|  |  |
| --- | --- |
| 翻译前字数 | 603 |
| 目录 | * Unity Manual/Graphics/Graphics Reference/Rendering Pipeline Detail/Forward Rendering Path Detail |
| 链接 | https://docs.unity3d.com/Manual/RenderTech-ForwardRendering.html |

**Forward Rendering Path Details**

**正向渲染路径详情**

This page describes details of **Forward** [rendering path](http://docs.unity3d.com/540/Documentation/Manual/RenderingPaths.html).

这项页面描述了正向渲染[路径](http://docs.unity3d.com/540/Documentation/Manual/RenderingPaths.html)的细节

Forward Rendering path renders each object in one or more passes, depending on lights that affect the object. Lights themselves are also treated differently by Forward Rendering, depending on their settings and intensity.

正向渲染路径使用一个或者更多的passes渲染每一个对象，取决于影响对象的灯光。在正向渲染中，光源本身也会根据他们的设置和强度受到不同的对待。

**Implementation Details**

**实现细节**

In Forward Rendering, some number of brightest lights that affect each object are rendered in fully per-pixel lit mode. Then, up to 4 point lights are calculated per-vertex. The other lights are computed as Spherical Harmonics (SH), which is much faster but is only an approximation. Whether a light will be a per-pixel light or not is dependent on this:

在正向渲染中，影响物体的最亮的几个光源使用逐像素光照模式。最多有四个点光源会以逐个渲染的方式计算。其他光源会以球面调和（Spherical Harmonics (SH)）的方式计算。球面调和计算速度很快但是只是一个近似值。判断一个光源是否为逐像素光源会依赖以下几点：

* Lights that have their Render Mode set to Not Important are always per-vertex or SH.
* 渲染模式设为不重要（Not Important）的光以逐定点或者球面调和的方式计算。
* Brightest directional light is always per-pixel.
* 最亮的方向光总是以像素光源计算。
* Lights that have their Render Mode set to **Important** are always per-pixel.
* 渲染模式设置为重要（Important）的光源设置为像素光源
* If the above results in less lights than current Pixel Light Count [Quality Setting](http://docs.unity3d.com/540/Documentation/Manual/class-QualitySettings.html), then more lights are rendered per-pixel, in order of decreasing brightness.
* 如果呈现的结果比当前[Quality Setting](http://docs.unity3d.com/540/Documentation/Manual/class-QualitySettings.html)中像素光源的数量（Pixel Light Count）少，为了减少亮度，更多的光源会以逐像素模式渲染。

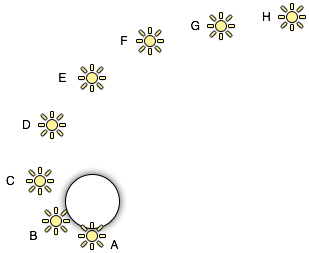
Rendering of each object happens as follows:

每个对象的渲染情况如下：

* Base Pass applies one per-pixel directional light and all per-vertex/SH lights.
* 基础Pass渲染单像素方向光和所有逐定点/球面调和光照。
* Other per-pixel lights are rendered in additional passes, one pass for each light.
* 其他逐像素光在额外的passes中渲染，每个光源需要一个pass。

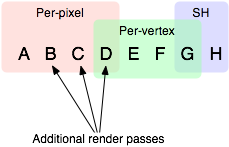
For example, if there is some object that’s affected by a number of lights (a circle in a picture below, affected by lights A to H):

举个例子，如果一些对象他们受几个光源的影响（下图是一个圆，受到光源A到H的影响）：



Let’s assume lights A to H have the same color and intensity and all of them have Auto rendering mode, so they would be sorted in exactly this order for this object. The brightest lights will be rendered in per-pixel lit mode (A to D), then up to 4 lights in per-vertex lit mode (D to G), and finally the rest of lights in SH (G to H):

假设光源A到光源H有相同的颜色和强度，他们都有自动的渲染模式，那么它们会以自身的顺序排序。最亮的光以逐像素的光照模式渲染（A到D），最多有四个光采用逐顶点的光照模式渲染（D到G），剩余的光源以球面调和方式渲染（G到H）。



Note that light groups overlap; for example last per-pixel light blends into per-vertex lit mode so there are less “light popping” as objects and lights move around.

注意光组重叠；例如最后的逐像素光源也是逐顶点光照模式渲染，可以减少物体和灯光移动时产生的光照跳跃。

**Base Pass**

**基础Pass**

Base pass renders object with one per-pixel directional light and all SH/vertex lights. This pass also adds any lightmaps, ambient and emissive lighting from the shader. Directional light rendered in this pass can have Shadows. Note that Lightmapped objects do not get illumination from SH lights.

基础Pass渲染采用一个逐像素方向光和所有球面调和光渲染物体。这个pass也可以从shader中添加任何的光照贴图，环境光，自发光。方向光在这个pass中渲染可以有阴影。

注意使用了光照贴图的对象不会得到球面调和光的光照。

Note that when “OnlyDirectional” [pass flag](http://docs.unity3d.com/540/Documentation/Manual/SL-PassTags.html) is used in the shader, then the forward base pass only renders main directional light, ambient/lightprobe and lightmaps (SH and vertex lights are not included into pass data).

注意当”OnlyDirectional”的 [pass flag](http://docs.unity3d.com/540/Documentation/Manual/SL-PassTags.html).使用在着色器中，前向基础的pass之渲染主要的方向光，环境光，光探头和光照贴图（球面调和和顶点光照不会包含在pass数据中）

**Additional Passes**

**附加的passes**

Additional passes are rendered for each additional per-pixel light that affect this object. Lights in these passes by default do not have shadows (so in result, Forward Rendering supports one directional light with shadows), unless [multi\_compile\_fwdadd\_fullshadows](http://docs.unity3d.com/540/Documentation/Manual/SL-MultipleProgramVariants.html) variant shortcut is used.

附加的passes用于渲染影响物体的逐像素光源。光照在这些passes中默认没有阴影（因此，正向渲染支持一个带阴影的方向光,除非[multi\_compile\_fwdadd\_fullshadows](http://docs.unity3d.com/540/Documentation/Manual/SL-MultipleProgramVariants.html)变体被使用）。

**Performance Considerations**

**性能的注意事项**

Spherical Harmonics lights are very fast to render. They have a tiny cost on the CPU, and are actually free for the GPU to apply (that is, base pass always computes SH lighting; but due to the way SH lights work, the cost is exactly the same no matter how many SH lights are there).

渲染球谐函数的光照是非常快的。CPU消耗很小，并且实际无需消耗GPU的时间。（基础pass会计算球面调和光照，无论有多少球面调和光。计算他们的时间都是相同的。）

The downsides of SH lights are:

球面调和的缺点：

* They are computed at object’s vertices, not pixels. This means they do not support light Cookies or normal maps.
* 他们计算的是对象的定点，不是像素。意思是他们不支持light Cookies或者法线贴图。
* SH lighting is very low frequency. You can’t have sharp lighting transitions with SH lights. They are also only affecting the diffuse lighting (too low frequency for specular highlights).
* 球面调和光是非常低的频率。球面调和光照不可以共享光照变换。他们也只影响漫反射照明（太低的频率为了镜面高光）。
* SH lighting is not local; point or spot SH lights close to some surface will “look wrong”.
* 球面调和光不是局部的。靠近曲面的球面调和点光和聚光可能会"看起来不正确"。

In summary, SH lights are often good enough for small dynamic objects.

总的来说，球面调和光的效果对小的动态对象足够好。