

PBR学习笔记

PBR学习笔记

- 一、PBR的渲染方程：
- 二、BRDF(双向反射函数)：
 - 1.正态分布函数D
 - 2.几何函数G
 - 3.菲涅尔方程F
 - 3.1物理学上的菲涅尔：
 - 4.总结
- 三、金属工作流材质
 - 1. 金属度
 - 2. 粗糙度
 - 3.漫反射
- 四、总结
 - 1.
 - 2.
 - 3.

一、PBR的渲染方程：

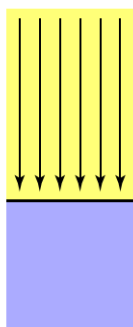
光作用在物体表面有一种简单的理解：物体先**吸收**各个角度传来的光，然后**再辐射**到相机（or眼睛）。那么每个角度传入的光对于辐射出去的光的贡献是多少呢？有如下公式来描述：

$$f_r(w_i \rightarrow w_o) = \frac{dL_r(w_r)}{dE_i(w_i)} = \frac{dL_o(w_r)}{L_i \cos \theta_i dw_i}$$

对以上公式做一个微分乘法，再在 w_i 上积分（这个变换是个人理解），得到：

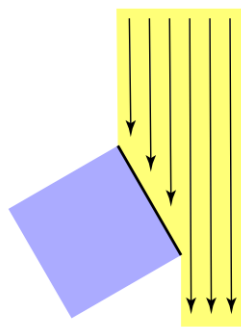
$$L_r(p, w_o) = \int_{H^2} f_r(p, w_i \rightarrow w_r) L_i(p, w_i) \cos \theta_i dw_i$$

这个就是**反射率方程**，一个基于物理的渲染方程。公式中 $L_i(p, w_i)$ 和 $L_o(p, w_o)$ 分别代表传入的光和辐射出去的光。公式中的 $\cos \theta_i$ 表示当光**斜射**到点上，这个点接受的能量会减弱。其实这个再之前的*phong*模型中已经用到了，漫反射强度会乘以这个数， θ 角度是光与点的法线的角度，这种漫反射模型叫lambert漫反射模型。可视化的理解如下图：



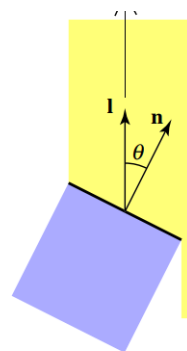
Top face of cube receives a certain amount of power

$$E = \frac{\Phi}{A}$$



Top face of 60° rotated cube receives half power

$$E = \frac{1}{2} \frac{\Phi}{A}$$



In general, power per unit area is proportional to $\cos \theta = l \cdot n$

$$E = \frac{\Phi}{A} \cos \theta$$

二、BRDF(双向反射函数):

BRDF可以近似的求出每束光线对一个给定了材质属性的平面上最终反射出来的光线所作出的贡献程度。举例来说，对于一个完全光滑的平面（镜面），那么对于所有的入射光线 w_i ，BRDF函数都会返回0.0，除了出射光线方向与 w_o 相同时，此时函数返回1.0。接下来主要了解下实时渲染中常用的Cook-Torrance BRDF模型：

$$f_r = k_d f_{\text{lambert}} + k_s f_{\text{cook-torrance}}$$

其中，漫反射有很多模型，但是都比较昂贵。Lambert漫反射从效果上来看足够应付实时渲染，其定义如下：

$$f_{\text{lambert}} = \frac{c}{\pi}$$

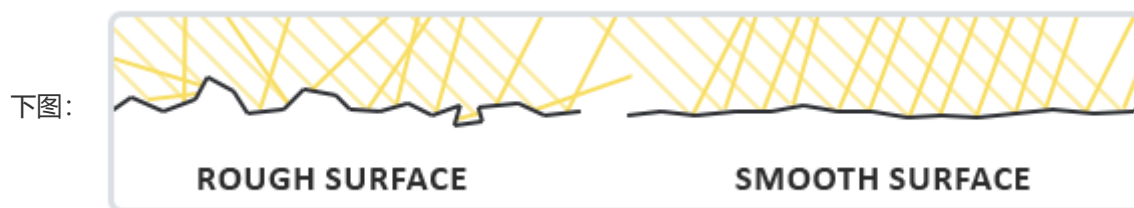
BRDF的镜面反射更加复杂，定义如下：

$$f_{\text{cook-torrance}} = \frac{DFG}{4(w_o \cdot n)(w_i \cdot n)}$$

接下来主要介绍公式中的DFG的意义。

1.正态分布函数D

解释这个函数之前，需要理解一下微平面这个模型，就是用微观下细小的镜面来模拟宏观的一个面，如

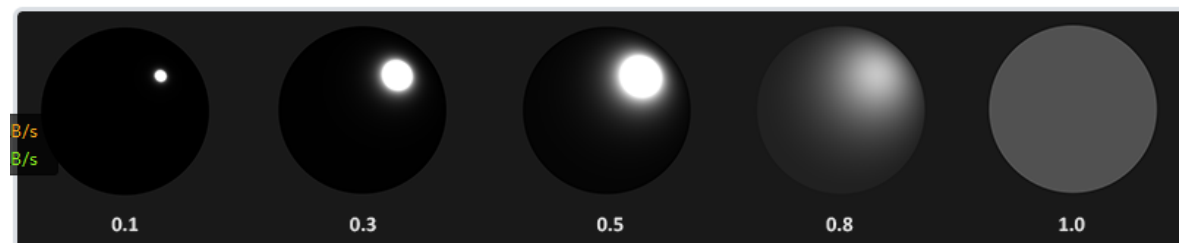


这个模型只需要一个参数，粗糙度 α 。

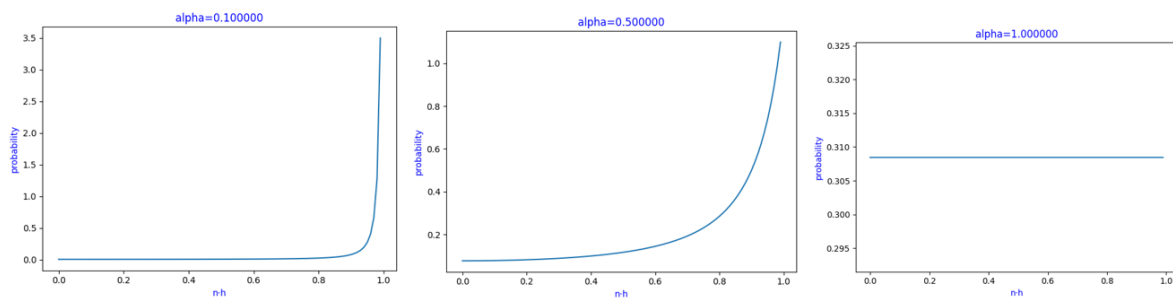
正太分布函数就是计算平面中微平面的 n 与 h 一致的概率。这样的函数有很多个，主要介绍Trowbridge-Reitz GGX正太分布函数：

$$NDF_{GGXTR} = \frac{\alpha^2}{\pi((n \cdot h)^2(\alpha^2 - 1) + 1)^2}$$

对于不同的粗糙度，渲染效果如下：



当粗糙度一定的时候，可以绘制出概率随 $n \cdot h$ 值变化的函数图像：



可以看出当 $\alpha=0.1$ 的时候，反射集中在很小一块。但是当 $\alpha=1.0$ 的时候，反射均匀分布在表面，跟渲染图一致。但是值得注意的一点是，在 $\alpha = 0.1, n \cdot h = 1.0$ 的时候，算出的概率值竟然是3.5左右。而Cook-Torrance BRDF除数中有一个常数4，是不是用在正太分布函数的呢？

2.几何函数G

几何函数从统计学给出微平面之间相互遮蔽比率，主要使用的模型是Schlick-GGX模型，其定义如下：

$$G_{schlick-GGX}(n, v, k) = \frac{n \cdot v}{(n \cdot v)(1 - k) + k}$$

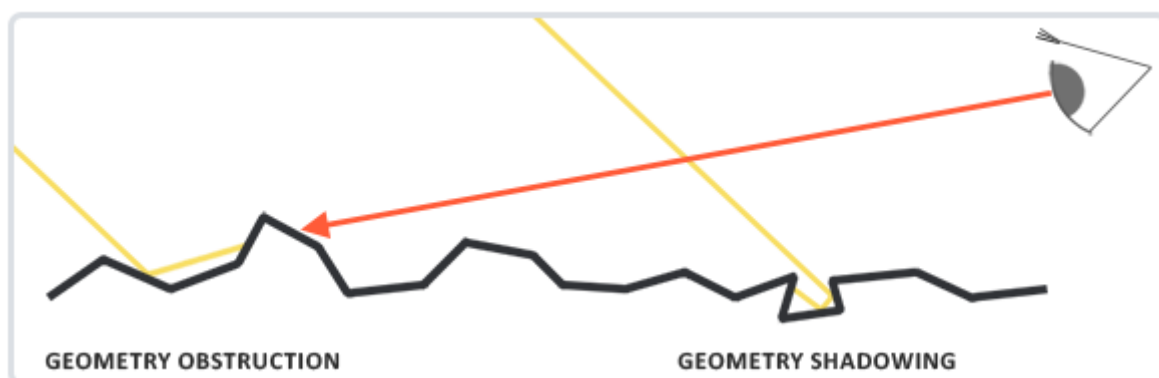
其中 k 是对粗糙度的一个重映射，也就是说，几何函数也是一个跟粗糙度有关的函数。映射关系如下公式所示：

$$k_{direct} = \frac{(\alpha + 1)^2}{8}$$

$$k_{IBL} = \frac{\alpha^2}{2}$$

为了更加贴近物理，几何函数需要将几何遮蔽和几何阴影都考虑进去。主要使用Smith模型：

$$G(n, v, l, k) = G_{sub}(n, v, k) \cdot G_{sub}(n, l, k)$$



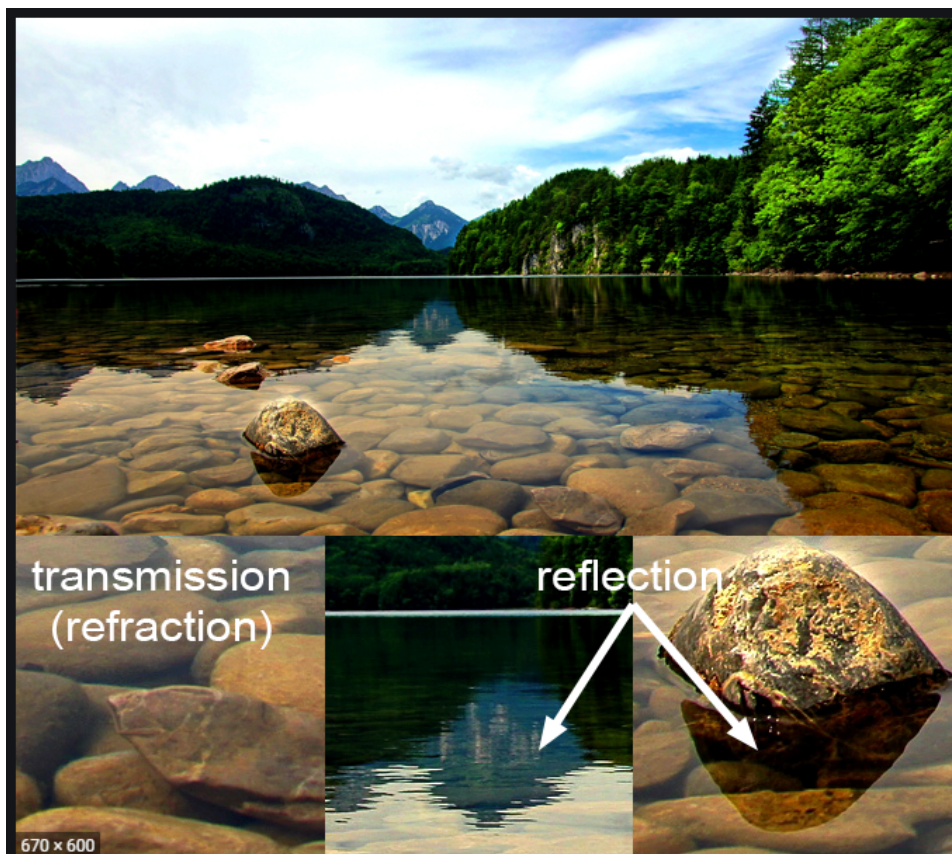
加入几何函数后，渲染效果如下图：



从图中可以看书，当粗糙度越大的时候，光强越小，也就是遮蔽效果越明显。

3.菲涅尔方程F

3.1物理学上的菲涅尔：



菲涅尔现象简单的解释为：随着视线观察表面的角度不同，表面反射出来的光线与折射出来光线的比率也随着变化。视线与表面法线的角度越大，反射的比率越大，这种表现在现实生活中，当你观察一个非金属表面的时候，会越亮（用塑料表面亲测效果明显）。

菲涅尔方程的数学定义为：

$$F_{Schlick}(h, v, F_0) = F_0 + (1 - F_0)(1 - (h \cdot v))^5$$

4.总结

综合上述的，我们可以得到渲染方程为：

$$L_r(p, w_r) = \int_{H^2} (k_d \frac{c}{\pi} + k_s \frac{DFG}{4(w_o \cdot n)(w_i \cdot n)}) L_i(p, w_i) \cos \theta_i dw_i$$

在加上物体本身的自发光项，可以得到最终的渲染方程：

$$L_r(p, w_r) = L_e(x, W_o) + \int_{H^2} (k_d \frac{c}{\pi} + k_s \frac{DFG}{4(w_o \cdot n)(w_i \cdot n)}) L_i(p, w_i) \cos \theta_i dw_i$$

三、金属工作流材质

1. 金属度

理论上来说，表面的金属度应该是二元的，也就是是或者否。但是在图形学中可以选择0到1之间的值作为金属度，这样做是为了表面上锈迹和刮痕等一些效果。要理解金属工作流材质，必须得深入的理解前文简单提到的菲涅尔现象。

当我们垂直观察任何表面的时候，都会有一个基础反射率 F_0 ，这个值表示的是光从表面反射的比率为 F_0 ，折射进入表面的比率为 $(1 - F_0)$ 。对于非金属来说， F_0 是一个一维的值（大多数非金属的范围在0.02-0.05之间）。对于金属来说， F_0 是一个三维的值，也就是说，它对于三原色的反射比率并不一致（看很多资料说带反射颜色，因为金属的 F_0 与非金属的 F_0 计算方式不一样，但其应用方式并无差别，因此我统一理解成对颜色的反射比率，这个理解需要再阅读材料来验证）。

材料	F_0 (线性)	F_0 (sRGB)	颜色
水	(0.02, 0.02, 0.02)	(0.15, 0.15, 0.15)	
塑料/玻璃 (低)	(0.03, 0.03, 0.03)	(0.21, 0.21, 0.21)	
塑料 (高)	(0.05, 0.05, 0.05)	(0.24, 0.24, 0.24)	
玻璃 (高) / 红宝石	(0.08, 0.08, 0.08)	(0.31, 0.31, 0.31)	
钻石	(0.17, 0.17, 0.17)	(0.45, 0.45, 0.45)	
铁	(0.56, 0.57, 0.58)	(0.77, 0.78, 0.78)	
铜	(0.95, 0.64, 0.54)	(0.98, 0.82, 0.76)	
金	(1.00, 0.71, 0.29)	(1.00, 0.86, 0.57)	
铝	(0.91, 0.92, 0.92)	(0.96, 0.96, 0.97)	
银	(0.95, 0.93, 0.88)	(0.98, 0.97, 0.95)	

在渲染过程中，有一个叫BaseColor的贴图，也叫做Albedo贴图。当一个着色点的金属度为1的时候，从Albedo采样出来的就是 F_0 。当某一个着色点的金属度为0的时候， F_0 是有引擎指定（一般是非金属材料的平均 $F_0=0.04$ ），而不是计算出来的。

当一个点的金属度在0.0到1.0之间，就需要对Albedo采样出来的值进行插值（这个时候我们认为采样出来依旧是基础反射率，而不是表面颜色）。插值代码如下：

```
vec3 F0 = vec3(0.04);
F0 = min(F0, albedo, metallic); // 插值
vec3 f = fresnelSchlick(max(dot(H, V), 0.0), F0); // 菲涅尔计算函数
```

以上的代码是learnrn-opengl学习web得到，至于ue4是如何处理这部分的，需要阅读它的源码。

2. 粗糙度

粗糙度这个参数是用来实现渲染方程中的正态分布函数和几何函数，直接按照公式实现代码就可以了，没有困惑的地方。

3. 漫反射

BRDF方程中除了反射项，还需要一个漫反射项。其实漫反射项的比率 k_d 已经由菲涅尔方程指定了，也就是 $(1 - F_0)$ 。但是需要注意的是，漫反射的效果只对非金属有效，金属是没有漫反射的。物理原因是，折射进入金属表面内的光并不会在辐射出来。那么对于金属流工作材质中的金属度介于1和0之间的该如何计算呢，就是利用金属度这个数值对于 F_0 插值。具体代码如下：

```
vec3 ks = f;
vec3 kd = vec3(1.0) - f;
kd *= 1.0 - metallic;
```

漫反射表现出的是物体表面颜色，对于金属来说，我们可以当作是黑的，从物理角度解释就是，金属吸收所有折射进入金属的光。而对于非金属来说，可以从BaseColor贴图采样而来，物理角度上，非金属会对折射进入物体的光再辐射出去，辐射的时候因为表面属性，光的不同波长辐射比率不同。

那么为什么黄金会是黄色的呢，这个其实是反射现象，从菲涅尔想象可以解释。对于非金属的反射，对于rgb的反射比率一致，那么反射表现出来的就是光减弱了，但颜色不变。对于金属的反射，rgb的反射比率不一样，对于黄金来说，红色全部反射，那么就会再表面看见颜色。

四、总结

1.

以上是基于opengl学习材料、闫老师视频课以及其他网络资料做的总结。现在积分公式里边的一个微分项，可以了解代码实现，也就是说，对于数量一定的光源，可以实现BPR的效果。**当使用IBL技术或者说必须要用积分的时候，虽然知道理论上可以使用蒙特卡洛积分法，但是实践层面还有疑问。**

2.

对于不同的效果，即不同的渲染模型，比如二次元风格，是不是需要找到合理的BRDF即可呢？

3.

接下来准备再学习下IBL技术，然后做一下opengl的Demo。在看闫老师的课程的时候，对path-trace技术渲染出来的效果图感觉很厉害。一切的学习都是从模仿开始的，想尝试自己手撸一下。



Photo



Path traced:
global illumination