

- 渲染方程
- BRDF散射部分
 - 重要性采样
- BRDF高光部分
 - 法向分布函数(Normal Distribution Function)
 - 菲涅尔函数
 - 几何阴影函数
 - 高光BRDF的若干性质
 - IBL预算方案
 - Split Integral Approximation
 - DFG预算方案
 - LD预算方案
 - 1. 原始LD
 - 2. 假设1：D()不随视角改变，一直使用视角垂直于平面的lobe形状
 - 3. 假设2：半球积分域变成视角的反射向量的上半球面
 - 4. 实时渲染时获取预算方案结果
 - 重要性采样
- Filtered Importance Sampling
- 参考

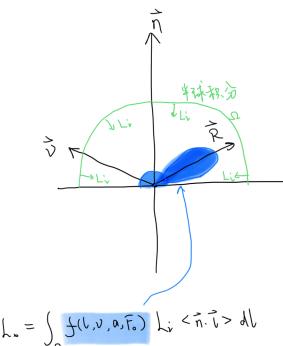
符号说明：

l ：入射光向量 L_i : 入射光强度 L_o : 出射光强度 v : 视角向量/出射光向量 a : 粗糙度 f : brdf函数 $\langle x \cdot y \rangle$: 向量 x 点成向量 y $\int_{\Omega} f(x \cdot y) dy$

渲染方程

$$L_o = \int_{\Omega} f(l, v, \Theta) L_i \langle n \cdot l \rangle dl$$

其中 Θ 是参数集合, 包含 菲涅尔系数 F_0 , 粗糙度, 等。



其中 f 有 diffuse 散射和 specular 高光两部分组成, 其中 specular 是入射光直接从物体表面反射出去的部分, diffuse 是入射光折射进入到物体内部后再次反射出去的部分。折射的反射的比例可以用菲涅尔函数 $F(v, h, F_0, F_{90})$ 来获得 (下面会介绍), 则基于能量守恒定律有:

$$f(l, v, \Theta) = (1 - F(v, h, F_0, F_{90})) f_d(l, v, h, a) + f_s(l, v, h, a)$$

其实上面的表述并不完全正确, 因为光线折射进入物体后, 并不一定会再次跑出物体。对于非金属, 光线是会再次跑出物体; 对于金属体, 进入物体的光线会被完全吸收, 不再跑到外边去。基于此, 将上述公式基于金属度 $metallic$ 参数改写成

$$f(l, v, \Theta) = (1 - F(v, h, F_0, F_{90})) (1 - metallic) f_d(l, v, h, a) + f_s(l, v, h, a)$$

BRDF散射部分

散射部分使用 Lambert Diffuse Brdf

$$brdf_d = c_{diffuse}/\pi;$$

c_diffuse表示材料的漫反射颜色，具体实物可以通过Spectralon漫反射标准板测量，gltf把这个值存储在baseColor贴图中。

散射部分的渲染方程

$$L_o = \frac{c_{diff}}{\pi} \int_{\Omega} L_i < n \cdot l > dl$$

可以看出渲染方程的积分结果只和法向量相关，这样我们可以将不同法向量的积分预计算然后存储到环境图，使用时基于法线量查询预计算的环境图 (textureCube(normal)) 就可以。

重要性采样

均匀半球采样

在球面上进行均匀的重要性采样的概率密度函数

$$p_d = \frac{1}{2\pi}$$

假设 ξ_1, ξ_2 是在[0, 1]范围内均匀采样的数值，则基于 p_d 进行重要性采样的方向向量的球坐标参数 ϕ 和 θ 可用如下公式获得：

$$\theta = \arccos(\xi_1)$$

$$\phi = 2\pi\xi_2$$

则散射部分的渲染方程进行基于重要性采样（均匀半球采样）的蒙特卡洛积分

$$L_o = \frac{1}{N * \pi} \sum_{i=1}^N L_i$$

BRDF高光部分

高光brdf

$$f_s(l, v, h, a) = \frac{D(n, h, a)F(v, h, F_0, F_{90})G(l, v, h, a)}{4 < n \cdot v > < n \cdot l >}$$

其中， $a = roughness * roughness$ 。

法向分布函数(Normal Distribution Function)

法向分布函数表示物体微表面的法线方向的概率分布。

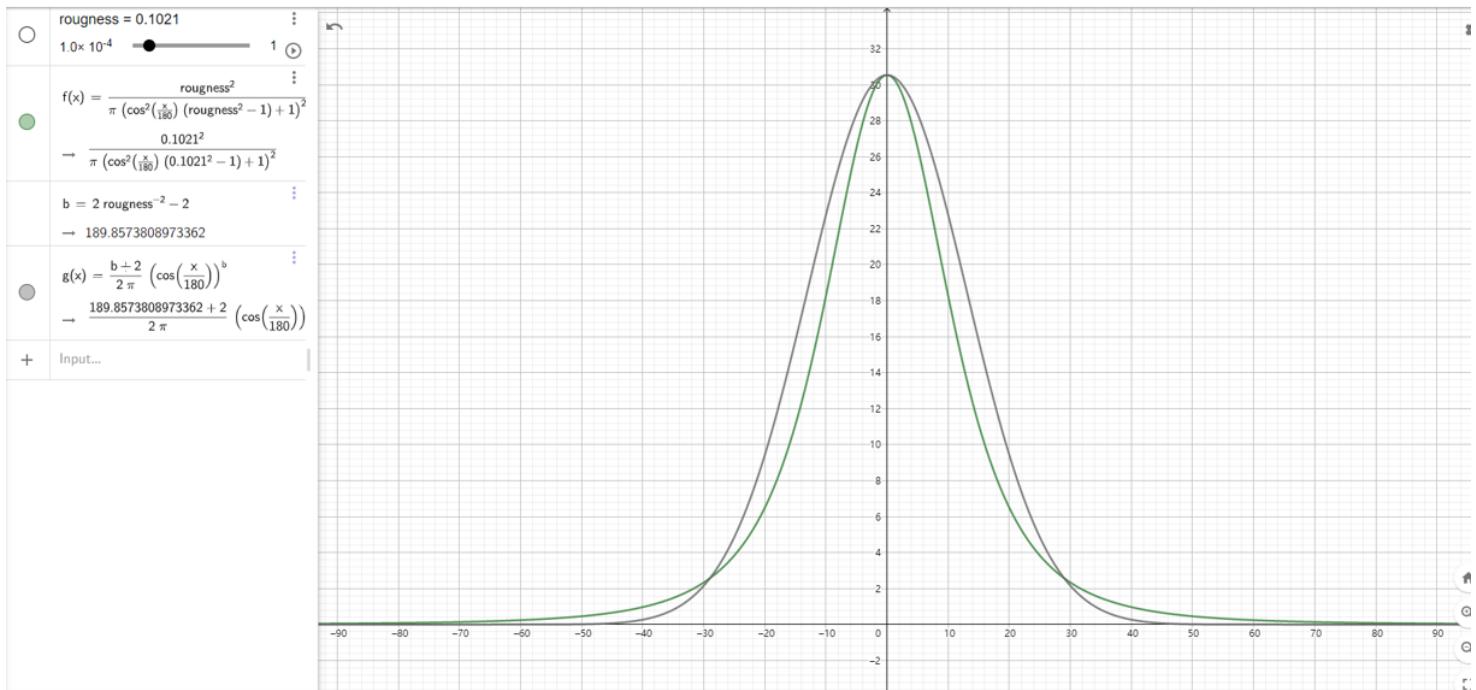
具体的，使用GGX法向分布函数

$$D(n, h, a) = \frac{a^2}{\pi(< n \cdot h >^2 (a^2 - 1) + 1)}$$

其他可选的法向分布函数有blinn phong。

$$D(n, h, a) = \frac{a_p + 2}{2\pi} (< n \cdot h >)^{a_p} \text{ 其中 } a_p = 2a^2 - 2$$

和GGX的比表结果如下图。可以看出当法线n和半角向量h的夹角比较大时，GGX的值更大，这种在渲染上表现为拖尾的效果，更接近真实的情况。



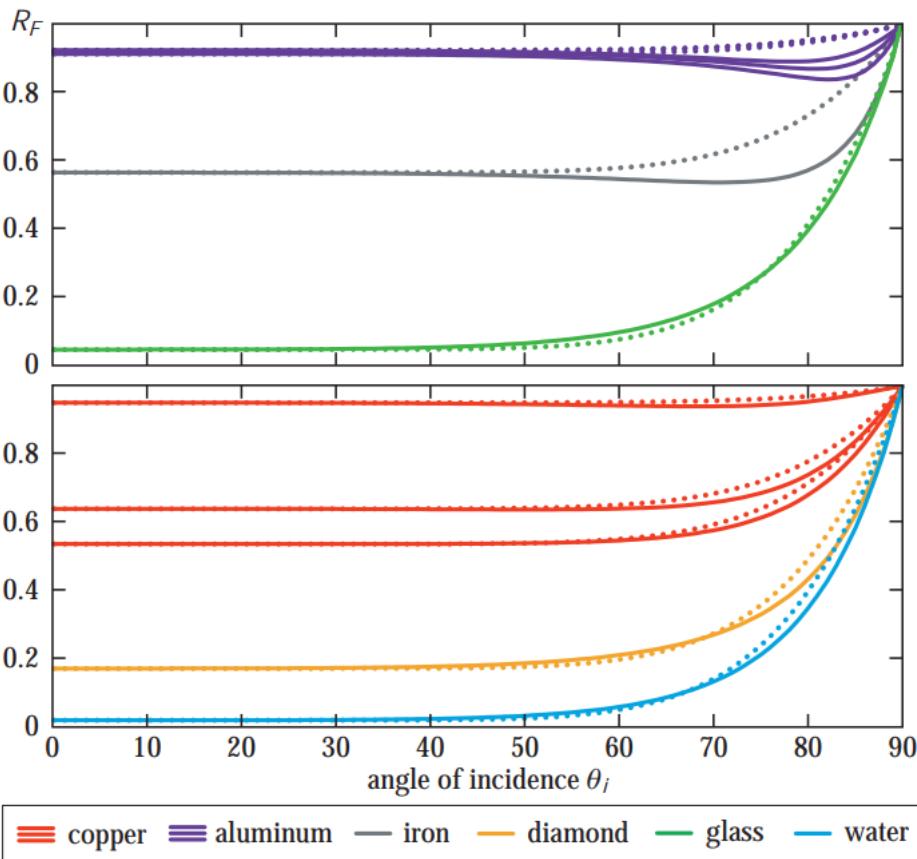
from Real Shading in Unreal Engine 4

菲涅尔函数

菲涅尔函数描述入射光照射到物体平面后反射光和折射光的比例，是一个rgb三维的值。



物体平面的菲涅尔值会随入射光和法线的夹角增加逐渐增加至1。如上图所示的湖面，视线看到越远处的湖面，入射光和法线的夹角越大，菲涅尔值越大，反射光比折射的光越多；视线看到越近处的湖面，入射光和法线的夹角越小，菲涅尔值越小，反射光比折射的光越少。下图是一些物体的菲涅尔值随入射光和法线夹角的变化曲线，其中铜 铝有三条曲线是因为其对光的rgb三通道的菲涅尔系数是不一样的。



Schlick's approximation的近似结果（虚线）和实际值的比较

这条曲线可以用一个近似函数表示(Schlick's approximation)

$$F(v, h, F_0, F_{90}) = F_0 + (F_{90} - F_0)(1 - \langle v \cdot n \rangle^5)$$

F90(夹角90度)一般直接取1.0, F0 (夹角0度) 的取值可参考如下表格 , 或者根据公式 , 代入折射率ior计算。

$$F_0 = \left(\frac{ior - 1}{ior + 1} \right)^2$$

Insulator	$R_F(0^\circ)$ (Linear)	$R_F(0^\circ)$ (sRGB)	Color
Water	0.02,0.02,0.02	0.15,0.15,0.15	
Plastic / Glass (Low)	0.03,0.03,0.03	0.21,0.21,0.21	
Plastic High	0.05,0.05,0.05	0.24,0.24,0.24	
Glass (High) / Ruby	0.08,0.08,0.08	0.31,0.31,0.31	
Diamond	0.17,0.17,0.17	0.45,0.45,0.45	

上图是非金属的F0

Metal	$R_F(0^\circ)$ (Linear)	$R_F(0^\circ)$ (sRGB)	Color
Gold	1.00,0.71,0.29	1.00,0.86,0.57	
Silver	0.95,0.93,0.88	0.98,0.97,0.95	
Copper	0.95,0.64,0.54	0.98,0.82,0.76	
Iron	0.56,0.57,0.58	0.77,0.78,0.78	
Aluminum	0.91,0.92,0.92	0.96,0.96,0.97	

上图是金属的F0

在GLTF标准中 , 对于金属工作流 :

非金属 , F0统一取0.04。

金属 , F0取baseColor贴图的颜色。

因为金属的baseColor的颜色存的就是F0 , 具体测量方式还不清楚。

所以F0就可以统一由以下公式获得 :

$$F_0 = vec3(0.04) + metalness * (albedo.rgb - vec3(0.04))$$

对于微表面模型, 参与光照的有效微表面需满足 $n==h$, 菲涅尔函数有

$$F(v, h, F_0, F_{90}) = F_0 + (F_{90} - F_0)(1 - \langle v \cdot h \rangle^5)$$

注 : 菲涅尔函数部分的图表均来自Realtime Rendering 3rd edition。

几何阴影函数

Height-Correlated Smith Geometric Shaow Function [Heitz14]

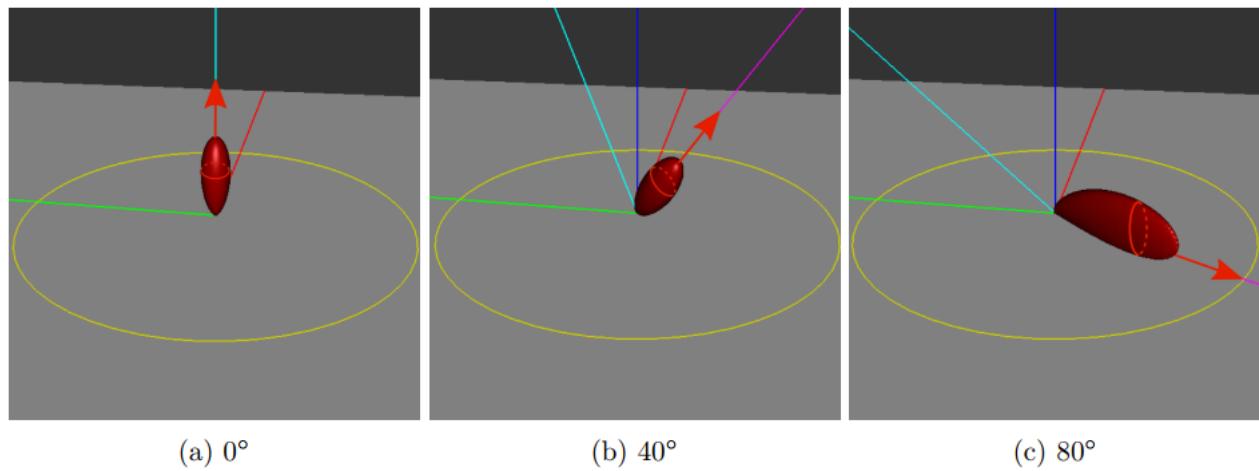
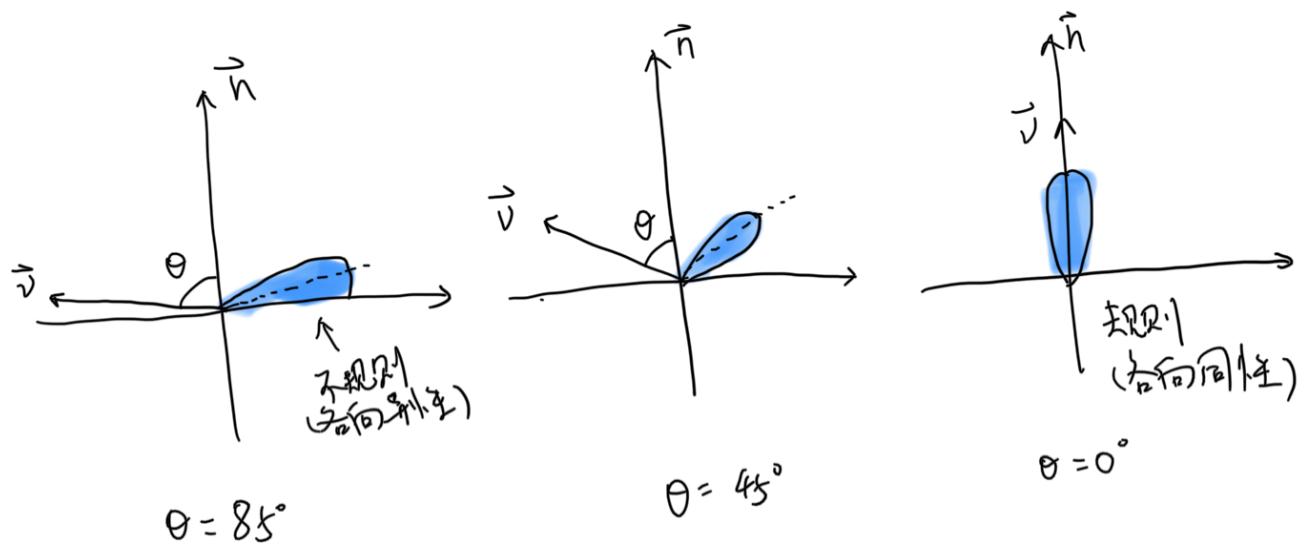
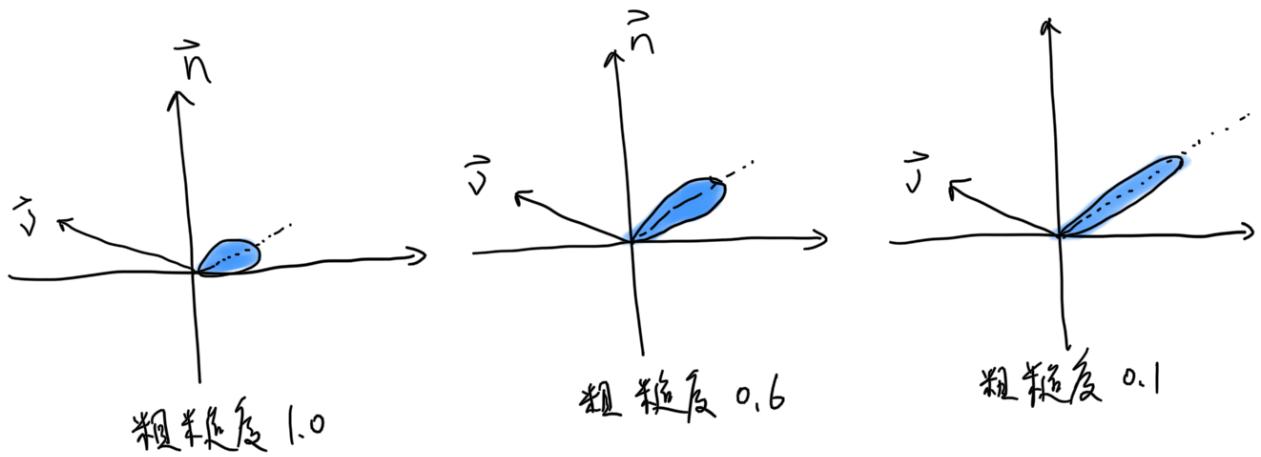
$$G(n, v, l, a) = \frac{2 \langle n \cdot l \rangle}{\sqrt{\langle n \cdot l \rangle + \sqrt{a^2 + (1 - a^2) \langle n \cdot l \rangle}}} \frac{2 \langle n \cdot v \rangle}{\sqrt{\langle n \cdot v \rangle + \sqrt{a^2 + (1 - a^2) \langle n \cdot v \rangle}}}$$

高光BRDF的若干性质

粗糙度越大 , 高光BRDF的形状越钝 , 如下图所示。

其次 , 其分布只有在 $v=n$ 时是比较接近各向同性的 , 在大多数情况下是非常不均匀的。

在 v 和 n 的夹角接近于90度时 , 分布尤其不均匀。



上图是一个使用了GGX NDF的BRDF的可视化结果，来自 [Charles 2014]

IBL预算方案

Split Integral Approximation

split Integral Approximation：将渲染方程的积分近似拆成环境图光照相关的以及环境图光照无关的两部分

这个近似并没有理论依据，是一个经验性的近似。这样可以把F0和环境图进行解耦，对于F0相关的预积分不再需要和环境图相关联。

$$L_o = \int_{\Omega} f_r(l, v, \Theta) L_i < n \cdot l > dl$$
$$L_o \approx \underbrace{\left(\int_{\Omega} f_r(l, v, \Theta) < n \cdot l > dl \right)}_{DFG} \underbrace{\left(\frac{\int_{\Omega} D(l, v, n, a) L_i < n \cdot l > dl}{\int_{\Omega} D(l, v, n, a) < n \cdot l > dl} \right)}_{LD}$$

DFG预算

代入高光brdf

$$DFG = \int_{\Omega} \frac{D(n, h, a) F(v, h, F_0, F_{90}) G(l, v, h, a)}{4 < n \cdot v > < n \cdot l >} < n \cdot l > dl$$

由于DFG的积分没有环境光照，所以DFG积分结果只受brdf的lobe的形状的影响，brdf的lobe的形状由粗糙度，F0还有视角的法向量的夹角theta。这里有三个变量，还是太多，可以代入菲涅尔项的具体公式将F0从积分中分理出，拆分出的DFG1和DFG2都和F0无关，只和粗糙度和夹角theta相关。

具体地，做积分的重要性采样，概率密度函数使用如下ps

$$p_s = \frac{D(n, h, a) < n \cdot h >}{4 < v \cdot h >}$$

有

$$DFG = \frac{1}{N} \sum_{i=1}^N \frac{F(v, h, F_0, F_{90}) G(l, v, h, a) < v \cdot h >}{< n \cdot v > < n \cdot h >}$$

定义一个可见性函数 Visibility Function

$$V(l, v, h, a) = \frac{G(l, v, h, a) < v \cdot h >}{< n \cdot v > < n \cdot h >}$$

代入有

$$DFG = \frac{1}{N} \sum_{i=1}^N F(v, h, F_0, F_{90}) V(l, v, h, a)$$

将F菲涅尔项

$$F(v, h, F_0, F_{90}) = F_0 + (F_{90} - F_0)(1 - < v \cdot h >^5)$$

代入DFG有

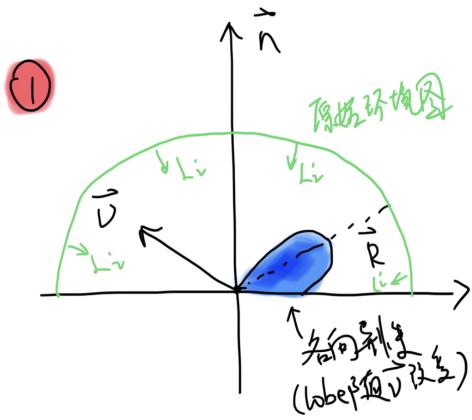
$$DFG = \frac{1}{N} \sum_{i=1}^N V(l, v, h, a) (F_0 + (F_{90} - F_0)(1 - < v \cdot h >^5)) DFG = F_0 \underbrace{\frac{1}{N} \sum_{i=1}^N (1 - (1 - < v \cdot h >^5)) V(l, v, h, a)}_{DFG_1} + F_{90} \underbrace{\frac{1}{N} \sum_{i=1}^N (1 - < v \cdot h >^5) V(l, v, h, a)}_{DFC}$$

DFG1和theta粗糙度a相关。类似地DFG2也和theta，粗糙度a相关。针对不同的v和粗糙度，我们对其进行积分，并将积分结果存储在纹理上，纹理的横坐标是粗糙度，纵坐标是v。

LD预算

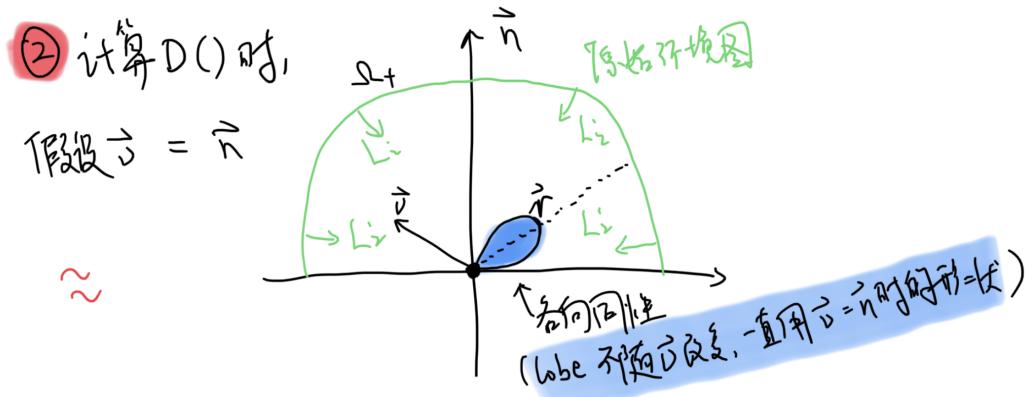
LD的预算算主要思路如下：

1. 原始LD



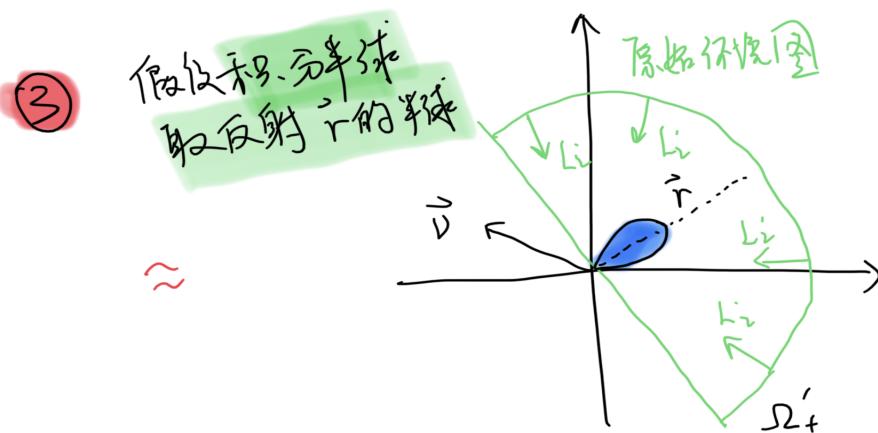
$$LD = \frac{\int_{\Omega} L_i D(l, v, n, a) < n \cdot l > dl}{\int_{\Omega} D(l, v, n, a) < n \cdot l > dl}$$

2. 假设1：D()不随视角改变，一直使用视角垂直于平面的lobe形状



$$LD = \frac{\int_{\Omega} L_i D(l, n, n, a) < n \cdot l > dl}{\int_{\Omega} D(l, n, n, a) < n \cdot l > dl}$$

3. 假设2：半球积分域变成视角的反射向量的上半球面



注意：LD的积分域变更到r的上半球面后，LD积分中的法线n变成上图中的反射向量r。

做LD积分的重要性采样，概率密度函数使用如下ps

$$p_s = \frac{D(l, v, n, a) < n \cdot h >}{4 < v \cdot h >} \text{ let } v = np_s = \frac{D(l, n, n, a)}{4}$$

有

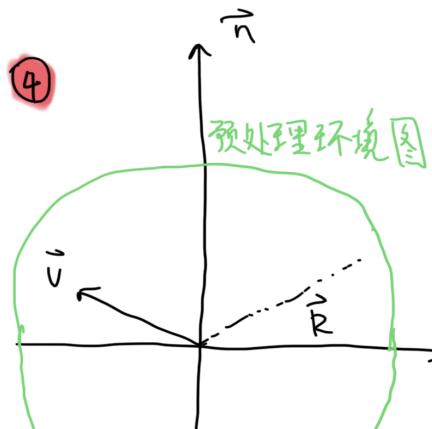
$$LD \approx \frac{\sum_{i=1}^N \frac{D(l, n, n, a) < n \cdot l > L_i}{p_s}}{\sum_{i=1}^N \frac{D(l, n, n, a) < n \cdot l >}{p_s}} = \frac{\sum_{i=1}^N < n \cdot l > L_i}{\sum_{i=1}^N < n \cdot l >}$$

将不同的r和粗糙度对应的积分结果预计算后存储在环境图里。



[上图prefiltered cubemap 来自 learn-opengl](#)

4. 实时渲染时获取预计算结果



在实时渲染过程中，通过计算视角向量v的反射向量r，根据当前的粗糙度，去范围预计算的环境图得到LD的值。

重要性采样

在具体使用时，pdf是GGX NDF* Jacobian()

$$p_s = \frac{D(h, a) < n \cdot h >}{4 < v \cdot h >} \\ \text{假设 } n = v, \\ p_s = \frac{D(h, a)}{4}$$

其中

$$D(n, h, a) = \frac{a^2}{\pi(< n \cdot h >^2 (a^2 - 1) + 1)}$$

固定n和a，D是关于h的函数

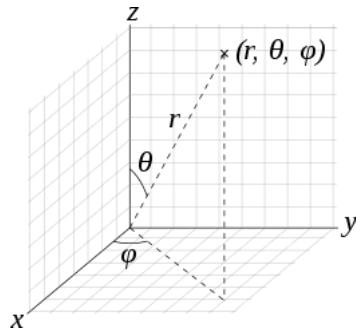
要想获得上述pdf的重要性采样的向量，可以通过**Inversion Method**来实现。

通过若干推导，可得

ξ_1, ξ_2 是在 $[0, 1]$ 范围内均匀采样的数值，则基于 p_s 进行重要性采样的方向向量的球坐标 ϕ 和 θ 可用如下公式获得：

$$\phi = 2\pi\xi_1 \\ \theta = \arccos \sqrt{\frac{1 - \xi_2}{\xi_2(a^2 - 1) + 1}}$$

其中方向向量可以用球坐标系来表示：

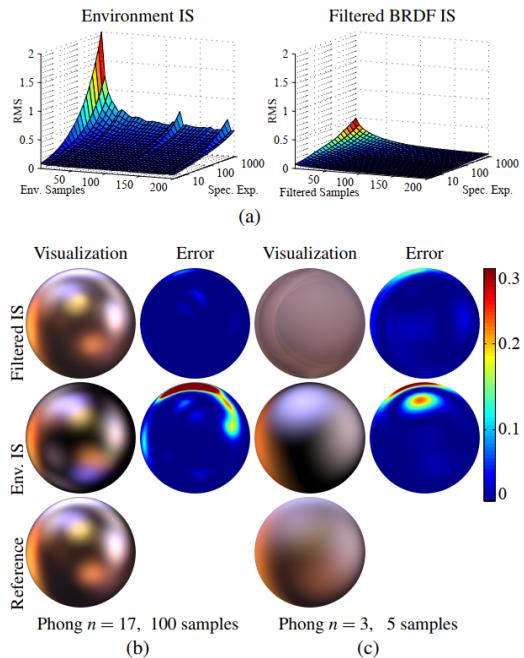


$$\begin{aligned}x &= \sin(\theta)\cos(\phi) \\y &= \sin(\theta)\sin(\phi) \\z &= \cos(\theta)\end{aligned}$$

Filtered Importance Sampling

基于上述的重要性采样方法进行渲染方程的积分计算依然十分耗时，Colbert等[Colbert 2007]提出了Filtered Importance Sampling。其核心想法是，在做重要性采样时，pdf越小的样本，采集环境图的层级越高。这样重要性采样收敛的更快。

由下图可以看出，(a) Filter Importance Samping 比 Environment Importance Sampling的RMS收敛的更快。(b) 在高频BRDF, Filtered Importance Sampling的RMS误差和视觉效果都更好；在低频BRDF, Filter Importance Samping的RMS误差更小，但是视觉效果不如Environment Importance Sampling。



参考

- [Tomas 2008] Tomas, Eric, Naty Hoffman, "Real-Time Rendering", Third Edition, 2008
- [Burley 2012] Brent Burley, "Physically Based Shading at Disney", SIGGRAPH 2012, PBR Course
- [Karis 2013] Brian Karis, "Real Shading in Unreal Engine 4", SIGGRAPH 2013, PBR Course
- [Charles 2014] Charles de Rousiers, "**Moving Frostbite to PBR**", SIGGRAPH 2014, PBR Course [链接](#)
- [Bruno 2019] Bruno Ospenica, Image Based Lighting with Multiple Scattering, [链接](#)
- [Filament by Google](#)
- <https://dassaultsystems-technology.github.io/EnterprisePBRShadingModel/spec-2022x.md.html>
- <https://autodesk.github.io/standard-surface/>
- [Colbert 2007] Mark Colbert, GPU-Based Importance Sampling, 2007, [链接](#) [链接](#)