

基于物理着色（二）- Microfacet材质和多层材质



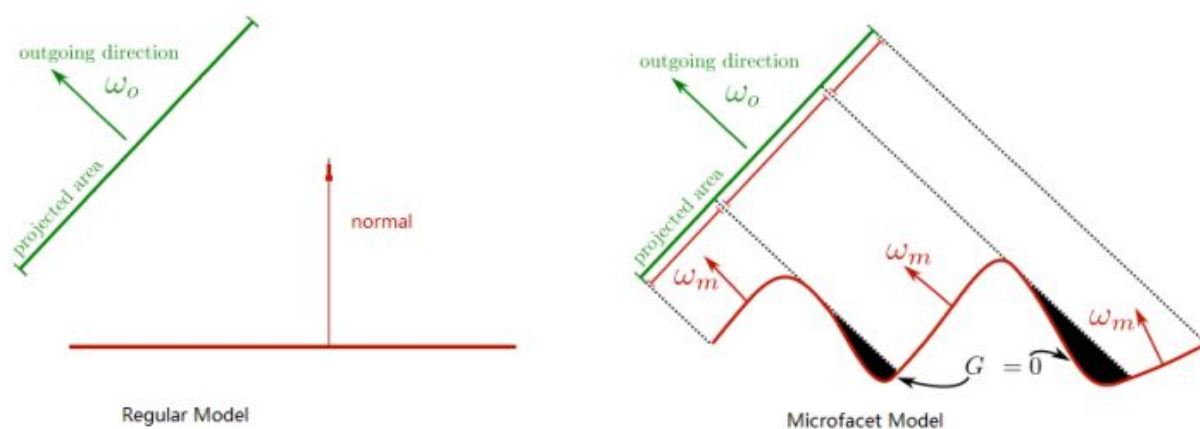
文刀秋二

计算机图形学 话题的优秀回答者

上一篇文章介绍了基于物理着色的基本概念以及最基本的漫反射与镜面反射模型，这一篇会介绍利用Microfacet模型对粗糙镜面反射的模拟，以及多层材质的模拟例如塑料，陶瓷，车漆等。有了这些模型我们就已经可以很真实的模拟出很大一部分现实中的材质了。

Microfacet

最常用的Microfacet模型是Cook Torrance。与普通的着色模型的区别在于，普通的着色模型假设着色的区域是一个平滑的表面，表面的方向可以用一个单一的法线向量来定义来定义。而Microfacet模型则认为 1) 着色的区域是一个有无数比入射光线覆盖范围更小的微小表面组成的粗糙区域。2) 所有这些微小表面都是光滑镜面反射的表面。因为这种模型认为着色区域的表面是由无数方向不同的小表面组成的，所以在Microfacet模型中，着色区域并不能用一个法线向量来表示表面的方向，只能用一个概率分布函数 D 来计算任意方向的微小表面在着色区域中存在的概率。



因为每一个微小表面都做完美镜面反射，上一节提到过完美镜面反射只在入射光和反射光根据发现镜面反射对称的时候才有能量，所以在计算微小表面的BRDF值时，只用先根据入射方向 w_o 和出射方向 w_i 计算出中间向量 w_m 也就是微小表面的法线方向，然后用就可以用法线分布函数计算出这个能够完美镜面反射当前入射和出射光 $D(w_m)$ 。同样因为是完美镜面反射的缘故，Cook Torrance模型的另一个因素就是菲涅尔（Fresnel）项 $F(w_o, w_m)$ ，这个项和上一节镜面反射BRDF中通到的一样，用于计算不同入射角度的情况下反射的光线的强度，对于导体和电介材质记得使用不同的Fresnel公式。最后，为了更好的模拟着色区域的凹凸不平， $G(w_o, w_i, w_m)$ 项则模拟了凹凸表面间的遮挡因素，如上图所示。省略掉复杂具体的推导过程，Cook Torrance模型的BRDF就是：

$$f(w_o, w_i) = \frac{D(w_m)F(w_o, w_m)G(w_o, w_i, w_m)}{4 \times \cos(w_o \cdot w_n) \times \cos(w_i \cdot w_n)}, \text{ 其中 } w_m = \frac{w_o + w_i}{|w_o + w_i|}$$

是微小表面的法线，就是入射和出射光线中间的方向， w_n 是着色区域实际表面的法线，和普通着色模型中的法线是一样的。

人们也根据这种模型设计出了许多不一样的法线分布函数 D 。大家熟知的Blinn-Phong就是一种很简单的分布函数，相信大家在学最简单的Lighting Shader的时候肯定都看过这段代码

```
// 计算镜面反射强度
float NdotH = dot(normal, Wm); // Wm = normalize(Wo + Wi)
float specularIntensity = pow(saturate( NdotH ), e);
```

这就是基于Blinn-Phong的法线分布

$D_{Blinn}(w_m) = (w_n \cdot w_m)^e$ ，其中 e 代表粗糙程度。

Blinn-Phong并不是一个非常真实的分布，上面的公式也并不满足能量守恒定律（ D 需要满足 w_m 在半球面上积分为1，Blinn-Phong还需乘以一个常量 $\frac{e+2}{2\pi}$ 才满足这个条件）。近年来有许多新的分布函数被发明出来例如Beckmann，GGX等，他们的能量分布都更接近用光学仪器测量的反射数据。Disney BRDF也使用了GGX分布。本文后面的所有渲染结果也使用了GGX。 G 项也有各种不同的选择，本文使用了有粗糙度作为参数的Smith模型。网上GGX和Smith G项的实现很多，这里就不贴公式了，直接贴两段伪代码。

```
float GGX_D(Vector3 wm, float alpha) // alpha为粗糙度
{
    float tanTheta2 = TanTheta2(wm),
          cosTheta2 = CosTheta2(wm);

    float root = alpha / (cosTheta2 * (alpha * alpha + tanTheta2));

    return INV_PI * (root * root);
}

float Smith_G(Vector3 wo, Vector3 wi, Vector3 wm, float alpha)
{
    auto SmithG1 = [&](Vector3 v, Vector wm)
    {
```

```

float tanTheta = abs(TanTheta(v));

if (tanTheta == 0.0f)
    return 1.0f;

if (Dot(v, wm) * CosTheta(v) <= 0)
    return 0.0f;

float root = alpha * tanTheta;
return 2.0f / (1.0f + Sqrt(1.0f + root*root));
};

return SmithG1(wo, wm) * SmithG1(wi, wm);
}

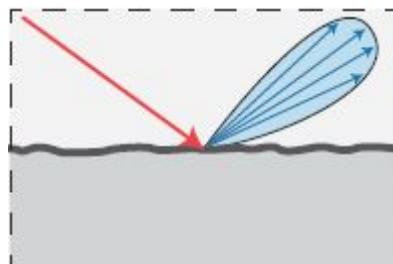
```

粗糙镜面反射

有了Microfacet，我们就可以模拟现实中由非常细小的粗糙表面组成的镜面反射表面了。生活中这种材质很常见，各种粗糙的金属，iPhone6的背面也是：



这种材质将入射光根据表面的粗糙程度分散的反射到镜面反射方向周围的方向，表面越粗糙反射的分布越分散。

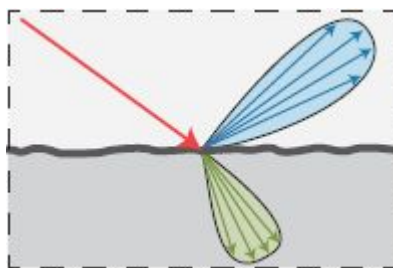


下面的两张渲染结果都是用GGX分布计算的，第二张比第一张更加粗糙（个人觉得第二张很像深空灰iPhone6）。

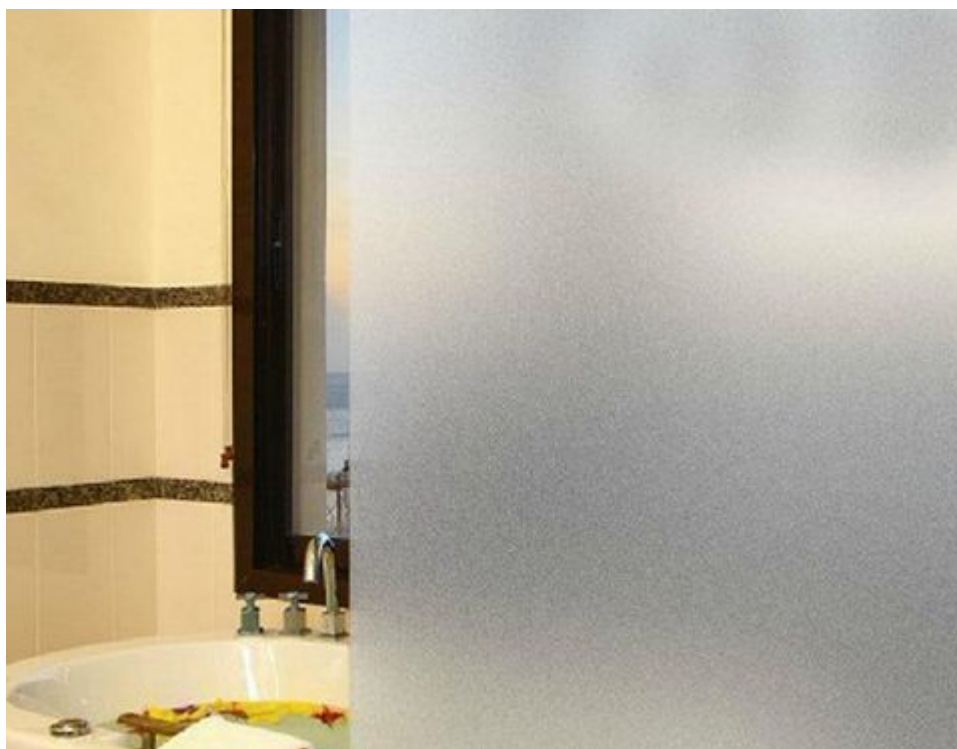


粗糙镜面折射

Microfacet模型同样可以用于模拟粗糙的折射表面。顾名思义这一种材质则会将入射光线同时反射和折射到镜面反射和折射周围的方向，分布大小取决于表面的粗糙程度。



这种材质在现实中的例子就是各种毛玻璃：



折射光线能量的计算相比反射的略微复杂，想了解详细和看推倒过程的可以读这篇 Paper：[Microfacet Models for Refraction through Rough Surface](#)。

在BRDF计算的时候先要根据入射的方向决定当前发生的是反射还是折射，然后用不同的方法算出两种情况下的 w_m ，然后再用反射或者折射的Microfacet公式计算出结果。简单写一下这个过程的伪代码：

```
float f(Vector3 wo, Vector3 wi, Vector3 wn)
{
    bool IsReflection = true;
    // 根据出入射方向和法线方向判定是反射还是折射
    if(Dot(wo, wn) * Dot(wi, wn) < 0.0f)
        IsReflection = false;

    Vector3 wm;
    if(IsReflection)
```

```

        wm = Normalize(wo + wi);
    else // 如果是折射则根据折射率算出wm
        wm = -Normalize(etai * wo + etat * wi);

    float D = GGX_D(wm, Roughness);
    float F = Fresnel(Dot(wo, wm), etai, etat);
    float G = GGX_G(wo, wi, wm, Roughness);

    // 有力D, F和G就可以套入Microfacet的公式计算当前BRDF的返回值。
}

```

下面贴上用GGX分布计算的粗糙折射的结果。注意在这个Shader Ball的模型的表面下方故意放置了一个Bar，可以看到他因为表面粗糙的关系而被模糊了。第二个比第一个更为粗糙。





当然想真正渲染出更有趣更真实的图片还是少不了Artists给你提供好的Roughness map和法线图。本篇文章的封面就是用Microfacet的粗糙折射材质配合好看的Roughness map渲染的。

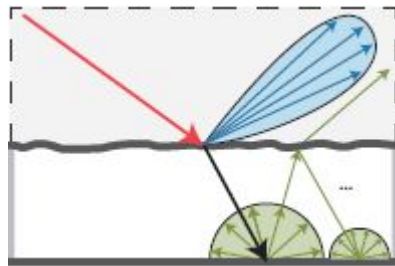
写到这里，已经介绍过了漫反射，完美镜面反射和折射，以及基于Microfacet的粗糙反射和折射。然而只有这些还不足以模拟现实中的许多材质。下面介绍另一类很重要的材质，多层材质。

多层材质 (Multi-Layered Material)

现实中其实绝大多数的材质都有超过一层以上的组成部分，例如木地板。木头本身是粗糙的漫反射表面，但是木地板都会在表面涂上一层镜面反射的透明油漆。这一类的材质比比皆是，塑料，大理石，瓷器等等。而且严格来说许多漫反射的材质都有些许的镜面反射组成部分在里面。这大概也是当年最早的固定管线渲染可以让你设置镜面光和漫反射光的颜色，渲染的结果就像塑料一样的原因吧。



所以想要渲染出上图一样的材质，就也一定要在构建材质模型的时候考虑多层的信息。幸运的是，我们现在手上已经有之前介绍过的材质作为我们的building block。光是漫反射的表面上增加一层透明的镜面图层（可以是粗糙也可以是平滑）就可以模拟出许多新的材质了！光线传递示意图如下。

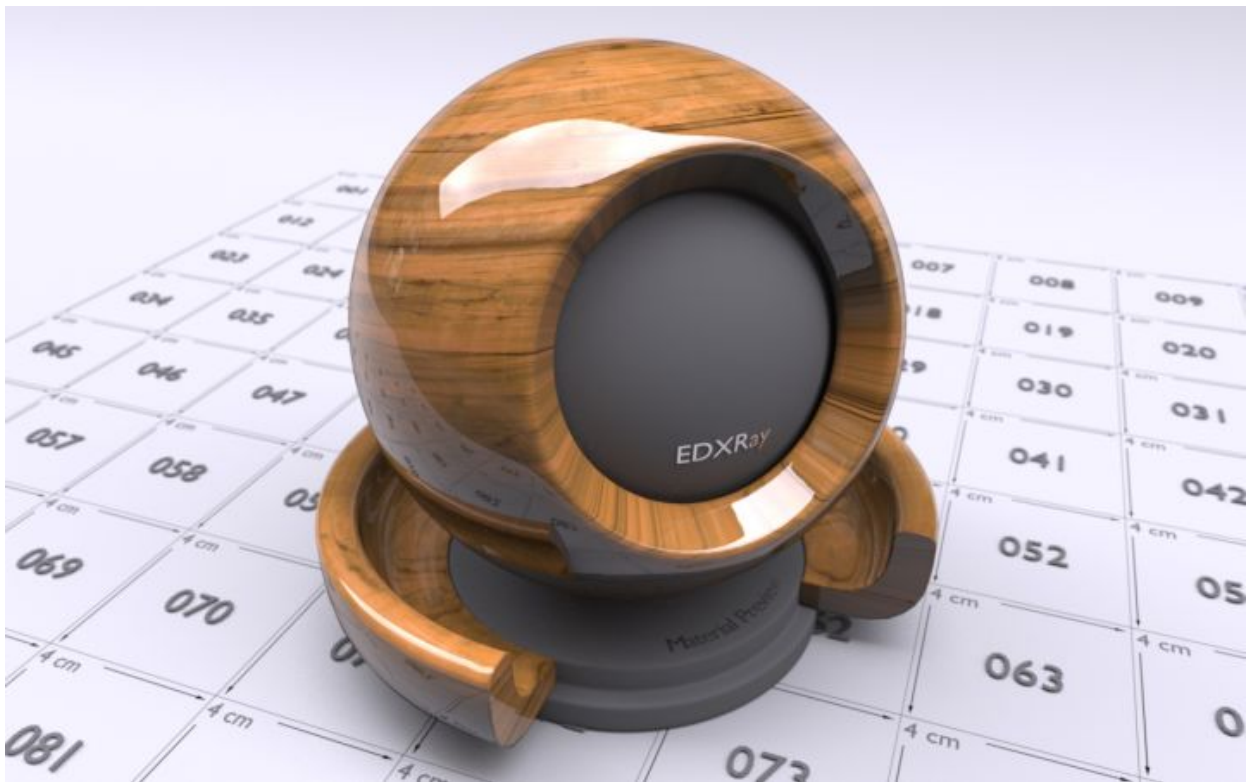


对于以上这种多层次的材质，如果不考虑不咋的层次间的反射和折射的话（通常这一部分对视觉的影响也非常小），直接将Microfacet和漫反射的部分相加就可以得到非常不错的效果。

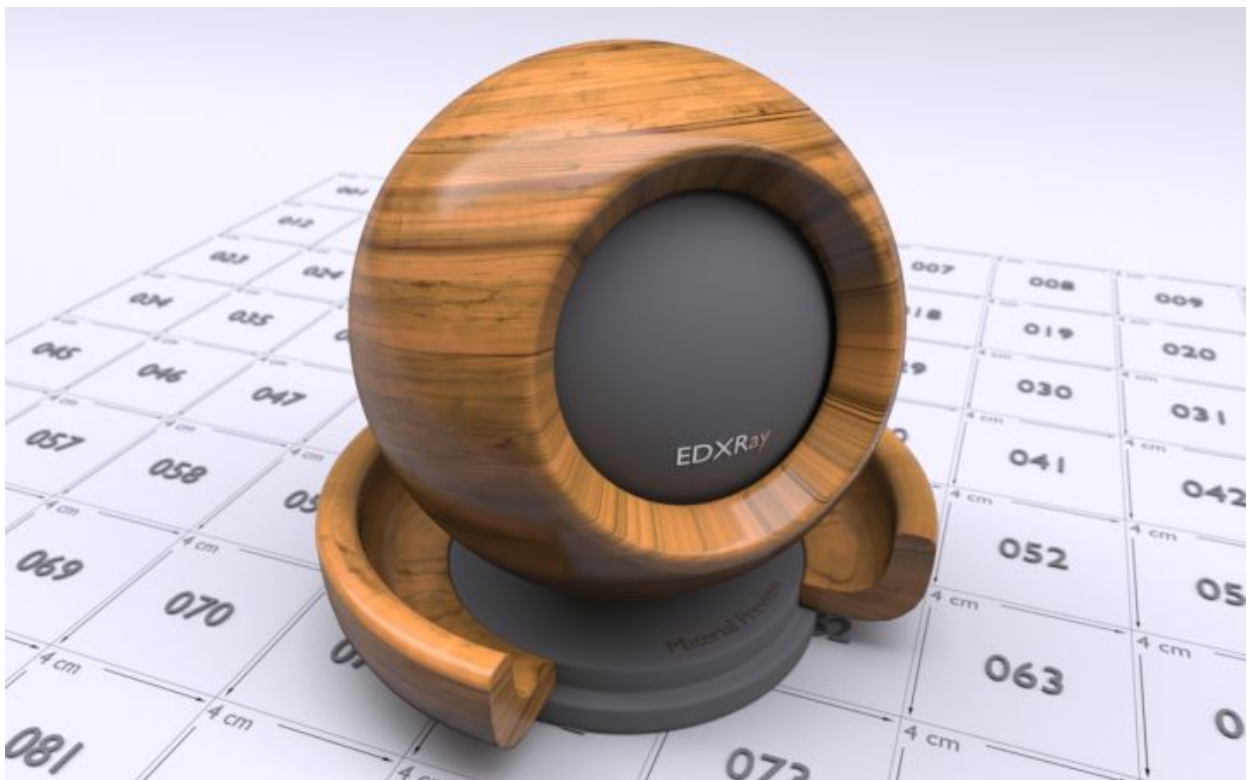
$$f(w_o, w_i) = f_{diffuse}(w_o, w_i) + \frac{D(w_m)F(w_o, w_m)G(w_o, w_i, w_m)}{4 \times \cos(w_o \cdot w_n) \times \cos(w_i \cdot w_n)}$$

镜面反射和漫反射的能量的权重需要用电介材质的菲涅尔项进行计算。如果要更加精确的计算不同材质间内部的反射，吸收，以及考虑材质厚度的影响，推荐阅读这一篇论文：[Arbitrarily Layered Micro-Facet Surfaces](#)

下面是直接将漫反射项和Microfacet项用Fresnel权重相加得到的表面涂有清漆的木材渲染结果。



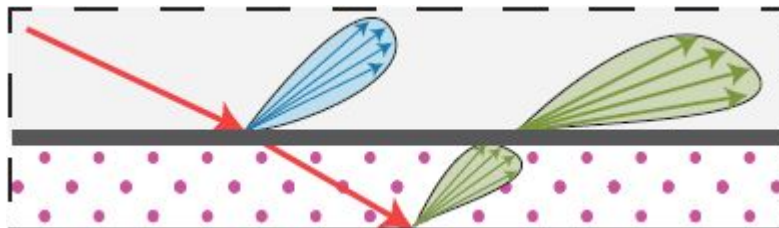
这是表面的清漆不那么光滑的渲染结果。



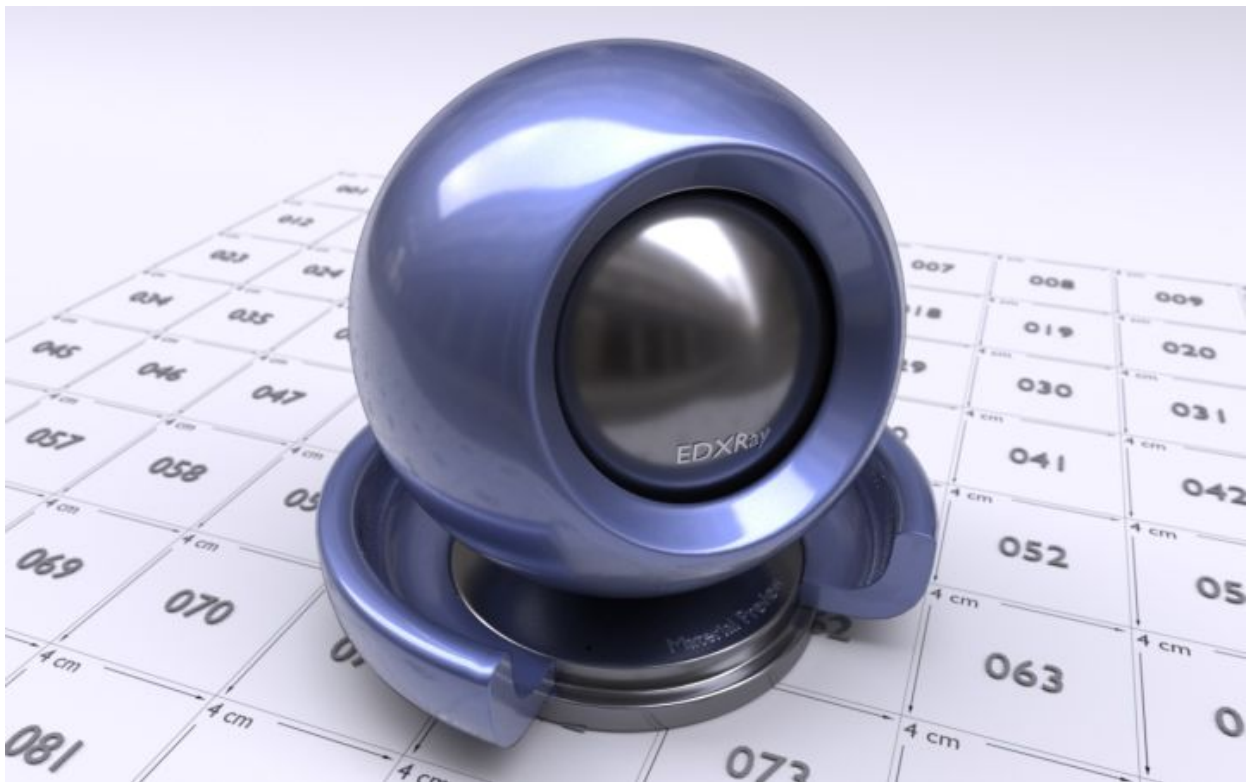
有了这种模型，就已经可以模拟很大一部分现实中的材质了。但是还有相当一部分的材质，一层漫反射加上一层镜面反射是不足以模拟的，一个非常常见的例子就是车漆。通常车漆是在一层粗糙的金属材质外层涂上一层平滑的透明清漆，所以看起来才有下图的样子，首先有金属的质感，但是同时表面又有非常细致的镜面反射。



这一类材质的光线传递示意图如下：



所以对于这种材质来说则需要多次用Fresnel项去叠加两个Microfacet镜面反射。就可以渲染出下图中的材质。



可采样性 (Being Samplable)

最后再补充一点，一个材质模型除了需要能计算出特定入射和出射方向的反射的能量大小，还必须是可采样的。可采样意味着要能够给定一个入射方向的时候，重要性采样 (Importance Sampling) 散射之后的出射方向，以及告诉你当前被采样到的出射方向的概率密度是多少。这是Physically Based Shading非常重要的一点，因为所有的渲染算法都依赖于在材质表面进行散射构造光线传递的路径。所以本文提到的所有材质模型在被设计的时候都是可以进行Analytical的采样的。采样出射光线也被虚幻引擎4在Prefilter环境贴图的时候用到了，这也是虚幻引擎使用Physically Based Shading的好处之一。当然如果有一天游戏想用光线追踪的话，所有的材质都必须满足可采样性才可以。所以基于物理着色的大趋势还是很好的。

写到这里基本上我已经覆盖了所有我这个系列想写的材质了。总的来说手上能用的工具就是这几个模型，通过不同的叠加 (用Fresnel)，或者是插值，改变粗糙度等就可以模拟生活中绝大部分材质了。在下一节 (基于物理着色 (三) - Disney和UE4的实现) 我则会谈一谈Disney Principled BRDF是如何运用这两篇文章介绍的这些模型设计出一个Artists友好的材质系统，希望能带来一些启发。

当然我这两篇文章覆盖的材质模型还遗漏了现实中一大类材质。那就是Participating Media，也就是光线会在材质内部散射的材质，又叫做次表面散射。例如皮肤，玉，蜡

烛等等。这个系列就不写了，因为无论从模拟的模型和计算量的要求都和本文提及的材质区别很大。以后会单独写一篇文章聊聊那一类材质，敬请关注！

本文中光线传递的示意图出自：[mitsuba-renderer.org/re...](https://mitsuba-renderer.org/research/light-tracing/)，其他实物图片来自网络搜索

本文中其他所有渲染结果均使用本人自己开发的渲染器EDXRay。（链接是非常旧的介绍，见谅）