|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

PBR已成为一个常见的行业术语，但其含义在游戏引擎之间存在很大差异。 对我们来说，核心PBR原则之一是材质和照明信息的分离，这是确保场景中所有对象之间视觉一致性的关键。通过这种方法，相同的光照应用于所有物体及其材质层，没有诸如负光之类的任何不正确行为，或诸如光贡献的“重复计算”之类的artifacts。从生产的角度来看，这有助于以透明的方式在不同环境中重用assets和照明设备。同时减少了向艺术家公开的参数数量，并使创作更直观。但是，正如我们将在本文后面看到的那样，这种分离仅从创作的角度来看是正确的，因为出于性能原因，照明和材质紧密耦合在代码中。

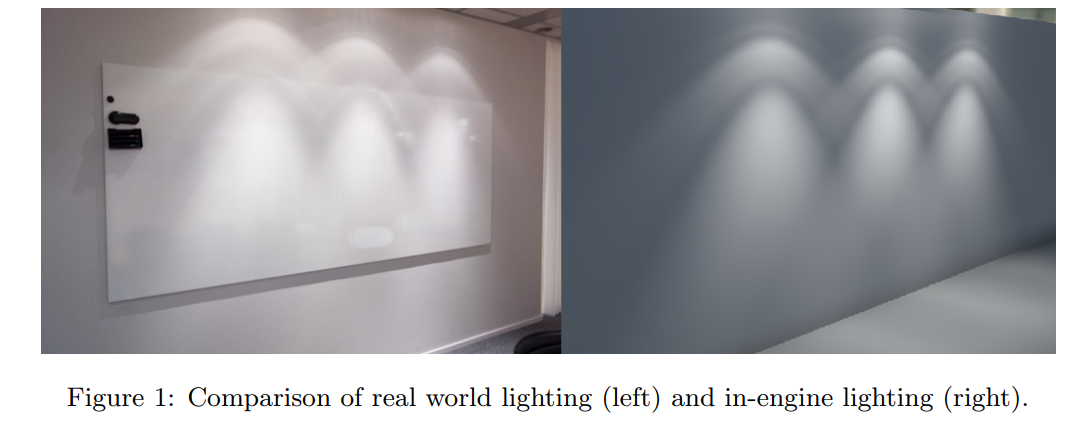
在整合PBR时，需要明白整个图形管道（渲染器和工具）需要升级。考虑到这一点，我们这些课程笔记的目标是涵盖大规模生产引擎所需的所有不同升级，包括文献中通常省略的许多小细节。首先，第2节更详细地解释了地面实况参考在PBR背景下的重要性。接下来 第3节介绍材料，并回顾光与物质的相互作用。在此之后，第4节描述了如何定义和发射光。第5节重点介绍相机和输出图像，包括如何将亮度转换为最终像素值。最后，我们以第6节结束，回顾我们如何安排过渡到PBR以及我们在此期间考虑的内容。

2 参考

2.1 验证模型和假设

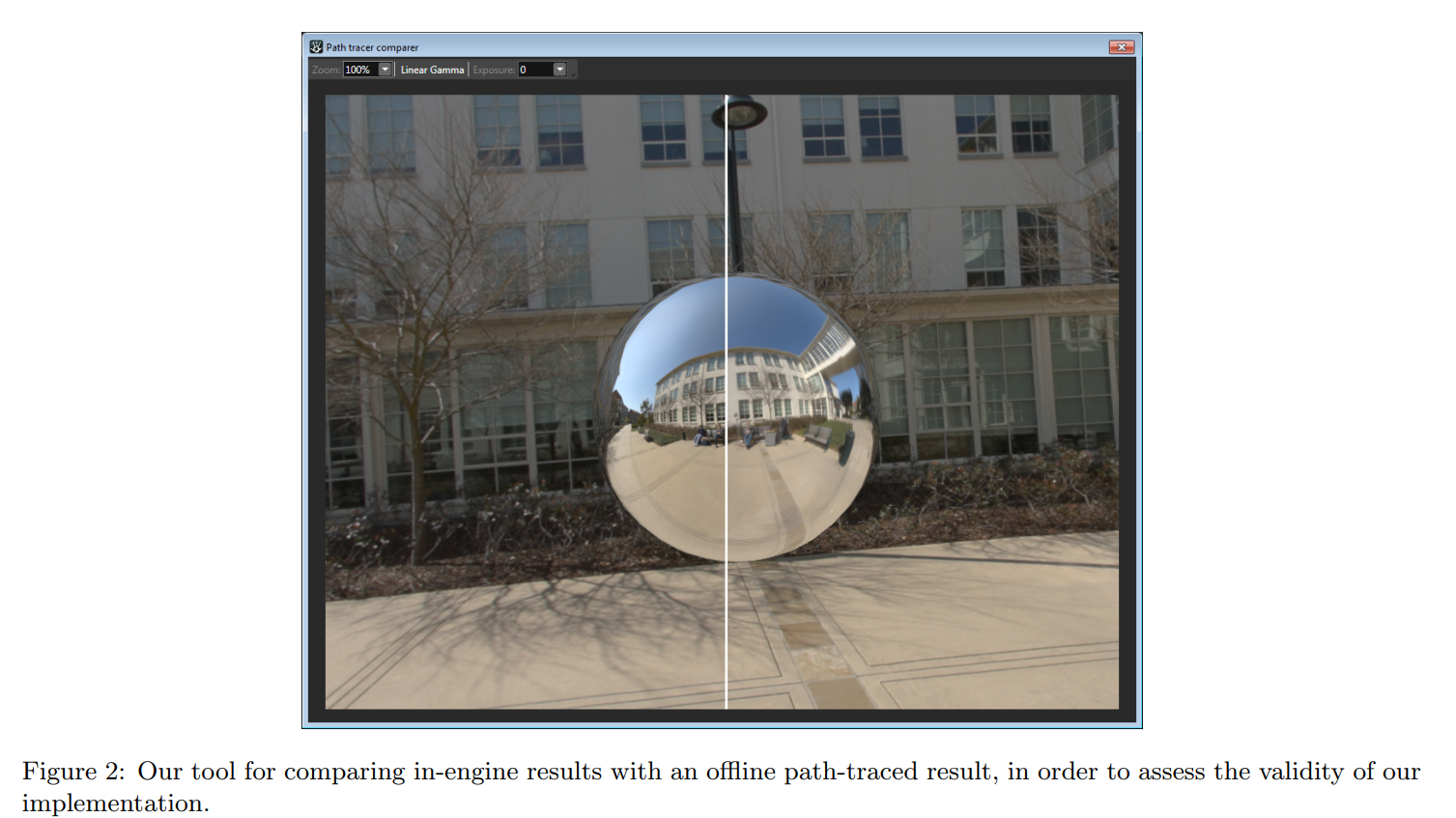
几十年来，视频游戏行业一直试图获得更真实的图像。但是，照片并没有说明用于创建此类图像的数据和方法。与基于物理的渲染不同，渲染试图模拟真实世界的行为和属性，其结果是定量判断。这对Ground-truth数据提出了额外要求。仔细选择正确的模型和正确的假设，即良好的参考，这一点很重要。通过观察和与现实世界比较是做出正确选择和判断技术或方法的相关性的最佳方法。在粗略的水平观察现实世界，我们可以快速掌握高光的形状，湿表面的行为，光强度的差异，以及许多其他视觉特征，参见图1.当采用真实世界的参考材料，重要的是以多个尺度拍摄照片以捕捉每个尺度上呈现的不同照明行为。

然而，准确地测量实际数据通常非常复杂或太耗时。某些数据库（如MERL [MER]）提供对此类数据的访问，这使我们能够快速评估模型。在我们的方法中，我们尝试测量和验证实际数据，例如光强度和衰减，天空亮度和相机效果。但是所有这些步骤都非常耗时且并不总是易于设置。



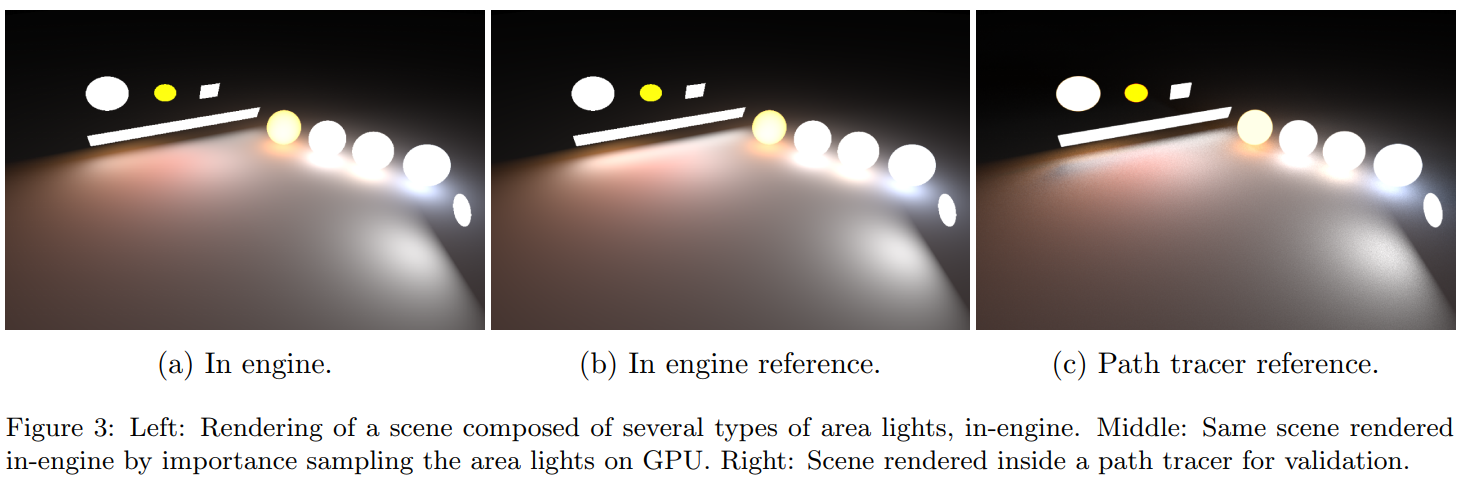
2.2 验证引擎内部近似

现代PBR路径追踪器如Mitsuba [Jak10]（由Wenzel Jakob提供）实现了最先进的渲染技术，可以创建令人难以置信的逼真图像。使用此类软件是评估模型准确性的更简单方法。在Frostbite中，我们为Mitsuba编写了一个简单的导出器，它允许我们快速评估近似值的有效性。导出器能够导出几何和恒定材质信息（即没有纹理）和所有光源。通过这种设置，可以轻松检查材质模型，光积分和光强度。此外，该输出器允许我们验证更复杂现象的准确性，如全局照明，环境遮挡和反射。图2显示了导出后自动触发的widget，允许用户快速滑动并比较引擎结果和离线参考之间的像素值，并完全控制曝光。最后一点很重要，因为渲染器输出的强度范围很广。为了保留此范围，两个渲染器都将其最终图像导出为线性HDR格式OpenEXR [Opea]。



2.3 验证引擎参考模式

如上所述，导出器很有用，但它需要几秒到几分钟的时间来导出和渲染场景。 为了快速迭代各种近似值并选择正确的近似值，我们在GPU上通过蛮力采样（基于图像的灯光和区域灯光）为我们的照明集成添加了引擎内参考模式，如图3所示。渲染时间不快，但迭代时间比我们的简单导出器快一个数量级。 附录A包含用于评估参考模式中几种类型光的列表。



注意：使用正确的参考很重要。这是显而易见的，如果参考不好，则近似值也不会太好。如果近似毛发着色模型，请使用真实世界最近的模型作为参考。当近似公式时，一定要使用原始方程而不是已经近似的公式，例如Oren Nayar或Schlick对菲涅耳方程的近似可能导致错误。唯一可以完全信任的参考永远是真实世界。

3 材质

3.1 材质模型

3.1.1 外观

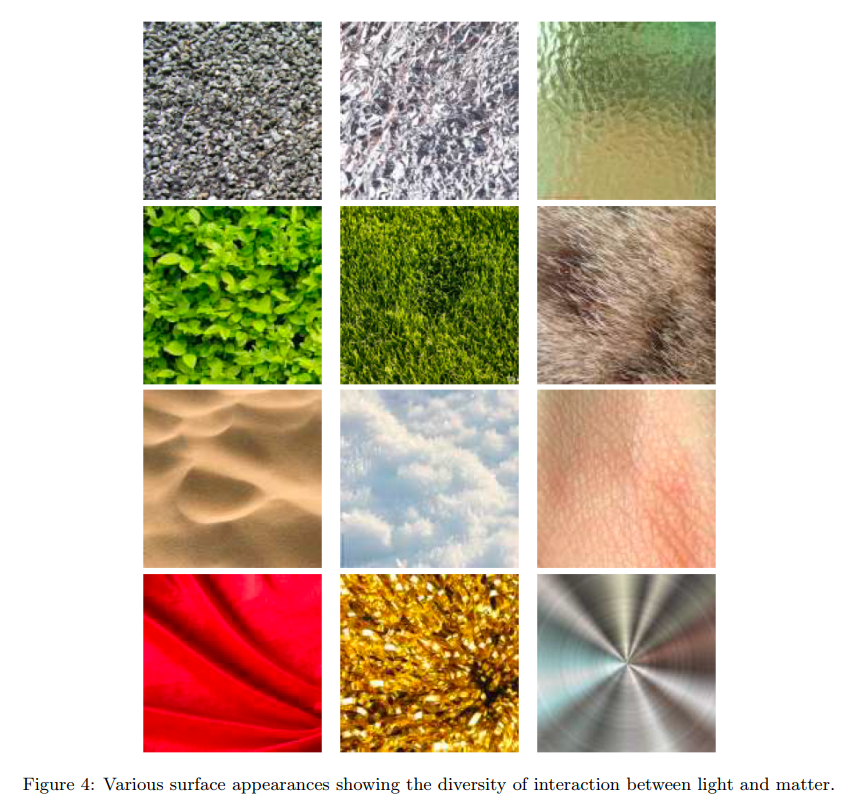
表面外观由入射光与表面材质属性之间的相互作用产生。在现实世界中可观察到的外观非常广泛，从简单的均匀材质到复杂的分层和异质材质，见图4。

这些不同的外观可以通过某些固有的物理性质来分类，例如导电性，平均自由程和吸收。 基于这些材料特性，文献已经揭示了能够在全光谱中表现出一定范围的外观的各种材料模型。 材料模型文献广泛，存在许多不同的模型，具有各种权衡和准确性。 被称为BSDF（双向散射分布函数）的材料模型可以分解为两个部分：反射部分（BRDF）和透射部分（BTDF）。 在本文中，我们将重点关注反射部分，特别是能够代表“标准”外观的材料模型，即我们日常生活中遇到的绝大多数表面。 因此，我们将自己局限于具有短平均自由路径的反射，各向同性，介电/导体表面。

3.1.2 材质模型

在这种标准材料模型的背景下，表面反射通常被分解为两个不同的术语：称为“漫反射”（）的低角频率信号和称为“镜面反色和”（）的由低到高角频率部分，参见图5。界面将两种介质分开：空气和物质。由平面组成的物体表面可以很容易地用菲涅耳定律[Wikd]表示，并且用于介质表面和导体表面。当界面不规则时，参见图6，文献显示基于微面模型[CT82]非常适合特征化这类表面与光的相互作用。微面体模型由等式1描述，关于推导的更多细节参见[Hei14]：

项目是微面分布模型(即NDF，法向分布函数)。项目是微面遮蔽(阴影遮蔽)模型。该公式对漫反射项和镜面反射项均有效。二者之间的差异存在于微面BRDF 。对于镜面反射项，是完美镜像因此使用菲涅尔定律模拟，这就引导出已知的下列公式：



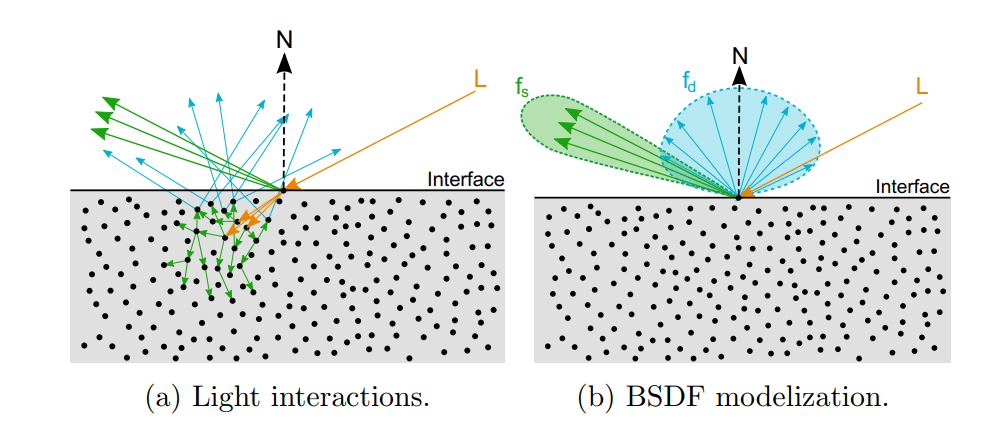
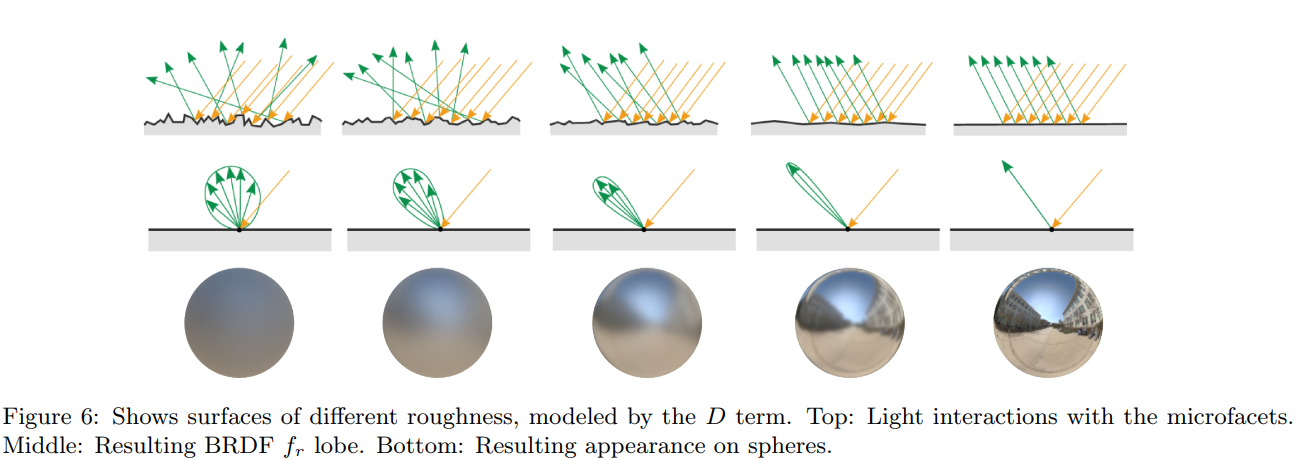


图5 光与“标准”物质——石板的交互。左图：光的交互。右图：漫反射项和镜面反射项的BSDF交互模型。

函数D在表面外观中起着重要作用，如图6所示。文献[Wal +; Bur12]最近指出，“长尾”NDF，如GGX分布，擅长捕捉真实世界的表面。G项对高粗糙度值也起着重要作用。 Heitz [Hei14]最近表明Smith能见度函数是正确且准确的G项。他还指出，文献通常倾向于使用Smith能见度函数的近似版本，而更精确的掩蔽阴影函数形式模拟由于微面高度引起的掩蔽和阴影之间的相关性，参见公式3 图7显示了简单Smith函数和高度相关Smith函数之间的差异。



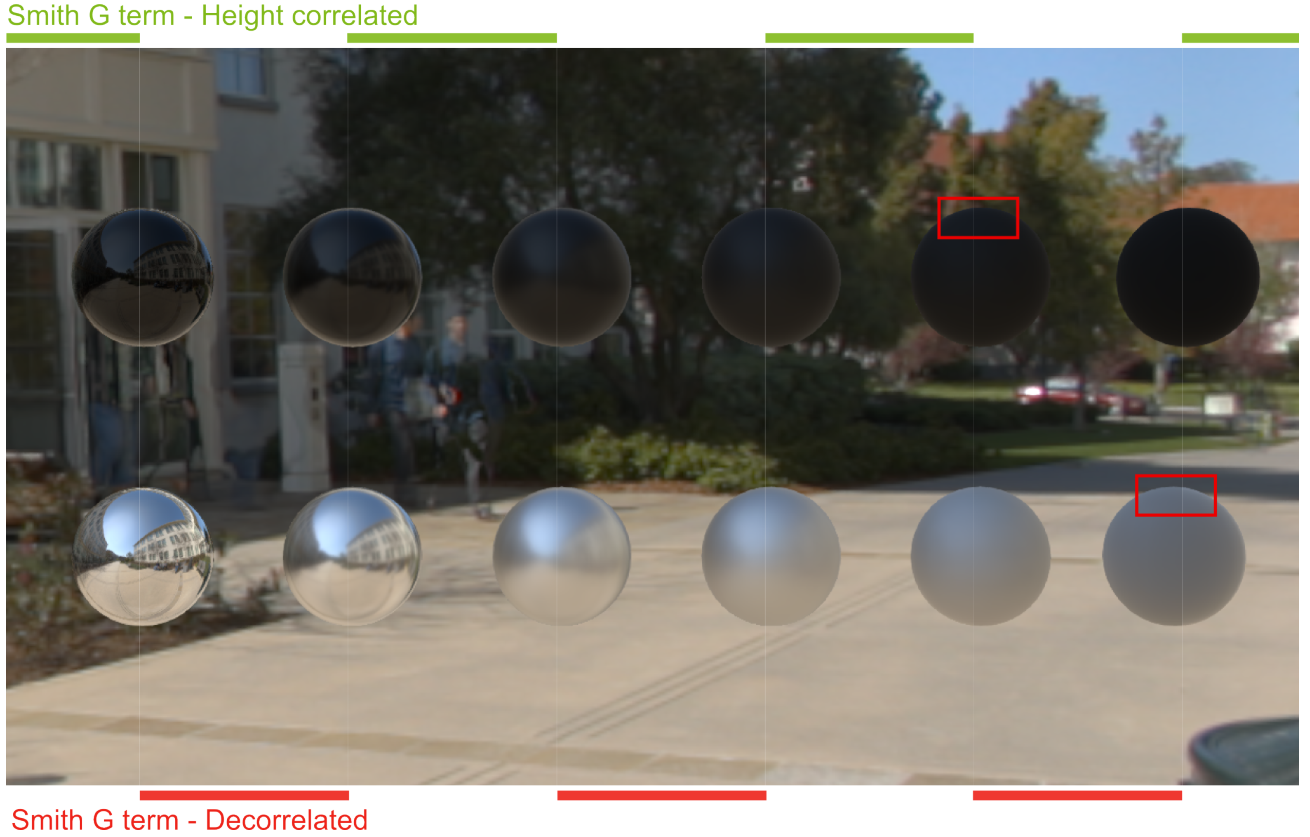
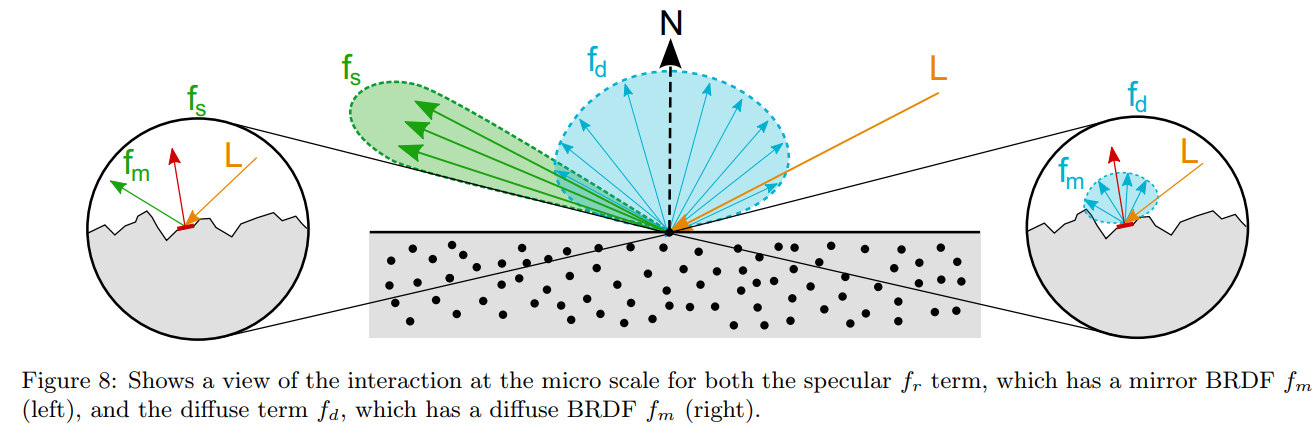


图7 在一组黑色电介质（顶部）和铬金属（底部）球体上的高度不相关和高度相关的Smith能见度函数之间的比较，从左到右粗糙度递增。请注意高度相关版本如何为较大粗糙度值带来更多的能量回馈。

对于漫反射项，如果是Lambertian模型，则公式1可简化为：

直到最近，漫反射项依然被假设是简单的Lambertian模型。 但是，除了分层材料外，漫射部分需要与镜面项连贯并且必须考虑表面的粗糙度[Bur12]（即镜面和漫反射术语应使用相同的粗糙度），如图8所示。公式4没有解析解。Oren等人[ON94]发现使用高斯NDF分布和V型微面G项作为该方程的经验近似，这就是经典的Oren-Nayar模型。为了正确支持我们的模型，我们应该使用GGX NDF对公式4进行等效近似，如Gotanda [Got14]中所述。 附录B详细介绍了我们对这种漫反射模型的一些分析，但需要进一步研究。

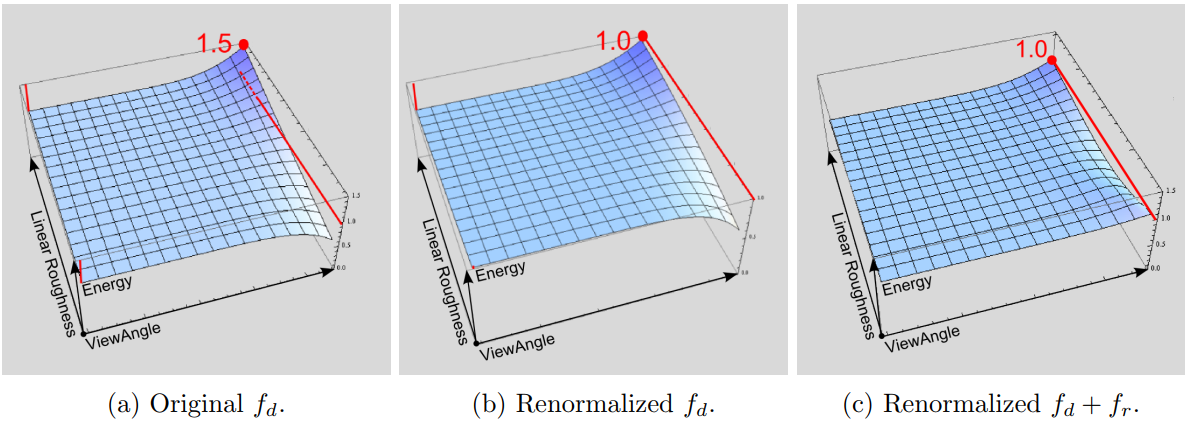


Burley [Bur12]提出了另一种基于现实世界Groud-Truth漫反射模型，见方程5.虽然这个模型是经验的，但它允许我们重现MERL数据库材料的主要特征。 出于这个原因，并且由于其简单性，我们选择在Frostbite中使用此模型。 该漫射项考虑了材料的粗糙度，并在掠射角处产生一些后向反射。

3.1.3 能量守恒

能量守恒非常重要，避免增加额外光能。 此外，它允许我们正确处理掠射角度的行为，对于这种情况，光线倾向于更多地被镜面项散射而不是满散射项。 在 Frostbite中，我们保持计算简单并且保证能量守恒，通过确保半球形定向反射率（由于半球上的恒定照明而在给定方向上给出总反射率）低于我们整体的一个 BRDF（漫反射+镜面术语）：

由于我们的镜面模型和漫反射模型之间没有直接关系，因此做出正确的推导并不容易（对于镜面和漫反射项均基于微平面模型的情况，请参阅附录C）。 迪士尼漫反射模型的一个重要警告是缺乏能量守恒。图9a显示了迪斯尼漫反射模型的半球方向反射率。 我们可以清楚地看到这种BRDF不是能量守恒的，因为得到的反射率值高于1.我们稍微修改它以补偿能量增益，同时保留其反光特性。 清单1显示了具有重归一化因子的迪士尼评估函数。 图9c示出了完整的半球方向反射率，由镜面微平面模型和迪斯尼漫射模型组成。虽然不完全等于1，但它足够接近。图10比较了迪士尼的原始漫反射术语及其重整化版本。



**图9**

float Fr\_DisneyDiffuse ( float NdotV , float NdotL , float LdotH , float linearRoughness )  
 {  
 float energyBias = lerp(0, 0.5, linearRoughness);

float energyFactor = lerp(1, 1.0 / 1.51, linearRoughness);

float fd90 = energyBias + 2 \* LdotH \* LdotH \* sd.linearRoughness;

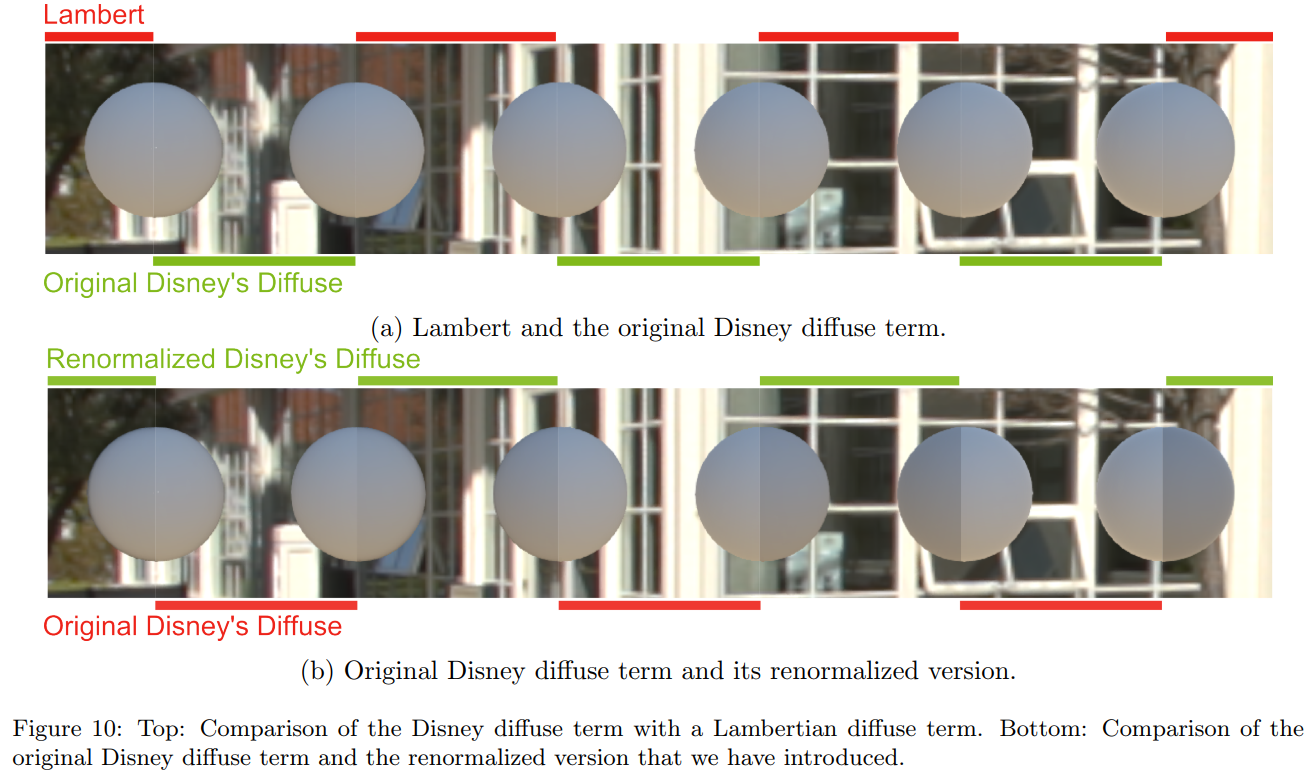
float fd0 = 1;

float lightScatter = fresnelSchlick(fd0, fd90, NdotL).r;

float viewScatter = fresnelSchlick(fd0, fd90, NdotV).r;

return (viewScatter \* lightScatter \* energyFactor \* M\_INV\_PI) \* sd.diffuse.rgb;

}  
Listing 1: Disney’s diffuse BRDF code with renormalization of its energy. linearRoughness is the perceptually linear roughness (See section 3.2.1)



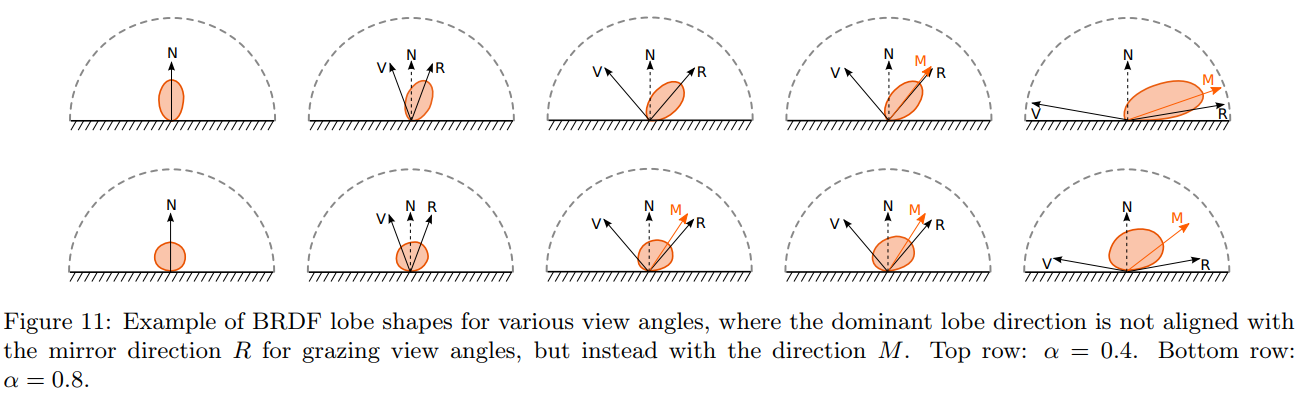
3.1.4 波瓣特征

基于微面的镜面BRDF具有某些特性，这些特性经常被绕过但对最终外观具有强烈影响。 特别是两个重要现象：

**半角参数化**：这种参数化意味着BRDF形状从法相入射角的各向同性朝着掠射角各向异性的转变是非线性。有关此部分的更多信息，请参见第4.9节。

**镜面偏离**：通常假设BRDF波瓣以反射方向为中心（也称镜像方向）。然后，由于和阴影-遮挡项G，当粗糙度增加时，BRDF波瓣会朝着法向量偏移，如图11。这种偏移被称为“非镜面峰值(Off-specular peak)”并且它在粗糙表面外观中扮演着重要角色。

非镜面峰值可导致高粗糙度值的巨大差异。 为了考虑这一重要特征，我们尝试对在区域光和基于图像的光评估中使用的“主导方向”进行建模，参见区域光4.7节和基于图像的照明部分4.9。



3.1.5 Frostbite标准模型

寒霜的标准材质模型与其它游戏引擎使用的模型非常接近。包括：

* **镜面项**: 镜面微面模型，Smith correlated visibility function, GGX NDF.
* **漫反射项**: 能量守恒的迪斯尼漫反射项

代码实现参数12页

3.2 材质系统

3.2.1 材质

Frostbite用于各种游戏，从体育到赛车，从第一人称射击游戏到开放世界游戏。为了满足这些游戏的不同要求，引擎需要提供有关照明和材料支持的灵活控制。此外，在转向PBR期间的一个限制是确保与我们的旧非PBR照明模型兼容，以便简化过渡。照明路径是可控的，支持：延迟渲染，前进或混合渲染。该路径将在4.11节中详细说明。

在Frostbite，一个“材质”被定义为：

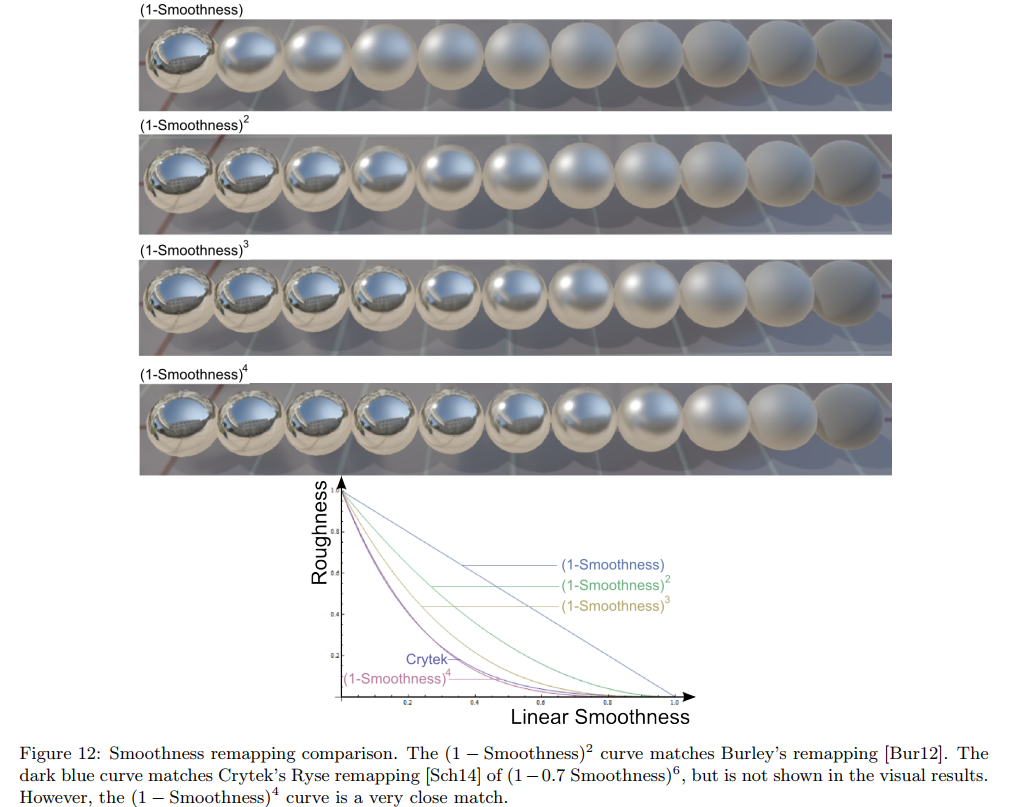
* 一条**光照路径**：延迟渲染、前向渲染或两者都有。
* 一组**输入参数**：漫反射，光滑度，厚度等等。
* 一个**材质模型**：粗糙表面，半透明，皮肤，头发，例如非PBR粗糙表面。这些是shader代码。
* 在延迟路径情况下，需要考虑**GBuffer布局**需要。

游戏团队可以从给定光路可用的材料中选择一组材质。每种材料都使用游戏的materialID属性进行标识。覆盖最常见情况的基础材料（我们称之为“标准”材料）始终存在并定义与其他材料共享的参数（例如粗糙度）。对于延迟着色，基本材料通常设置为“迪士尼”模型，参考Burley的模型[Bur12]。但是，我们还支持另外两种基材：“双色”材料和“旧”材料。

迪斯尼基础材质：

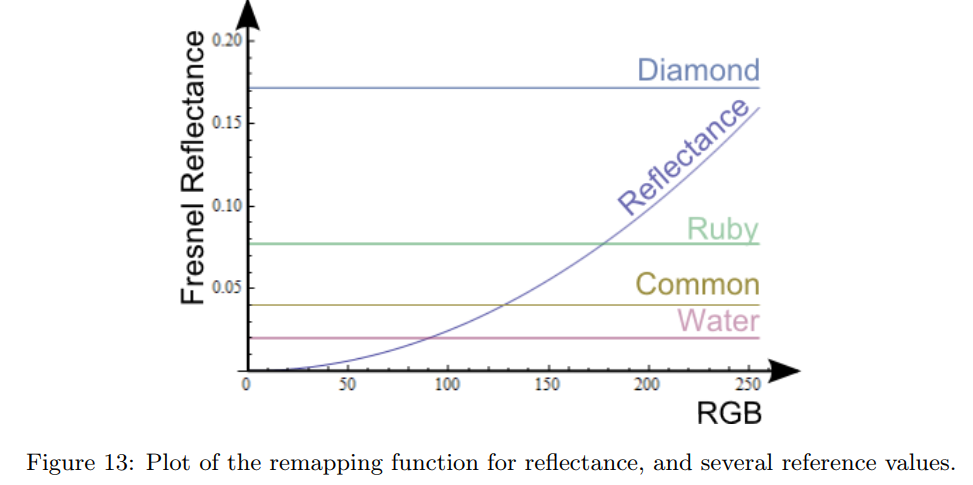
|  |  |
| --- | --- |
| Normal | 标准法向量 |
| BaseColor | 如Burley文献中建议那样，对非金属物体定义漫反射反照率，对金属物体定义法向入射角()的菲涅尔反射值。对于金属物体，该属性的低部分定义了微镜面遮挡。 |
| Smoothness | 定义对象的粗糙度。 我们选择使用平滑度而不是粗糙度，因为将白色映射到平滑值对于艺术家来说更直观，并且他们已经习惯使用Frostbite的非PBR材质模型。 与Burley的演示类似，平滑度被重新映射到感知线性平滑度（1-）。 |
| MetalMask | 定义表面（即电介质/导体）的“金属度”或电导率，如Burley的表述。 我们将它命名为金属蒙版，以向艺术家建议此变量的二元性质。 |
| Reflectance | 对非金属材质定义在法向入射角()的菲涅尔反射, 该属性的低部分定义了微镜面遮挡。 |

为了重新映射平滑度，我们分析了不同的重映射函数并测试。图12显示了不同重映射函数和结果的图。与Burley的演讲类似，我们选择了“平方”重映射，这对于我们的艺术家来说似乎是比较喜欢的。对于反射率，我们选择了以下重映射函数



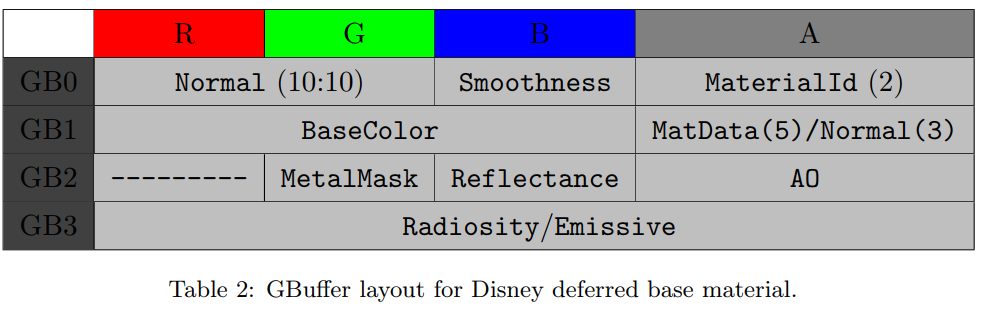
目标是将映射到可能包括宝石的高菲涅耳值的范围，其约束是将RGB 128重新映射到公共电介质4％反射率上。对于宝石，大约从红宝石的8％到钻石的17％。我们选择将函数限制为16％作为近似值。与常见值的比较如图13所示。实际上，根据我们的实时限制，的变化1％或2％几乎不可察觉。高于4％的快速增长值非常适合这一点。

对于非金属（反射）和金属（BaseColor）物体在法向入射角()的菲涅耳反射率，我们还使用<2％（水反射率）的下部来提供微镜面控制（更多细节请参见第4.10节））。 请注意，由于我们的特定编码，这意味着对于非金属物体我们要使用不同范围的微镜面遮挡值。



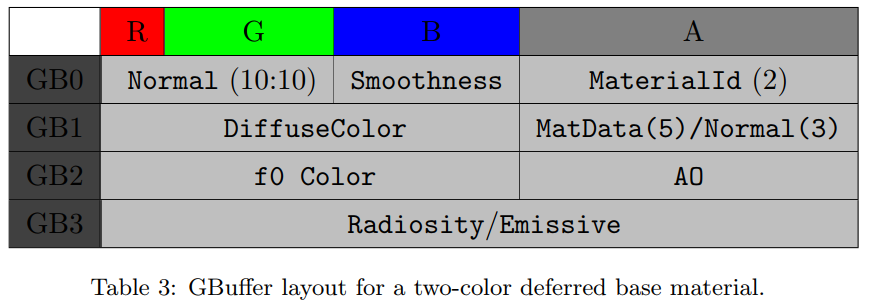
迪士尼材料使用3.1节中描述的“标准”材料模型。 相关的GBuffer布局如表2所示。此布局遵循以下约束：

* 所有基本属性（Normal，BaseColor，Smoothness，MetalMask，Reflectance）都需要可混合以支持延迟贴花。 像MaterialId这样的不可混合属性存储在alpha通道中。 我们还避免了可能影响混合质量的压缩和编码机制。
* MSAA支持可防止使用BaseColor属性[MP12]的色度子采样。
* 常用参数需要收集到同一个缓冲区中。
* MaterialId需要存储在所有材质的相同位置。
* 性能要求意味着我们的基础/标准材质仅使用四个Gbuffer纹理（不包括深度）。



MaterialId用于解释MaterialData（MatData）参数。例如，对于延迟皮肤材料，它存储扩散轮廓指数。对于各向异性材料，它存储各向异性强度。AO参数是一个始终存在的环境遮挡项（与材料类型无关），更多详细信息，请参见第4.10节。Radiosity是一种存储间接漫射照明的照明缓冲区，在GBuffer创建过程中进行评估。法向量分为两部分，使用仍然允许正常混合的有损编码方法（此算法可在未来的谈话中解释）。

双色基材：迪斯尼参数化（使用单个BaseColor，金属蒙版和标量反射值）在表达混合材料的镜面高光时不足，例如：在金属和电介质的界面处，如金属氧化物。出于这个原因，我们支持具有彩色和漫反射颜色的“双色”延迟基础材质。相关的GBuffer布局如表3所示。项支持其较低(精度)范围内的微遮挡，类似于Reflectance属性。根据上下文，需要在两个参数化之间进行转换。根据定义，从迪士尼参数化到这种“双色”参数化的转换是微不足道的，并且在我们的GBuffer解包期间已经发生用于照明计算。相反的转换（即从“两种颜色“参数化到迪士尼参数化）涉及更多，因为它需要非线性优化。这仅在资产需要从一种引擎模式转换为另一种引擎模式时执行。我们提供附录D中有关此转换的详情。



旧基础材质：是我们旧的非PBR引擎的基础材料，已在[Cof10]中详述。为了支持传统内容并简单地向PBR的过渡，我们根据基本的艺术规则在非PBR材料和PBR材料之间添加了自动转换。在存储输入参数之前，转换在shader中进行。与正确创作的资产相比，这种自动转换可以获得低质量的结果。 我们还没有找到一种自动转换这些资产同时保持质量的方法。

图14突出显示了材质参数的不同转换，并显示了材质模型和照明功能之间的依赖关系。 对于性能，区域灯（第4.7节）和基于图像的灯（第4.9节）依赖于预积分，这取决于材质模型。这意味着光照和材质模型在引擎内部耦合。PBR方法声称的光照和材质之间的分离仅对asset创建有效：艺术家无法访问shader中的任何照明信息。但在引擎盖下，这种分离并不成立，添加新材料通常意味着添加新的照明代码。

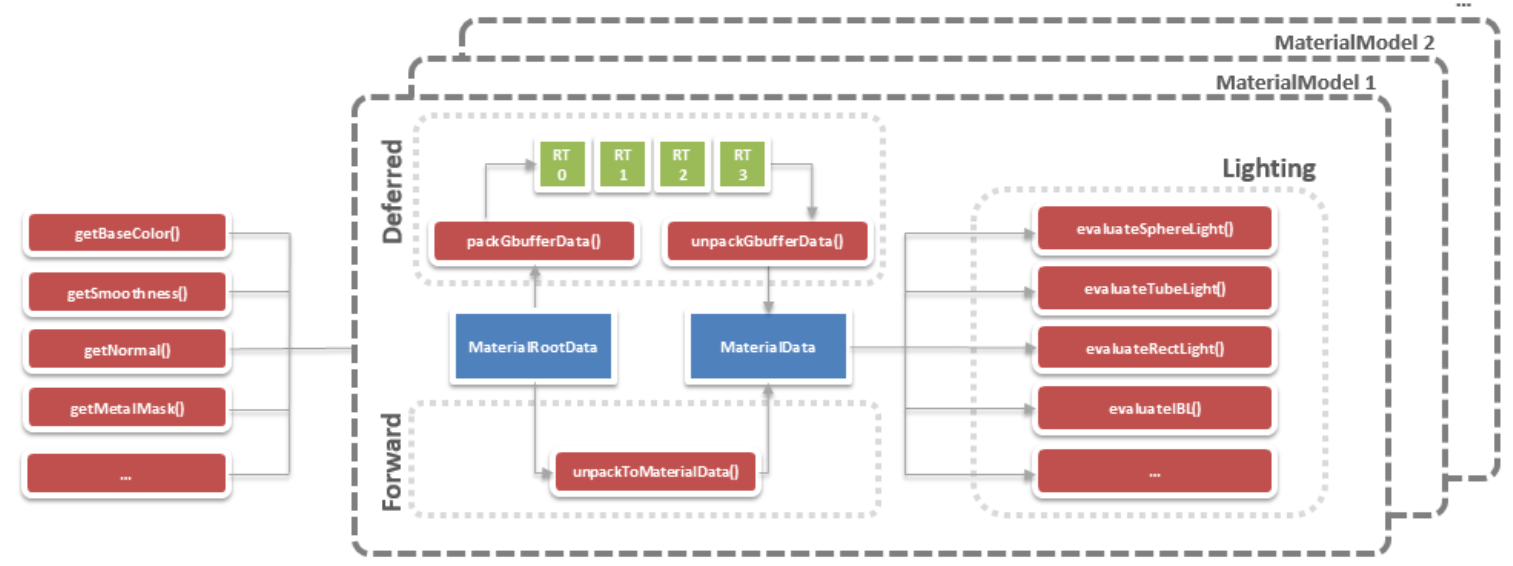


图14 寒霜材质示意图。输入参数列表（MaterialRootData）将转换为光照计算结构（MaterialData），用于计算材质模型的光照函数。光照功能包括所有灯光类型：精确光源，区域光源和IBL。如果材质使用延迟光照路径，它将被打包到GBuffer布局中，然后解压缩并转换。如果材质使用前向照明路径，则会直接转换。如果可以将不同的材料解压缩到相同的光照计算结构（MaterialData），则不同的材料可以共享相同的光照代码。

3.2.2 渲染Loop

上一节提供了材质的定义。本节将介绍材料的呈现方式。对于使用前向光照路径的表面：我们设置材质模型shader代码，传输参数列表，并渲染表面。对于使用延迟着色过程的表面，它更复杂。为了能够有效地管理一组材料，需要考虑一些因素：

材质模型尽可能地尝试使用相同的光照代码，依靠动态分支进行小调整。差异太大的光照代码需要使用模板缓冲区的不同光照pass。

材质尽可能于基础材质布局保持一致，并依靠动态分支进行小的调整以存储参数。材质差异太大将需要不同的GBuffer传递，再次使用模板缓冲区在光照过程中识别它们。

我们尝试通过执行“fix-up pass”来共享具有不同GBuffer布局的材质的照明pass。

对于大量几何形状传递，希望减少GBuffer创建成本，但仍然使用相同的光照pass。 一个典型的用例是植被，为此我们只填充基本GBuffer布局的两个缓冲区，然后使用清单3中所示的“修复”过程修补丢失的GBuffer参数。修复过程可能需要根据读取和写入同一缓冲区的硬件容量，在多次传递中完成。

1 void psMain (  
2 in float4 pos : SV\_Position ,  
3 in float2 texCoord : TEXCOORD0 ,  
4 out float4 outGBuffer2 : SV\_Target0 ,  
5 out float4 outGBuffer3 : SV\_Target1 )  
6 {  
7 float4 gbuffer0 = FETCH\_TEXTURE ( g\_gbufferTexture0 , ( texCoord .xy , 0);  
8 float4 gbuffer1 = FETCH\_TEXTURE ( g\_gbufferTexture1 , ( texCoord .xy , 0);  
9 float3 worldNormal = normalize ( unpackNormal ( gbuffer0 .rg , gbuffer1 .a));  
10  
11 // Read value for radiosity from a global diffuse probe  
12 float3 indirectLight = calcShL2Lighting ( worldNormal , ...) ;  
13  
14 // Fix smoothness , metalMask and reflectance with a global common value  
15 // ambient occlusion set to 1  
16 outGBuffer2 = float4 ( g\_smoothness , 0, g\_reflectance , 1);  
17 outGBuffer3 = packLightingRGBA ( indirectLight );  
18 }

Listing 3: Sample Gbuffer fixup

渲染流程如下：

// GBuffer creation

ForEach different stencil bit do

Render GBuffer pass for deferred material with n number of buffers

Do branching in shaders if needed

Render GBuffer pass for deferred material with (n + 1) number of buffers

Do branching in shaders if needed

// Fix-up pass

ForEach deferred material requiring sharing the deferred shading pass

Render GBuffer fix-up pass

// Decals

Render deferred decals affecting common parameters

// Deferred shading

ForEach individual deferred shading pass required do

Render deferred shading

Do branching in shaders if needed

// Forward rendering and lighting

ForEach ForwardMaterialModel do

Render forward lighting

贴花将在下一节中讨论。 实际上，由于材质参数和光照函数之间的耦合，游戏团队很难自定义所有内容。MaterialId提供了一种定制材料的简单方法，但是必须重用光照代码并依赖动态分支。例如，游戏团队已经添加了需要对照明进行两次采样的透明涂层模型。该系统允许我们从非PBR引擎执行转换，以满足用户的不同需求。

3.3 PBR和贴画

贴花可以看作是用于分层材质属性的动态系统，允许我们创建丰富的外观和变化。在此背景下，具有“正确”贴花的重要性随着PBR而增加。“正确”是指在任何照明发生之前正确组合贴花材料参数和表面的能力，即使在多个重叠贴花的情况下也是如此。 在Frostbite中，我们主要使用延迟着色来渲染贴花，但是在硬件和性能约束下正确处理它们几乎是不可能的。我们没有找到任何正确混合贴花的好解决方案，同时考虑到各种基于物理的材料，但为了完整起见，我们将列出一些主要缺陷和我们的选择。

* **正确性**：混合操作的正确性很重要，即混合操作后的属性恢复必须正确。例如，对于标准编码的Normal，如或只包含normal.x和normal.y，只有替换（即线性插值）才能正确恢复法线。 像添加细节[BH12]这样的有效操作会产生错误的结果。 这是可编程混合的一个很好的例子，但很少有平台支持它。 我们选择避免对我们的一些属性进行压缩，并在这种情况下依赖简单的线性插值alpha混合。
* **交互**：要正确混合，贴花和目标曲面应使用相同的材质模型（即相同的参数列表）。从铲平和性能的角度来看，这是不可管理的，因为它需要通过MaterialID创作贴花并为每个情况执行模板传递。我们选择将贴花参数限制为游戏所选材料中的常用参数集。通常是Normal，BaseColor，Smoothness。其他参数如MetalMask，Reflectance（或f0颜色）仅在选择迪士尼（或双色）基础材料时才考虑。但这还不够。影响除默认值之外的其他MaterialID的贴图将产生artifacts。MaterialData和MaterialID不是可混合数据，只能替换。一种选择是使用alpha通道的单独混合因子来强制中性值，如0。另一种选择是禁止贴花影响除MaterialID = 0之外的其他材料。
* **间接照明**：在任何照明计算之前，表面需要受到贴花的影响。但是在Frostbite中，间接漫反射在GBuffer创建过程中进行计算，防止除漫反射albedo之外的任何材料属性的进一步修改。一种解决方案是在延迟pass中计算所有间接漫射照明，例如使用覆盖水平[Val14]的光探测器体积纹理。 另一种解决方案是使用由UE4 [UE4]提出的预贴花pass，以在创建GBuffer之前将贴花属性存储在缓冲器中。这需要渲染两次接收贴花的对象。在Frostbite中，由于此类解决方案隐含的开销，我们选择保留artifacets。
* 自发光：由于Frostbite中的GBuffer存储有限，出于性能原因，有时会将自发光信息与radiosity buffer结合使用，请参见第4.8节。radiosity buffer稍后与反照率组合。在这种情况下，修改反照率的贴花将修改自发光颜色。在Frostbite中，我们接受自发光和贴花的artifacets处理。
* 高光形状保留和镜面反射：任何正常的参数修改都意味着在像素占用空间下方更改NDF。 这意味着它将偏向在贴花应用之前执行的任何形状保存（由Toskvig或LEAN映射等技术处理），参见第5.3节。 解决方案是在贴花应用后执行正常的过滤通过作为后处理[Sch14]。
* 前向渲染曲面：延迟贴花与透明和任何前向渲染对象不兼容。 在这种情况下，常见的解决方案是依靠前向贴花照明结果与表面照明作混合。支持前向渲染物体的贴花参数混合是可能的，但是在艺术方面需要很多约束（限制混合模式，限制纹理阵列的纹理大小等）。此问题与照明应用程序有一些相似之处。 应用已经发现的用于照明的解决方案可以打开平铺延迟和平铺前向贴花的方式。

4 光照

4.1 普通光照

光照管线必须遵从文献中已经广泛讨论过的重要基础。光照管线应支持高动态范围（HDR），必须在线性空间[Gd08]中完成。管道的所有输入和输出都应进行伽马校正（mip映射，混合，过滤等）。对于Frostbite，与所有游戏引擎一样，由于其硬件支持，我们选择依赖sRGB约定。

一个可信的场景是基于照明的连贯性和正确性：每个物体都应该从周围环境接收光线，以适当的强度反射光线并产生阴影。游戏引擎通常提供许多照明工具而没有在它们之间形成链接。艺术家难以操纵一个照明工具而无需重新调整其他照明组件以获得正确的结果，从而打破了可信度和空间参考。我们与Frostbite的主要指导是默认情况下一切正确，然后让艺术家有可能调整他们想要的东西。艺术家偏离正确的结果应该比获得正确的结果更难。但是，艺术控制不能忘记解决引擎限制和艺术原因。

Frostbite支持几种类型的光照模型：精确光源，光度灯，区域光源，自发光表面和基于图像的光照（IBL）。在术语“IBL”中，我们包括远光探测器（代表天空），以及局部光探测器和屏幕空间反射（SSR）。当所有照明组件耦合并与材质属性正确交互时，实现了相干性。这里有些例子：

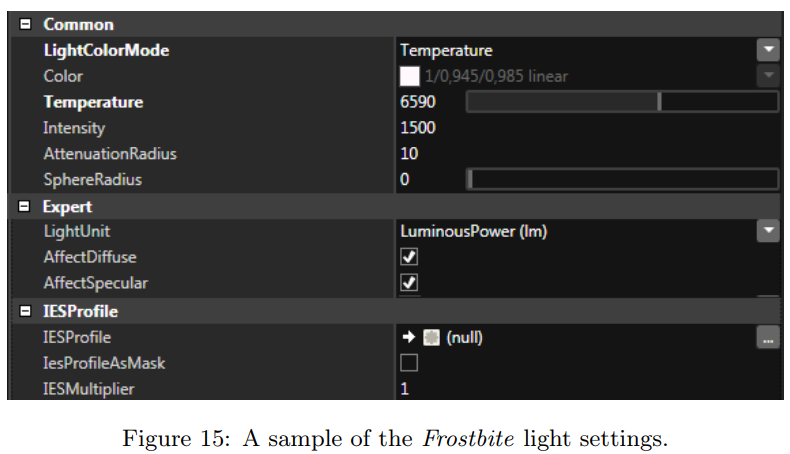
* **相干材质照明**：所有BSDF都应与所有光源类型正确交互。这包括直接照明，例如精确灯光和区域灯光，以及间接照明，例如IBL。我们的粗糙漫反射材质被所有光照模型和光能传递系统支持，以显示其特定的外观。
* **相干间接漫反射照明**：光能传递系统必须考虑所有光照模型。太阳和天空照明是非常重要的一部分，但也应包括其他光照类型。
* **相干间接镜面反射（即反射）**：SSR，局部光探测器和远光探测器必须正确组合在一起。
* **相干光单元**：所有光应以相同的单位表示，以实现真实的比率。
* **相干贴花**：贴花应该受到所有灯光类型的正确影响，包括间接照明。

4.2 解析光参数

为了方便艺术家，在Frostbite中，精确光源和区域光源共享相同的界面和设置。这些设置的一部分如图15所示。我们将光的色掉（或色相，称为颜色）与强度分开。我们使用强度这个词来指代灯光发出能量的大小，而不是它的严格定义。为了区分白色的各种色调，人造光源用色温（白炽灯和卤钨灯）或相关色温（CCT）（几乎所有其他）标记。色温是理想黑色辐射体发出相似色调的光的温度。CCT是黑体辐射体的色温，其对人类的颜色感知最接近地与灯的光相匹配。色温和CCT通常以开尔文（K）为单位测量。为简化本文档，我们使用术语色温来表示色温或CCT。我们也称前面的参数色温。表4显示了不同的光类型及其相关的温度和感知颜色。

色温转RGB颜色的算法：http://www.tannerhelland.com/4435/convert-temperature-rgb-algorithm-code

光谱渲染器（即使用光谱值而不是RGB三元组来表示颜色的渲染器）可以直接使用由黑体定义的辐射光谱。在Frostbite中，我们选择仅从色温中获取色调，让艺术家独立控制光强度。从色温中获取色调是一种相对复杂的操作。 艺术家也可以指定RGB颜色。



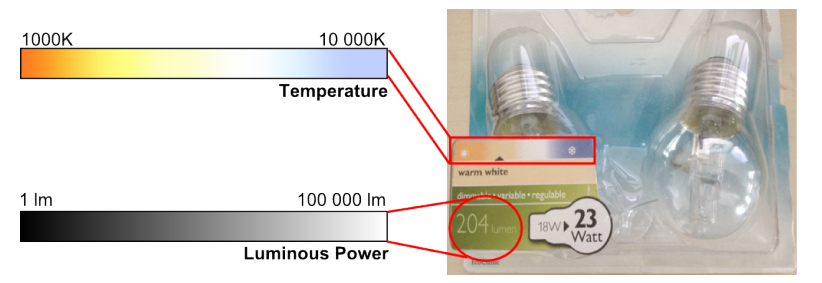


图16：白炽灯包装盒上的信息可直接用于寒霜引擎：色温和照明功率。

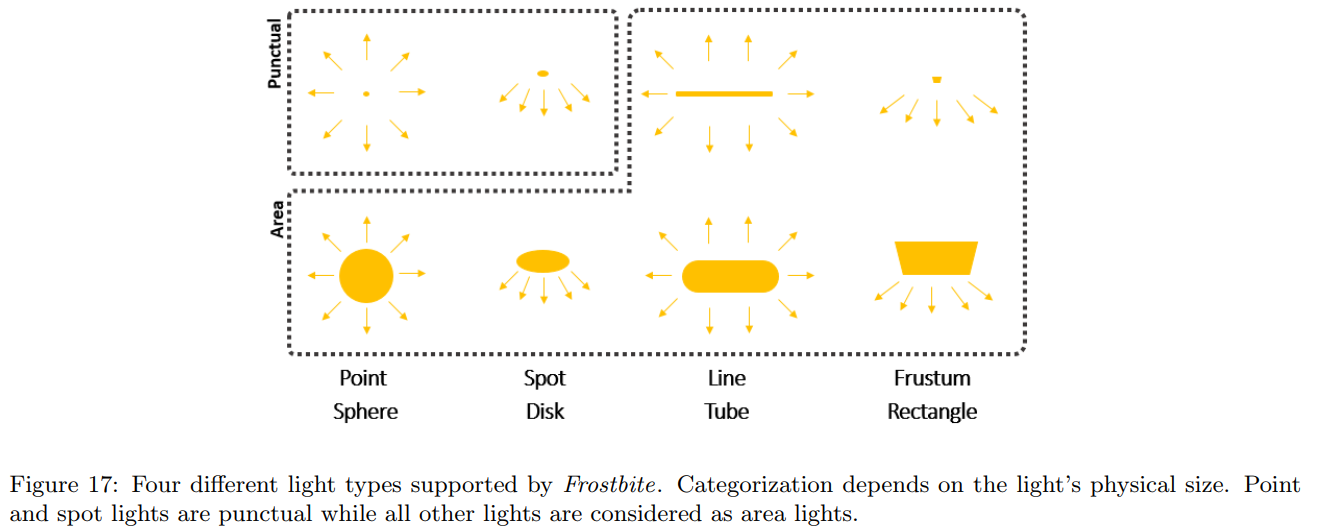
后面将针对每种光类型描述“光强度”参数。通过色温和强度，艺术家可以从制造商的网站上选择参考值并将其直接放入Frostbite。参见图16.灯光设置还包括通常的衰减范围和对物理光源尺寸的控制。光的物理尺寸（例如球半径，圆盘半径，管长等）允许艺术家定义光是区域光还是精确光。图17显示了Frostbite支持的灯光：点光源和聚光灯是唯二的精确灯光，所有其它灯光都被视为区域灯光。分离精确灯和区域灯对于性能至关重要，Frostbite支持这两种类型之间的平滑过渡。

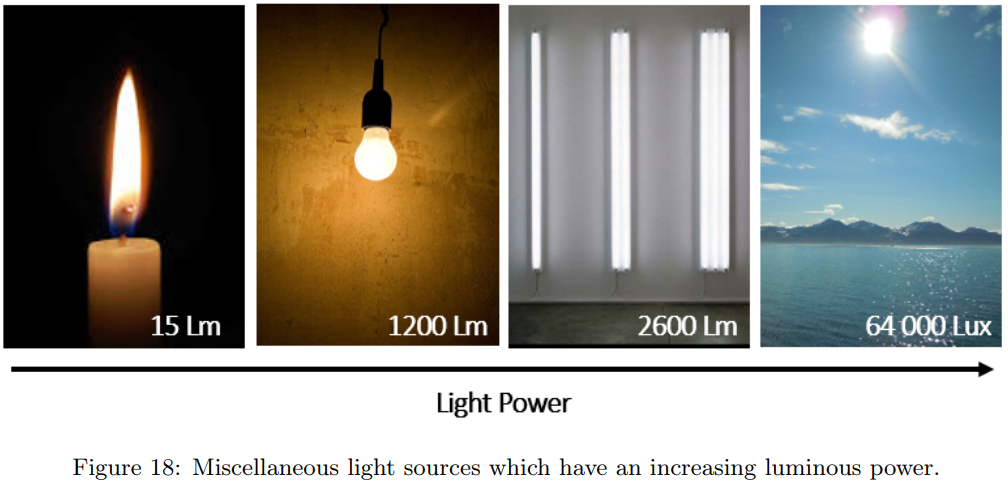
4.3 光照单位

为了具有相干照明，必须考虑光强度的比率，因此要考虑共同的单位系统。光强度可以跨越很大范围，如图18所示，保全它很重要。场景的感知丰富来自于照明的正确平衡。 图19显示了室内和室外的混合照明。展示过程将在管道末端将这种宽范围的强度转换为标准化的像素值。 见5.1节。

通常照明艺术家通过定义一个参考来设置光比率，如太阳光强度。该参考发出一定强度(通常是一个小数字，大约5到10)的光辐射，所有其他光源的值将基于它。然而，大多数情况下，这些设置被当前场景环境（地下，室内，室外等）所偏向，使得照明设备无法用于其他场景。为了引入正确的照明比例，我们在Frostbite中为灯光采用了物理单位。这允许我们：

* 与不同的光照类型协调一致。
* 在多个场景使用重复照明设备。
* 从物理材料中获得更好的响应：高对比度照明有助于揭示材料的丰富性。
* 能够使用基于物理的相机并依赖摄影师的知识。 见5.1节。





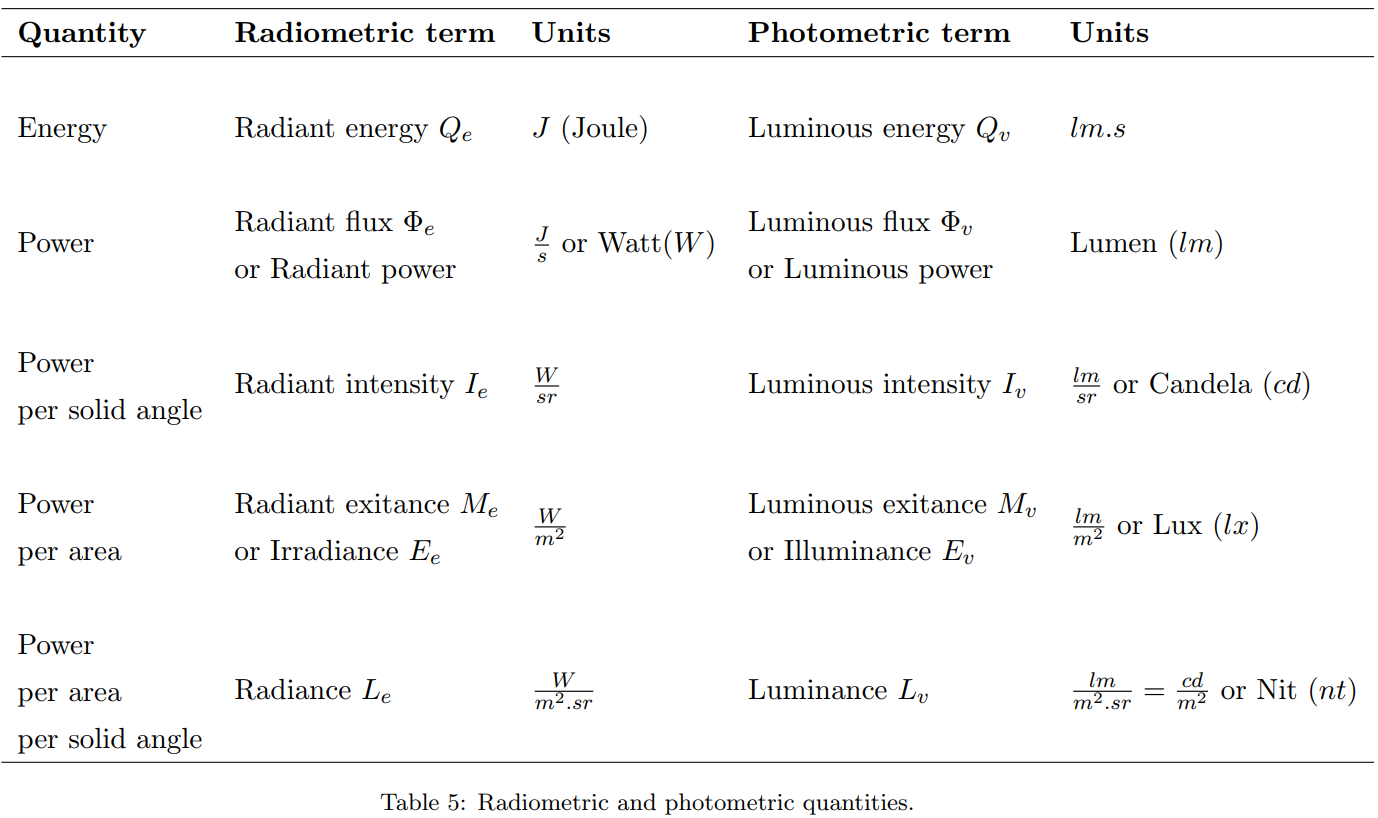
光照单元与光照测量有关，它们分为两类：

辐射测量：处理“纯”物理量，并用于光辐射测量和光谱渲染

光度学：仅关注可见光谱内的辐射。

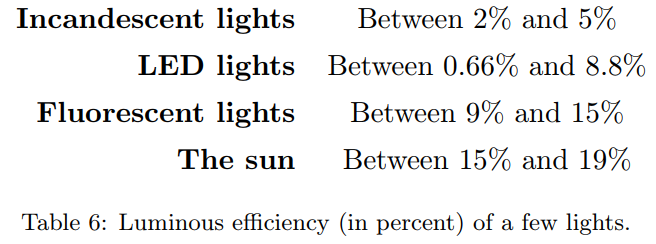
在辐射测量和光度测定中得到的量是密切相关的：光度基本上是辐射度由人眼的感知度加权得到的。这两种形式已在文献[Rei + 08]中广泛报道。最常用的辐射和光度量列于表5.能量下标e用于辐射量，视觉下标v用于光度量。

人眼的感知度由CIE光度曲线表示。它遵循钟形曲线，表示我们的眼睛如何有效地拾取某些波长的光，见图20。人眼的灵敏度在555nm处达到峰值，这对我们来说是绿色的。 在该波长下，灵敏度函数值为1个单位，意味着100％的效率。光度量与辐射量有关，在可见光谱（380nm至780nm）上进行以下积分：



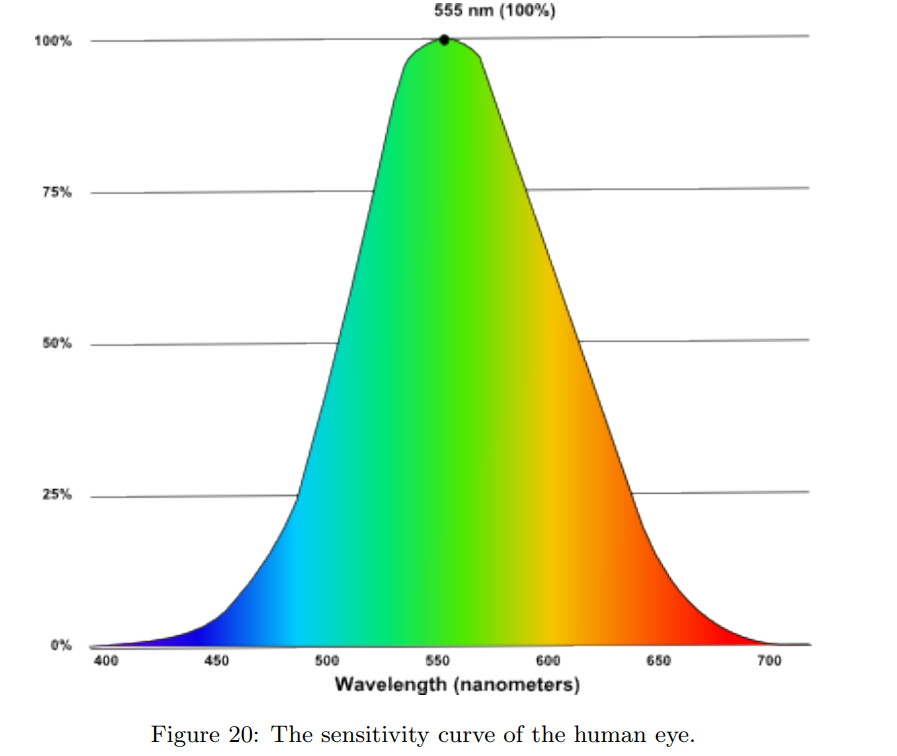
其中被称为光学视觉辐射最大光谱发光效率，它的值基于坎德拉(candela)，是发光强度测量的SI单位。它的定义为：是一光源在给定方向上的发光强度，该光源发出频率为540×10^12赫兹(波长为555nm)的单色辐射，且在此方向上的辐射强度为1/683瓦特/球面度。意思是 当处理灯泡强度时，通常以辐射功率表示，上述公式可以改写成一个简单的短语：“1瓦的绿色555nm光是683流明”。 光度曲线还允许我们推断出光的发光效率。 这可以解释为光源产生多少可见光。 计算发光效率的公式如下:

其中η以流明（lm）/瓦（W）为单位。该公式意味着绿色（555nm）光具有683lm / W的效率。这相当于100％的发光效率。不同光照类型的发光效率可能不同。结果是具有相同瓦数的两个灯可以产生不同的感知强度。表6显示了几种不同灯的发光效率。



在非光谱渲染（如游戏）的背景下，可以进行一些简化以消除辐射测量单位和光度单位之间的转换。提供光的辐射功率和发光效率使我们能够将其功率从瓦特转换为流明。

当此信息不可用时，通常假设灯的效率为100％，η= 683.此公式通常会扩展到其他数量：

在光谱渲染器中，需要以辐射测量单位描述光。使用光谱加权辐射的连续操作仅在辐射测量域内是正确的。 将该光谱辐射转换为像素值的后续阶段通常需要光度加权过程（包括自动曝光或色调映射）。

辐射测量或光度测量？对于非光谱渲染，与大多数游戏引擎一样，在辐射单位和光度单位之间来回需要发光效率信息。由于提供两个值（光强度和发光效率）因而增加了艺术家的复杂性，所以最好使用假设η= 683。通过这样的近似，每个量是另一个的线性变换，因此它们的处理将是相同的。然而，我们希望在引擎中使用真实世界的光强度能够匹配在现实世界中观察到的。100％发光效率的假设阻止了辐射参考值的使用，因为这会产生过亮的照明。 幸运的是，商业灯提供了光度数量的特征。此外，如第4.9.1节所述，HDRI通常以照片度量单位（亮度）提供。

在Frostbite中，所有灯光都使用光度单位，这意味着我们的渲染管道将亮度值存储在渲染目标中。照明强度也可以以相对方式（不同光类型的比率）或绝对方式（真光单位）表示。对于Frostbite，我们发现处理绝对值更容易，以便能够匹配灯具制造商网站上提供的值和任何参考数据。

备注：有些设备可以测量发射光。例如，使用入射光度计，可以测量到达表面的实际明度，图21.也可以用聚光灯/亮度计测量。这本质上是一种入射光度计，使用不透明屏蔽将入射光限制在单一方向。我们无法直接测量发光强度。相反，我们必须测量距离光源已知距离的明度并计算其等效流明。

**曝光值**：曝光值（EV）通常由具有摄影知识的艺术家用于描述光强度[Ree14]。 最初，EV为给定的镜头定义了一组相机设置（由光圈，快门速度和灵敏度组成）。在点测光仪的帮助下，摄影师可以确定自动调整的相机设置，以最大限度地提高相机胶片的使用率，有关详细信息，请参阅第5.1节。但是EV现在已经被滥用，并被用作照明装置。以基数为2的对数标度表示，称为f-stop，一个正步骤对应于亮度的两倍（亮度的两倍），一个负步长对应于一半（亮度的一半）。该刻度几乎是感知线性的，这意味着+0和+ 1 EV之间的差异看起来几乎与+1和+2 EV之间的差异相同。EV不是作为灯单元设计的，其定义取决于每个设备的校准常数K：

其中是场景平均亮度，是ISO算术，是反射光测量仪校准常熟。ISO 2720：1974建议的范围在10.6和13.4之间。最常用的两个值：12.5（Canon，Nikon和Sekonic）和14（Minolta，Kenko和Pentax）[Wikg]。似乎是渲染器中最常用的值[Wikc]。 表7显示了EV和亮度之间的对应关系。用于区域/发射光的基础2对数光单元是处理高值的理想选择。由于艺术家习惯于EV单位，我们决定采用的EV作为可选单位。因此，从EV转换为亮度：

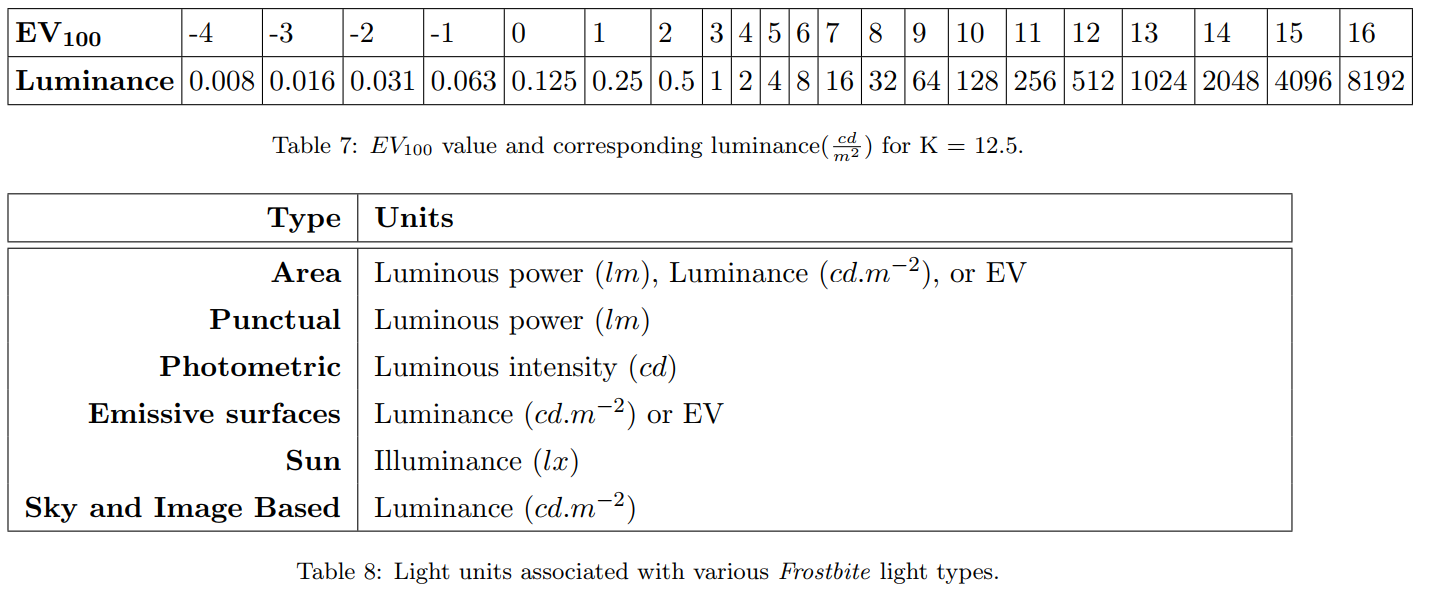


表8显示了Frostbite支持的不同光照类型的光源单位。用于每种光照类型单元的相应优点将在以下部分中说明。

4.4 精确光源

Frostbite仅支持两种类型的精确光源：点光源和聚光灯。为了使精确光源在物理上是正确的，它必须遵循所谓的“平方反比定律”[Wikf]，参见图22.从来自恒定辐射强度的光源观察到的光强度与距离物体的距离的平方成比例地减小。平方反比定律仅适用于点光源，不适用于：

分布很窄泛光灯和探照灯，因为光束高度融合。

区域灯或类似于菲涅耳透镜的特定灯。

流明功率：把平方反比定律转换成公式：

其中表示光照强度，代表明度。该等式要求在整个照明计算中距离单位是均匀的（m，cm，mm，...）。当距离趋于0时，这些值可以变为无穷大，但这在现实生活中不会发生，因为光总是有一个大小。 然而，在实时图形中，通常会添加一个小偏差以避免数字问题。 一个好的经验法则是将准时灯视为具有最小尺寸，并且永远不要让物体穿透它们。在Frostbite 1单位= 1米，我们定义准时灯的大小为1厘米（为简化表示法，我们将在本文档的提醒部分省略“max”运算符）：

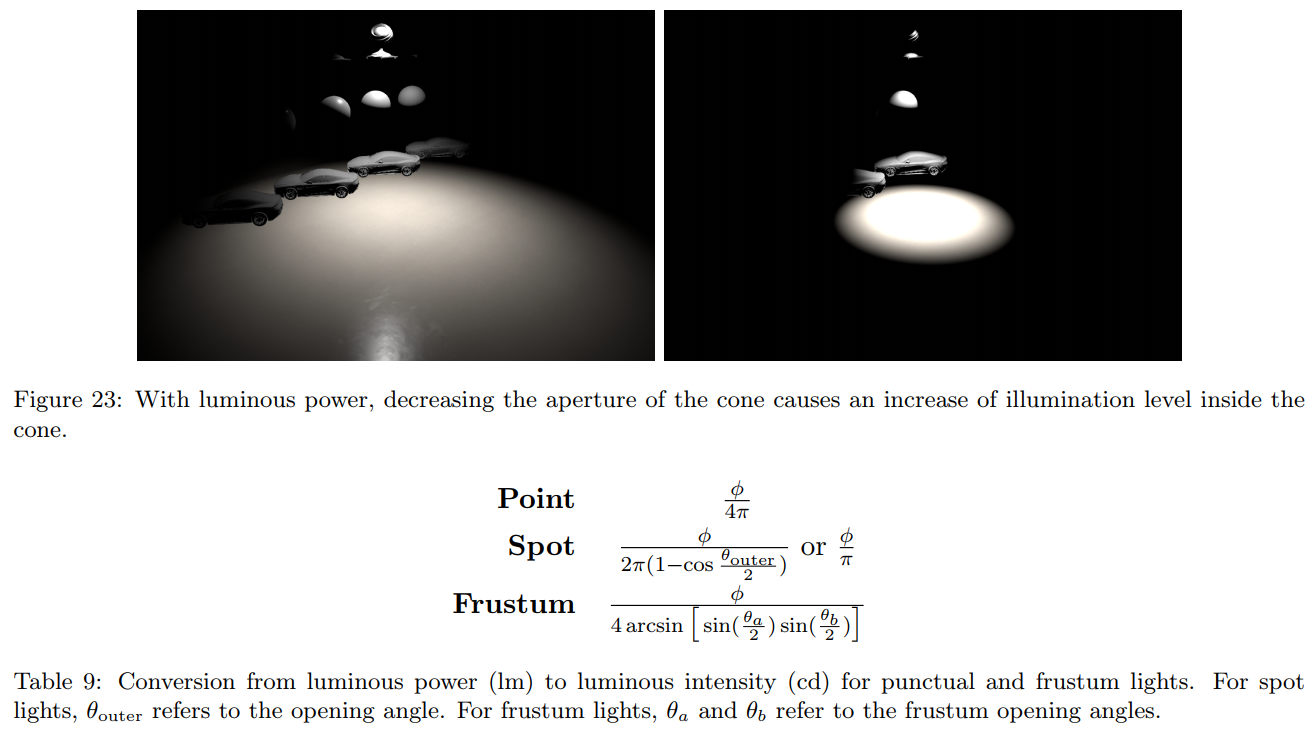
在Frostbite中，艺术家可以使用发光功率单位或发光强度单位来控制精确光强度，请参阅第4.5节。发光功率总是被转换为发光强度以进行照明计算。通过将发光强度在立体角上积分，可以根据发光强度计算发光功率

点光源:

聚光灯(开放角)：

等式16是锥体的精确立体角。如果是半角则积分变为。这个公式的结果是，对于相同的功率，锥角更小，照明更亮，因为小角度会导致照明将更集中，见图23。这样的耦合使得聚光灯操作对于艺术家来说更难，并且在优化时会引起麻烦。我们选择放下聚焦光耦合，就像点反射器会吸收光一样，就像一个简单的掩模。考虑到这些因素，并且为了平滑圆盘区域光线的过渡，我们将Frostbite中聚光灯的发光功率定义为：

表9总结了精确光源从发光功率到发光强度的转换。为了完整性，我们还添加了截头光[Wiki]（即矩形金字塔）的转换。截头光和线性光被视为Frostbite中的区域光，因此他们不使用这些公式。它们仅偶尔使用，无法共享点光源和聚光灯的有效照明路径。



光照计算：这些光源的照明计算如下：

点光源：

考虑一个表面法向直接指向光源，则照明结果遵循平方反比定律：

聚光灯：

测量：为了验证两种关系，我们使用入射光度计进行了一组测量。 设置如图24所示。表10中提供了结果，验证了我们对发光功率单位的处理以及我们对能够代表灯泡的点光源的假设。注意具有两倍流明量的光具有两倍的明度。 最接近的测量是对灯泡最高亮度值的测量，因此取决于灯泡的大小。我们没有对聚光灯进行任何测量，因为设置更复杂，并且不符合我们对聚光灯的定义。 Frostbite中的点光源是我们用于所有强度校准的参考灯。它们符合现实世界的测量值，因此是值得信赖。



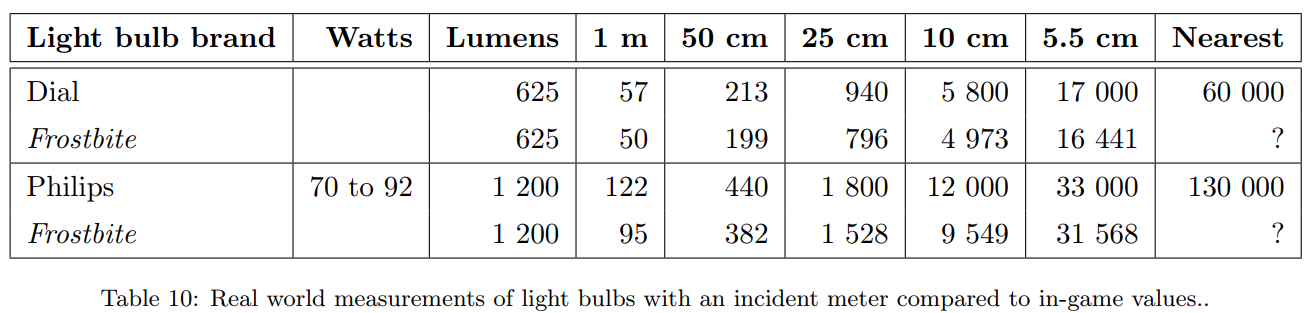
图24：照明测量设置：位于没有窗户的音响室内; 墙壁被减少声音的材料覆盖，低反照率，减少光线反弹。右下图灯泡底部是自阴影，因此强度不是完全各向同性的。因此，测量是对灯丝水平方向上进行的。对于每次测量，我们等待一定的时间让灯泡升温。请注意，入射光度计对任何障碍物（如人体）或反射（如白色表面）都非常敏感。所有测量距离都是从入射光度计的传感器到灯泡内的灯丝。

衰减：平方反比定律的一个问题是它永远不会达到零。出于性能原因，渲染器必须实现有限光范围以支持光剔除算法。在一定限度下，明度应平稳地达到零。解决这个问题的一种方法是以大部分功能不受影响的方式锁定衰减[Kar13]。为此，我们将根据距离阈值使用基本线性插值零：

为了保持功能不变，我们按如下方式调整距离阈值：

这种简单的方法有效，但会导致过渡区域生硬，看起来不自然。第二种方法是将函数通过阈值偏移并将其重新映射到其初始范围（0-1）[Mad11]。

虽然结果更好但是这种方法在0处具有非零梯度值，这导致可见的不连续性。更好的方法是窗函数并确保lightRadius处的零梯度。这可以通过提高窗口函数的指数来实现。见图25



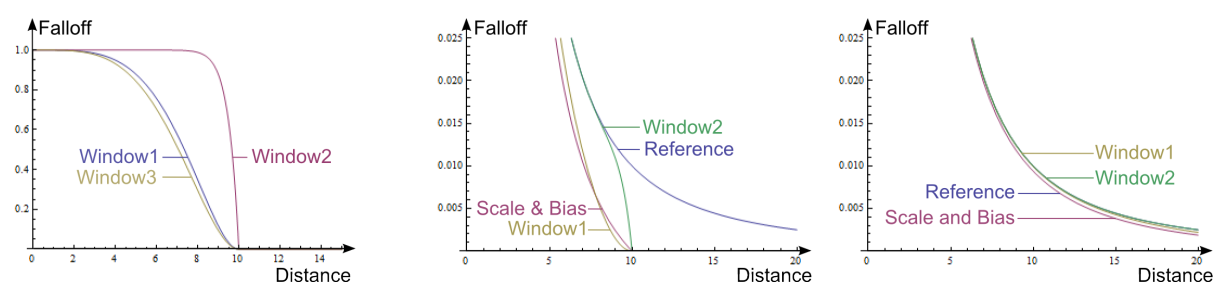


图25，明度：第一张图显示了不同的窗口函数，以突出跨越距离轴时的不连续性。 Window1是[Kar13]中使用的，Window2是不连续的，Window3是平滑的。 第二幅和第三幅图显示了应用了灯光半径衰减指数分别为10和40的窗函数。显示了当灯光半径增加时曲线与参考的拟合程度。

允许我们调整过渡平滑度。对于Frostbite，我们采用了值为的Window1，如Karis [Kar13]所示。所有精确光以及区域灯都采用了这种平滑函数。但是，由于lerp尺度是径向（xd22），因此对于非球形形状（如管或矩形灯）效果不佳。出于性能原因，我们接受这种权衡。如果需要精确度，艺术家仍然可以增加灯光半径。

*清单4显示了本节中讨论的不同主题的结果代码。 原文P.33，限于篇幅和排版这里不再列出*

4.5 光度学灯光