直接光照

**LightingPhysicallyBased** = **brdf** \* **radiance**

**radiance** = lightColor \* (lightAttenuation \* NdotL)

lightAttenuation = light.**distanceAttenuation** \* light.**shadowAttenuation**

NdotL = dot(normalWS, lightDirectionWS)

**brdf** = diffuse + specular \* DirectBRDFSpecular

diffuse = albedo \* (1 - metallic)

specular = lerp(0.04, albedo, metallic)

DirectBRDFSpecular = roughness2 / (d2 \* max(0.1, LoH2) \* (roughness \* 4 + 2))

d = NoH2 \* (roughness2 -1) + 1.00001f

halfDir = Normalize(lightDirectionWS + viewDirectionWS)

NoH = dot(normalWS, halfDir)

LoH = dot(lightDirectionWS, halfDir)

**GlobalIllumination**

**GlobalIllumination** = indirectDiffuse \* diffuse + indirectSpecular \* EnvironmentBRDFSpecular

indirectDiffuse = bakedGI \* **occlusion**

**bakedGI = SampleLightmap or SampleSHPixel**

EnvironmentBRDFSpecular = 1.0 / (roughness2 + 1.0) \* lerp(specular, grazingTerm, fresnelTerm)

grazingTerm = smoothness + reflectivity

half NoV = saturate(dot(normalWS, viewDirectionWS));

half fresnelTerm = Pow4(1.0 - NoV);

indirectSpecular = GlossyEnvironmentReflection

GlossyEnvironmentReflection = irradiance \* occlusion

mip = PerceptualRoughnessToMipmapLevel(perceptualRoughness);

encodedIrradiance = SAMPLE\_TEXTURECUBE\_LOD(unity\_SpecCube0, samplerunity\_SpecCube0, reflectVector, mip);

half3 irradiance = encodedIrradiance.rgb OR

half3 irradiance = DecodeHDREnvironment

return irradiance \* occlusion;