

18.1.2 双向反射分布函数 (BRDF)

在了解了上面的理论基础后,我们现在来学习如何用数学表达式来表示上面的光照模型。这意味着,我们要对光这个看似抽象的概念进行量化。

我们可以用辐射率 (radiance) 来量化光。辐射率是单位面积、单位方向上光源的辐射通量,通常用 L 来表示,被认为是单一光线的亮度和颜色评估。在渲染中,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 ,这个过程也往往被称为是着色 (shading) 过程。

而要得到出射辐射率 L_o ,我们需要知道物体表面一点是如何和光进行交互的。而这个过程就可以使用 BRDF (Bidirectional Reflectance Distribution Function, 中文名称为双向反射分布函数) 来定量分析。大多数情况下, BRDF 可以用 $f(\mathbf{I}, \mathbf{v})$ 来表示,其中 \mathbf{I} 为入射方向和 \mathbf{v} 为观察方向 (双向的含义)。这种情况下,绕着表面法线旋转入射方向或观察方向并不会影响 BRDF 的结果,这种 BRDF 被称为是各项同性 (isotropic) 的 BRDF。与之对应的则是各向异性 (anisotropic) 的 BRDF。

那么, BRDF 到底表示的含义是什么呢? BRDF 有两种理解方式——第一种理解是,当给定入射角度后, BRDF 可以给出所有出射方向上的反射和散射光线的相对分布情况;第二种理解是,当给定观察方向 (即出射方向) 后, BRDF 可以给出从所有入射方向到该出射方向的光线分布。

344

18.1 PBS 的理论和数学基础

一个更直观的理解是,当一束光线沿着入射方向 \mathbf{I} 到达表面某点时, $f(\mathbf{I}, \mathbf{v})$ 表示了有多少部分的能量被反射到了观察方向 \mathbf{v} 上。

据此,我们给出基于物理渲染的技术中,第一个重要的等式——反射等式 (reflection equation):

$$L_o(\mathbf{v}) = \int_{\Omega} f(\mathbf{I}, \mathbf{v}) \times L_i(\mathbf{I})(\mathbf{n} \cdot \mathbf{I}) d\omega$$

BDRF

而能量守恒则要求表面反射的能量不能超过入射的光能,即

$$\forall \mathbf{I}, \int_{\Omega} f(\mathbf{I}, \mathbf{v})(\mathbf{n} \cdot \mathbf{I}) d\omega \leq 1$$

基于这些理论, BRDF 可以用于描述两种不同的物理现象:表面反射和次表面散射。针对每种现象, BRDF 通常会包含一个单独的部分来描述它们——用于描述表面反射的部分被称为是**高光反射项 (specular term)**,以及用于描述次表面散射的**漫反射项 (diffuse term)**,如图 18.5 所示。

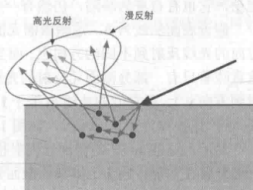


图 18.5 BRDF 描述的两种现象。高光反射部分用于描述反射,漫反射部分用于描述次表面散射

```
152 BDRF
153 {
154   漫反射 : 折射部分
155   高光反射
156   {
157     F : 菲涅尔方程 ==> 描述的是在不同的表面角下表面所反射的光线所占的比率
158     G : 几何函数 ==> 描述了微平面自成阴影的属性。当一个平面相对比较粗糙的时候,
159       平面表面上的微平面有可能挡住其他的微平面从而减少表面所反射的光线。
160     D : 正态分布 ==> 估算在受到表面粗糙度的影响下,取向方向与中间向量一致的微平面的数量
161       微平面的取向方向与中间向量的方向越是一致,镜面反射的效果就越是强烈越是锐利
162   }
163 }
164
165
166
```

满足 (a) 的微面元会在反射方向 \mathbf{v} 上被其他微面元挡住,因此,这部分反射光也不会被看到

这就是著名的 Torrance-Sparrow 微面元模型 [5]。上面的式子看起来难以理解,实际上其中的各个项对应了我们之前讲到的不同现象。 $D(h)$ 是微面元的法线分布函数 (normal distribution function, NDF),它用于计算有多少比例微面元的法线满足 $\mathbf{m}=\mathbf{h}$ 。只有这部分微面元才会把光线从 \mathbf{I} 方向反射到 \mathbf{v} 上。 $G(\mathbf{I}, \mathbf{v}, \mathbf{h})$ 是阴影-遮挡函数 (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 (active microfacets) 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。 $F(\mathbf{I}, \mathbf{h})$ 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。事实上,现实生活中几乎所有的物体都会表现出菲涅尔现象,读者可以在一篇很有意思的文章 Everything has Fresnel 中看到一些这样的例子。最后,分母 $4(\mathbf{n} \cdot \mathbf{I})(\mathbf{n} \cdot \mathbf{v})$ 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。

18.1.5 Unity 中的 PBS 实现

在之前的内容中，我们提到了 Unity 5 的 PBS 实际上是受 Disney 的 BRDF[2]的启发。这种 BRDF 最大的好处之一就是很直观，只需要提供一个万能的 shader 就可以让美工人员通过调整少量参数来渲染绝大部分常见的材质。我们可以在 Unity 内置的 `UnityStandardBRDF.cginc` 文件中找到它的实现。

总体来说，Unity 5 一共实现了两种 PBS 模型。一种是基于 GGX 模型的，另一种则是基于归一化的 Blinn-Phong 模型的。这两种模型使用了不同的公式来计算高光反射项中的法线分布函数 $D(h)$ 和阴影—遮掩函数 $G(l,v,h)$ 。在默认情况下，Unity 5 使用基于归一化后的 Blinn-Phong 模型来实现基于物理的渲染（尽管很多引擎选择使用 GGX 模型）。

如前面所讲，Unity 使用的 BRDF 中的漫反射项使用的公式如下：

347

18.2 Unity 5 的 Standard Shader

当我们在 Unity 5 中新创建一个模型或是新创建一个材质时，其默认使用的着色器都是一个名为 Standard 的着色器。这个 Standard Shader 就使用了我们之前所讲的基于物理的渲染。

Unity 支持两种流行的基于物理的工作流程：金属工作流（Metallic workflow）和高光反射工作流（Specular workflow）。其中，金属工作流是默认的工作流程，对应的 Shader 为 Standard Shader。而如果想要使用高光反射工作流，就需要在材质的 Shader 下拉框中选择 Standard (Specular setup)。需要注意的是，通常来讲，使用不同的工作流可以实现相同的效果，只是它们使用的参数不同而已。金属工作流也不意味着它只能模拟金属类型的材质，金属工作流的名字来源于它定义了材质表面的金属值（是金属类型的还是非金属类型的）。高光反射工作流的名字来源于它可以指定表面的高光反射颜色（有很强的高光反射还是很弱的高光反射）等，而在金属工作流中这个颜色需要由漫反射颜色和金属值衍生而来。在实际的游戏制作过程中，我们可以选择自己更偏好的工作流来制作场景，这更多的是个人喜好的问题。当然也可以同时混用两种工作流。

替相应的结构体）

1. 金属材质

- 几乎没有漫反射，因为所有被吸收的光都会被自由电子立刻转化为其他形式的能量；
- 有非常强烈的高光反射；
- 高光反射通常是有颜色的，例如金子的反光颜色为黄色。

2. 非金属材质

- 大多数角度高光反射的强度比较弱，但在掠射角时高光反射强度反而会增强，即菲涅耳现象；
- 高光反射的颜色比较单一；
- 漫反射的颜色多种多样。

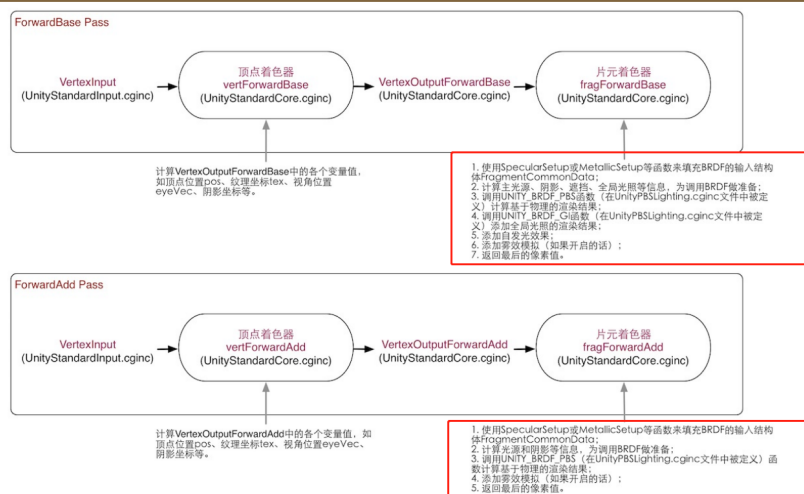
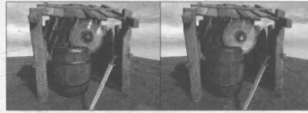


图18.7 Standard Shader中前向渲染路径使用的Pass（简化版本的PBS使用了VertexOutputBaseSimple等结构体来代替相应的结构体）

18.3.4 线性空间

在使用基于物理的渲染方法渲染整个场景时，我们应该使用线性空间（Linear Space）来得到最好的渲染效果。默认情况下，Unity 会使用伽马空间（Gamma Space），如果要使用线性空间的话，我们需要在 Edit→Project Settings→Player→Other Settings→Color Space 中选择 Linear 选项。图 18.17 显示了分别在线性空间和伽马空间下场景的渲染结果。



▲图 18.17 左边：在线性空间下的渲染结果。
右边：在伽马空间下的渲染结果

从图 18.17 中可以看出，使用线性空间可以

356

它的支持并不好。这种情况下，我们往往只能退而求其次，选择伽马空间进行渲染和计算。

那么，线性空间、伽马空间到底是什么意思？为什么线性空间可以得到更加真实的效果呢？这就需要介绍伽马校正（Gamma Correction）的相关内容了。实际上，当我们在默认的伽马空间下进行渲染计算时，由于使用了非线性的输入数据，导致很多计算都是在非线性空间下进行的，这意味着我们得到的结果并不符合真实的物理期望。除此之外，由于输出时没有考虑显示器的显示伽马的影响，会导致渲染出来的画面整体偏暗，总是和真实世界不像。

尽管在 Unity 中我们可以通过之前所说的步骤直接选择在线性空间进行渲染，Unity 会在背后为我们照顾好一切，但了解伽马校正的原理对我们理解渲染计算有很大帮助，读者可以在 18.4.2 节找到更多的解释。

全局光照 = 直接光照 + 间接光照

18.4.1 什么是全局光照

在上面的内容中，我们可以发现全局光照对得到真实的渲染结果有着举足轻重的作用。全局光照，指的就是模拟光线是如何在场景中传播的，它不仅会考虑那些直接光照的结果，还会计算光线被不同的物体表面反射而产生的间接光照。在使用基于物理的着色技术时，当渲染表面上一点时，我们需要计算该点的半球范围内所有会反射到观察方向的入射光线的光照结果，这些入射光线中就包含了直接光照和间接光照。

通常来讲，这些间接光照的计算是非常耗时间的，通常不会用在实时渲染中。一个传统的方法是使用光线追踪，来追踪场景中每一条重要的光线的传播路径。使用光线追踪能得到非常出色的画面效果，因此，被大量应用在电影制作中。但是，这种方法往往需要大量时间才能得到一帧，并不能满足实时的要求。

还是有性能瓶颈

Unity 采用了 Enlighten 解决方案来让全局光照能够在各种平台上有不错的性能表现。事实上，Enlighten 也已经被集成在虚幻引擎（Unreal Engine）中，它已经在很多 3A 大作中展现了自身强大的渲染能力。总体来讲，Unity 使用了实时+预计算的方法来模拟场景中的光照。其中，实时光照用于计算那些直接光源对场景的影响，当物体移动时，光照也会随之发生变化。但正如我们之前所说，实时光照无法模拟光线被多次反射的效果。为了得到更加真实的渲染效果，Unity 又引入了预计算光照的方法，使得全局光照甚至在一些高端的移动设备上也可以达到实时的要求。

Unity 全新的全局光照解决方案可以大大提高一些基于 PC/游戏机等平台的大型游戏的画面质量，但如果要在移动平台上使用仍需要非常小心它的性能。一些低端手机是不适合使用这种比较复杂的基于物理的渲染，不过，Unity 会在后续的版本中持续更新和优化。而且随着手机硬件的发展，未来在移动平台上大量使用 PBS 也已经不再是遥不可及的梦想了。更多关于 Unity 中全局光照的内容，读者可以在 Unity 官方手册的全局光照（<http://docs.unity3d.com/Manual/GIIntro.html>）一文中找到更多内容，本章最后的扩展阅读部分也会给出更多的学习资料。