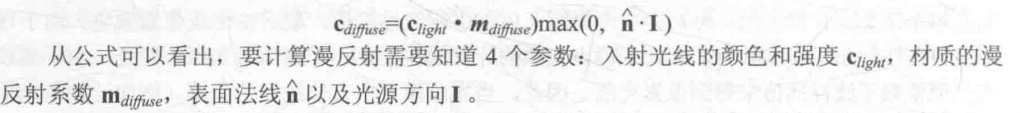
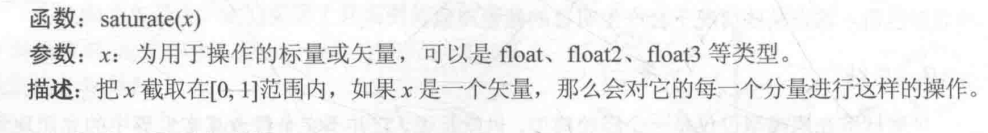
# Shader笔记

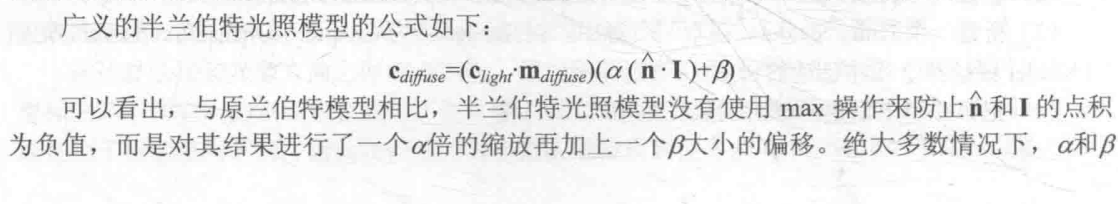
## 逐顶点漫反射

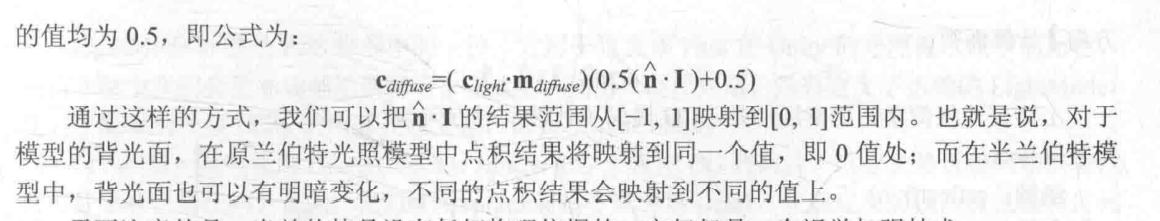
1. 兰伯特定律 计算漫反射





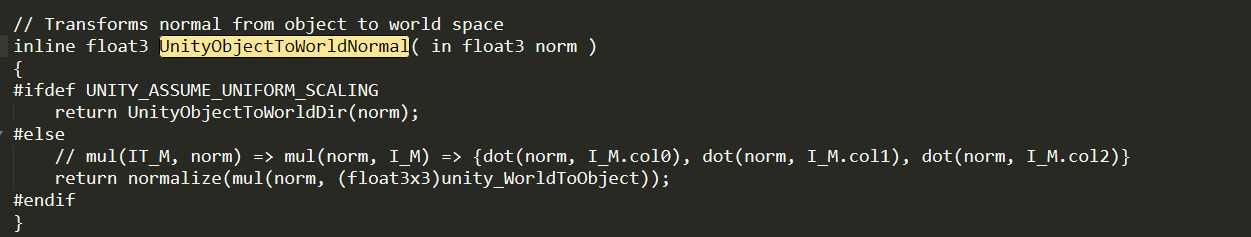
半兰伯特定律

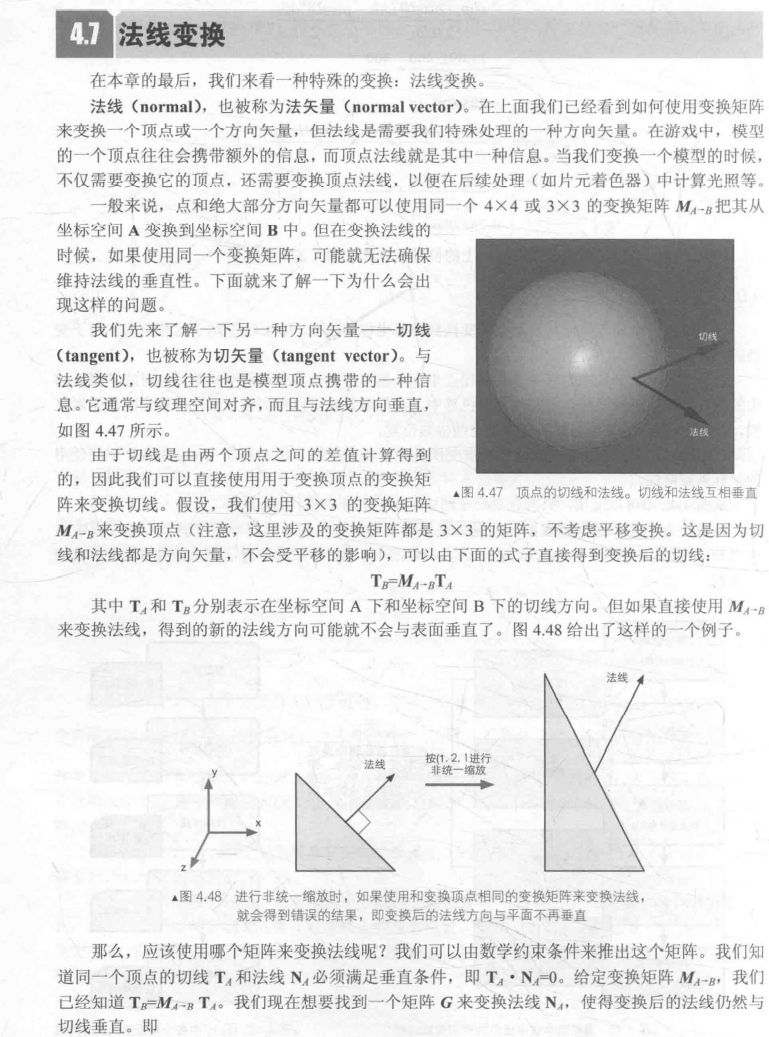


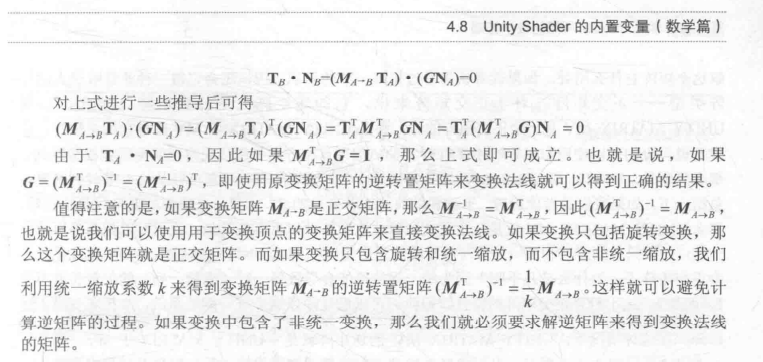


环境光颜色 UNITY\_LIGHTMODEL\_AMBIENT.xyz

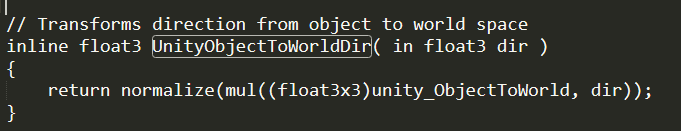
1. float4 UnityObjectToClipPos(float4 vertex); 将对象空间中的点变换到齐次坐标中的摄像机裁剪空间等同于mul(UNITY\_MATRIX\_MVP, float4(pos, 1.0))
2. float3 UnityObjectToViewPos(float3 pos)；将对象空间中的点变换到视图空间。这等效于 \_\_mul(UNITY\_MATRIX\_MV, float4(pos, 1.0)).xyz，应该在适当的位置使用。
3. float3 UnityObjectToWorldNormal(in float3 norm )







1. float3 UnityObjectToWorldDir( in float3 dir )

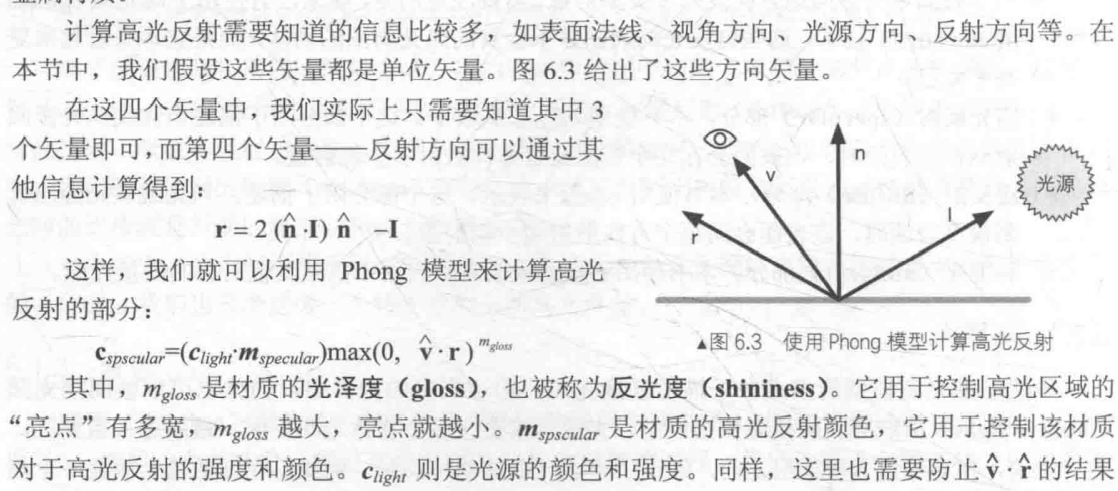


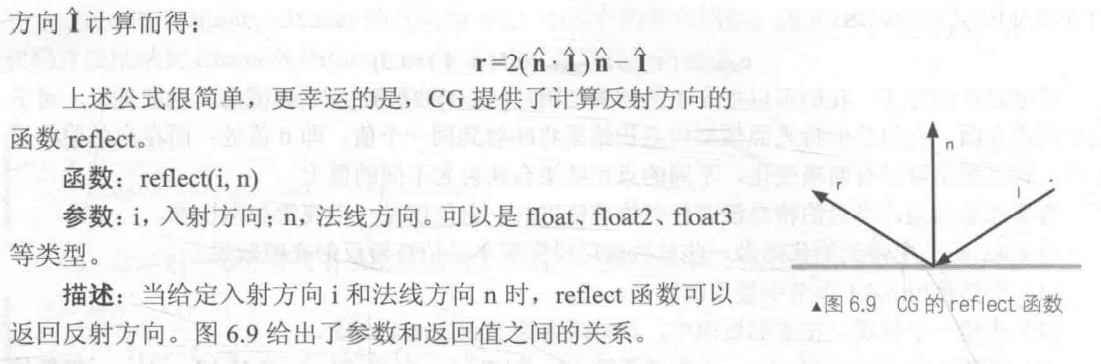
1. 光照方向

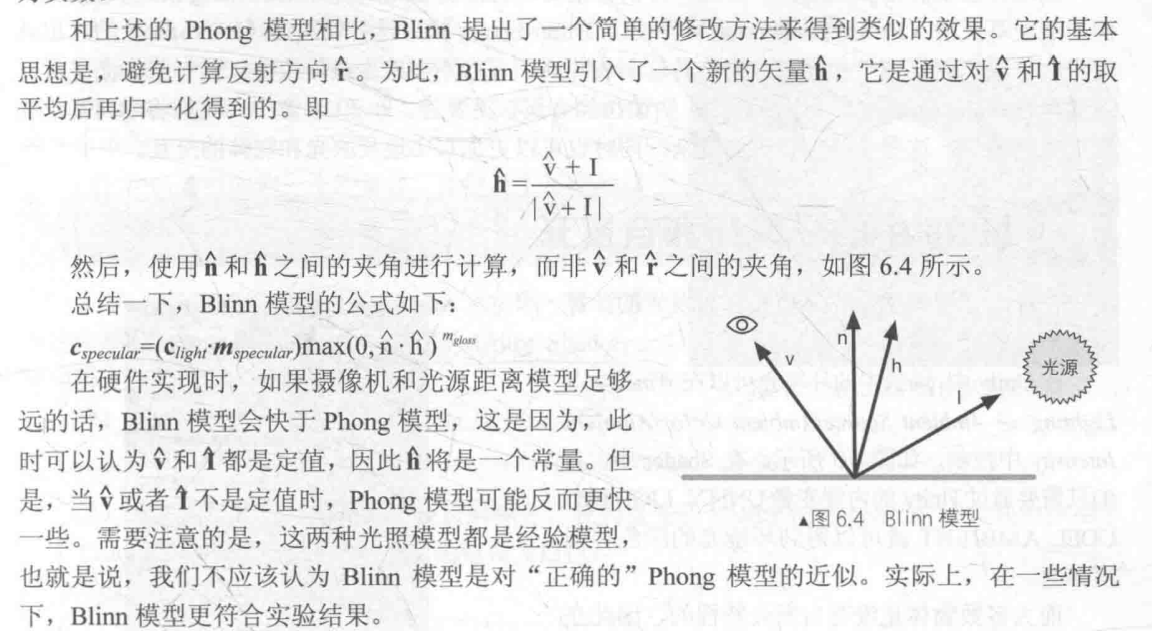
\_WorldSpaceLightPos0.xyz 必须引入 #include "Lighting.cginc"

1. 高光反射

Phong模型



Blinn-Phong模型



1. 相机位置获取 \_WorldSpaceCameraPos.xyz