# 从《全境封锁》中讨论软阴影技术

## 硬阴影与软阴影

在实时绘制中，阴影被用于体现物体的层次感，增加绘制的真实感。在传统的局部光照模型下，任何物体都直接地和光源进行亮度计算，而忽略了物体，光源之间地互相遮挡关系。阴影正是由于这种互相遮挡关系而产生的。因而在绘制一个物体时，不仅要考虑其和光源之间的关系，还要考虑其和光源之间是否存在其它物体产生了遮挡。为了解决这样的遮挡问题，诞生了两种主要的硬阴影算法。阴影图算法以及阴影体算法。阴影图利用了Z缓冲，记录了以光源为相机视角的深度图，因而当任意物体需要计算光照时，可以追溯到阴影图中判断是否为阴影。阴影体利用光源照射到物体后在物体背面形成一个阴影锥体区域，而将锥体信息映射到Z缓冲上，可以做到直接在相机坐标下计算阴影。

而上述算法产生的是硬阴影，即阴影边界的变化比较突兀，并且边界可能会存在较多的锯齿。因而在实际应用中，取而代之的是相应的软阴影算法。软阴影即边界较为平滑的阴影，边界部分会存在一个逐渐变化的过程。这也与现实中的情况比较相符，因而在实时阴影绘制中，主要采用的是软阴影技术。

## 百分比邻近滤波

百分比邻近滤波（Percentage Closer Filtering，PCF）是一个最早在1987年提出的一系列软阴影算法。其中心思想是对阴影图进行滤波。[1]提出了调换深度比较和滤波的顺序。在传统的纹理滤波中，对于任意一个传入坐标，我们都可以利用插值或滤波得到该坐标对应的纹理值，然后用该纹理值作为深度和物体深度作比较，由于比较的步骤在后，因而总是会返回二值结果，之前的滤波操作毫无意义，得到的阴影依然是硬阴影。而在PCF算法中，首先对于滤波核内的所有深度值，和物体深度进行比较，得到一张新的二值纹理，之后再对该二值纹理进行滤波得到阴影系数。假设完全处于阴影中计为1，完全暴露在外计为0，那么该滤波理论上可以提供在区间内的任意值作为阴影的混合系数，得到更多层次的阴影效果。而混合的方式是对光线进行衰减，1代表光线完全衰减至0，0代表光线完全没有衰减。其余的值按比例对光线进行削弱。如果对应多个光源，对每个光源都进行这样的操作，最终将所有光源的强度进行叠加用于计算该像素的最终颜色。

而作为PCF算法的扩充，Nvidia于2005年提出了百分比邻近软阴影算法（Percentage-Closer Soft Shadows，PCSS）[2]。PCSS对PCF进行了修改，将整体步骤分为3步：首先在被着色的点附近选取一片区域，获得区域中所有遮挡了被着色物体的光源空间中的深度；然后根据获取的深度计算半影系数，其中为被着色点在光源空间的深度，为上一步获得的遮挡体的深度，由于上一步对区域内的各点进行了选择，因而遮挡体必然比着色点的深度浅；最后一步则是对得到的半影系数进行PCF中的滤波得到该点的最终阴影系数。可以注意到，半影系数和物体与光源的距离以及光源的大小有关，光源越大，遮挡体离光源越紧，半影系数就越大，着色点位于阴影中的比例就越大，因为其引入了更多的控制参数，因而得到的阴影就更加具有平滑性，这也使得PCSS成为了主要的软阴影算法之一。

## 不规则Z缓冲

不规则Z缓冲（Irregular Z-Buffer， IZB）原则上是一个硬阴影算法，其概念最早与2005年[3]提出。由于传统的Z缓冲是均匀规则的，记录的是相机空间内均匀采样的深度值。然而其应用于阴影图时，由于阴影图是位于光源空间的，如果对光源空间进行均匀采样，那么这些采样点变换到相机空间后显然会变得不均匀，从而会使得部分区域采样点过少，最终产生出来的阴影会产生锯齿。而不规则Z缓冲可以弥补采样不均匀的缺陷，使得最终计算出来的阴影具有更少的锯齿。

IZB解决的问题是分布不均匀的采样点的存储问题。一般地，要存储不均匀地分布在空间中的物体主要使用空间加速结构，例如BSP树，K-D树，等，对这些点的访问进行加速。而IZB采用了一种网格与链表的方式进行存储。类似于均匀剖分的网格内记录了一个变长的数组。这种方案曾经被用于实现光线跟踪[4]以及光子映射[5]。采用这种方法的考量是基于：适应传统的光栅化管线；K-D树本身的面积重叠测试开销要大于采样点判断本身的开销；GPU指令集更适合做链表的遍历。

每个网格里的采样点以链表的形式存储，计算阴影时，首先获取相机空间内各点的深度；然后将每个点转换到光源空间，获得光源空间投影面的坐标以及光源空间内的深度；随后将该采样点加入IZB；利用IZB获取遮挡体的深度；最后用从IZB中获取的遮挡体深度和物体深度比较，从而计算是否处于阴影中。

在这之后，在IZB的基础上，又有人提出了基于锥体跟踪的IZB阴影。其基本思想来源于光想跟踪，不同之处是：光线跟踪追踪的是从摄像机向目标像素射出的光线，而锥体跟踪追踪的是摄像机朝目标像素占据的矩形发射的锥体，最后锥体会和由光线跟踪得到的交点所在的且平面相交。锥体和平面相交得到的区域中会有采样点，这些采样点相对于光源做可见性检测，根据被遮挡的采样点数对阴影进行抗锯齿，得到去锯齿的硬阴影。

尽管IZB实现的是硬阴影，我们可以对IZB进行模糊处理来实现软阴影。因为实质上PCSS就是通过一个动态大小的滤波器对阴影图进行滤波，从而实现软阴影，因而同样的方法也可以用在IZB的阴影图上使得IZB也可以生成软阴影。《全境封锁》中就是使用了这种技术来生成IZB下的软阴影。

## 《全境封锁》中的软阴影技术

在育碧的RPG游戏《汤姆克兰西：全境封锁》中，使用了一种混合的锥体跟踪阴影（Hybrid Frustum-Traced Shadows）。在这项技术中，其混合了上述两类经典的软阴影算法。其既采用了锥体追踪的方案，使用IZB实现软阴影，同时又保留了PCSS方法，通过计算半影系数近似地实现柔和的阴影。

由于规则的阴影图的偏移和锯齿在目标物体距相机比较近的时候更加明显，因而对于近处的物体，需要采用IZB的方法来减少错误的阴影，而为了实现软阴影，对不规则的阴影图进行模糊处理。然而IZB的方法相比于PCSS方法其运行效率更低，因而《全境封锁》实现了两种方法之间按照物体深度进行切换。当目标物体和遮挡物体的距离较近时，为了获得较好的效果，需要采用IZB方法；而当目标物体和遮挡物体距离较远时，则采用更高效的PCSS算法。而为了使在转换处附近PCSS的结果不会影响锥体阴影的结果，育碧使用了插值的方式来实现两种方法计算得到的阴影系数的平滑过渡。



图1 多种软阴影在游戏场景中的效果

图1 表示了不同方法的实现效果，对于近处的阴影，通过不规则的采样，阴影部分可以得到更为准确的结果，而PCSS由于其基于规则阴影图，尽管做了软阴影，但无法消除原本就有问题的部分。混合模型弥补了这一缺陷，使得近处的阴影能够正确显示，不过其模糊的效果则没有PCSS那样明显。

而在运行效率方面，由于引入了不规则的阴影图，其构建和查询比基于滤波的软阴影算法更为费时，因而整个场景的绘制中，混合模型的效率会明显低于只使用PCSS的效率。图1中左侧场景在1920\*1080的分辨率下的阴影绘制，混合模型需要8.1ms，而PCSS只需要5.7ms，但是混合模型带来的效果提升是很明显的。

## 纯软阴影算法

上述软阴影算法存在一个共同的特性，那就是都是基于传统硬阴影的方法改进而来，例如PCSS使用传统阴影图，利用动态大小滤波实现软阴影，IZB则本身就是硬阴影算法，加上了模糊才使得边缘变得平滑。正因为如此，上述方法实现的阴影都是和传统阴影的形状类似的。而除了这些方法之外还存在不利用真实阴影图进行阴影计算的方法。

球面谐波（Spherical Harmonic，SH）[6]可以用于计算软阴影。SH分解原本用于预计算光照中，将绘制方程中的光传输按照SH基进行分解，将积分化简为SH基的线性组合，从而通过存储每个基的系数，降低了光照计算量，实现静态物体的快速光照计算。而SH同样可以用于阴影计算[7]。对于任意一点，我们可以计算其四周的可见性函数。其原始结果完整记录了遮挡体的边界，因而若按照原始可见性函数进行绘制，得到的阴影就是硬阴影。若先对可见性函数进行SH分解，然后进行重构，原来遮挡体的边界就会被模糊化，此时若用新的可见性函数代替原有函数进行绘制方程求解，得到就是软阴影。

而接下来的问题则是如何对可见性函数进行采样。由于计算的离散性，我们无法获得连续的可见性函数，因而需要对物体周围进行采样，从而得到类似光场分布的采样结果，称为阴影场[8]。光场描述了光源发射的光线分布，而阴影场描述了遮挡体的遮挡分布。我们可以用预计算光照的方式得到预计算遮挡。对每个光源，其光场和相关的遮挡场相乘，然后乘以最终表面的BRDF可以得到该光源在该点的反射值，最后将这些反射值累加起来得到添加了阴影的最终颜色。而由于遮挡场经过了SH基的分解和重构，因而边缘是平滑的，反映到最终的光照反射上就可以形成软阴影。

此外，对于可见性函数的模糊还可以进一步优化。在遮挡场的采样的过程中我们使用了物体的真实边界，而物体往往不规则，因而其求交运算也较为复杂与低效。一个优化的手段为讲遮挡体用一组球体来近似[7]，由于球体的求交相当简单，因而对整个物体的可见性采样的复杂度就可以大大降低。而由于计算得到的可见性函数最终要经过SH分解重构，本身会进行模糊化，因而球体可以在效果损失较小的情况下获得更快的计算效率。更进一步，遮挡场的乘法由于是张量乘法，开销明显地高于加法，因而可以将原有的SH空间转变到对数空间，从而将乘法退化为加法来加速。

## 总结

软阴影技术目前大致上可以分为两种：基于原有硬阴影形状而柔化而成的阴影；根据绘制方程对遮挡场本身进行近似与模糊。前者在效果上更多地保留了遮挡物体的形状，而后者更多地描述了阴影的范围。两类方法几乎实在同一个时期获得的发展，而从近几年的情况来看，实际应用于3D游戏中的阴影，还是以前者为主。PCSS，IZB等使用的Z缓冲是一般绘制流水的一部分，而SH近似应该属于PBR体系的一部分。可以考虑到两者的结合：用遮挡体的球包围盒来更新Z缓冲从而获得近似的阴影图，再对阴影图进行模糊得到使用传统技术但是阴影更“软”的结果。而从另一个角度看，原本作为向准确度妥协的近似操作，在软阴影中恰好变成了我们需要的东西，从中可以看出合理的近似对于获得既符合要求又高效的重要性。

## 引用

[1] William T. Reeves, David 1-1. Salesin, Robert L. Cook. 1987. Rendering Antialiased Shadows with Depth Maps

[2] Randima Fernando. 2005. Percentage-closer soft shadows. In ACM SIGGRAPH 2005 Sketches.

[3] JOHNSON, G. S., LEE, J., BURNS, C. A., AND MARK, W. R. 2005. The irregular z-buffer: Hardware acceleration for irregular data structures. ACM Trans. Graph. 24, 4 (Oct.), 1462–1482.

[4] PURCELL, T. J.,BUCK, I.,MARK, W. R., AND HANRAHAN, P. 2002. Ray tracing on programmable graphics hardware. In SIGGRAPH’02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press, New York, NY, 703–712.

[5] PURCELL, T. J., DONNER, C., CAMMARANO, M., JENSEN, H. W., AND HANRAHAN, P. 2003. Photon mapping on programmable graphics hardware. In Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Conference on Graphics Hardware. Eurographics Association, Aire-la-Ville, Switzerland, 41–50.

[6] KAUTZ, J., SLOAN, P., AND SNYDER, J. 2002. Fast, arbitrary BRDF shading for low frequency lighting using spherical harmonics. In Proc. of the 13th Eurographics Workshop on Rendering, 291–296.

[7] REN, Z., WANG, R., SNYDER, J., ZHOU, K., LIU, X., SUN, B., SLOAN, P.-P., BAO, H., PENG, Q., AND GUO, B. 2006. Realtime soft shadows in dynamic scenes using spherical harmonic exponentiation. ACM Transactions on Graphics 25, 3 (July), 977–986.

[8] ZHOU, K., HU, Y., LIU, S., GUO, B., AND SHUM, H. 2005. Precomputed shadow fields for dynamic scenes. ACM Transactions on Graphics 24, 3, 1196–1201.