algorithms [J]. *Computer Graphics Forum*, 2003, 22(4): 753-774. [DOI: 10.1111/j.1467-8659.2003.00722.x]
- [6] Crow F C. Shadow algorithms for computer graphics [C]//*Proceedings of Annual Conference on Computer Graphics and Interactive Techniques*. ACM SIGGRAPH. New York: ACM Press 1977: 242-248 [DOI: 10.1145/965141.563901]
- [7] Williams L. Casting curved shadows on curved surfaces [C]//*Proceedings of Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM Press, 1978: 270-274. [DOI: 10.1145/965139.807402]
- [8] Akenine-Möller T, Assarsson U. Approximate soft shadows on arbitrary surfaces using penumbra wedges [C]//*Proceedings of the 13th Eurographics Workshop on Rendering*. Aire-la-Ville: Eurographics Association Press, 2002: 297-306.
- [9] Assarsson U, Akenine-Möller T. A geometry-based soft shadow volume algorithm using graphics hardware [J]. *ACM Transactions on Graphics*, 2003, 22(3): 511-520. [DOI: 10.1145/882262.882300]
- [10] Assarsson U, Dougherty M, Mounier M, et al. An optimized soft shadow volume algorithm with real-time performance [C]//*Proceedings of the Eurographics Conference on Graphics Hardware*. Aire-la-Ville: Eurographics Association Press, 2003: 33-40.
- [11] Forest V, Barthe L, Paulin M. Realistic soft shadows by penumbra-wedges blending [C]//*The 21st Eurographics Symposium on Graphics Hardware*. New York: ACM Press, 2006: 39-46. [DOI: 10.1145/1283900.1283907]
- [12] Chan E, Durand F. An efficient hybrid shadow rendering algorithm [C]//*The 15th Eurographics Symposium on Rendering*. Aire-la-Ville: Eurographics Association Press, 2004: 1-11.
- [13] Wyman C, Hansen C. Penumbra maps: approximate soft shadows in real-time [C]//*Proceedings of the 14th Eurographics Workshop on Rendering*. Aire-la-Ville: Eurographics Association Press, 2003: 202-207.
- [14] Lv W W, Meng W L, Xue G C, et al. Real-time approximate soft shadow rendering on GPU [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2009, 21(3): 275-288. [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): 275-288.]
- [15] Blythe D. The Direct3D 10 system [C]//*Proceedings of Annual Conference on Computer Graphics and Interactive Techniques*. New York: ACM Press, 2006: 724-734. [DOI: 10.1145/1179352.1141947]
- [16] Sun M Y, Lu W W, Liu X H, et al. Approximate soft shadow algorithm based on depth peeling and GPU [J]. *Journal of Image & Graphics*, 2010, 15(9): 1391-1397. [孙明彦, 吕伟伟, 刘学慧, 等. 深度剥离与 GPU 结合的近似软影算法[J]. *中国图象图形学报*, 2010, 15(9): 1391-1397]
- [17] Atty L, Holzschuch N, Lapierre M, et al. Soft shadow maps: efficient sampling of light source visibility [J]. *Computer Graphics Forum*, 2006, 25(4): 725-742. [DOI: 10.1111/j.1467-8659.2006.00995.x]
- [18] Schwarz M, Stamminger M. Bitmask soft shadows [J]. *Computer Graphics Forum*, 2007, 26(3): 515-524. [DOI: 10.1111/j.1467-8659.2007.01074.x]
- [19] Yang B, Feng J, Guennebaud G, et al. Packet-based hierarchical soft shadow mapping [J]. *Computer Graphics Forum*, 2009, 28(4): 1121-1130. [DOI: 10.1111/j.1467-8659.2009.01489.x]
- [20] McGuire M. Observations on silhouette sizes [J]. *Journal of Graphics Tools*, 2004, 9(1): 1-12. [DOI: 10.1080/10867651.2004.10487594]
- [21] Laine S, Aila T, Assarsson U, et al. Soft shadow volumes for ray tracing [C]//*Proceedings of Annual Conference on Computer Graphics Interactive Techniques*. New York: ACM Press, 2005: 1156-1165. [DOI: 10.1145/1186822.1073327]