

# UE4 材质系统中兰伯特光照模型研究优化与 高光反射模型的制作

## 一、兰伯特光照模型

(一) 兰伯特光照模型的计算公式：

$$C_{diffuse} = (C_{light} \cdot M_{diffuse}) \max(0, \hat{n} \cdot \hat{l})$$

C light: 场景灯光颜色

M diffuse: 模型漫反射颜色

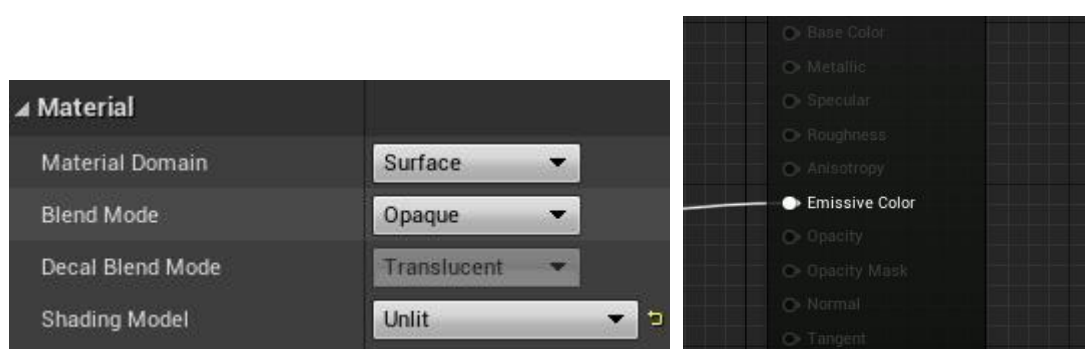
N: 归一化的模型表面法线向量

L: 归一化的场景灯光方向向量

## （二）使用 UE4 材质系统实现兰伯特光照模型：

### 1、材质球基本属性设置：

首先需要将我们新建的材质球设置为不透明不受光材质，后续可以通过将计算结果输出到 EmissiveColor 节点，利用自发光颜色来表现计算结果。



### 2、计算模型法线与环境光照方向的点积：

#### （1）计算公式与函数节点：

$$\max(0, \hat{n} \cdot \hat{l})$$

- VertexNormalWS 节点获取模型表面法线
- skyAtmosphereLightDirection 节点获取到环境光的方向
- Normalize: 归一化函数
- Dot: 点积
- Saturate: 将数值限制在 0-1 范围内

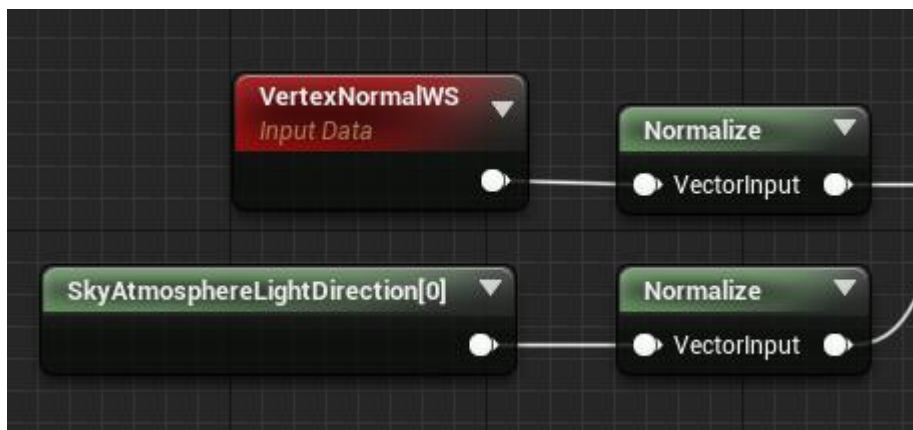
#### （2）Normalize 归一化函数：

UnityShader 入门精要一书中提到的的归一化公式：

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|}, \mathbf{v} \text{ 是任意非零矢量}$$

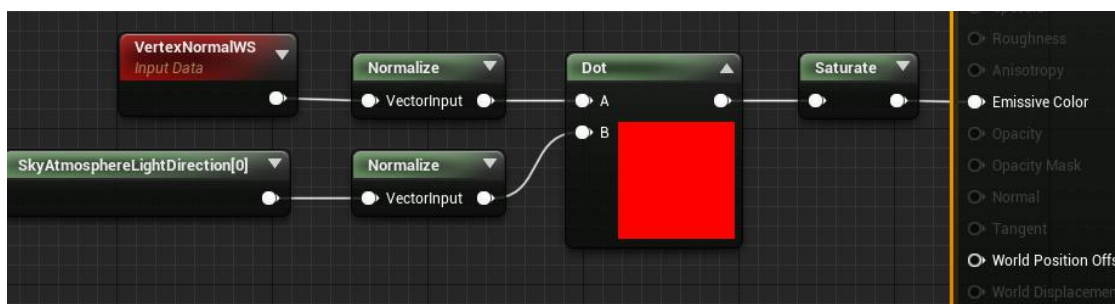
通过用该矢量除以该矢量的模来得到归一化后的向量。

在 UE4 中, 我们使用 **Normalize** 函数对获取到的模型表面法向量和环境光方向向量进行归一化

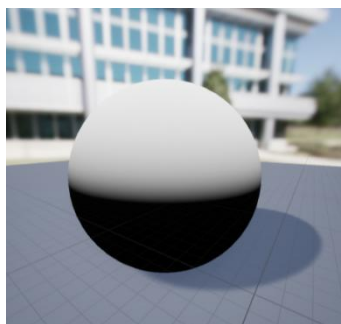


(3)、将归一化后的模型表面法线向量和环境光方向向量进行点积。

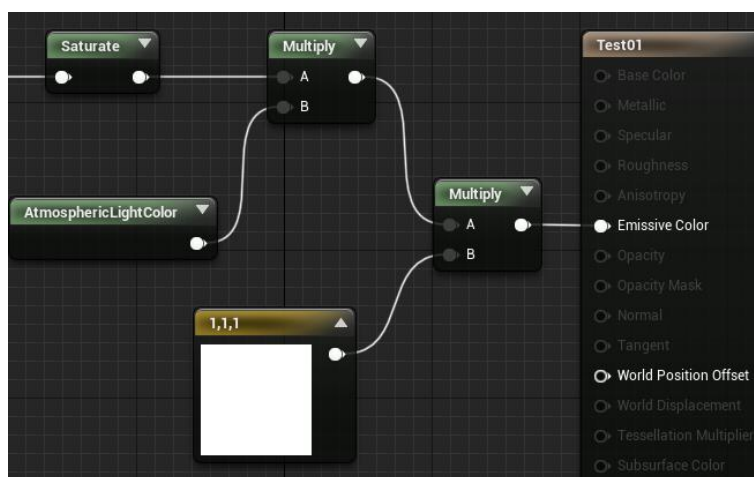
因为当模型表面法线和光照方向相反时, 所得到的结果会是负数, 所以这里使用 Saturate 节点将计算结果限制在 0-1 的范围内



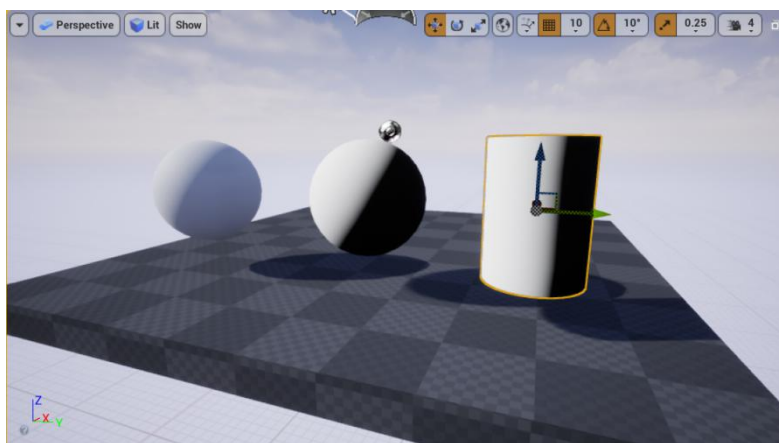
得到计算结果直接反映在 Sphere 模型上:



最后再乘以模型漫反射颜色和环境光颜色：

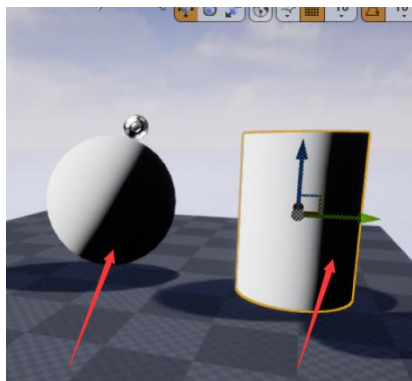


得到以下右一、右二模型与左一 UE4 默认光照模型的对比效果：



## 二、半兰伯特光照模型

(一)、在以上的兰伯特光照模型计算中，模型的非受光面为全黑色的，效果不好。



在现实场景中因为光线在进入模型表面时会经过不断的反射，所以模型的非受光面不会是全黑色的（全局光照技术）。

我们在兰伯特模型的基础上进行一些简单的修改，也称为**半兰伯特模型**。

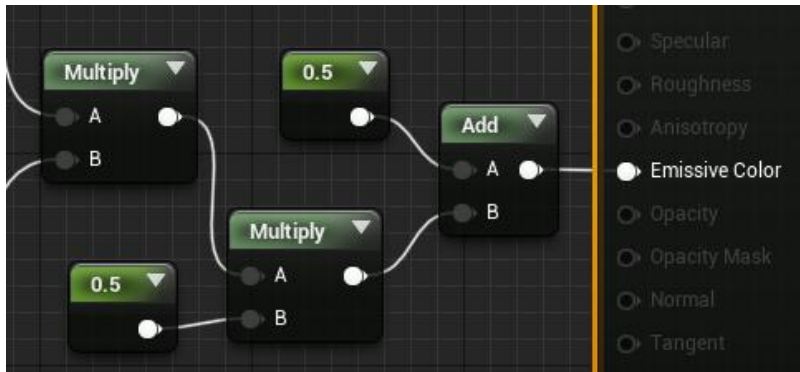
(二)、广义半兰伯特模型计算公式：

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

与原兰伯特模型相比，半兰伯特光照模型没有使用 **Max** 操作来防止归一化的模型法线和归一化的环境光照方向向量的点积结果为负值，而是对其进行一个  $\alpha$  倍的缩放后再加上一个  $\beta$  大小的偏移，绝大多数情况下  $\alpha$  和  $\beta$  的值均为 0.5，所以公式为：

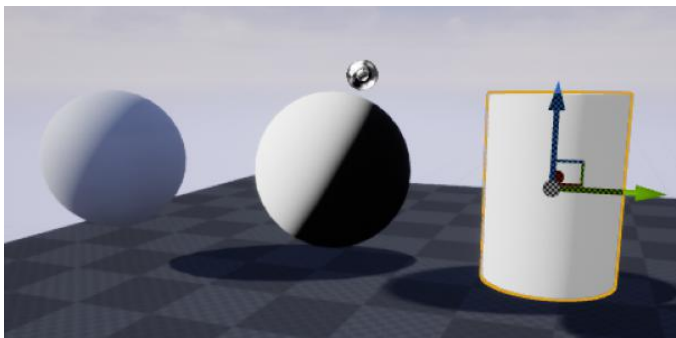
$$c_{diffuse} = (c_{light} \cdot m_{diffuse})(0.5(\hat{n} \cdot \hat{l}) + 0.5)$$

通过这个公式我们将模型法线向量与环境光向量的点积结果范围从  $[-1, 1]$  映射到了  $[0, 1]$  的范围内



得到一个视觉效果加强了的结果:

左一为 UE4 默认光照模型，右一半兰伯特光照模型，中间为兰伯特光照模型



### 三、在半兰伯特模型的基础上再进行一定的修改优化

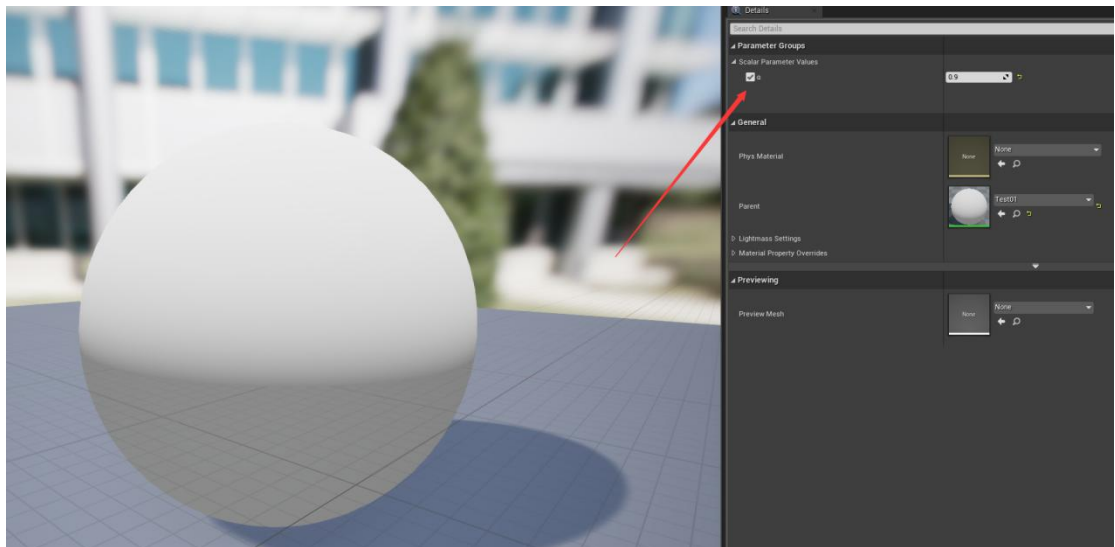
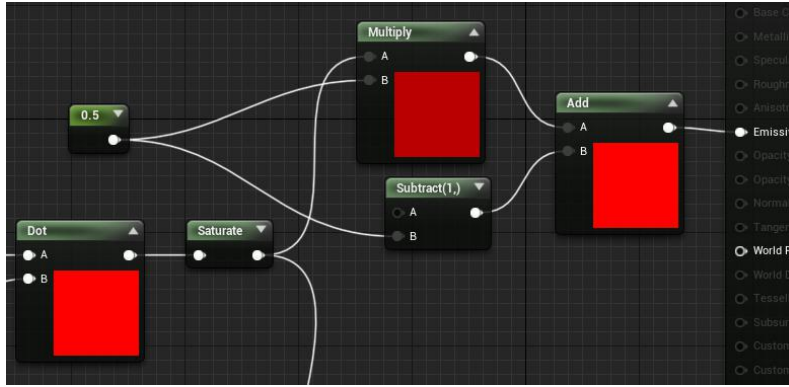
公式:

$$C_{diffuse} = (C_{light} \cdot M_{diffuse}) (\alpha (N \cdot L) + (1 - \alpha)) \quad 0 < \alpha \leq 1$$

N: 归一化的模型法线

L: 归一化的环境光照方向

这样既可做到通过修改  $\alpha$  的值来动态的调整背光面的亮度值



## 四、对半兰伯特模型再进一步修改优化（平滑渐变过渡）

公式：

$$C_{diffuse} = (C_{light} \cdot M_{diffuse}) (\alpha (\text{pow}(N \cdot L)) + (1 - \alpha)) \quad 0 < \alpha \leq 1$$

N: 归一化的模型法线

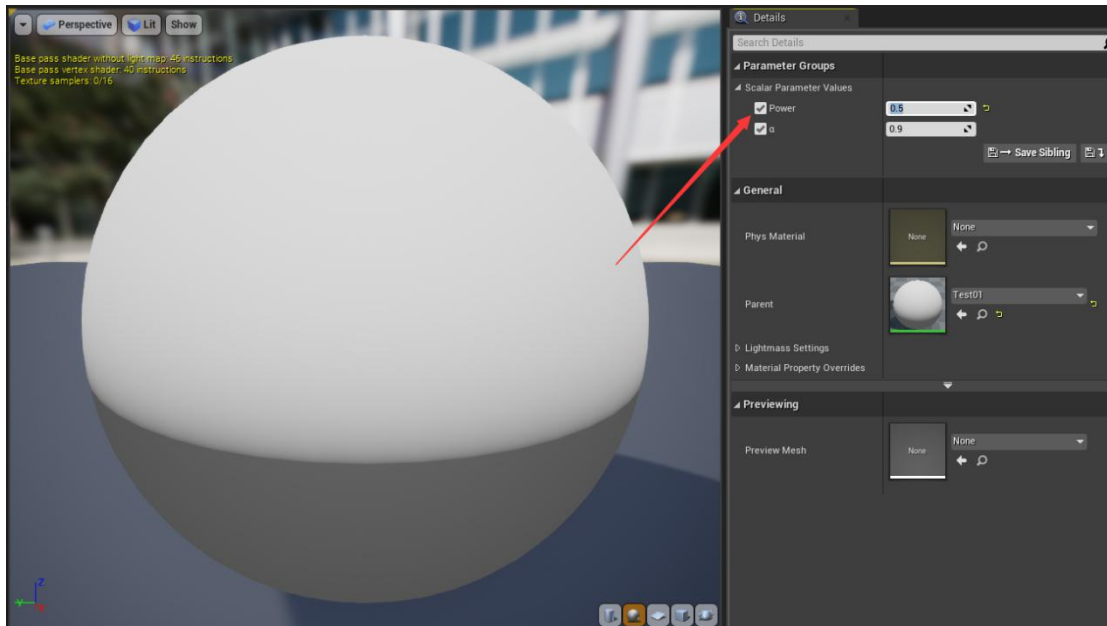
L: 归一化的环境光照方向

在原半兰伯特光照模型的优化基础上再新增加了一个 **Power** 函数处理模型法线与光照方向的点积结果。

（一）、从 power 幂函数的规律中我们知道，当指数  $c$  的值越小时，结果越接近于 1，当指数  $c$  时越大时，结果越靠近 0。

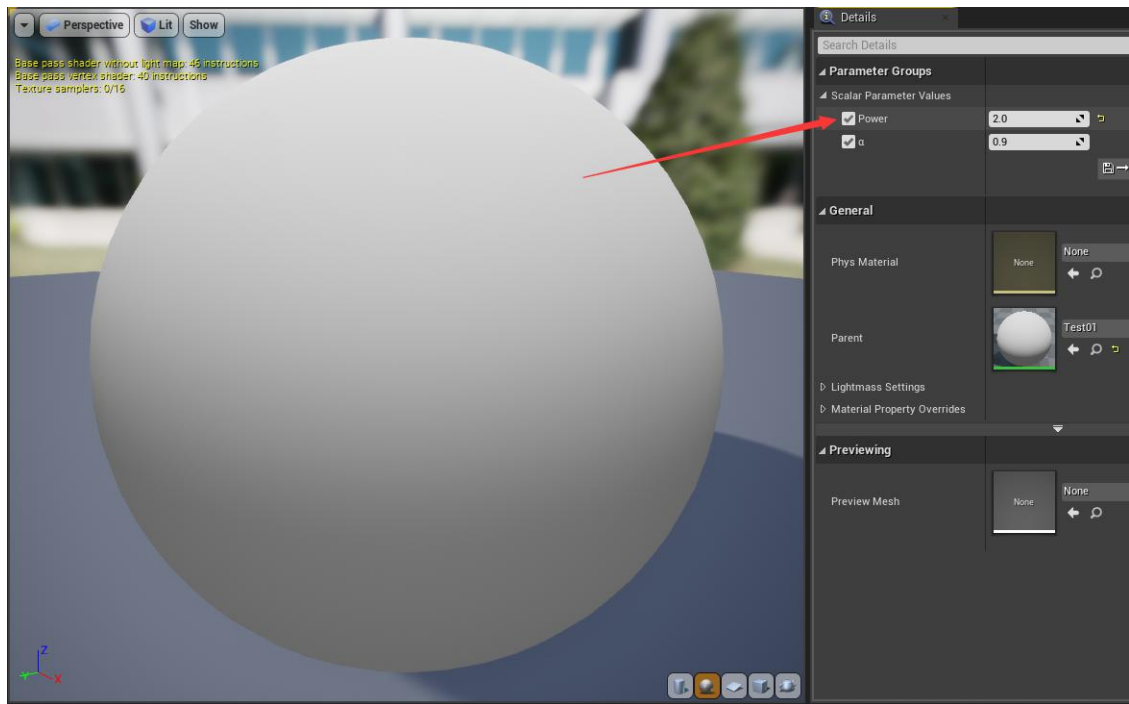
（二）在 UE4 中调整 power 的参数

当指数等于 0.5 时，过渡较为生硬：

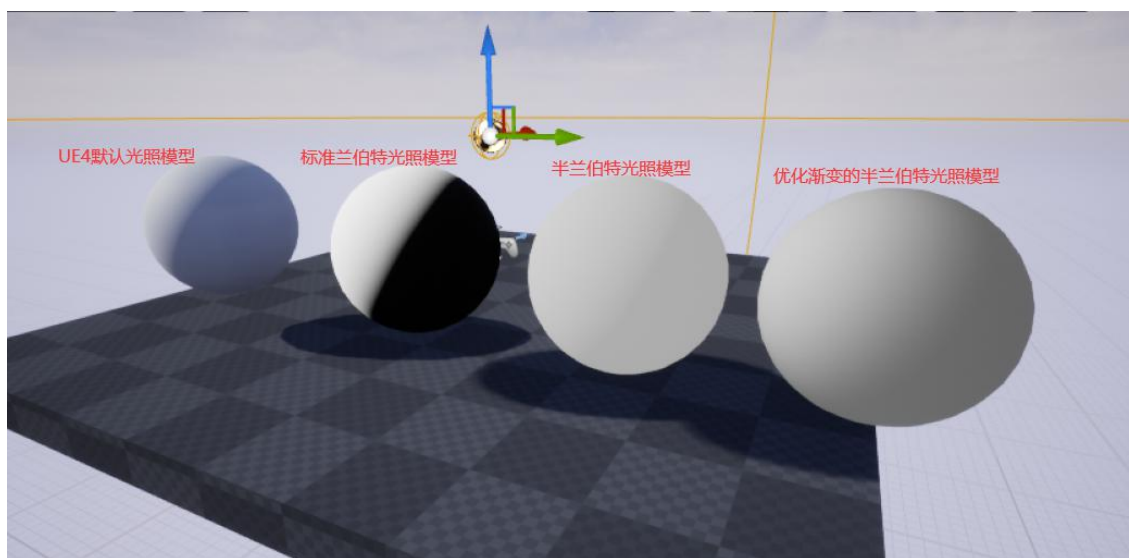


当指数等于 2 时，过渡就变的更加柔和：



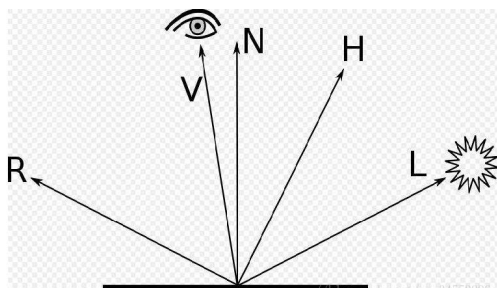


## 五、结果对比



## 六、添加高光反射效果

### （一）、高光反射部分计算公式：



$$C_{specular} = (C_{light} * M_{specular}) * \text{pow}(\max(0, \text{dot}(V, R)), M_{gloss})$$

$C_{light}$ : 入射光线的颜色和强度

$M_{specular}$ : 材质高光反射系数

$V$ : 视角方向

$R$ : 反射方向

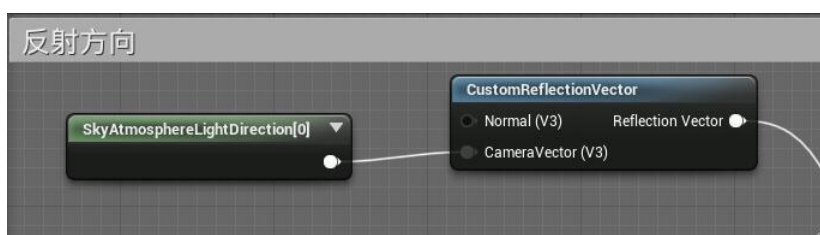
$M_{gloss}$ : 高光范围大小

### （二）、计算反射方向

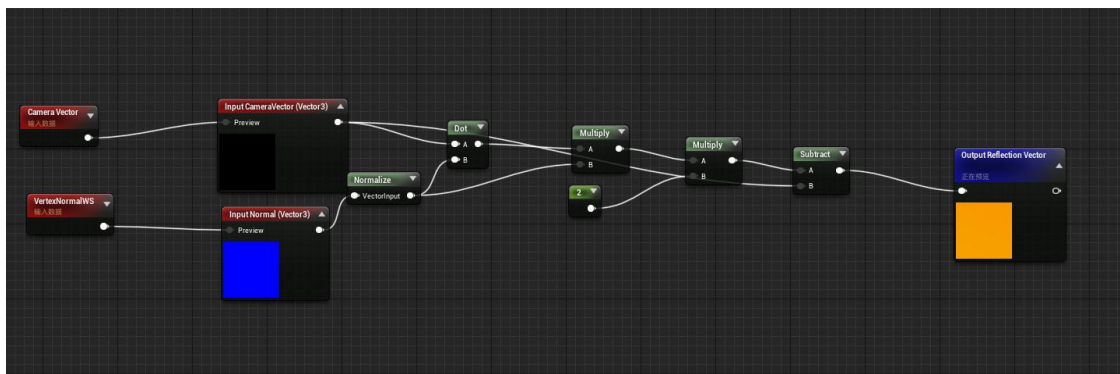
反射方向  $R$  可以由表面法线  $N$  和光线方向  $L$  计算得到：

$$R = L - 2(N \cdot L)N$$

在 UE4 中使用该节点可以计算得到反射方向  $R$  的结果：

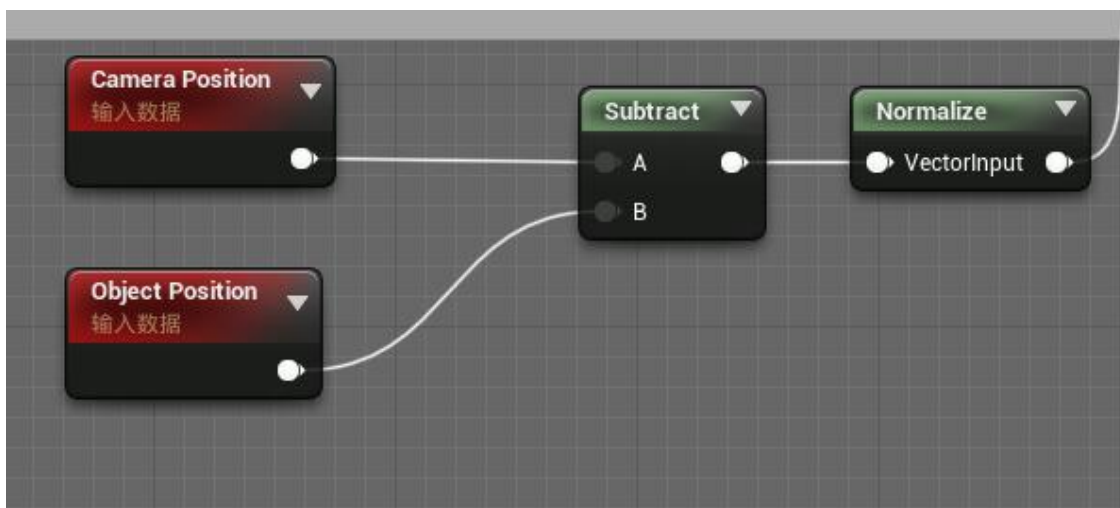


双击进入 **CustumReflectionVector** 节点，可以看到节点内部算法便是在计算表面法线与光线方向的结果：

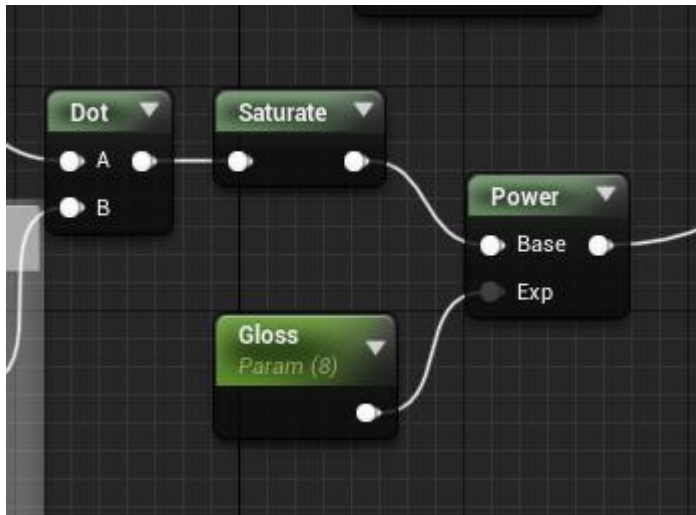


### （三）、计算视角方向

视角方向可通过归一化世界空间相机位置减去世界空间模型位置的结果得到：



将反射方向与计算机视角方向点积后，通过 **Power** 节点控制高光的大小，**gloss** 的值越大，高光范围越小：



(四)、最后将高光与前面计算的光照相加，得到以下结果：

