

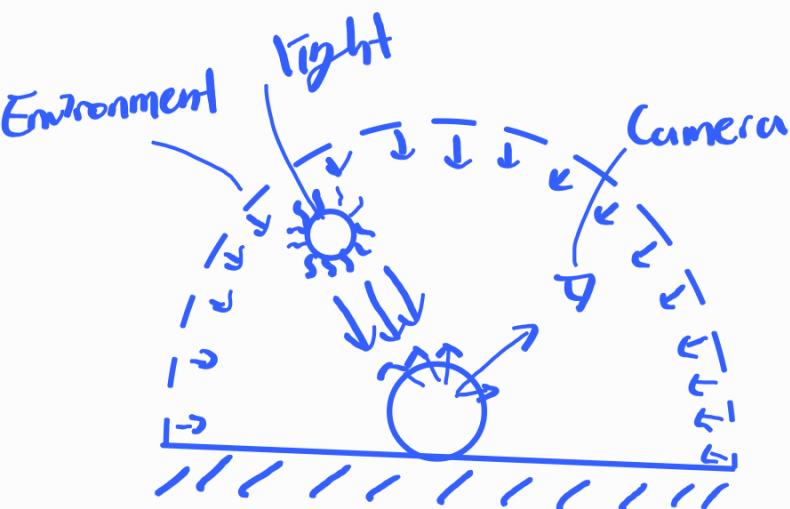
PBR (Physically Based Rendering)

PBR 구현 시 가장 복잡한 부분은 BRDF 구현이다.

$$f(l, v) = \frac{D(h) F(l, h) G(l, v, h)}{4(n \cdot l)(n \cdot v)}, \quad l = \text{light vector}, \quad v = \text{camera vector}$$

$h = \text{halfway vector}$
 $n = \text{normal vector}$

$F(l)$ = Fresnel 방정식 (반사계수와 투과계수에 관련 계수) 를, 흐리기 등



간접광은 빛이 물체에 반사되거나 (camera)로 들어오는 걸 면밀히 살것

이때 빛(Light)을 2가지로 분류

1) Environment Light (주변광)

전반적인 주변환경에서 오는 조명

2) Direct Light (직접광)

Spot/Point light 같은 직접적인 조명

Unreal PBR에서는 주변에서 들어오는 간접광을 IBL으로 처리(HDR)

남은 부분은 Directional / Point / Spot light들로 조명 효과를 더해나간다.

물리적인 관점(Physics)



빛은 파동의 성질을 가지고 있다

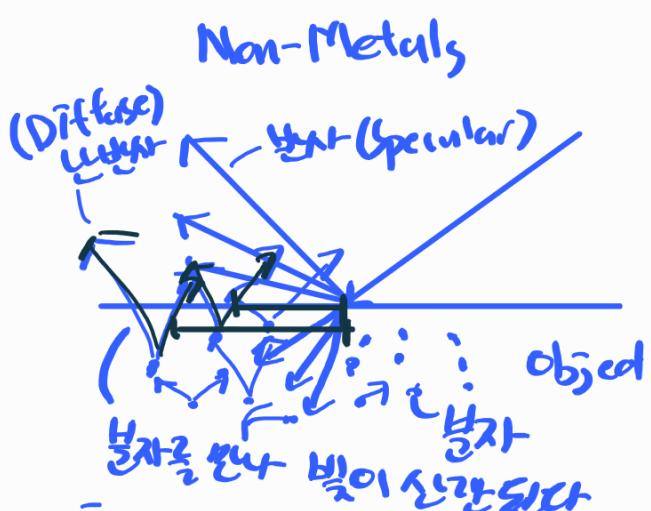
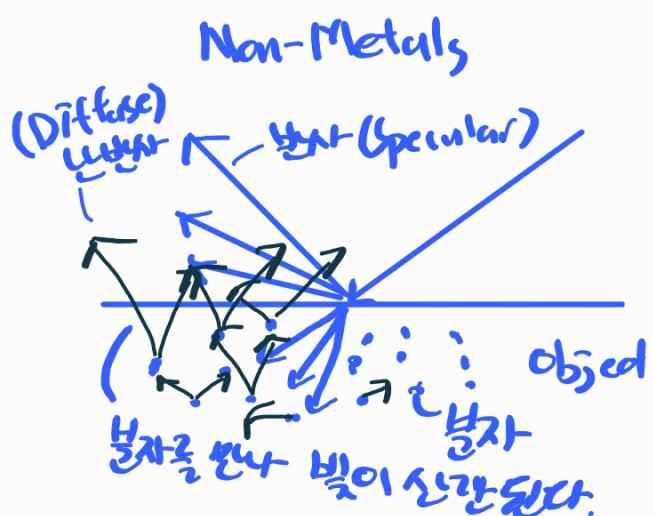
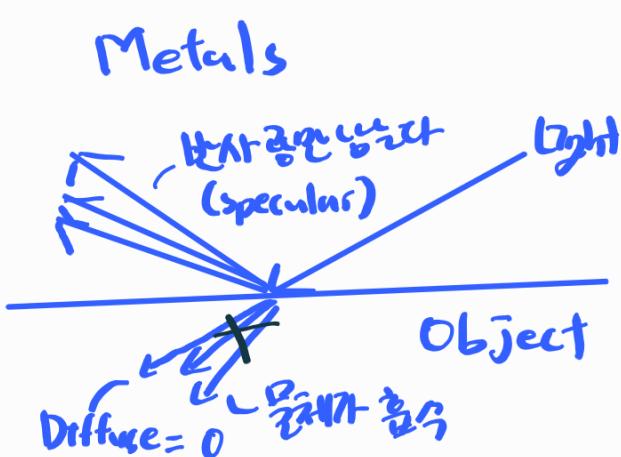
빛은 다양한 wavelength을 가진 여러 파동들이 혼합되었을 때 생긴다.
파장에 따라 나타나는 색이 다르다.
(적색 = 빨간색, wavelength가 길다;
검정 = 파란색, wavelength가 짧다)

빛이 어떤 물질을 만나서 흩어지는 걸 Scattering이라 한다.

물리적인 고정(Physics)

물체는 아주 다양한 성질을 가지고 있는데 이런 재질을 크게 3가지로 나눌 수 있다. 그중 가장 많이 쓰는 2가지 재질이다.

- 1) Metal (금속)
- 2) Non-Metal (금속이 아닌 것)



빛이 물체에 부딪힐 때 물체의 분자들에 의해 산란되거나 다시 나오는 복합까지의 현상이 일어나며 있다.

- a) 사람의 피부, 머리카락 등 사람의 피부같은 경우 빛이 나오기까지의 거리가 멀다
- b) 표면 밑에서 일어나는 Scattering을 SubSurface Scattering으로 부르기 때문에 있다.

일반적인 물체는 빛이 Scattering으로 나가는 거리가 짧다.
때서 이거라 한 Places에서 이걸 어려면 광이 신경쓸 필요가 없다.

PBR Math (Physics → Math)

Radiance = 빛이 얼마나 들어오는지 (조명이 빛을 얼마나 강하게 쏘는지)

Single Ray = 한 Pixel에서 고려하는 하나의 광선

Spectral / RGB = 어떤 빛안에 어떤 주파수 영역이 몇이 있고 적게 있는지를 Spectrum이라 한다

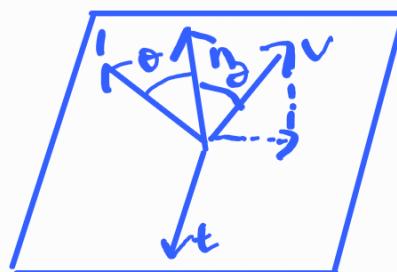
빛의 전자기자, Spectrum까지 고려한 렌더링 기술을 Spectral Rendering이라 한다.

하지만 결국 Shader(PC)로 표현하면서면 RGB 색으로 대체해 준다(혹은 흑백화면)

Bidirectional = 양방향 조명

$f(l, v)$, l=Light, v=Camera

Reflectance = 반사광

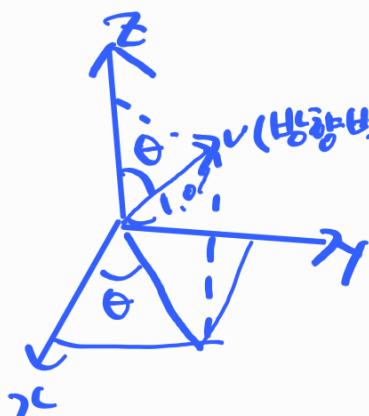


Distribution = 반반사광

Function = 가능 조명

반사광을 할 때 많이 사용하는 좌표계는 Sphere(3) Coordinate이다

3차원 좌표계는 대개 브라스터드 좌표계 → Spherical 좌표계

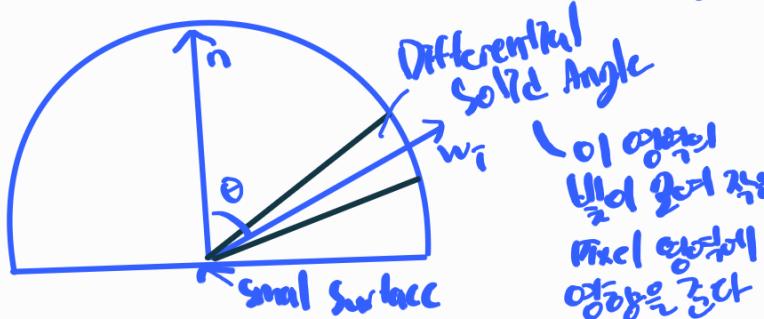


방향 벡터(v)는 Unit Vector, 벡터가 길이가 1인 제약을 가져 각도 2개(2개의 자유도)로 표현할 수 있다
(theta, phi는 가지고 표현 가능)

BRDF는 결국 비율이다.

$$\text{BRDF} = \frac{L_o}{E_i}$$

특정 방향으로부터 들어온 빛이 그 방향을 향해 어떤 비율로 반사되는지를 의미합니다



빛이 들어올수 있는 모든 방향에 대해 BRDF 함수를 적용하면 눈으로 향하는 빛의 양을 정확히 알 수 있다.

The Reflectance Equation - BRDF

$$L_o(v) = \int_{\Omega} f(l, v) \otimes L_i(l)(n \cdot l) d\omega_i$$

Component-wise Multiplication (HLSL에서 float3 사용)
곱하기

$L_o(v)$ = 빛의 방향, v 에 대해 빛의 방향 결과 (RGB)
(Pixel Shader에서 계산되는 색깔)

$L_i(l)$ = l 방향으로 부터 들어오는 빛의 양 (그린시 RGB값) (RGB)

$(n \cdot l)$ = (Normal vector • Light direction vector) (scalar)

$L_i(l)(n \cdot l)$ = 어떤 조명으로 부터 표면으로 빛을 쏘았을 때 들어오는 양

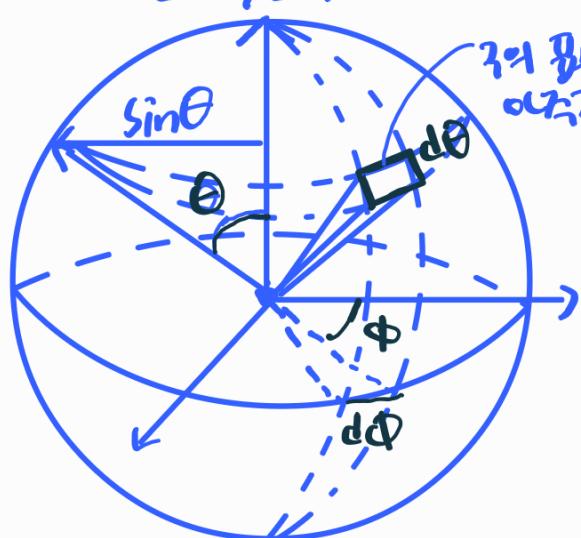
$f(l, v)$ = BRDF, 빛의 방향 (RGB)과 색상 (Vector) 형태로 나온다

$L_i(l)(n \cdot l)$ 에 대해 빛의 양이 실시간으로 눈에 들어올 수 있는 빛이 되어 $(L_o(v))$ → BRDF에 따라 전달되는 결과가 달라진다

\int_{Ω} = 빛이 여러 방향에서 들어온 것에 따른 방향에 대해 적분해준다.

$d\omega_i$ = Solid Angle, $d\omega_i$ 의 면적분

빛이 들어올 수 있는 모든 방향에 대해 전부 더하여면 구를 Pixel(표면)을 감싸는 구(Sphere)의 표면상의 각을 지정하는 방법에 대해서면 빛의 흡수율을 계산할 수 있다.



구의 표면의
인접하는 면적 (d ω_i) = 저점의 지점을 통과해 들어오는
빛의 양

Microfacet(미세표면)

rough Surface

Smooth

Microfacet Specular BRDF - Cook-Torrance Model

$$f(l, v) = \frac{\bar{F}(l, h) G(l, v, h) D(h)}{4(n \cdot l)(n \cdot v)} = \text{Specular}$$

Fresnel Reflectance - 빛을 쪽으로 볼 때 달라지는 반사 정도

$\bar{F}(l, h)$ = 빛의 방향에 따라 색이 바뀌거나 달라진다

The Schlick Approximation to Fresnel, $F_o = 1 - \sqrt{2}$ (ex: 물 = 0.02)

$$F_{\text{Schlick}}(F_o, l, n) = F_o + (1 - F_o)(1 - (l \cdot n))^5$$

- For microfacet BRDFs ($m \approx h$)

$$F_{\text{Schlick}}(F_o, l, h) = F_o + (1 - F_o)(1 - (l \cdot h))^5$$

Normal Distribution Function - roughness에 의한 반사 분포

$D(h)$ = 미세표면(microfacet) 중에서 봄 방향이 Normal인 표면의 비율 (봄 방향으로 빛을 반사해줄 확률은 $1 - e^{-\pi h^2}$)
 (미세표면들은 거울처럼 반사하는 경우는 제거 가능하다.)
 (Specular와 highlight를 결정해준다)

= Normal의 h 인 비율

최근에 많이 사용하는 $D(h)$ 는 GGX다

$$D_{GGX}(n, h, \alpha) = \frac{\alpha^2}{\pi ((n \cdot h)^2 (\alpha^2 - 1) + 1)^2}, \quad \begin{array}{l} n = \text{normal} \\ h = \text{half way} \\ \alpha = \text{roughness}^2 \end{array}$$

Geometry Function - 미세표면 그림자

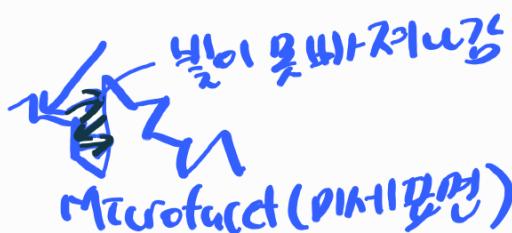
$G(l, v, h) =$ 미세표면의 Shadowing과 Masking을 표현하는 함수

Shadowing : 미세표면에서 다른 미세표면에의 빛이 빠져들어오지 못해 그림자가 생기는 부분



Microfacet(미세표면)

Masking : 빛이 둘러싸인 반사면에 빛이 빠져나오지 못하는 부분



Microfacet(미세표면)

많이 쓰는 함수(Smith function)

$$G_s(l, v, h) = G_{s1}(l, h) G_{s2}(v, h)$$

G : Smith's Schlick-GGX 함수

$$k = \frac{(\text{Roughness} + 1)^2}{8}$$

$$G_{s1}(v) = \frac{n \cdot v}{(n \cdot v)(1-k) + k}$$

$$G_s(l, v, h) = G_{s1}(l) G_{s1}(v)$$

즉, Specular은

$$L_o(v) = \int_{\Omega} f_{BRDF}(l, v) \otimes L_i(l)(n \cdot l) d\omega_i, \text{ 가중치 } -$$

Diffuse

Lambert

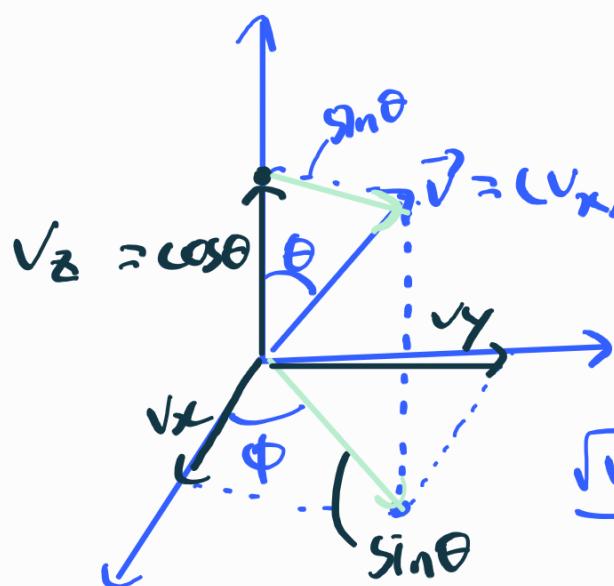
$$f_{\text{Lambert}}(I, v) = \frac{C_{\text{diff}}}{\pi}, \quad (C_{\text{diff}} = \text{fraction of light reflected or diffuse color})$$

즉, Diffuse는

$$L(v) = \int_{\Omega} f_{\text{Lambert}}(I, v) \otimes L_i(I)(n \cdot I) d\omega_i, \quad \text{가 된다}$$

(이제까지 본 모든 빛은 모두 빛을 나누는 것 같다)

3 차원에서 변환



$$\|\vec{v}\| = 1, \text{unit vector}$$

$$v_x = \sin \theta \cos \phi$$

$$v_y = \sin \theta \sin \phi$$

$$v_z = \cos \theta, \quad v \text{가 unit vector} \\ \text{이므로 } v_z = \cos \theta$$

$$\frac{v_y}{v_x} = \tan \phi, \quad \phi = \arctan \left(\frac{v_y}{v_x} \right)$$

$$\frac{\sqrt{v_x^2 + v_y^2}}{v_z} = \tan \theta, \quad \theta = \arctan \left(\frac{\sqrt{v_x^2 + v_y^2}}{v_z} \right)$$

$$\pm \arccos(v_z)$$

Spherical \rightarrow Cartesian

$$\begin{cases} v_x = \cos \phi \sin \theta \\ v_y = \sin \phi \sin \theta \\ v_z = \cos \theta \end{cases}$$

$$\sqrt{v_x^2 + v_y^2 + v_z^2} = 1$$

$$\begin{cases} \phi = \arctan \left(\frac{v_y}{v_x} \right) \\ \theta = \arctan \left(\frac{\sqrt{v_x^2 + v_y^2}}{v_z} \right) = \arccos \left(\frac{v_z}{\sqrt{v_x^2 + v_y^2 + v_z^2}} \right) \end{cases}$$