Real-Time Shadows

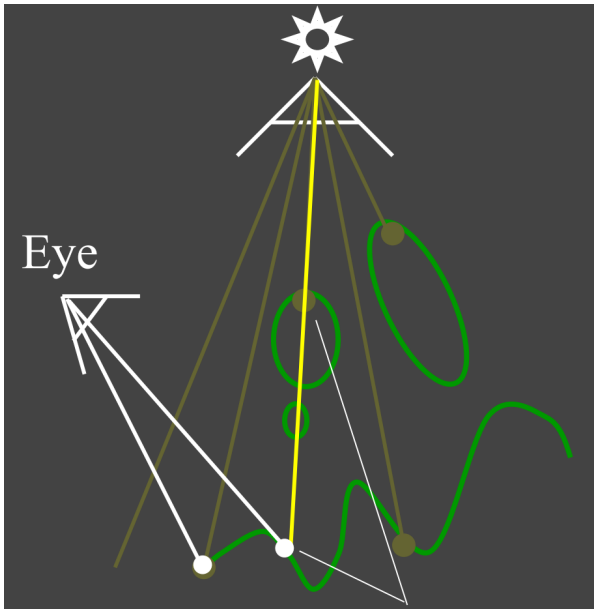
Mengzhu Wang

2024 年 5 月 13 日

1 Shadow Mapping

Shadow mapping 是一种 2-Pass 算法，即对场景渲染两次：

- 第一个 pass 从光源看向场景并记录深度图 shadow map
- 第二个 pass 从相机看向场景，像素对应的世界坐标系进行光源的 MVP 变换到光源的屏幕空间中，其 z' 值与 shadow map 的对应深度值 depth 比较，若 depth 小于 z'，则该像素的光被遮住形成阴影。



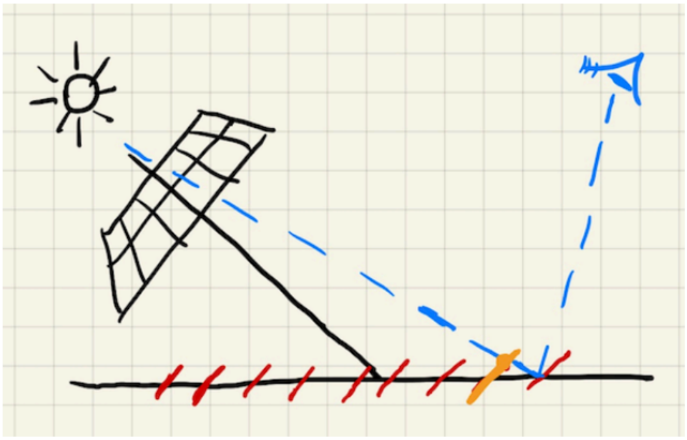
1.1 Problem

Shadow mapping 存在自遮挡（self occlusion）和走样（aliasing）问题。

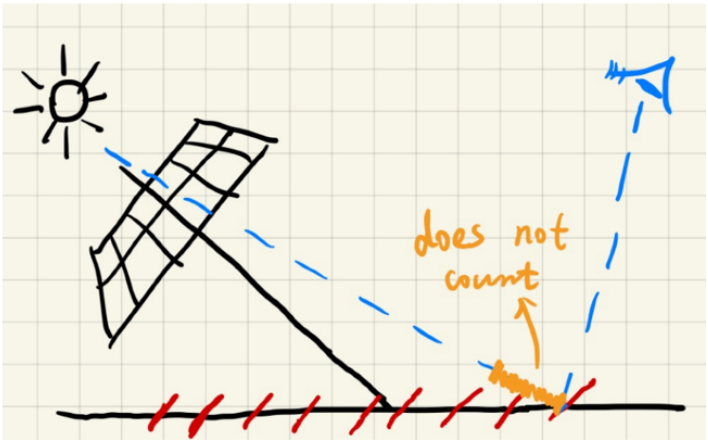
a) self occlusion

由于数值精度的限制且 shadow map 保存的是离散的深度值，也就是说 shadow map 上的每个采样点代表一块范围内图元的深度值，光线斜射到平面时，实际相当于发射到多个垂直的小平面上，

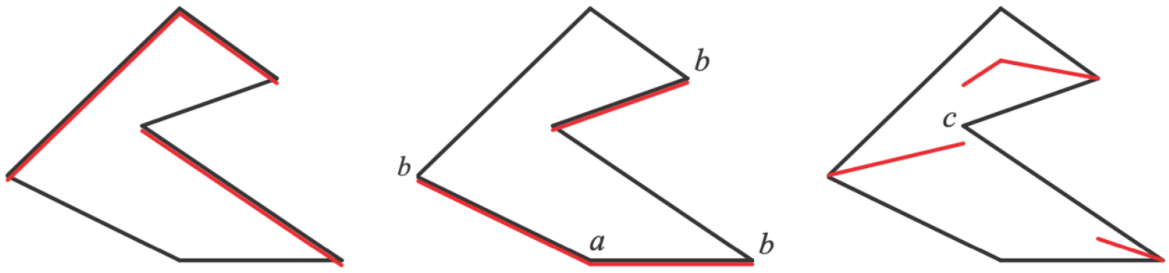
因此第二个 pass 时 shadow map 的深度可能会略小于物体表面的深度，从而误算为阴影，出现 shadow acne。



解决该问题的一个方法是在比较深度时增加一个 bias，认为只有当前点的深度大于 shadow map 一定阈值后才判断是阴影。但这会导致 detached shadow/peter panning，可以将 bias 与入射角关联，入射角越大，bias 越大。



另外一种方法是 second-depth shadow mapping，计算最小深度和第二小深度的中间值作为 shadow map。这导致阴影计算的开销增加（RTR does not trust in COMPLEXITY），而且物体必须是非面片（watertight）。



front faces second-depth midpoint

b) aliasing

走样的原因是由于 shadow map 的分辨率不足，在对 shadow map 采样时，多个不同的顶点都采样到同一个 texel，从而产生锯齿。可以用级联阴影（cascade shadow map）解决，即距离近的采用高分辨率，远处采用低分辨率。

1.2 Math

实时渲染往往进行近似计算，把不等式当作等式，其中一个重要的约等式是

$$\int_\Omega f(x)g(x)dx \approx \frac{\int_\Omega f(x)dx}{\int_\Omega dx} \cdot \int_\Omega g(x)dx$$

在 $g(x)$ 的积分域很小或者 $g(x)$ 足够光滑（变化不大）时，上式近似准确。将其代入到渲染方程中，得到

$$\begin{aligned} L_o(p,\omega_o) &= \int_{\Omega^+} L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)\cos\theta_i V(p,\omega_i)d\omega_i \\ &\approx \frac{\int_{\Omega^+} V(p,\omega_i)d\omega_i}{\int_{\Omega^+} d\omega_i} \cdot \int_{\Omega^+} L_i(p,\omega_i) f_r(p,\omega_i,\omega_o)\cos\theta_i d\omega_i \end{aligned}$$

渲染方程变为 visibility 和 shading 相乘的结果。同样地，使得该式准确需要满足两个要求：

- 积分域小（点光源、方向光源）
- 光滑，即光照在各个区域强度变化不大（diffuse dsbf，radiance 恒定的面光源）

2 PCSS (Percentage Closer Soft Shadows)

2.1 PCF (Percentage Closer Filtering)

PCF 最初是用于阴影边缘的抗锯齿反走样。PCF 不是直接对第一次 pass 得到的 shadow map 做滤波，因为对其滤波再比较，结果仍是二值化数据；也不是对两次 pass 后得到的存在锯齿的着色结果滤波，这会导致模糊。PCF 的滤波对象是 shading point 的深度和对应 shadow map 深度的比较得到的二值图。

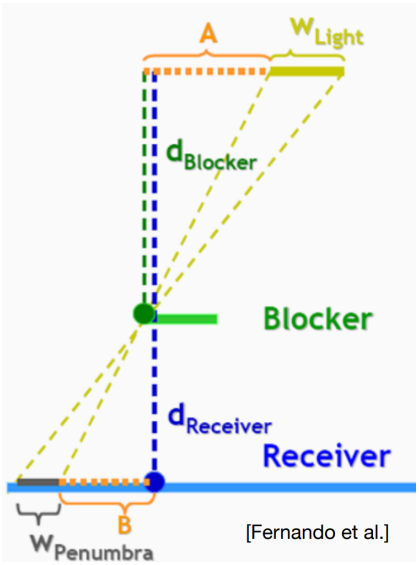
PCF 考虑采样一定范围的内的点而非单个点，对其周围一定范围内的 shading point 和对应点的 shadow map 深度值进行比较，得到二值结果，然后求平均作为当前点的 visibility，实现了软化阴影的效果。filter size 越大，阴影越软。

2.2 PCSS

实际生活中观察可知，投影物与阴影之间的距离越远，则阴影越软，所以可以通过计算遮挡物和投影平面的关系来确定 filter size。

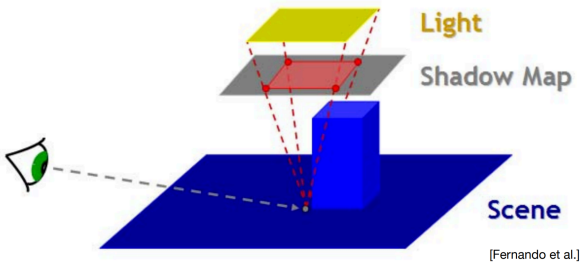
半阴影（penumbra）是指有些区域被遮住一部分光同时也接收到一部分光，在二维平面中实际上就是光源与遮挡物连线打在投影面的区域。根据相似三角形原理可以得到半阴影的大小

$$w_{Penumbra} = (d_{Receiver} - d_{Blocker}) \cdot w_{Light} / d_{Blocker}$$



在这里唯一的未知量是遮挡物的深度值。如果投影点在 shadow map 中只进行单次采样，相当于单一直线连接光源面的中心，若没有碰到遮挡物则全亮，而实际上投影点和光源面的处处相连后可能有一部分被遮住，即属于半阴影。所以可以对 shadow map 的一定范围做局部深度测试，找到范围内的 blocker 并计算平均深度，作为当前 blocker 的深度。这个过程被称作 blocker search。

blocker search 的半径同样需要确定。在光源处设置视锥，shadow map 置于近平面上，连接 shading point 和光源，截得的 shadow map 部分作为计算平均深度的范围。离光源越远，遮挡物越多，计算 blocker 所用的样本范围就越小；而离光源越近，遮挡物越少，计算 blocker 所用的样本空间就越大。



PCSS 的步骤可以总结为

- Blocker search (getting the average blocker depth in a certain region)
- Penumbra estimation (use the average blocker depth to determine filter size)
- PCF

2.3 Math

滤波或卷积是指对点 p，以其周围点到该点的距离关系作为权重，计算的加权平均得到最终结果，可以表示为

$$[w * f](p) = \sum_{q \in \mathcal{N}(p)} w(p,q) f(q)$$

上式代入到 PCSS 计算深度过程中，得到

$$V(x) = \sum_{q \in \mathcal{N}} w(p,q) \cdot \mathcal{X}^+[D_{SM}(q) - D_{scene}(x)]$$

因此 PCF 并不是对 shadow map 进行滤波后再比较，即

$$V(x) \neq \mathcal{X}^+[w * D_{SM}(q) - D_{scene}(x)]$$

也不是对得到的二值结果 visibility 滤波，即

$$V(x) \neq \sum_{q \in \mathcal{N}} w(p,q) V(q)$$

在整个 PCSS 过程中，第一步和第三步需要遍历范围内的每个 texel 使得算法变慢。对于越软的阴影，需要越大滤波范围，进一步导致了变慢。

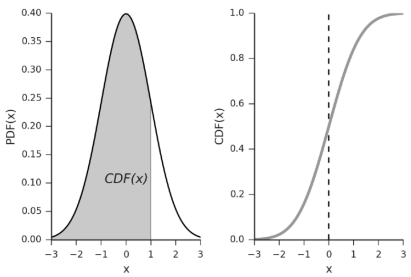
3 VSSM (Variance Soft Shadow Mapping)

3.1 加速 PCSS 第三步

PCSS 的第三步 PCF 需要比较某一区域内所有 shading point 的深度和 shadow map 上对应的深度，该过程遍历耗时。VSMM 则假定样本服从正态分布来估计。

定义一个正态分布需要均值和方差。均值计算的方法包括硬件 MIPMAP 和区域求和表 Summed Area Tables (SAT)。方差计算公式为 $Var(X) = E(X^2) - E^2(X)$，所以只需要增加一张深度平方的 shadow map，将其和原来的 shadow map 分别存储在 R 通道和 G 通道，无需额外的纹理图。

定义完均值和方差，就能得到概率密度函数 PDF。我们要统计的是深度比 shading point 浅的 texel 百分比，实际上也就是比 shading point 深度小的 PDF 的阴影面积，即求累积密度函数 $CDF(x) = \int_{-\infty}^{x} PDF(t)dt$。



为了求解 CDF，VSSM 用单边切比雪夫不等式进行近似，即

$$P(x > t) \leq \frac{\sigma^2}{\sigma^2 + (t - \mu)^2}$$

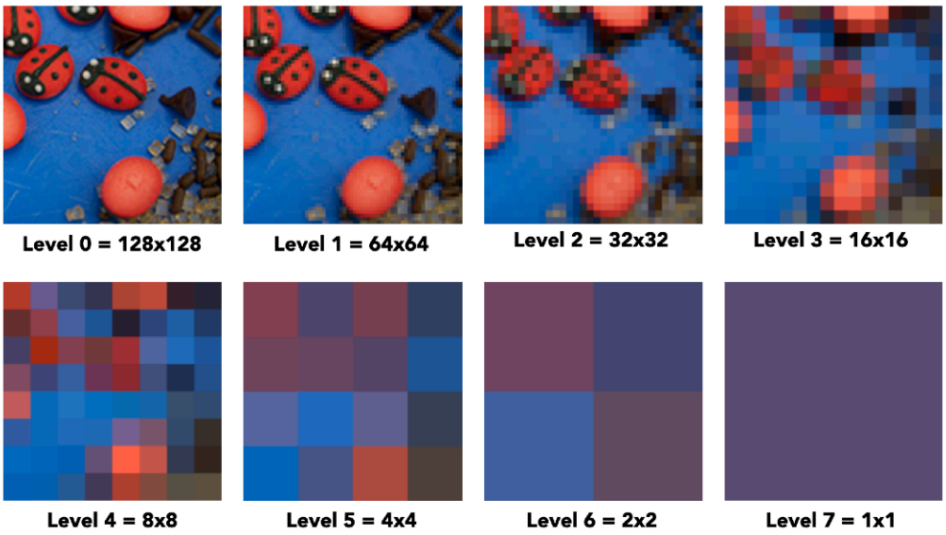
对于上述 CDF，只需要求出 1-$P(x > t)$ 即可。该公式不仅适用于正态分布，还适用于任何单峰函数，因此 VSSM 也并不局限于服从正态分布的样本。

总结来说，square depth map 和 shadow map 可以并行得到，MIPMAP 是硬件层面的速度很快，分别计算 map 的均值时间复杂度为 $O(1)$，运用切比雪夫不等式得到结果也仅仅是 $O(1)$，不需要多次采样和循环对比。但是若物体和光源不是静止的，需要更新 shadow map，mipmap 和 square depth map。

P.S. 均值

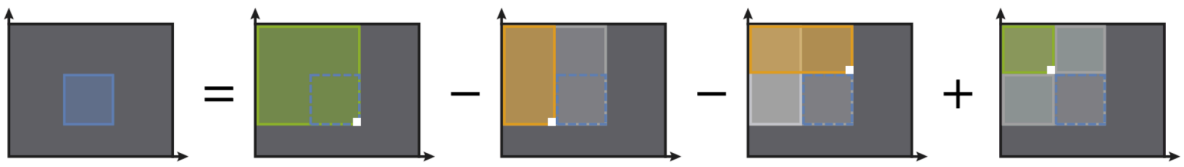
a) MIPMAP

首先通过双线性插值生成多级纹理，让纹素和像素大致对应。查询时确定层级号 $D = log_2 L$，在层与层之间做一次插值作为结果写入像素。三线性插值会使得准确度不高，且只能做正方形查询。



b) Summed Area Table

SAT 就是数据结构中的前缀和数组，在二维情况下可计算出当前矩形区域的和，进一步求得样本均值。该方法精确，但需要 $O(n)$ 的时间和空间。



3.2 加速 PCSS 第一步

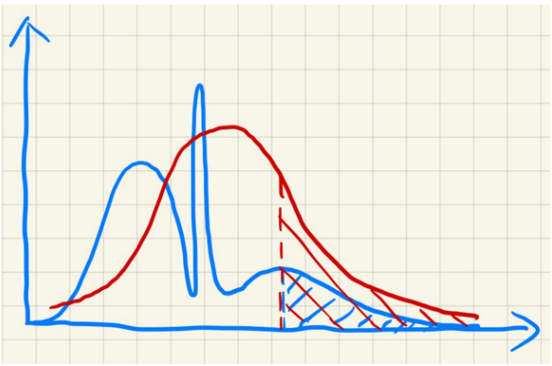
PCSS 的第一步 blocker search 是将 shading point 的深度和一定范围内的 shadow map 深度比较，若 shadow map 更浅表示该 texel 为 blocker，找到 blcoker 的所有深度并计算均值。将 blocker (z<t) 的平均深度记作 z_{occ}，非 blocker (z>t) 的平均深度记作 z_{unocc}，整个范围的平均深度记作 z_{avg}，可以得到关系式

$$\frac{N_1}{N} z_{occ} + \frac{N_2}{N} z_{unocc} = z_{avg}$$

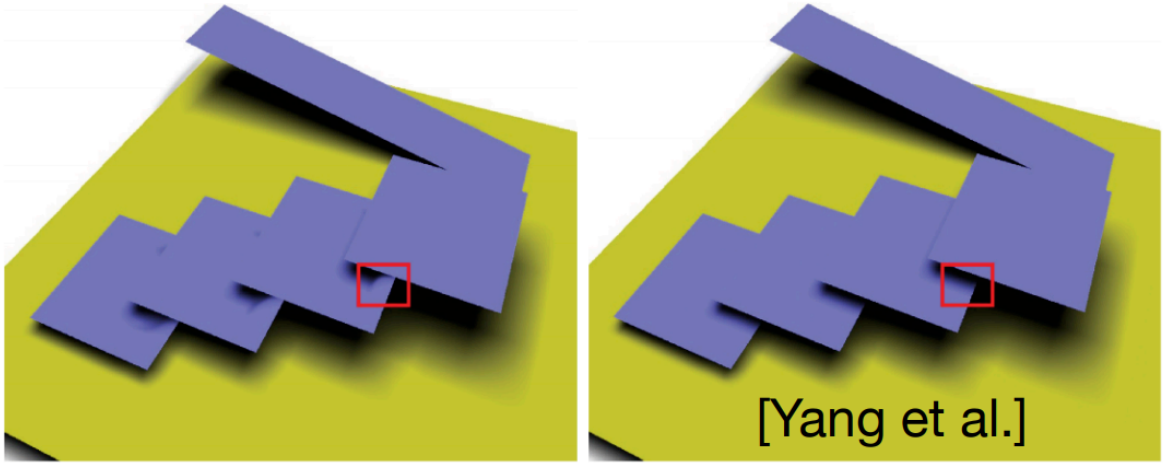
其中 $\frac{N_1}{N} = P(x > t)$，且 $\frac{N_2}{N} = 1 - P(x > t)$，可以用切比雪夫不等式估算。另外，$z_{unocc}$ 可以近似为 shading point 的深度，因为多数阴影的接收物都是平面。由此可以进一步算出 z_{occ}。

3.3 缺点

VSSM 将遮挡物的分布认为是单峰分布，但实际情况并非永远如此。如果实际非遮挡区域占比较小，而 VSSM 结果会过大，导致用 1 减去之后得到的遮挡区域变小，发生漏光 (light leaking)。

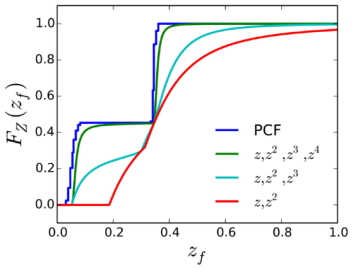


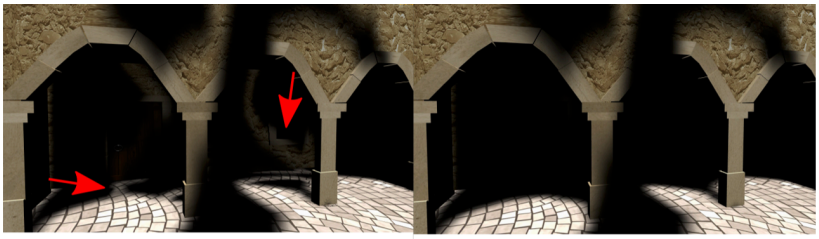
在计算 blokcer 深度时，将非遮挡平面深度都近似为 shading point 的深度，当接受阴影的物体是曲面、非连续表面时，阴影会出现中断。



4 MSM (Moment Shadow Mapping)

VSMM 存在由于分布描述不准确导致的漏光，VSSM 使用了二阶矩表示，而 MSM 使用高阶矩更加准确地表达分布，不过它也需要相当的额外空间开销和性能开销。MSM 恢复了 VSSM 在拟合 PCF 与 blocker search 中的 CDF 丢失。





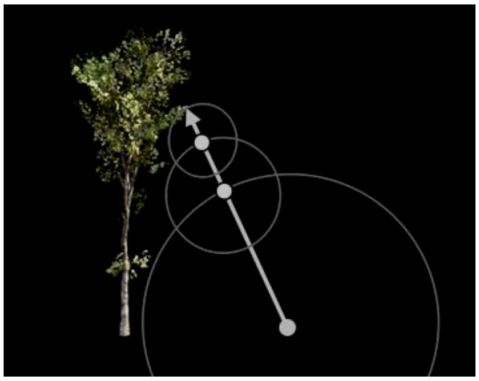
Variance shadow mapping, 1.15ms　　Moment shadow mapping, 1.46ms

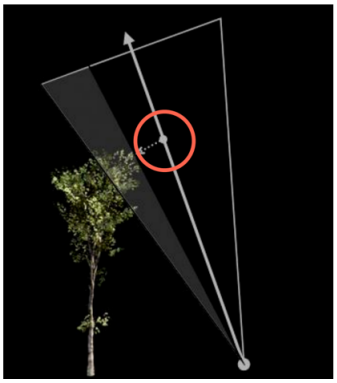
5 DFSS (Distance Field Soft Shadows)

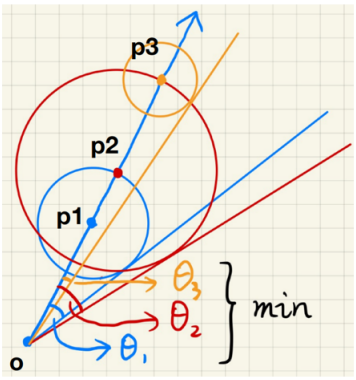
有向距离场 (SDF) 是指空间中任意一点到物体表面的最近距离，并且用正负号表示该点在物体内还是在物体外，若距离为 0，则认为该点在物体表面上。

向场景中发射一根光线，ray marching 根据场景中每个物体的 SDF，在投射点位置选取一个最小距离作为安全距离当做步长。即在以投射点为球心、安全距离为半径的球形范围内，不论光线方向为何，它都不会与场景内物体相交。当光线按原来方向走过这一段步长后，它又可以在新的位置依据 SDF 计算出新的安全距离，以此类推，直到步进次数达到一定值或者步长小到一定范围，即可认为光线与物体相交，退出循环。

安全距离和累加的步长可以进一步得到安全角度，也就是从 shading point 看向光源，在安全角内光线不会被遮挡。连接面光源的中心与着色点，在这个方向上做 ray marching，得到一系列安全角度后取个最小值作为最终的安全角度，这个角度越小，就说明光线被遮挡的概率越高，阴影越黑。



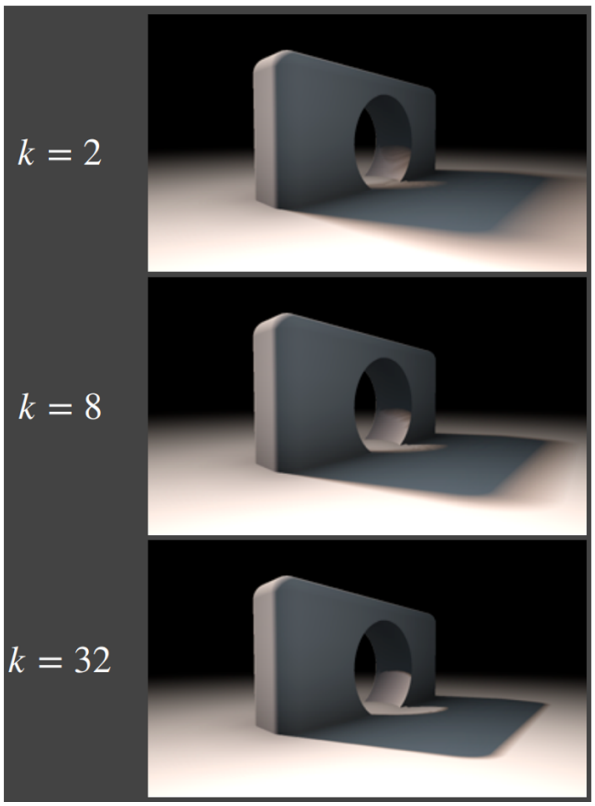




为了求得安全角度，直接用反三角函数计算开销较高，用如下方法近似

$$arc\sin\frac{SDF(p)}{p-o} \to min\left\{\frac{k\cdot SDF(p)}{p-o}, 1.0\right\}$$

同等安全角度下，k 越小，visibility 越小，半影区越大，阴影越软；k 越大，visibility 越容易被 1 截断，阴影越硬。



距离场的优点在于速度快且质量高，但是它需要预计算，存储量也很大。