阴影质量

阴影过渡

上图中还有一个问题，突然切断阴影最大距离处的阴影会显得很突兀，我们通过一种线性淡化的方式使阴影过渡变得柔和自然一些。阴影淡化应从阴影最大距离之前的一段距离开始，直到最大距离时阴影强度为0。

saturate( (1 – d / m) / f )

d是表面深度

m是阴影最大距离

f是阴影过渡范围

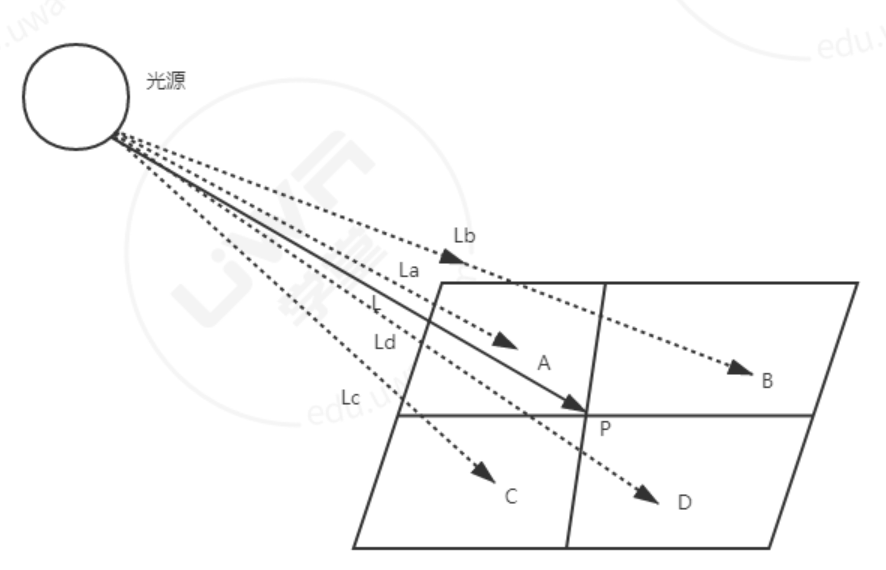
级联过渡

我们也可以使用和阴影过渡相同的方法，在最后一个级联边缘对阴影进行平滑过渡，而不是硬切。

(1 – d2 / r2) / f

深度偏差

交错条纹状阴影的情况，这种现象被比喻为“痤疮（Acne）”，专业术语叫做阴影渗漏（Shadow Acne）。

  
因为阴影贴图分辨率太小，导致A、B、C、D这4个在光源空间处于不同位置坐标的片元对应同一个阴影贴图的位置P上，并且P所对应的深度值为L，即光源到这一被照亮的位置点距离为L。如果点P所对应的片元位置与光源的距离小于L，该片元会被照亮，大于L就会遮住。

上图因为4个片元都没有被其它物体遮住，所以La、Lb、Lc、Ld长度无论是多少，都应该能被光源照亮。但实际计算中，因为阴影贴图分辨率太小，4个片元都只能使用L作为判断能否被照亮的距离，最终La<L，Lc<L，片元A和C被照亮，Lb>L，Ld>L，片元B和D被遮挡，于是导致了交错的条纹状阴影。

解决阴影渗漏最直接的办法就是计算出La、Lb、Lc和Ld的长度，沿着这些线的反方向往回拉一拉，即减去一个微小的偏移值，使得最终La、Lb、Lc和Ld的长度都小于L，这样原本应该能被照亮的地方确实被照明了，这种方法叫做调整阴影偏差。

​调整阴影偏差有个问题，就是比较难定量的针对当前被照明的物体的表面凹凸程度设置准确的偏差值，随着深度偏差将阴影投射推离光线，采样阴影也会向同一方向移动。如果偏差设置的过小，依然会有一些应被照亮的片元没被照亮，如果偏移值过大就会导致影物飘离（Peter Panning），即原本某些应该被遮住不被照亮的片元反而被照亮，显得物体和它的影子分开了一样，

所以要找到一个刚好能消除阴影痤疮的值需要一定的技巧和算法。Unity中采用的阴影偏差值的计算方法是基于物体斜度（Slope）的，称为“基于斜度比例的深度偏差值”（Slope Scale Based Depth Bias）算法。大部分改善对阴影深度贴图采样误差的算法，其核心思想是分析待绘制场景中各部分内容对采样误差的影响程度。

法线偏差

既然无法调整深度偏差来达到我们想要的效果，那我们尝试另外一个方法，即尝试在采样阴影时使表面沿法线方向偏移一点，然后对表面的一点进行采样，如果距离足够远就可以避免阴影痤疮，这虽然会让阴影的位置发生稍微的改变，可能导致边缘不对齐或添加假阴影，但这些改变远没有影物飘离（Peter Panning）来的明显。

接下来我们要沿表面的法线方向稍微移动表面的位置来采样阴影。如果只考虑一个维度，那么移动距离等于一个世界空间中纹素大小的偏移就足够了。

阴影平坠

可能导致阴影痤疮的另一个潜在问题是如果在Unity中应用了阴影平坠（Shadow Pancaking）技术，那么就可以剔除那些不希望看到的阴影。该技术的想法是渲染方向光的阴影时，通过剪裁光照空间，给该空间设定阴影视椎体近裁剪平面，只有位于该平面内的物体才能投射阴影，且阴影视椎体近裁剪平面会尽可能的向前移动，意在减少沿光照方向渲染阴影贴图时使用的光照空间范围，这可以提高阴影贴图的精度，减少阴影痤疮。

百分比切近滤波(PCF)

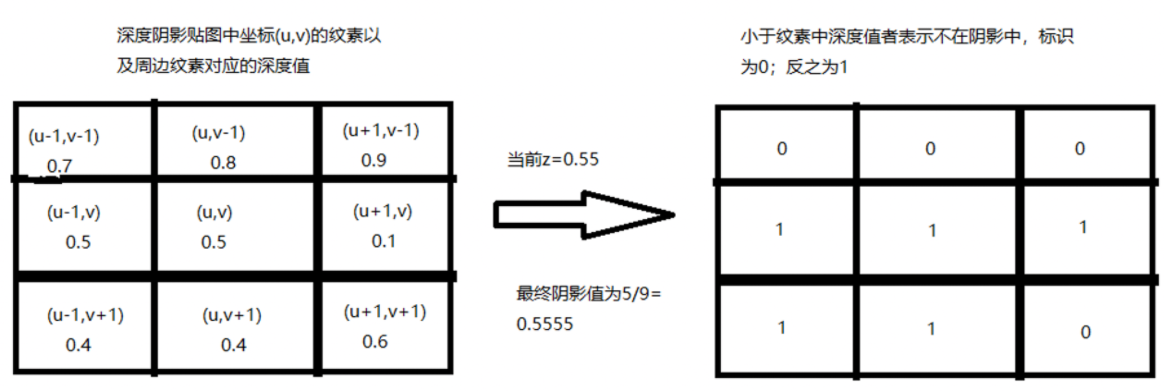
接下来该解决阴影的锯齿问题了。产生锯齿的原因在于，判断一个片元是否在阴影内而进行深度测试时，要把该片元从当前摄像机观察空间转换到光源空间，因为转换矩阵不一样，且阴影贴图分辨率不大，导致观察空间中多个片元对应阴影贴图同一个纹素。例如两个黑色锯齿中间的空白部分，本来这部分也应该处于黑色阴影中的，但因为采样到的阴影贴图中的纹素刚好不是黑色的，即那个纹素刚好不在黑色阴影下，就会导致阴影锯齿。

解决锯齿最直接的办法就是提高阴影贴图分辨率，但内存占用会变大，这种方法也只是减轻问题。实际开发中通常采用适中的分辨率的阴影贴图加上区域采样方法改善锯齿现象。

因为阴影贴图的纹素存储的不是颜色信息而是深度信息，对深度值取均值会产生不正确的深度结果，所以锯齿不能通过对某纹素周边邻接的纹素取值然后求平均来消除。百分比切近滤波（Percentage-Close Filtering，PCF）方法，是对阴影比较测试后的值进行滤波，可以使生成的阴影边缘平滑柔和。

PCF方法具体步骤是：在片元着色器中，把当前操作的片元f先变换到光源空间，然后经过投影和视口变换到阴影深度贴图空间中，假设变换后深度值为z，对应的贴图坐标为（u，v），该坐标对应的纹素深度值为z0。进行到这一步，如果不使用PCF方法，那么直接就根据z和z0的大小判断该片元在阴影中全黑还是不在阴影中不黑。PCF是对贴图坐标（u，v）处周边纹素也进行采样获取其深度值，再和当前片元的深度值z比较。如果在阴影中标识为1，不在阴影中标识为0，并把这些01值每项累加求平均值，这些平均值落在［0，1］中，这样阴影就有浓淡之分而不像未使用PCF方法之前的非明即暗，从而达到柔化边缘，减少锯齿的效果。

下图演示了PCF 3X3方法的采样效果：



混合级联

现在的阴影已经比较柔和了，但是会使得各级联之间的过渡更加明显。可以通过在级联之间添加一个过渡区域来进行相邻级联之间的混合从而使级联过渡更柔和一些。

级联过渡抖动

虽然从级联混合效果上看起来不错，但这会使我们必须在级联混合区域中对阴影贴图多采样一次。有一种替代方法是基于抖动模式始终从一个级联中采样。虽然这看起来不太好，但是可以提升计算效率，也会减少一次采样，尤其在使用大的PCF过滤模式时。

剔除偏差

使用级联阴影贴图有一个缺点，我们不止一次对每个光源渲染相同的投影。如果大的级联中的一些投影数据能被小的级联中的投影数据覆盖，就可以从大的级联中剔除这些投影。在Shadows脚本的RenderDirectionalShadows方法中将splitData的shadowCascadeBlendCullingFactor属性设置为1来实现这点。在渲染方向光阴影之前执行这个操作。

该值是调节用于执行剔除的上一个级联的半径的因子。在剔除时，Unity相当的保守，但我们应该通过级联过渡比例降低它，确保过渡区域中的投影不会被剔除。在RenderDirectionalShadows方法中我们使用0.8减去级联过渡值，最小值限制到零。如果看到在级联过渡的阴影中出现孔洞，则必须进一步减少孔洞。