

# 实时渲染中的光照模型及BRDF公式介绍

2018年1月22日 王欣捷

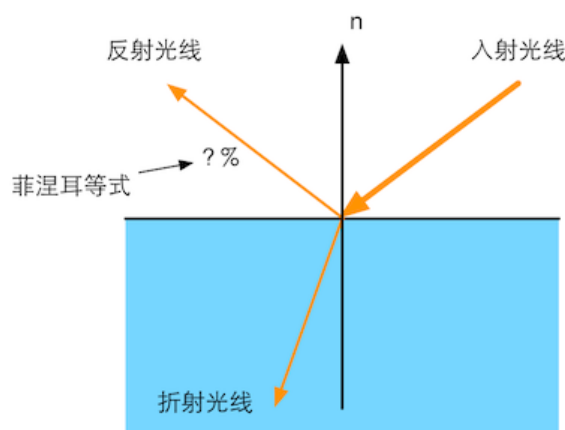
## 1. 光照基础

### 1.1. 成像的过程

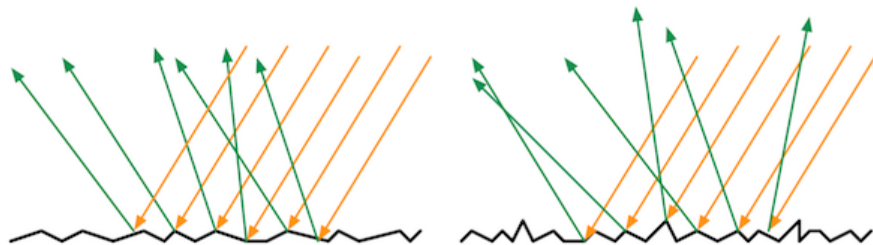
1. 光线从光源(light source)被发射出来。
2. 光线和场景中一些物体相交，一些光线被物体吸收了，另一些光线被散射到其他方向。
3. 摄像机吸收了一些光，产生了一张图像。

### 1.2. 光与物体的交互过程

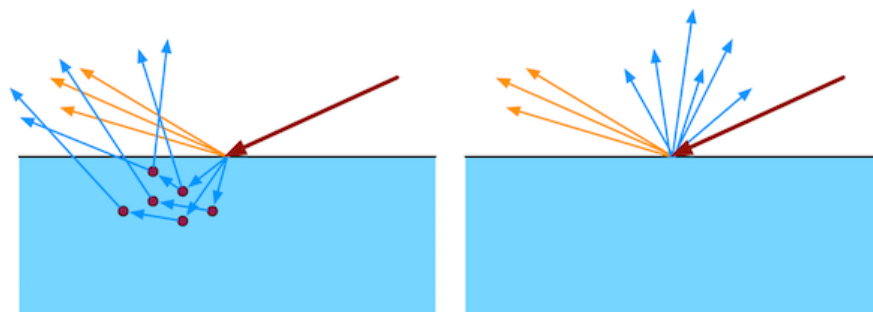
- 通过上面的过程，我们知道物体和光线相交会产生两种物理现象：散射和吸收（还有自发光，此处不考虑）。吸收会改变光的总能量，但不会改变光的方向；散射则不会改变光的总能量，但会改变它的反射方向。
- 现实生活中的光和物体交互过程是非常复杂的，但在渲染中，我们简化这个过程，只考虑**散射**。而散射又有两种情况：**表面反射**和**折射**。当光线从一种介质进入另一种介质时，我们假设光会被分割成两个方向，反射方向和折射方向。在传统模型中，通常只考虑表面反射而不考虑折射；在PBR中，用**菲涅尔等式**（Fresnel equation）来描述有百分之多少的光被反射，剩下的光就是被折射了。



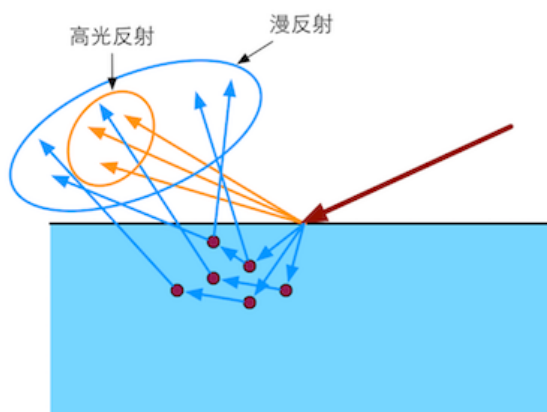
- 对于表面反射而言，除了绝对光滑的镜面物体，并不是所有的入射光线都可以完美地沿着轴对称的方向反射出去。不同材质的物体表面，其粗糙程度是不一样的，如果我们用一个放大镜去观察这些物体表面，会发现有许多凹凸不平的细节，这些细节被称为**微表面**。在微表面下，光线会被局部的法线反射到不同的方向，光滑物体的表面法线变化角度小，所以更容易反射到同一个方向，而粗糙物体的表面法线变化剧烈，光线会反射到各个方向。



- 图解：光滑表面的微平面的法线变化较小，反射光线的方向变化也 smaller。右图：粗糙表面的微平面的法线变化较大，反射光线的方向变化也更大
- 除了表面反射，一些折射到物体内部的光也会被散射出来，这些光被称为次表面散射光（subsurface-scattered light）。次表面散射光会从不同于入射点的位置从物体内部再次射出，根据这些光离入射点的距离值和屏幕像素的大小之间的关系，会产生两种建模结果，如果散射距离足够小，小于一个像素的距离，则渲染可以在局部进行，即我们可以在一个像素内把反射分成表面反射和次表面散射两部分来求解。如果散射距离大于一个像素，那么要采用特殊的渲染模型了，也就是所谓的次表面散射渲染技术 (subsurface scattering rendering technique)



- 图解：次表面散射。左图：次表面散射的光线会从不同于入射点的位置射出。如果这些距离值小于需要被着色的像素大小，那么渲染就可以完全在局部完成（右图）。否则，就需要使用次表面散射渲染技术
- 传统光照模型和标准PBR模型都不考虑散射距离大于一个像素的情况，而是只考虑表面反射和散射距离足够小的次表面散射。并且会对这两部分进行分别求解。在实时渲染中，我们会把表面反射当做高光反射，次表面散射光当做漫反射来求解。



- 图解：光照模型描述的两现象。高光反射部分用于描述反射，漫反射部分用于描述次表面散射

## 2. 传统光照模型（Blinn-Phong模型）

- 在基于物理渲染(PBR)之前，主要受硬件计算水平的影响，人们不能像PBR那样使用真实世界的物理法则进行光照渲染。转而根据自己的经验使用一些开销不那么大的近似计算进行渲染。

- 在1973年，裴祥风(Bui Tuong Phong)提出标准光照模型背后的基本理念。后来Blinn简化了其中关于高光的计算，改进后的光照明模型被称为Blinn-Phong光照明模型。
- 其理念为:把进入到摄像机的光线分为3个部分，每个部分使用一个方法来计算它的贡献度。这3个部分是：
  - 环境光(ambient)
  - 漫反射(diffuse)
  - 高光反射(specular)
- 因此，传统光照模型可表达为:  $C_{result} = C_{ambient} + C_{diffuse} + C_{specular}$

## 2.1 属性定义

- 指向光源的方向，用 $\mathbf{l}$ 表示。用 $\hat{\mathbf{l}}$ 表示指向光源的单位向量
- 光的强度和颜色，统一用 $\mathbf{c}_{light}$ 表示，三个分量分别为RGB
- 物体表面法线，用 $\mathbf{n}$ 表示。用 $\hat{\mathbf{n}}$ 表示单位法线
- 物体表面的漫反射颜色，用 $\mathbf{m}_{diffuse}$ 表示
- 物体表面的高光反射颜色，用 $\mathbf{m}_{specular}$ 表示
- 由物体表面指向摄像机（视点）方向，用 $\mathbf{v}$ 表示。用 $\hat{\mathbf{v}}$ 表示单位向量

## 2.2. 逐顶点和逐像素

- 逐顶点光照（per-pixel lighting）
  - 在每个顶点上计算光照，然后在片元内对光照信息进行插值，速度很快，但是精度不高
  - 这种技术被称为**高洛德着色**（Gouraud Shading）
- 逐像素光照
  - 在片元内对每个像素上计算光照
  - 这种技术被称为**Phong着色**（Phong Shading）

## 2.3. 环境光

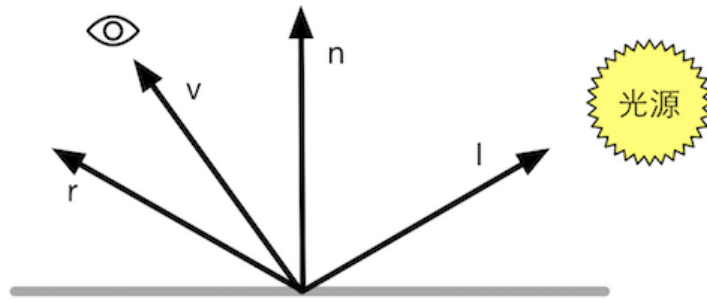
- $C_{ambient} = g_{ambient}$
- $g_{ambient}$ ：环境光常量 (在Lighting面板中可以找到)

## 2.4. 漫反射

- **兰伯特余弦定理**：发光强度的空间分布符合余弦定律的发光体（不论是自发光或是反射光），其不同角度的辐射强度会依余弦公式变化，角度越大强度越弱。
- 传统光照模型不考虑微表面的细节，而认为反射是**各向同性**的。因此在传统模型中，漫反射光照是用于对那些被物体表面随机散射到各个方向的辐射度进行建模的，视角的位置不重要，因为反射是完全随机的。因此可以认为在任何反射方向上的分布都是一样的。但是入射光线的角度很重要。
- **兰伯特光照模型**：根据这些假设，兰伯特余弦定理可以用来描述漫反射，即反射强度与表面法线和光源方向之间夹角的余弦值成正比，反射颜色与入射颜色和物体表面颜色的乘积相关。

$$C_{diffuse} = (\mathbf{c}_{light} * \mathbf{m}_{diffuse}) * \max(0, \hat{\mathbf{n}} \cdot \hat{\mathbf{l}})$$

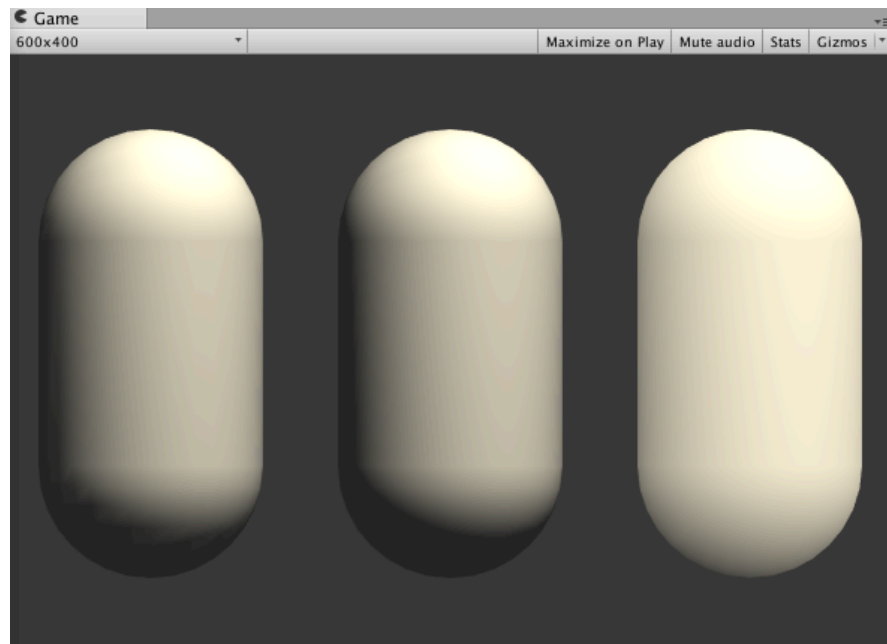
- $\hat{n}$ 是表面法线的单位向量， $\hat{l}$ 是指向光源的单位向量。为了防止法线和光源方向的点积为负值，使用 $\max$ 函数来保证结果始终大于等于0。这可以防止物体被从后面来的光源照亮。



- 半兰伯特（Half Lambert）光照模型：上述模型的问题在于，在光照无法到达的区域，模型的外观是全黑的，就算添加了环境光，背光面的明暗仍然是一致的，缺乏立体感。为了改善此问题，出现了半兰伯特光照模型：

$$C_{diffuse} = (C_{light} * m_{diffuse}) * (\alpha * (\hat{n} \cdot \hat{l}) + \beta)$$

- 可以看出，与原兰伯特模型相比，半兰伯特光照模型没有使用max操作来防止点积为负值，而是对其结果进行了一个 $\alpha$ 倍的缩放和一个 $\beta$ 大小的平移，通常情况下 $\alpha$ 和 $\beta$ 的值均为0.5。
- 半兰伯特光照模型没有任何物理依据，仅仅是视觉经验模型。



- 图解：逐顶点漫反射光照、逐像素漫反射光照、半兰伯特光照的对比效果

## 2.5. 高光反射

- 传统模型中的高光反射是一种经验模型，也就是说，它并不完全符合真实世界中的高光反射现象
- Phong模型中，高光反射与表面法线 $\hat{n}$ ，视角方向 $\hat{v}$ ，光源方向 $\hat{l}$ ，反射方向 $\hat{r}$ 有关。其中反射方向 $\hat{r}$ 可以通过 $\hat{n}$ 和 $\hat{l}$ 计算得到：

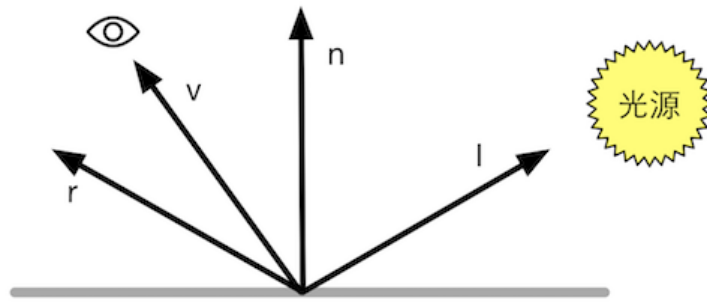
$$\hat{r} = 2(\hat{n} \cdot \hat{l}) * \hat{n} - \hat{l}$$

- 高光部分的公式如下：

$$C_{specular} = (C_{light} * m_{specular}) * \max(0, \hat{v} \cdot \hat{r})^{m_{gloss}}$$

- $m_{gloss}$  表示光泽度，用于控制高光区域的亮点有多大，这个值越大，亮点面积越小。

- 

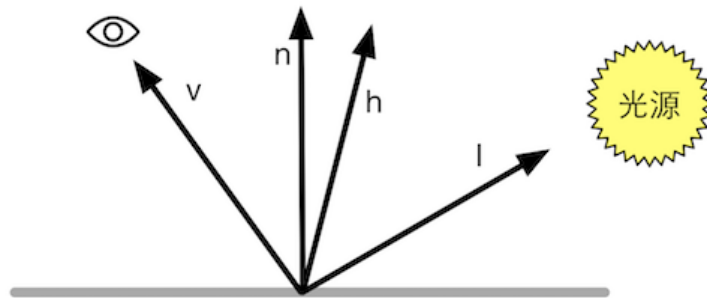


- **Blinn-Phong模型**：在Phong模型的基础上修改高光项，从而避免计算反射方向 $\hat{r}$

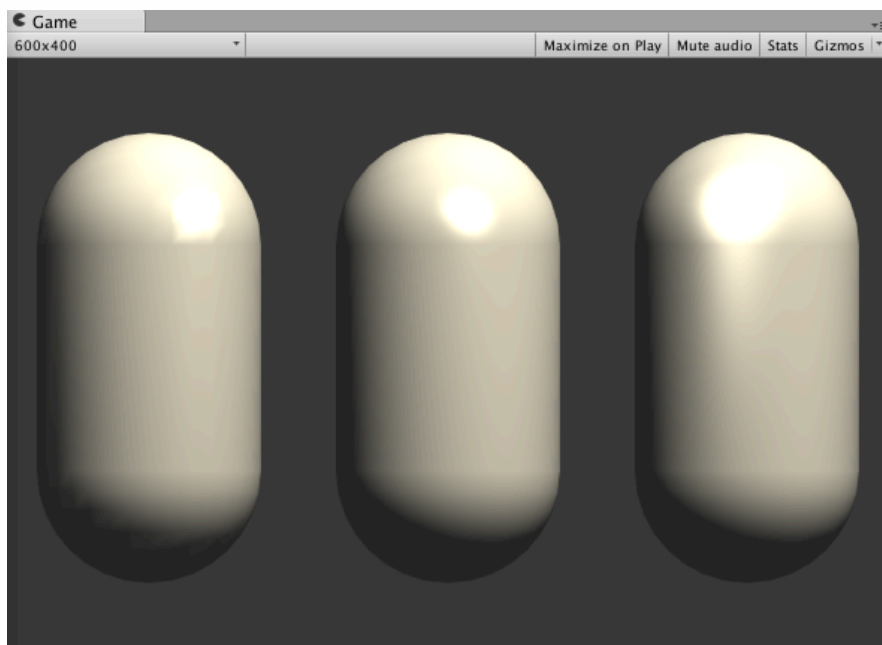
$$C_{specular} = (C_{light} \cdot m_{specular}) \cdot \max(0, \hat{n} \cdot \hat{h})^{m_{gloss}}$$

- 其中 $\hat{h}$ 的计算为

$$\hat{h} = \frac{\hat{v} + \hat{l}}{|\hat{v} + \hat{l}|}$$



- 传统的固定管线着色（fixed-function shading pipeline）采用的默认光照模型就是Blinn-Phong模型。之所以不采用Phong模型，根据WikiPedia的解释，当摄像机和光源距离模型足够远的时候，Blinn-Phong模型的效率要高于Phong模型。这是因为此时 $\hat{h}$ 将是一个常量，对于每个光源每帧只需要计算一次。



- 图解：逐顶点的高光反射光照、逐像素的高光反射光照（Phong光照模型）和Blinn-Phong高光反射光照的对比结果

## 2.5 总结

- 现在你知道Maya软件里的材质名称Lambert、Phong和Blinn-Phong的名字是怎么来的了吧
- 标准光照模型简单易用，适合初学者入门理解
- 标准光照模型仅仅是一个经验模型，不完全符合物理
- 一些重要的物理现象是无法用标准光照模型表现出来的，例如菲涅尔反射(Fresnel reflection)。其次，标准光照模型是各向同性(isotropic)的，也就是说，当我们固定视角和光源方向并旋转物体时，反射不会发生任何改变。但有些表面是具有各向异性(anisotropic)的，例如粗糙表面，毛发等。这些效果需要用基于物理的模型才可以实现，其公式也更复杂

## 2.6. 名词回顾

- **兰伯特光照模型**：描述漫反射项时，符合兰伯特余弦定理的光照模型
- **半兰伯特光照模型**：对兰伯特光照模型中的漫反射进行视觉改善的模型
- **Phong模型**：由Bui Tuong Phong总结的，使用兰伯特光照描述漫反射项，使用经验模型描述高光项的一种光照模型
- **Blinn-Phong模型**：修改了Phong模型中高光项计算公式的另一种经验模型
- **高洛德着色**：逐顶点计算光照，在片元中插值光照信息
- **Phong着色**：逐像素计算光照

## 3. 基于物理的光照模型

- 基于物理的光照模型（Physically based rendering，PBR）是为了对光和物体之间的行为进行更加真实的建模，PBR其实早已被应用到电影特效及动画等离线渲染中，但实时渲染中的PBR是近年才逐渐流行起来的（Unity 5中才正式引入PBR）。
- 需要明确的是，PBR渲染并不意味着渲染出来的画面一定像照片一样真实。PBR的计算量通常巨大，在实时渲染中受限于硬件计算能力，往往需要对公式进行简化，从而选择一些近似算法或者经验算法来替代某些项的计算。
- 在Unity中，PBR模型不是唯一的，而是根据运行平台的不同而编译出不同的代码，来获得性能

与效果之间的平衡。

### 3.1. 属性定义

- 入射辐射率：单位面积、单位方向上某光源的入射通量，称为辐射率(radiance)。通常用 $L_i$ 来表示，被认为是对单一光线的亮度和颜色的评估
- 出射辐射率：单位面积、单位方向上某光源的出射通量，通常用 $L_o$ 表示
- 辐照度：在垂直于光源方向 $\mathbf{l}$ 的某表面的单位面积上的入射通量，称为辐照度(irradiance)，通常用 $E_i$ 表示。
- 辐射度：某表面的单位面积上的出射通量，通常用 $E_o$ 表示。 $E_o \leq E_i$
- 物体表面的漫反射颜色，用 $m_{diffuse}$ 表示。在PBR中， $m_{diffuse}$ 是有物理意义的，它表示物体的“Albedo”色，即光线照在物体上，一部分波长被吸收后，反射出来的颜色值。金属没有漫反射，因为所有的次表面散射光都会被自由电子立刻转化为其他能量
- 物体表面的高光反射颜色，用 $m_{specular}$ 表示。非金属没有特定的高光反射颜色，通常是一个定值（白色）。金属的高光反射非常强烈，并且是有高光颜色的。 $m_{specular}$ 有时会用其他物理变量推算出来，例如在本文描述的PBR模型中，使用菲涅尔反射系数 $F_0$ 来代替高光颜色。
- 物体表面的粗糙程度，用 $m_{roughness}$ 表示，通常在 $[0, 1]$ 范围

### 3.2. BRDF定义

- 在基于物理渲染中，我们通常会基于表面的入射光线的入射辐射率 $L_i$ 来计算出射辐射率 $L_o$ ，然后再根据 $L_o$ 决定表面的颜色。
- 而要得到 $L_o$ ，我们需要知道表面的某个点是如何与光进行交互的，即：当入射通量为单位1的一束光线沿着入射方向 $\mathbf{l}$ 到达表面某点时，有多少出射通量被反射到了观察方向 $\mathbf{v}$ 上面。因此，使用 $f(\mathbf{l}, \mathbf{v})$ 来表示这个交互过程，这个函数就被称为**BRDF函数**（Bidirectional Reflectance Distribution Function，双向反射分布函数）。
- BRDF函数必须满足两个特性：
  - 交换律： $f(\mathbf{l}, \mathbf{v}) = f(\mathbf{v}, \mathbf{l})$
  - 能量守恒： $\forall \mathbf{l}, \int_{\Omega} f(\mathbf{l}, \mathbf{v})(\mathbf{n} \cdot \mathbf{l}) d\omega_0 \leq 1$

### 3.3. BRDF反射等式

- 通用的反射公式：

$$L_o(\mathbf{v}) = \int_{\Omega} f(\mathbf{l}, \mathbf{v}) * L_i(\mathbf{l}) * (\mathbf{n} \cdot \mathbf{l}) d\omega_i$$

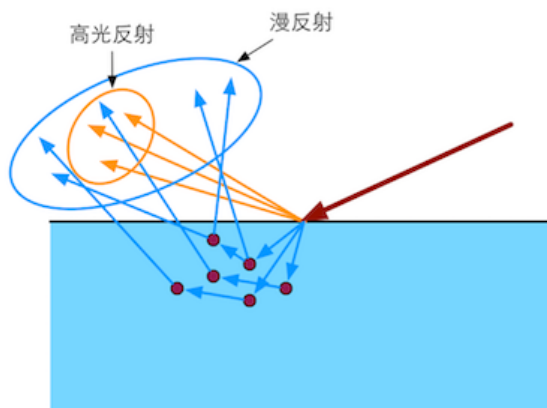
- 这个公式的含义是给定观察视角 $\mathbf{v}$ ，该方向上的出射辐射率 $L_o(\mathbf{v})$ 等于所有入射方向上的辐射率积分乘以它的BRDF值 $f(\mathbf{l}, \mathbf{v})$ ，再乘以强度因子 $\mathbf{n} \cdot \mathbf{l}$ 。
- 在游戏实时渲染中，求积分往往要花费很大的开销，如果我们能够将光源简化成为精确的，那么将大大简化上面的反射等式。对于一个给定的精确光源，我们使用以下等式来计算它在某个观察方向 $\mathbf{v}$ 上的出射辐射率：

$$L_o(\mathbf{v}) = \pi f(\mathbf{l}_c, \mathbf{v}) * c_{light} * (\mathbf{n} \cdot \mathbf{l}_c)$$

- 如果场景中包含了多个精确光源，则把它们分别代入上面的公式中计算，再把结果相加即可。

- 在BRDF中，高光和漫反射是分开计算的，这两部分是两个独立的项目，表面反射部分被称为高光反射项，次表面散射部分被称为漫反射项。

$$f(l, v) = f_{diff}(l, v) + f_{spec}(l, v)$$



- 图解：BRDF描述的两现象。高光反射部分用于描述反射，漫反射部分用于描述次表面散射
- BRDF函数  $f(l, v)$  不是唯一的。只要符合上述反射等式的光照模型，都可以称为PBR（Physically based rendering），或者被称为PBS（Physically based shading）。

### 3.4. 漫反射项

- Lambert模型就是最简单的BRDF漫反射项。其表示为

$$f_{diff}(l, v) = \frac{m_{diffuse}}{\pi}$$

- 可以看出，上述公式实际上是个定值，与光线方向和视点方向没有任何关系。其中原来Lambert中的余弦部分  $n \cdot l$  实际上是反射等式的一部分。这就是为什么Lambert模型是各向同性的。之所以要除以  $\pi$ ，是因为BRDF要在半球内积分值为1。将上述公式代入到反射等式里，可以得到：

$$L_0 = m_{diffuse} * c_{light} * (n \cdot l)$$

- Unity中的PBR模型（PC端）选择了Disney在2012年Siggraph的文章中使用的BRDF模型的漫反射项，表示为：

$$f_{diff}(l, v) = \frac{m_{diffuse}}{\pi} * (1 + (F_{D90} - 1)(1 - n \cdot l)^5) * (1 + (F_{D90} - 1) * (1 - n \cdot v)^5)$$

- 其中， $F_{D90} = 0.5 + 2 * m_{roughness} * (h \cdot l)^2$ 。
- 论文：Burley, Brent, and Walt Disney Animation Studios. "Physically-based shading at disney." ACM SIGGRAPH. Vol. 2012. 2012.

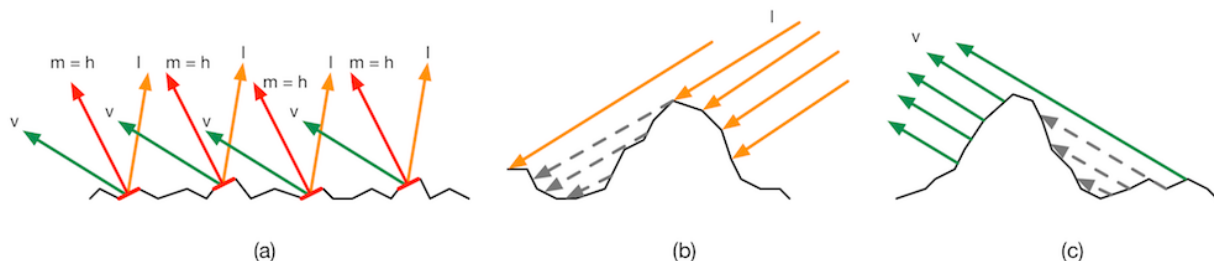
### 3.5. 高光反射项

#### 3.5.1. 微面元理论

- BRDF中的高光反射项大多数是建立在微面元理论（microfacet theory）的假设上的。微面元理论认为，物体表面是由许多人眼看不到的微面元组成的，理论中假设每个微面元都是足够平滑的，具有单一方向的高光反射。当光线和物体表面一点相交时，实际上是和一系列微面元交互的结果。当然，微面元理论也仅仅是真实世界的散射的一种近似理论，仍然有一些材质是无法使用微面元理论来描述的。



- 假设表面法线为 $\mathbf{n}$ ，微面元的法线 $\mathbf{m}$ 并不都等于 $\mathbf{n}$ 。因此，不同的微面元会把同一入射方向的光线反射到不同方向上。当我们计算BRDF时，入射方向 $\mathbf{l}$ 和视点方向 $\mathbf{v}$ 都会给定，这意味着只有一部分微面元反射的光线才会进入到我们的眼睛中，这部分微面元恰好会把光线反射到方向 $\mathbf{v}$ 上，即它们的法线 $\mathbf{m}$ 刚好把 $\mathbf{l}$ 和 $\mathbf{v}$ 分割成相等的两部分，用 $\mathbf{h}$ 表示，如下图(a)所示。然而这些微面元也并不会全部反射到眼睛中，因为其中一部分会在入射方向 $\mathbf{l}$ 上被其他微面元挡住（shadowing，阴影），如下图(b)所示，或者是在它们的反射方向 $\mathbf{v}$ 上被其他微面元挡住（masking，遮蔽），如下图(c)所示。



### 3.5.2. Torrance-Sparrow模型

- 基于微面元理论假设，高光部分采用了著名的Torrance-Sparrow模型，即：

$$f_{spec}(\mathbf{l}, \mathbf{v}) = \frac{F(\mathbf{l}, \mathbf{h}) * G(\mathbf{l}, \mathbf{v}, \mathbf{h}) * D(\mathbf{h})}{4 * (\mathbf{n} \cdot \mathbf{l}) * (\mathbf{n} \cdot \mathbf{v})}$$

- 论文：Torrance, Kenneth E., and Ephraim M. Sparrow. "Theory for off-specular reflection from roughened surfaces." *Josa* 57.9 (1967): 1105-1114.
- 该公式的理解如下：
  - $D(\mathbf{h})$ 是微面元的法线分布函数（normal distribution function, NDF），用于计算有多少比例的微面元满足 $\mathbf{m} = \mathbf{h}$ ，只有这部分微面元才会把光线从 $\mathbf{l}$ 方向反射到 $\mathbf{v}$ 上。
  - $G(\mathbf{l}, \mathbf{v}, \mathbf{h})$ 是阴影-遮掩函数（shadowing-masking function），它用于计算那些满足 $\mathbf{m} = \mathbf{h}$ 的微面元中有多少会由于遮挡而不会被人眼看到，因此它给出了活跃的微面元所占的浓度，只有活跃的微面元才会成功地把光线反射到观察方向上。
  - $F(\mathbf{l}, \mathbf{h})$ 则是这些活跃微面元的菲涅尔反射（fresnel reflectance）函数，它可以告诉我们每个活跃的微面元会把多少入射光线反射到观察方向上，即表示了反射光线占入射光线的比率。
  - 最后，分母 $4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})$ 是用于校正从微面元的局部空间到整体宏观表面数量差异的校正因子。

### 3.5.3. 法线分布函数 $D(\mathbf{h})$

- 传统模型里的Blinn-Phong模型的高光项实际上就是一种法线分布函数，可以表示为：

$$D_{blinn}(\mathbf{h}) = (\mathbf{n} \cdot \mathbf{h})^{gloss}$$

- Unity中的PBR模型（PC端）选择了GGX模型的NDF，表示为：

$$D_{GGX}(\mathbf{h}) = \frac{a^2}{\pi * ((a^2 - 1) * (\mathbf{n} \cdot \mathbf{h})^2 + 1)^2}$$

- 其中  $a = m_{roughness}^2$
- 论文：Walter, Bruce, et al. "Microfacet models for refraction through rough surfaces." *Proceedings of the 18th Eurographics conference on Rendering Techniques*. Eurographics Association, 2007.

#### 3.5.4. 阴影-遮掩函数 $G(l, v, h)$

- Unity中的PBR模型中有多个阴影遮蔽函数，例如Smith-Schlick、Kelemen、SmithJointGGX等。具体选择哪个模型是由计算复杂度决定的，不同的平台会使用不同的模型。例如Smith-Schlick模型：

$$G(l, v, h) = \frac{0.5}{((n \cdot l) * (1 - k) + k) * ((n \cdot v) * (1 - k) + k)}$$

- 其中  $k = \frac{m_{roughness}^2}{2}$
- 论文：Heitz, Eric. "Understanding the masking-shadowing function in microfacet-based BRDFs." *Journal of Computer Graphics Techniques* 3.2 (2014): 32-91.

#### 3.5.5. 菲涅尔反射公式 $F(l, h)$

- Unity中的PBR模型选择了图形学中经常使用的Schlick菲涅尔近似公式：

$$F(l, h) = F_0 + (1 - F_0)(1 - l \cdot h)^5$$

- 其中  $F_0$  表示菲涅尔反射系数，是由折射率算出来的，通常来说，非金属的  $F_0$  是一个定值，通常在0.02-0.05左右，因此Unity的Standard Shader为非金属设置了一个固定的  $F_0$  为0.04。金属的  $F_0$  通常在0.5-1.0之间，需要从外部输入得到。
- 论文：Schlick, Christophe. "An Inexpensive BRDF Model for Physically-based Rendering." *Computer graphics forum*. Vol. 13. No. 3. Blackwell Science Ltd, 1994.

#### 3.5.6. 扩展：菲涅尔反射现象

- 有一类材质，例如瓷砖表面有釉的材质，或者木头表面清漆的材质，透明的水晶及液体等，当光达到这些材质表面时，一部分光被反射，一部分发生折射，即视线垂直于表面时，反射较弱，而当视线非垂直表面时，夹角越小，反射越明显。所有物体都有菲涅尔反射，只是明显程度不同。
- 光滑的非金属的菲涅尔反射现象很明显。
- 而金属的菲涅尔反射则非常不明显，无论视线与平面夹角如何，反射强度几乎一样。



- 注意，图中的反射效果并不能从3.3小节中的反射公式中计算得出，因为反射的光源是环境光，而不是精确光源。在Unity中，除了使用3.3小节中的反射公式来计算精确光源的反射以外，还会使用GI（全局光照技术）来近似模拟环境光照，例如图中的反射效果就是用一张环境贴图中取得的高光颜色乘以 $FresnelLerp(F_0, m_{roughness}, \mathbf{n} \cdot \mathbf{v})$ 来模拟得到的。