第 42 卷　第 2 期　　计　算　机　工　程　　2016 年 2 月
Vol. 42　No. 2　　Computer Engineering　　February 2016

·图形图像处理·　　文章编号：1000-3428(2016)02-0242-07　　文献标识码：A　　中图分类号：TP391

# 混合场景中阴影绘制方法研究

陈国栋，叶东文，党琪琪

（福州大学物理与信息工程学院，福州 350116）

**摘　要：**为使基于远程与本地绘制的场景融合后更具真实感，结合移动终端场景简单、对象交互性强等特点，提出一种新的阴影绘制方法。服务器端采用平行分割阴影映射和百分比渐进滤波（PCF）算法使绘制出的阴影在真实感与实时性之间达到平衡，移动终端利用阴影图和 PCF 算法完成阴影绘制。根据服务器端和移动终端场景的实际相对位置关系在移动终端进行场景融合，使呈现出的阴影效果在保持一致性的同时更具真实感。实验结果表明，该方法可在移动终端实现不同场景的无缝融合，具有较高的绘制效率。

**关键词：**本地绘制；远程绘制；混合场景；阴影绘制；无缝融合

**中文引用格式：**陈国栋，叶东文，党琪琪. 混合场景中阴影绘制方法研究[J]. 计算机工程，2016，42(2)：242-248.

**英文引用格式：**Chen Guodong，Ye Dongwen，Dang Qiqi. Research on Shadow Rendering Method in Hybrid Scene[J]. Computer Engineering，2016，42(2)：242-248.

# Research on Shadow Rendering Method in Hybrid Scene

CHEN Guodong，YE Dongwen，DANG Qiqi

（College of Physics and Information Engineering，Fuzhou University，Fuzhou 350116，China）

**【Abstract】** In a hybrid scene of local and remote rendering，in order to make the scene more realistic，a shadow rendering method is proposed based on the characteristics of simple mobile terminal scene，and strong object interaction. On the server side，a combination of improved Parallel Split Shadow Mapping（PSSM）and Percentage Closer Filtering（PCF）algorithm is used to improve the unity of the reality and real-time shadow rendering. A combination of classic Shadow Mapping（SM）and PCF algorithm is used to draw a shadow on mobile terminals. Then the integration is done on mobile terminal based on the actual relative position of the two scenes which are processed on server side and mobile side. Experimental results show that the seamless fusion of scene can be realized on the mobile terminal through the proposed method，and it brings higher efficiency when drawing graph.

**【Key words】** local rendering；remote rendering；hybrid scene；shadow rendering；seamless fusion

**DOI：**10.3969/j.issn.1000-3428.2016.02.043

## 1　概述

近年来，随着通信技术的不断发展，以平板电脑和智能手机为代表的移动设备逐渐普及，人们对于在小屏幕移动设备上显示高质量图形图像的需求日益增加[1]。与资源配置较高的服务器和桌面 PC 相比，移动终端的各种资源极为有限，导致在移动终端绘制复杂三维场景存在诸多困难。

目前，虚拟场景中使用的阴影绘制算法很多都源于阴影图（Shadow Mapping，SM）算法和阴影体（Shadow Volume，SV）算法。阴影图算法是一种基于图像空间的阴影绘制方法，其绘制效率高，且不受场景复杂度和多边形数量限制，但容易引起走样问题[2]。阴影体算法是一种基于对象空间的精确算法[3]，该算法能产生精确的硬阴影，但随着场景复杂度的增加，资源消耗也会逐渐增加，绘制效率逐渐变低。根据以上 2 种算法的特点，近年来不少国内外学者在此基础上提出新的阴影绘制方法。文献[4]采用平均分割和对数分割的方法改进了阴影图算法中的透视走样问题。文献[5]结合阴影图和阴影体算法提出一种混合绘制方法，该方法先在物体空间中提取面光源的轮廓边，然后在图像空间中确定轮廓边的像素走向，最后通过反投影拟合的轮廓边计算最终的几何可见性。文献[6]提出一种基于实时

**基金项目：**福建省自然科学基金资助项目（2013J05090）。

**作者简介：**陈国栋（1979－），男，副研究员、博士，主研方向为图像分析与处理、计算机图形学；叶东文、党琪琪，硕士研究生。

**收稿日期：**2014-12-30　　**修回日期：**2015-02-12　　**E-mail：**fzucgd@126.com

卷积计算的阴影绘制方法来展现动态场景中物体的阴影细节。文献[7]通过压缩阴影图的方法实现阴影滤波过程中对大滤波核的加速。

由于上述方法计算比较复杂,对硬件资源的要求较高,因此当绘制设备为移动终端时,表现力欠佳。对此,结合移动终端场景简单以及对象交互性强的特点,本文在移动终端采用绘制效率较高的经典阴影图算法,并通过百分比渐进滤波(Percentage Closer Filtering,PCF)算法对其滤波,避免产生走样问题。在服务器端绘制阴影时,考虑到服务器端绘制的场景较大,同时服务器具有资源配置高的优点,故在服务器端采用平行分割阴影映射(Parallel Split Shadow Mapping,PSSM)算法进行阴影绘制,并根据实际情况采用不同大小的 PCF 滤波模板进行滤波。当服务器端把渲染好的场景发送至移动终端时,移动终端再根据 2 个场景的实际相对位置关系进行融合,从而实现 2 个场景的无缝融合。

## 2　阴影映射算法

### 2.1　标准阴影映射算法

阴影映射的原理是从光源可以看到场景中所有被光照亮的表面,所有隐藏元素都处于阴影区域。该算法的实现过程是对场景进行 2 遍绘制[8],具体步骤如下:

(1)以光源作为视点对场景进行绘制,创建相应的阴影图,阴影中的每一个元素记录了第 1 个可见表面的深度值,即与光源的距离。

(2)从真实视点的角度对场景进行绘制,对于每个绘制点,计算其在光源裁剪空间的深度值 $D_2$,将其与步骤(1)生成的阴影图中该点对应的深度值 $D_1$ 作比较,若 $D_2 > D_1$,则该点处于阴影中;若 $D_2 < D_1$,则该点不在阴影中。

虽然阴影图映射算法绘制效率高,但也存在基于图像空间固有的缺点。如果阴影图的分辨率低于采样的需求,则会产生走样,具体表现为锯齿状阴影边界[9]。

### 2.2　PSSM 算法

当绘制场景较大时,通过 SM 算法渲染出来的阴影走样较为严重,对此 Zhang Fan[10] 等人于 2006 年提出 PSSM 算法。其主要思想是首先利用与视椎体近平面相平行的若干个面将视椎体分割成多个不同区域,然后针对每个区域创建相应的深度纹理图,最后在深度测试时根据实际物体 $P$ 所处位置调用不同的深度纹理图。具体如图 1 所示,其中,物体 $P$ 垂直的箭头表示物体表面法向量。

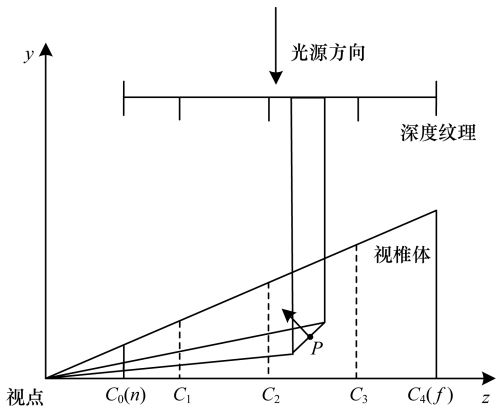



**图 1　场景分割示意图**

在图 1 中,视椎体被分割成了 4 个部分,$C_0$,$C_1$,$C_2$,$C_3$ 和 $C_4$ 分别对应分割面在 $z$ 轴上的坐标,每个区域对应一张深度纹理图。场景的分割方式不同,直接影响了场景对深度纹理的采样精度,分割数目越多,采样越精确,同时场景的绘制效率也越低。为了使 PSSM 算法在实际中有更好的应用,文献[10]提出一种将平均分割与对数分割相结合的方法来对场景进行划分。

$$C_i = \frac{n + (f-n)\frac{i}{m} + n\left(\frac{f}{n}\right)^{\frac{i}{m}}}{2} + \delta_{\text{bias}} \quad (1)$$

其中,$n$ 为场景中视椎体的近平面在 $z$ 轴上的坐标;$f$ 为远平面坐标;$m$ 为划分的区域数目;$i$ 的取值范围为 $0 \sim m$;$\delta_{\text{bias}}$ 为偏移量,用于调整分割面在实际中的位置。

## 3　远程阴影绘制

在上述对传统 PSSM 算法的介绍中可知,场景分割数量 $m$ 和各分割区域对应的深度纹理图的分辨率 $r_i$ 都需人为主观确定,而这 2 个量的取值对场景的绘制效率起到了决定性作用。在传统 PSSM 算法的基础上,将文献[11]中的场景分割方式应用到本文中,以初步完成阴影绘制中的反走样处理,然后再利用改进的 PCF 方法对阴影进行滤波,从而绘制出效果较为理想的软阴影。下面针对新的场景分割方法与 PCF 滤波方法进行说明。

### 3.1　新的场景分割方法

根据上文对 SM 算法中走样问题的分析可以看出,投影走样与物体表面法向量有直接关系,而物体的变化是无法控制的,导致投影走样难以消除,因此,针对透视走样提出新的场景分割方法。在标准的阴影图算法中,由于距离视点越近,走样越严重,因此当对场景分割时应保证每个分割区域中距离视点最近的点满足深度纹理的采样要求。其中,令每个分割区域对应的深度纹理图的分辨率为 $r_i \times r_i$;眼

睛空间的屏幕分辨率为 $r_{s_1} \times r_{s_2}$。

本文用 $C_i$ 和 $C_{i+1}$ 所在的 2 个平面之间的区域表示第 $i$ 个分割区域。为保证能消除场景中走样最严重的地方,对于第 $i$ 个分割区域而言,必须消除位于 $C_i$ 处的走样现象。具体如图 2 所示,其中,与物体 $P$ 垂直的箭头表示物体表面向量;与 $z$ 轴平行的箭头表示视线方向;与 $y$ 轴平行的箭头表示光的逆反方向。

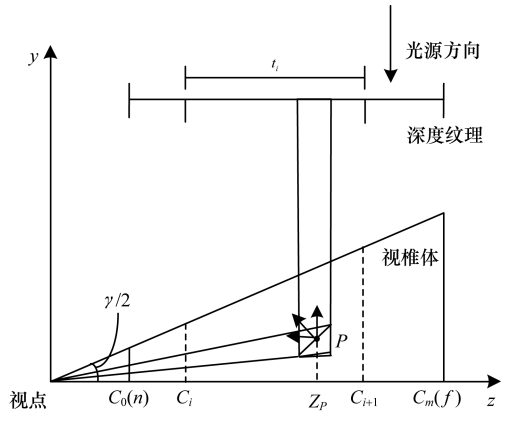



**图 2 场景分割后的走样原理示意图**

为了更好地说明问题,将物体 $P$ 及相关信息放大并在图 3 中进行展示,其中,$s$ 为物体 $P$ 所对应的屏幕长度;$l$ 为其在深度纹理中对应的长度。

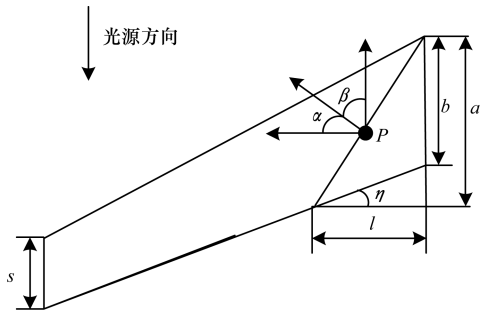



**图 3 局部区域放大示意图**

根据上文分析,可以得到物体 $P$ 的走样层度为:

$$m = \frac{s}{l} \cdot \frac{t_l}{t_s} \tag{2}$$

其中,$t_s$ 和 $t_l$ 分别为屏幕和深度纹理的实际长度;$m$ 值越大表示走样越严重。下面根据场景分割的情形推导 $m$ 关于 $z_p$ 的表达式。根据图 2、图 3,依据三角形相似原理有以下等式成立:

$$\frac{s}{b} = \frac{n}{z_P} \tag{3}$$

$$a = b + l\tan\eta \tag{4}$$

$$a = l\tan\beta \tag{5}$$

从而推出:

$$\omega = \frac{n(\tan\beta - \tan\eta)}{z_P} \cdot \frac{(C_{i+1} - C_i)}{2n\tan(\gamma/2)} \tag{6}$$

为消除物体 $P$ 的透视走样,必须使 $\omega$ 满足以下条件:

$$\omega \leqslant \frac{r_i}{r_{s_1}} \tag{7}$$

其中,$r_{s_1}$ 为屏幕在 $y$ 方向上的分辨率大小,又由图 3 可知 $\eta$ 的取值范围为 0 ~ $\gamma/2$,因此 $\omega$ 一定满足式(8):

$$\omega \leqslant \frac{\tan\beta(C_{i+1} - C_i)}{2z_P\tan(\gamma/2)} \tag{8}$$

为保证物体 $P$ 不出现走样,于是有式(9)成立:

$$\frac{\tan\beta(C_{i+1} - C_i)}{2z_P\tan(\gamma/2)} \leqslant \frac{r_i}{r_{s_1}} \tag{9}$$

此处,由于不考虑对投影走样的优化问题,因此令 $\tan\beta = 1$。同时为了能使物体 $P$ 在 $C_i$ 处也不出现走样,令 $z_P = C_i$。此时式(9)变成:

$$\frac{C_{i+1} - C_i}{C_i\tan(\gamma/2)} \leqslant \frac{2r_i}{r_{s_1}} \tag{10}$$

由于屏幕分辨率、眼睛空间以及视椎体的视角都是固定的,因此 $\tan(\gamma/2)$ 和 $r_{s_1}$ 可以看成常量。又由于场景的初始分割点为 $C_0 = n$,最终分割点为 $C_m = f$,因此可得各个区域的分割点在 $z$ 轴上的坐标与深度纹理分辨率之间的关系为:

$$\frac{C_{i+1}}{C_i} \leqslant d \cdot r_i \tag{11}$$

其中:

$$d = \frac{2\tan(\gamma/2)}{r_{s_1}} \tag{12}$$

由此可知,在确定每个分割区域深度纹理分辨率的情况下利用迭代快速地计算出每个分割区域的位置,或者在确定场景划分方式的情况下可以求出每个分割区域需要的最小深度纹理分辨率。因此,利用该方法不需要人为主观划分场景和确定深度纹理分辨率,可操作性强。

### 3.2 基于 PSSM 的分层 PCF 滤波

PCF 算法是一种先比较再过滤的算法。首先找到每一个屏幕像素点在深度纹理图中的对应点;然后读取该点周围的采样纹理坐标值;最后将经过深度比较后得到的值进行双线性插值,确定阴影标志值。例如,一个 3 ×3 像素区域的 PCF 算法。点到光源的距离 $z$ = 49.8,滤波模板大小为 3 ×3 像素,对于深度图上的每一点采样其周围 8 个点,然后将这9 个点的深度值与图 2 的 $z$ 值进行比较,即进行 9 次深度测试,得到 9 个光照亮度权值并进行双线性插值,最后得到该点的 55% 区域在阴影内。

而基于 PSSM 算法的分层 PCF 滤波算法[12]是在得到每个分割区域的深度图后,采用不同大小的 PCF 滤波模板对每个对应的深度图进行 PCF 比较,

再进行 PCF 滤波,从而得到最后的阴影图,如图 4 所示,其中,$v_1$,$v_2$,$v_3$ 为视锥体的分割面;4 ×4,8 ×8,16 ×16,512 ×512,1 024 ×1 024 为 PCF 采样区域尺寸;水平箭头表示视点方向。

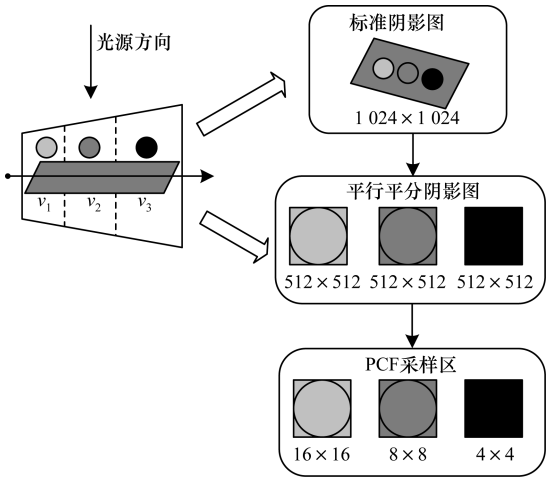



**图 4　基于 PSSM 的分层 PCF 滤波示意图**

在确定滤波模板 $F_i$ 前,必须先确定不同滤波模板间的渐进方式和渐进度,然后确定 $F_1$ 值,再根据选择的渐进度和渐进方式确定每个 $F_i$ 值,此处 $F_i \geq 1$,当 $F_i = 1$ 时,即为 1 ×1 PCF,此时不进行 PCF 滤波。例如,设置渐进度为 2 的等比渐进,同时令 $F_1 = 16 \times 16$,则可以知道 $F_2 = 8 \times 8$,$F_3 = 4 \times 4$,$F_4 = 2 \times 2$,$F_5 = 1 \times 1$。若设置渐进度为 1 的等差渐进,同时令 $F_1 = 5 \times 5$,则可以知道 $F_2 = 4 \times 4$,$F_3 = 3 \times 3$,$F_4 = 2 \times 2$,$F_5 = 1 \times 1$,如图 5 所示。

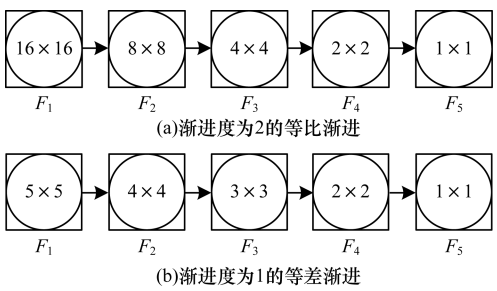



**图 5　滤波模板的选择**

在利用 PCF 算法进行滤波时,如果对屏幕上每一个像素点都进行过滤,则计算量会变得很大,导致绘制效率变低。因此,本文在利用 PCF 算法进行软影绘制时,只针对对视觉效果有重要影响的阴影边界的附近区域[13],此处的“附近区域”本文定义为距离图元边界 2 层像素宽度的区域。如图 6 所示,数字 1 所围成的区域为阴影区域,附近区域为由数字 0,1,2 表示的区域。判断附近区域的过程如下:在确定滤波模板的前提下判断距离当前像素最远的 4 个采样点的深度值,若 4 个采样点的值都为 0,则该像素点在阴影内部,不属于附近区域;若 4 个采样点的值都为 1,则该像素点在阴影外部,也不属于附近区域;若 4 个采样点的值中有 1 与 0,则表示该像素点在附近区域,需进行 PCF 滤波。

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | | | | | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | | | | | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | | | | | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | | | | | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 0 | 0 | |
| | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

**图 6　阴影边界附近区域**

### 3.3　远程阴影绘制实现步骤

图 7 为远程阴影绘制的整体流程。

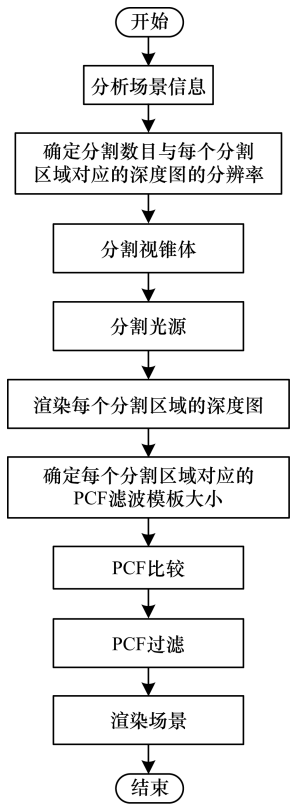



**图 7　远程阴影绘制流程**

远程阴影绘制的具体实现步骤如下:

(1)场景分析。在服务器端绘制阴影前首先需要分析场景信息,即确定场景中哪些物体会产生阴

影或者接受阴影。

(2)渲染深度图。首先根据场景大小利用标准阴影映射算法确定出分割块数目与每个分割区域对应的深度图分辨率,从而确定每个分割区域 $V_i$ 在 $z$ 轴上对应的坐标。然后利用传统 PSSM 算法中光源区域的分割方法对现有场景的光源进行分割,即将光椎体 $W$ 分割为更小的区域 $\{W_i\}$,通过计算轴对齐包围盒 $V_i \cap W$ 得到 $W_i$,且覆盖当前的深度块。最后在得到集合 $\{W_i\}$ 后,对每个分割部分 $V_i$ 在空间 $W_i$ 上按照标准 SM 算法渲染得到每个区域对应的深度纹理图 $T_i$。

(3)基于 PSSM 算法的分层 PCF 滤波处理。首先对步骤(2)中产生的深度纹理图 $T_i$ 指定渐进方式、渐进度以及 $F_1$ 值,从而计算出每张深度纹理图对应的滤波模板大小。然后确定每张深度纹理图阴影边界的附近区域。最后采用 PCF 算法,依据 $F_i$ 对每张深度纹理图进行 PCF 比较和过滤,从而得到最终的阴影图。

## 4 本地阴影绘制

由于本地移动终端具有硬件条件资源不足的问题,如果采用比较复杂的阴影绘制算法,将导致绘制效率极低,甚至出现死机现象,又加上本地绘制的对象交互性强,因此对阴影的实时性要求较高。考虑到以上因素,本文在进行本地阴影绘制时采用标准 SM 算法与 PCF 滤波相结合的方法以达到效率与效果的折中。图 8 为本地阴影绘制的整体流程。

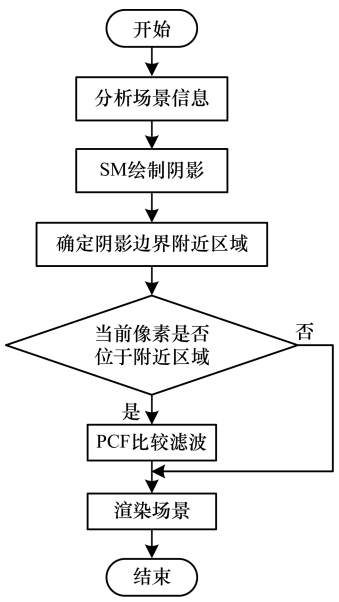



**图 8 本地阴影绘制流程**

本地阴影绘制的具体实现步骤如下:

(1)分析场景。由于本地场景一般只有一个交互对象,因此该对象在绘制时会产生阴影。

(2)渲染深度图。本地绘制过程中深度纹理图的渲染采用传统的 SM 算法,该算法的具体实现过程见 2.1 节中的描述。

(3)PCF 滤波。在进行 PCF 滤波时依旧只针对阴影边界的附近区域进行滤波处理,阴影边界的判断过程参考 3.2 节中所述方法。如果当前像素点位于阴影边界区域,则该像素的阴影深度系数采用指定的滤波模板来更新;如果当前像素不在阴影区域,则从真实视点的角度对场景进行绘制,计算出该像素在光源裁剪空间的深度值。

## 5 场景融合

为了使绘制的场景更具真实感,本文在实现了本地绘制场景与远程绘制场景共享同一光照环境的基础上,分别对其进行阴影绘制,并最终在移动终端上进行 2 个场景的融合。在场景融合后,为实现不同场景绘制的阴影具有相同或者相近的效果,在不同场景进行阴影绘制时采用相同的参数值,如相同的彩色空间(RGBA)等。由于远程绘制阴影时不同的分割区域所采用的 PCF 滤波模板不同,因此当移动终端所绘制的交互对象移动到相应的分割区域时,应使用该分割区域在远程绘制时所对应的滤波模板来更新原来的滤波模板,从而使得此时混合场景中的阴影效果保持一致性。考虑到移动终端硬件资源条件不足,在对本地阴影进行绘制时限制了所使用的滤波模板大小,规定最大为 4×4 像素。此时,若所处分割区域在远程绘制时所对应的滤波模板超过 4×4 像素,则在本地绘制时依旧采用 4×4 像素的滤波模板;当小于 4×4 像素时,则以实际分割区域所对应的滤波模板对原来的滤波模板进行更新,具体流程如图 9 所示。

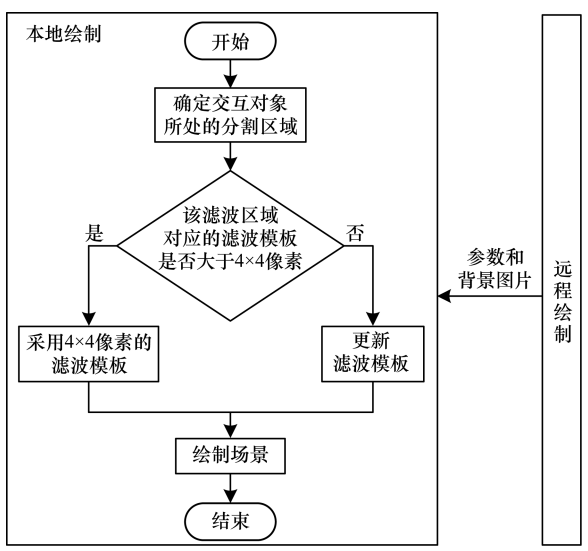



**图 9 混合场景阴影绘制流程**

## 6　实验结果与分析

本文采用上述方法在移动终端利用 Webstorm 开发平台和 WebGL API 实现本地场景的阴影绘制,在服务器端利用 VS2013 开发平台和 OpenGL 的高级着色语言(GLSL)实现远程场景的阴影绘制。移动设备采用 iPhone 5S,服务器配置如下:处理器为 Intel Xeon(至强) L5639 @ 2.13 GHz 双核(X4),主板为微软 Virtual Machine,内存 2 GB。远程阴影绘制时场景分割数目为 3,每个分割区域对应的阴影图分辨率分别为 1 024 × 1 024 像素、512 × 512 像素、256 × 256 像素,采用渐进度为 2 的等差渐进方式,且 $F_1 = 5 \times 5$,场景中三角形面片总数为268 936。本地阴影绘制时,所采用阴影图分辨率为1 024 × 1 024 像素,采用的初始滤波模板大小为4 × 4像素,场景中三角形面片总数为 4 079。

图 10 为在服务器端采用传统 PSSM + PCF 方法时的阴影边界图与采用本文所述方法时的阴影边界图[11]。可以看出,采用本文所述方法进行远程阴影绘制时可以绘制出更为柔和的阴影。表 1 为采用传统方法与本文方法时的速率对比,可以看到,采用本文方法进行远程阴影绘制时速度更快。



(a)传统方法



(b)本文方法

**图 10　远程阴影绘制效果比较**

**表 1　远程阴影绘制速率比较**

| 方法 | 场景三角形面片数 | 滤波模板大小/像素 | 绘制帧率/(f · s$^{-1}$) |
|---|---|---|---|
| 传统方法 | 268 936 | 5 × 5 | 68 |
| 本文方法 | 268 936 | 5 × 5 | 97 |

图 11 为手机端绘制的交互对象,图 12 为服务器端渲染的完整背景图片,两者在绘制时均采用相同的光照模型。图 13 为两者在移动终端融合之后的效果,此时在移动终端的阴影绘制速率达到38 f/s 左右。图 14 为整个完整场景都在服务器端渲染之后的效果,可以看出,采用本文方法在移动终端融合之后的视觉效果与整个完整场景都在服务器端渲染的视觉效果几近相同,说明通过本文方法融合效果较为理想。图 15 为当交互对象移动到其他位置时的融合效果,其中有一部分阴影出现重叠,与未重叠的阴影相比,重叠部分的阴影颜色较深。比较图 13 和图 15 可以看出,当交互对象处于不同位置时,交互对象与服务器端所渲染的场景都能共享同一个光照环境,且绘制出的阴影具有一致性。

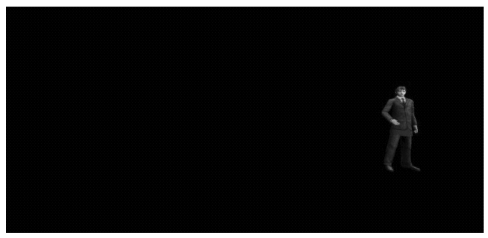


**图 11　移动终端所绘交互对象**



**图 12　服务器端所绘背景**



**图 13　阴影未叠加时的混合场景**



**图 14　服务器端所绘完整场景**



**图 15　部分阴影叠加时的混合场景**

## 7 结束语

在移动环境下,为了使远程绘制的场景与本地移动终端绘制的场景融合后具有更强的真实感,本文针对移动终端与服务器的特点分别采用不同的方法绘制阴影,并最终在移动终端进行场景融合,使得在移动终端上呈现出的阴影效果在保持一致性的同时更具真实感。实验结果表明,本文方法具有较高的绘制效率。今后将针对如何提高移动图形的绘制速率方面做进一步研究。

### 参考文献


[1] 潘志庚,何高奇,杨柏林,等. 移动设备图形技术的研究进展[J]. 计算机辅助设计与图形学学报,2008,20(3):273-280.

[2] 付燕丽,刘 循. 一种基于 GPU 的柔和阴影实现方法研究[J]. 计算机技术与发展,2013,23(2):52-56.

[3] 程利利. 三维室内场景中实时阴影算法的研究及应用[D]. 南京:东南大学,2010.

[4] 张 伟,奚俊峰. 基于 Shadow Maps 的实时阴影生成算法改进[J]. 兰州交通大学学报,2011,30(1):22-26.

[5] 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.

[6] Daniel P, Veronika Š, Jan M, et al. Instant Convolution Shadows for Volumetric Detail Mapping [J]. ACM Transactions on Graphics,2013,32(5):13-15.

[7] Erik S, Viktor K, Ola O, et al. Compact Precomputed Voxelized Shadows[J]. ACM Transactions on Graphics, 2014,33(4):1-8.

[8] 刘 洋,刘学慧,吴恩华. 基于自适应分割的动态面光源图像空间绘制[J]. 计算机工程与设计,2012,33(2):624-629.

[9] 李 晗,武奇生. 融合颜色相关性和纹理差异的阴影检测方法[J]. 计算机工程,2011,37(15):146-148.

[10] Zhang Fan, Sun Hanqiu, Xu Leilei, el al. Parallel-split Shadow Maps for Large-scale Virtual Environments[C]//Proceedings of ACM International Conference on Virtual Reality Continuum and Its Applications. New York, USA: ACM Press, 2006:311-318.

[11] 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.

[12] 王 英. 实时动态软阴影技术研究与应用[D]. 成都:电子科技大学,2010.

[13] 张方彦,杨 猛,刘金刚. 一种改进的实时软阴影算法的设计与实现[J]. 空军工程大学学报:自然科学版,2012,13(2):70-73.


编辑 陆燕菲

(上接第 241 页)

### 参考文献


[1] Yang Lianqiang, Zeng Xiaoming. Bézier Curves and Surfaces with Shape Parameters[J]. International Journal of Computer Mathematic,2009,86(7):1253-1263.

[2] Han Xuli. Cubic Trigonometric Polynomial Curves with a Shape Parameter [J]. Computer Aided Geometric Design,2004,21(6):535-548.

[3] Chen Xiaodiao, Ma Weiyin, Xu Gang, et al. Computing the Hausdorff Distance Between Two B-spline Curves [J]. Computer-aided Design,2010,42(12):1197-1206.

[4] 尹池江,檀结庆. 带多形状参数的三角多项式均匀 B 样条曲线曲面[J]. 计算机辅助设计与图形学学报,2011,23(7):1132-1138.

[5] 王文涛,汪国昭. 带形状参数的三角多项式均匀 B 样条[J]. 计算机学报,2005,28(7):1193-1198.

[6] 李军成. 一种构造任意类三次三角曲线的方法[J]. 小型微型计算机系统,2011,32(7):1442-1445.

[7] 王成伟. 带有参数的三次三角多项式样条曲线[J]. 北京服装学院学报,2008,28(3):51-55.

[8] 梁锡坤. 基于曲线组合的 3 次均匀 B 样条曲线的拓展[J]. 中国图象图形学报,2011,16(1):118-123.

[9] 屠 宏,耿国华. 一种基于局部特征的三维模型检索算法[J]. 计算机工程,2015,41(3):218-222.

[10] 刘华勇,李 璐,张大明. 任意阶参数连续的三角多项式样条曲线曲面调配[J]. 浙江大学学报:理学版,2014,41(4):413-418.

[11] 吕勇刚,汪国昭,杨勋年. 均匀三角多项式 B 样条曲线[J]. 中国科学:E 辑,2002,32(2):282-288.

[12] 张 莉,刘静静,檀结庆. 多形状参数的指数均匀 B 样条曲线曲面[J]. 图学学报,2013,34(3):29-35.

[13] Qin Xinqiang. A Novel Extension to the Polynomial Basis Functions Describing Bezier Curves and Surfaces of Degree n with Multiple Shape Parameters[J]. Applied Mathematics and Computation,2013,223(3):1-16.


编辑 刘 冰