### 辐射度算法

计算物体表面某一点的辐射度方程为：

其中， 代表处的辐射度

代表点自发光的辐射度

代表点的反射率

代表积分区域，即整个场景中所有的面

代表内的任意一点

代表到之间的距离

代表点的法向量与的夹角

代表点的法向量与的夹角

代表点到点的可见性

两点之间无不透明障碍物，可见时，否则

在等式中， 表示的是两个点（积分后为它们所在的面片）之间的几何关系，将它定义为形状因子，则有：

设场景中的面片总数为，第个面片的辐射度为，其自身辐射度为，漫反射系数为，面积为，那么对辐射度方程进行积分，可以得到：

对场景中的每一个面片建立如上所示的辐射度方程，可以得到一组方程组，求场景中所有面片的辐射度，就是对这组方程组进行求解。然而实际应用中，精确地求解这个方程组是不切实际的，需要对它进行一定的简化。

辐射度方程的另一个解决难点是如何求解形状因子。它表示了两个面片的辐射传播关系。具体求解方法有封闭形状解法（Closed Form Solutions）以及数学解法（Numerical Solutions）。

### 预处理辐射传播

通过球谐函数简化复杂的光线传播函数，并通过它计算光照和阴影。

预处理辐射度算法分三步（第一步为预处理计算，第二与第三步为实时计算）：

1. 对几何体上的任一点，计算它的漫反射传递向量
2. 对点，计算它的光源向量
3. 利用和实时计算点的光照

这里计算与指的是使用基函数近似这两个函数。

对于漫反射表面的任意点，假设它的双向反射函数为，那么使用下述的光照模型来形式化传递函数：

1. 不考虑阴影的漫反射光照：

其中， 表示光线的入射向量

(s) 表示方向的入射光线在点的传递函数，

表示点的法线

此时，

1. 考虑阴影的漫反射光照：

其中， (s) 表示点是否能够被来自方向的光线直接照射（即可见性）

此时，

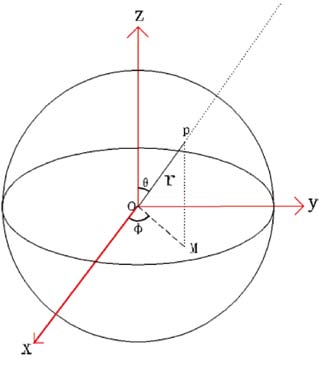
1. 考虑间接光照的漫反射光照

其中， 表示遮挡物在方向上对点的光照（光线照射遮挡物产生的漫射光）

预处理辐射传播算法除了可以计算漫反射表面的光照，还可以计算光滑表面的高光效果。（需要用到额外的转换矩阵）

### 球谐函数

球谐函数是定义在球面坐标系下的一组基函数，通常表示为或，其中



基函数：最常见的基函数是傅里叶变换，它利用三角函数或它们的积分的线性变换来近似模拟其他函数。

和傅里叶变换类似，球谐函数可以近似地模拟任意定义在球面坐标系中的函数。通常球谐函数都定义在复数域中，但在光照计算中，使用的都是实数值，不需要用到复数。

球谐函数是在单位球面坐标下定义的，单位球面坐标与点坐标的对应关系如下：

球谐函数在球面坐标系下表示为：

其中， 表示连带勒让德多项式

表示标量因子，

表示球谐函数的阶，它将球谐函数分割为独立的层

表示在层中的次数

连带勒让德多项式

是阶次的连带勒让德多项式，计算公式如下：

由于连带勒让德多项式的求解过于复杂，通常会简化为对如下三个递归式进行求解：

1. ，其中“!!”代表双阶乘（详情见百度百科）

计算的递归规则为：首先使用递归式2计算，如果，则取得最终解；否则即有，使用递归式3，求得，若，则取得最终解；否则将递归式2与递归式3的结果代入递归式1，不断计算更高阶数的，直到为止。

球谐函数的下标

下标和满足：

球谐函数也可以用一维的形式表示：

球谐函数示例

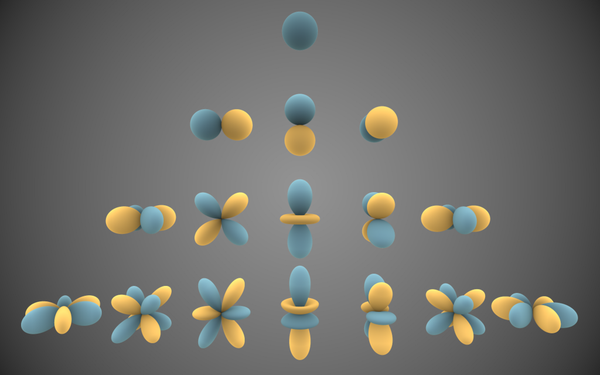
一个阶的球谐展开式包括0阶到阶的所有球谐函数，共有项。

2阶球谐函数共有9项，它们的值如下（对应到点坐标）：

注：

)

令球坐标，可以得到如下示意图（其中，蓝色表示球谐函数值为正数，橙色表示球谐函数值为负数；图中采用了右手坐标系，x轴垂直纸面水平向外，y轴平行纸面水平向右，z轴平行纸面垂直向上；行数代表球谐函数的阶数，列数代表次数，最中间是0次，向左次数依次减小，向右次数依次增大）：



球谐函数的投影

使用基函数线性变换表示函数的公式如下：

投影指的是计算的过程，指的就是对基函数数组中的每一个函数的线性变换量(可以理解为每一个基函数的系数)。对于任意一组基函数，投影的过程可以用下式来表示：

将球谐函数作为基函数，代入上式可得：

我们称上述等式中的为球谐系数。

球谐函数的基本性质

球谐函数具有一些特殊的性质，使得将一个定义在球面上的函数转化到频域的球谐函数后，可以简化积分操作，从而在计算环境光照时能够极大地简化计算。

1. **正交性**

在球面上对任意两个球谐函数的积求积分，得到的值不是0就是1。

1. **旋转不变性**

给定函数和，使为经过旋转矩阵变换所得，即。则对于和，的投影等于的投影的旋转。

1. **卷积性质**

对于任意两个定义在球面上的函数和，用阶的球谐函数来近似它们（可以分别得到个系数），分别记为和。根据球谐函数的正交性，可以得到球谐函数的卷积性质：

1. **回旋性**

一个圆对称的核函数与原函数复合得到一个新的函数，那么新函数的球谐系数可以使用和的球谐系数计算得到：

### 基于球谐函数的动态环境光照实时绘制算法

光照遵循叠加原理：有多个光源产生的光照效果，等价于每个光源单独产生的光照效果的和。

根据这个原理可以得到基本的光照方程：

其中， 表示点在方向的出射光线的光照

表示点自发光的光照

表示以点为中心的单位圆

表示BRDF（双向反射分布函数），其中是入射光方向

表示从点向方向发射光线后得到的第一个碰撞点的坐标

表示物体表面在点处的法线

表示光的波长

表示时间

光照方程的简化

为了简化光照方程，假设：

1. 不考虑光的波长与时间 忽略参数与
2. 物体只具有漫反射表面 为固定值
3. 没有自发光 为0
4. 忽略场景物体表面的多次漫反射 不计算
5. 计算环境光的贡献 计算

表示点向方向发出的追踪光线能否直接照射到环境贴图

此时光照方程可以简化为：

如果不考虑场景物体对环境光的遮挡，光照方程可以进一步简化（参数与此时已经没有意义）：

光照方程的球谐函数展开

此时，可以将和用球谐函数展开式代替：

注：这里的指的应该是点的法线方向

注：这里的指的是的方向

定义（这里的是与的夹角）。对进行球谐函数展开。因为只和极角相关而和方位角无关，所以在的球谐函数展开式中，当时，。因此，可以将下标省略，只使用下标。的球谐函数展开式如下：

因为函数是圆对称的，那么由球谐函数的回旋性可得：

令 ，则

将代入的球谐函数展开式：

其中，是常量（下一节有推导）。在环境光照一定的情况下（静态光源），的值也是常量。因此，点的颜色只与它的顶点法向量有关（点的法向量即为所指的方向）。

系数的计算

根据球谐函数的系数计算公式 ，可以对进行计算：

因为只考虑上半球面的光照，所以极角的积分上限为。

根据球谐函数的计算公式，可以对进行计算：

代入公式 ，可得：

将代入，可得：

由连带勒让德多项式的递归求解公式，可得：

……，

此时，，，，……

由此，可以通过规律推导出的一般计算公式：

注：[ ]为取整符号

由于随的增大衰减速度较快，通常情况下，只需要计算2阶球谐函数展开式即可，此时需要计算9个球谐函数系数。

使用蒙特卡洛积分计算系数

根据球谐函数的系数计算公式 ，可以得到：

其中表示环境光照，如果直接求解上述积分等式，对于每一个需要计算光照的像素点，都要对环境贴图上的每个像素点进行计算，再求它们的和，这个复杂积分过程的计算量显然是不可接受的。因此，需要采用蒙特卡洛方法来计算该积分。

蒙特卡洛积分是一种用随机数来解决积分问题的方法，它的基本思想是：任何一个定积分，都可以看作某个随机变量的数学期望，只要能实现该随机变量的抽样，就可以通过随机抽样的平均值来获取近似积分值。数学公式表述如下：

其中， 表示取到的概率

在采样点的数量足够多的情况下，计算得到的近似积分值和原始积分值的误差非常小。

在球面上取一组随机采样点，由于采样点是随机采样得到的，可以认为它们在球面上是均匀分布的，所以（S为单位球的表面积）。因此可以表示如下：

其中可以使用球谐函数计算公式计算得到。

总结

之前已经提到过，通常使用2阶的球谐函数对环境光照进行求解，设点处的法向量，则它的光照值为：

其中是与有关的常量，在确定的值情况下，可以计算得到：

简化后的光照计算具体步骤如下：

1. 预处理计算，得到系数
2. 绘制时，每帧实时计算
3. 对于每个顶点，根据公式计算该点的光照

UE4中的解法

考虑到效率问题，UE4只采用了1阶球谐函数求解光照方程，即只需要使用4个系数，刚好可以存在一个4维向量中，求解公式如下：

这样像素着色器中可以通过3次四维向量点积（RGB三个颜色通道各一次）求得环境光照。若使用2阶球谐函数（使用9个系数），计算过程会复杂很多。

注：UE4的BasePassForForwardShading着色器中并没有使用最后计算得到的最简公式。

### 环境遮挡

环境遮挡算法是一种通过模拟柔和阴影，从而模拟全局光照效果的算法。

环境遮挡表示了物体表面某个点被周围的几何环境遮挡的百分比，它代表了该物体不能被环境光照直接照射的比率。绘制时，将着色点的颜色乘以环境遮挡因子的逆，即可得到计算环境遮挡后的颜色。设物体表面的一个点，法向量为，那么该点的环境遮挡由以下的公式表示：

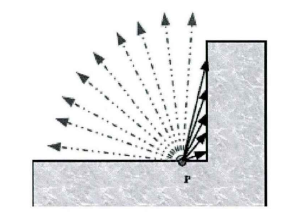
其中， 表示点为原点的方位角

表示点的法向量

表示点的可视函数的逆（被遮挡时，否则）

表示方向上的光线对遮挡的贡献程度

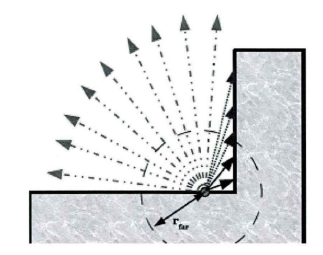
（方位角方向越接近法向量方向，光线对遮挡的贡献越大；反之，方向垂直于法向量的光线对点的遮挡贡献为0）



如上图所示，它是将环境遮挡算法以二维图像进行示例（实际的环境遮挡算法是在三维空间进行的）。其中，虚线表示的光线是没有被遮挡的，实线表示的光线在未到达环境光源的时候，被不透明物体遮挡，也就是说沿黑色光线的路径射向点的环境光是不能到达点的。环境遮挡就是计算黑色光线占总光线的百分比，并根据这个百分比模拟阴影效果。

而公式需要计算整个半球面的积分，积分的计算量太大，因此实际计算过程中，一般都使用采样的方法来模拟：通过对周围空间均匀发射若干条测试光线，计算光线被遮挡的概率，来近似获取积分值。

实际计算时，还有一个常用的近似优化方法：



如上图所示，在计算环境遮挡的过程中，定义一个计算范围半径，只考虑一点为中心，半径为范围内的遮挡，这样可以减少遮挡项的计算量 ，但副作用是会降低计算结果的精度（部分光线在范围外才与不透明物体相交）。

### 屏幕空间环境遮挡

环境遮挡算法的一种近似算法，它利用深度缓存计算遮挡，速度很快。

具体来说，就是通过对一个像素点周围进行采样，通过比较深度缓存值来计算环境遮挡因子，通过这种方法可以产生柔和的阴影，有比较好的真实感。

相对于常规环境遮挡算法，屏幕空间环境遮挡的优点有：

1. 时间复杂度低 时间复杂度为常数
2. 空间复杂度低 空间复杂度为常数
3. 不需要预处理
4. 不受场景因素的干扰 复杂场景消耗的计算资源与简单场景完全一样
5. 可以完全由GPU处理 不涉及CPU与GPU间的数据传输，极大提高了运算效率

屏幕空间环境遮挡的采样

采样有两个重要的方面：

1. 采样方式 决定选取哪些像素点
2. 采样点个数 与模拟效果成正相关

常用的采样方式如下图所示：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

其中，橙色像素点为需要计算环境遮挡的像素点，黑色像素点为采样点（这里共取了16个采样点）。

计算流程

对于屏幕中的每一个像素点，执行如下计算：

1. 取得该像素点的深度值
2. 根据采样方式取得采样点的深度值
3. 利用前两步取得的深度值计算环境遮挡值
4. 取得该像素点的颜色值
5. 令该像素点颜色值等于