

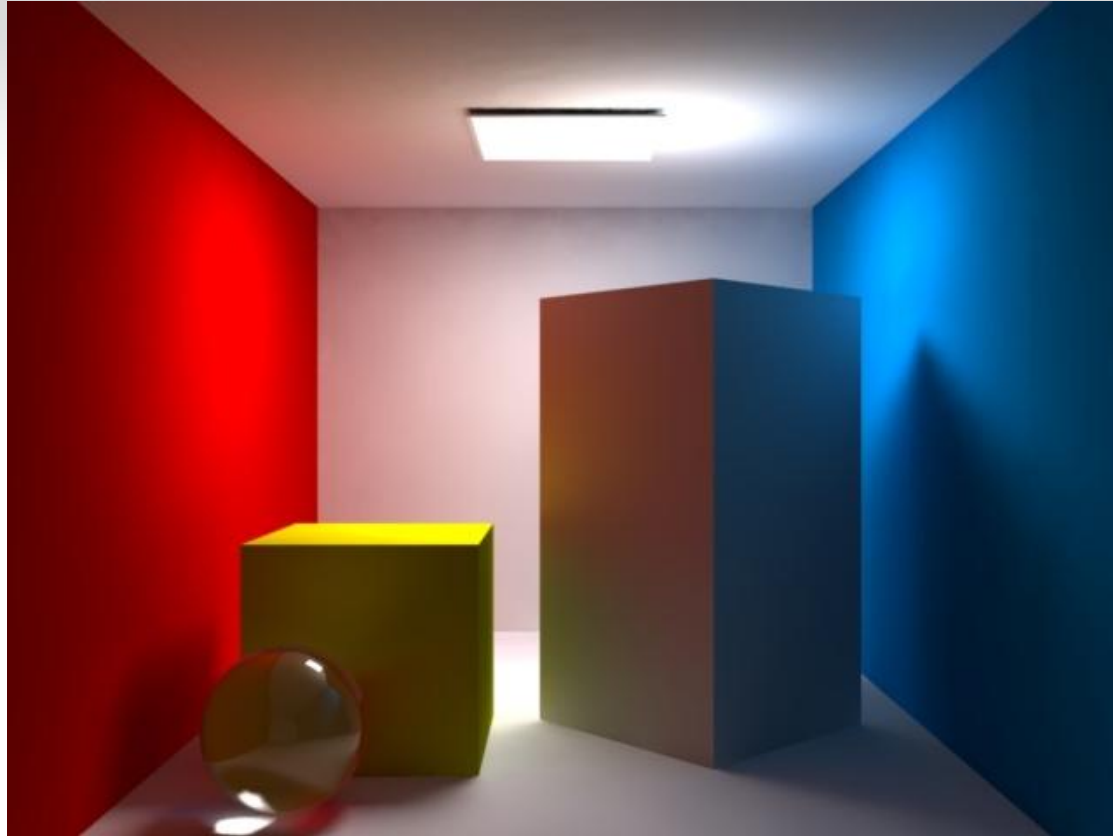
Iluminación y Sombreado

Luz y Superficies



La apariencia de los objetos depende de la interacción de las superficies con la luz

Rendering

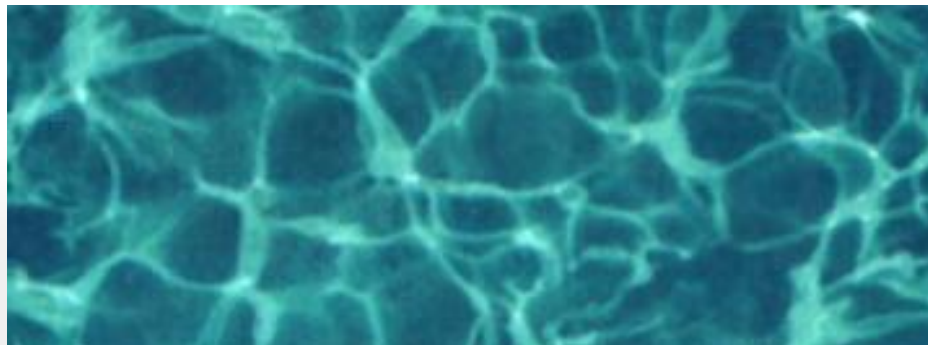
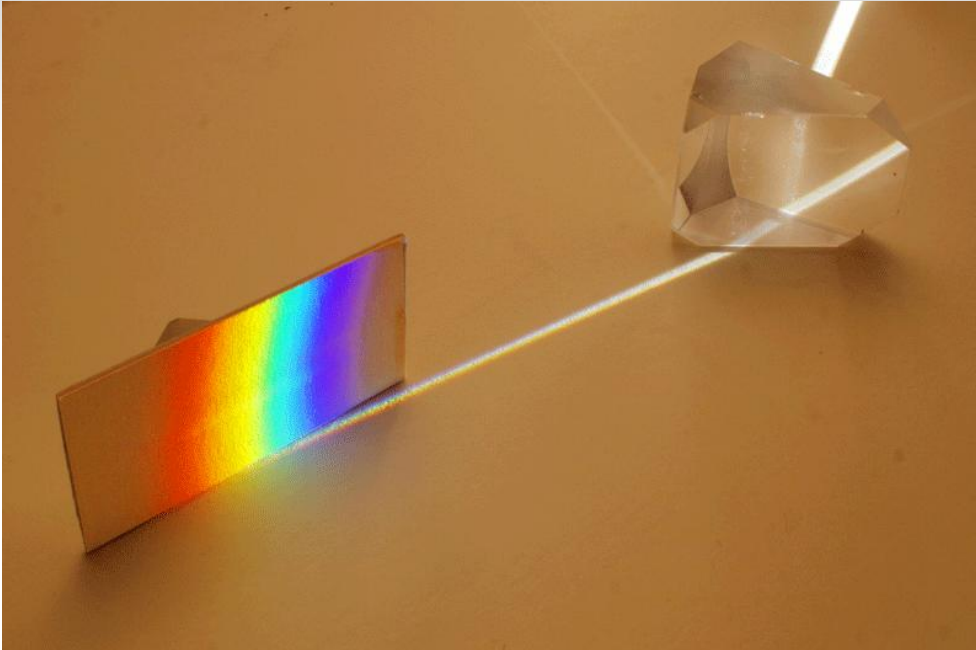


El proceso de **rendering** consiste en simular dichas interacciones.

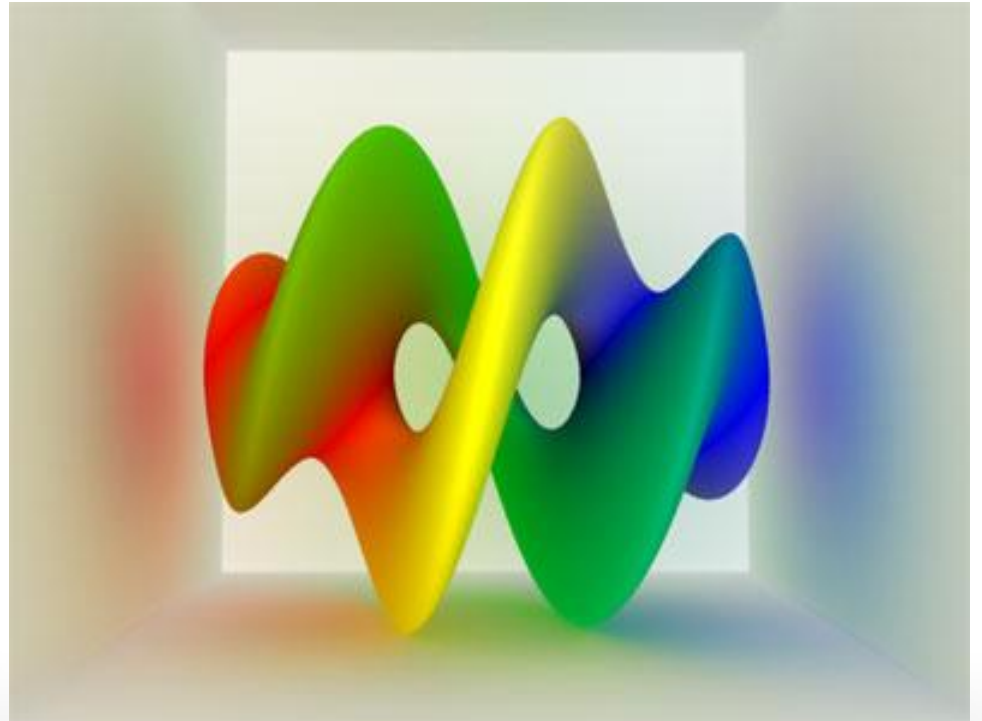
Fenómenos físicos - Reflexión



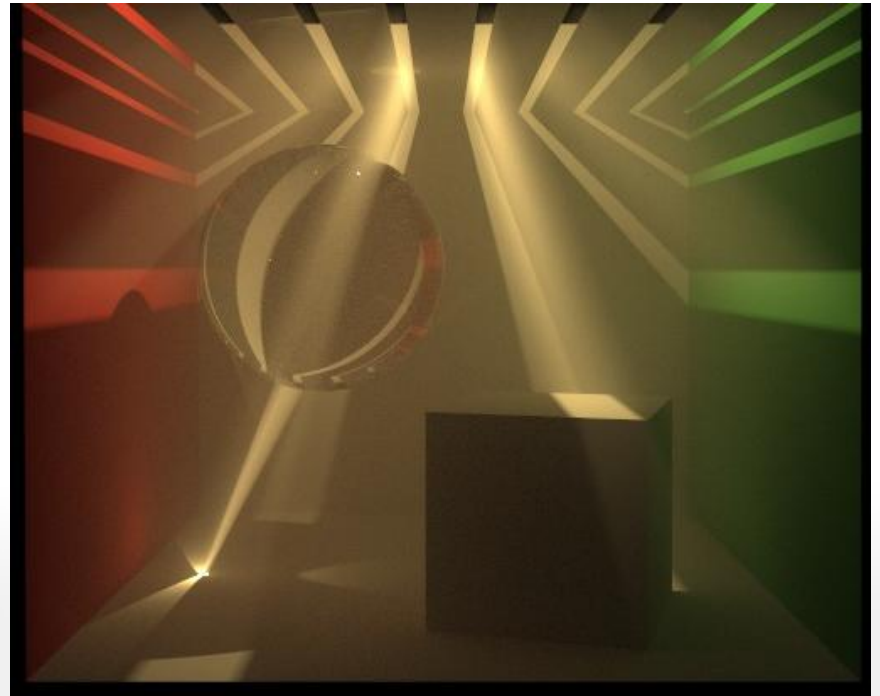
Fenómenos físicos - Refracción



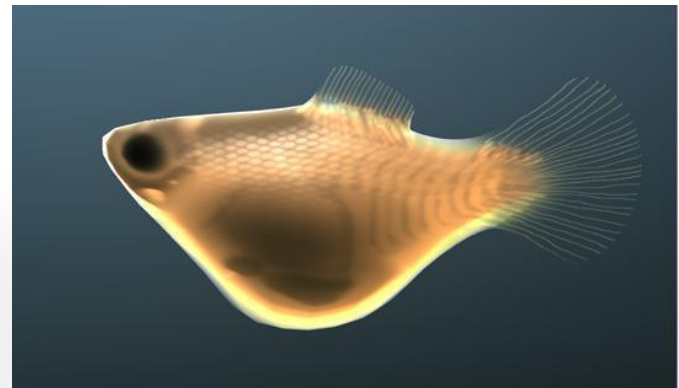
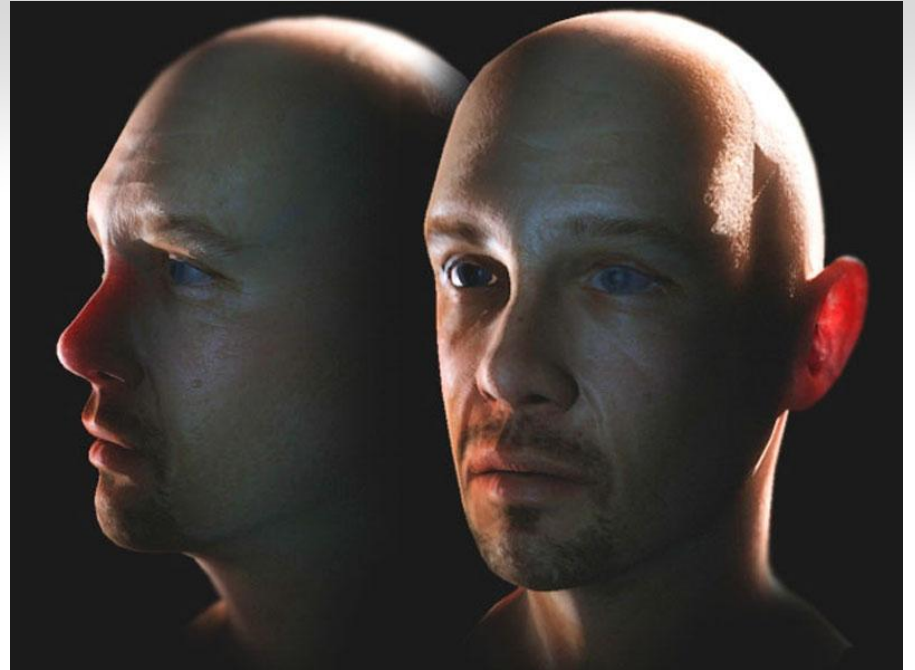
Interreflexiones



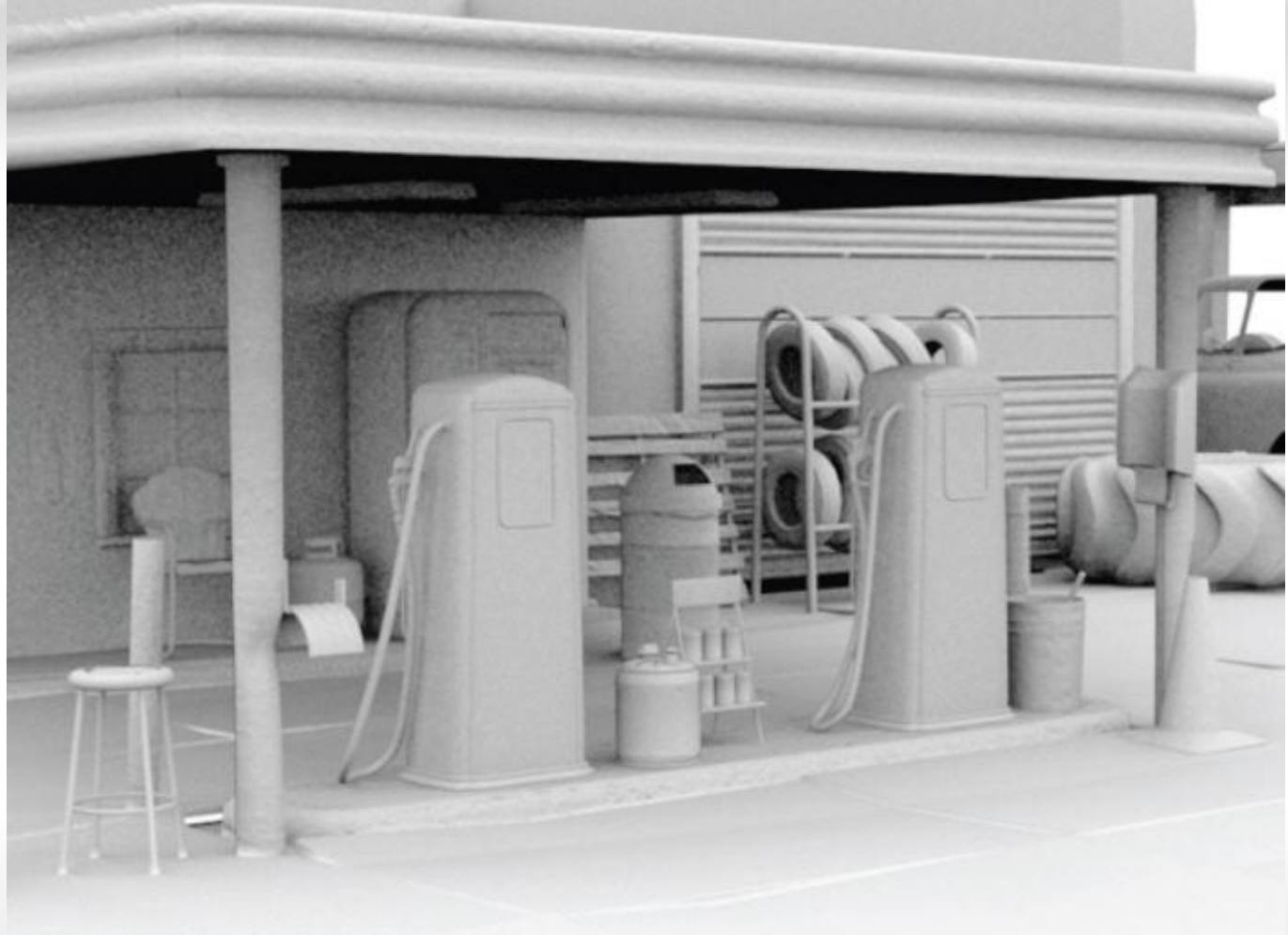
Efectos atmosféricos



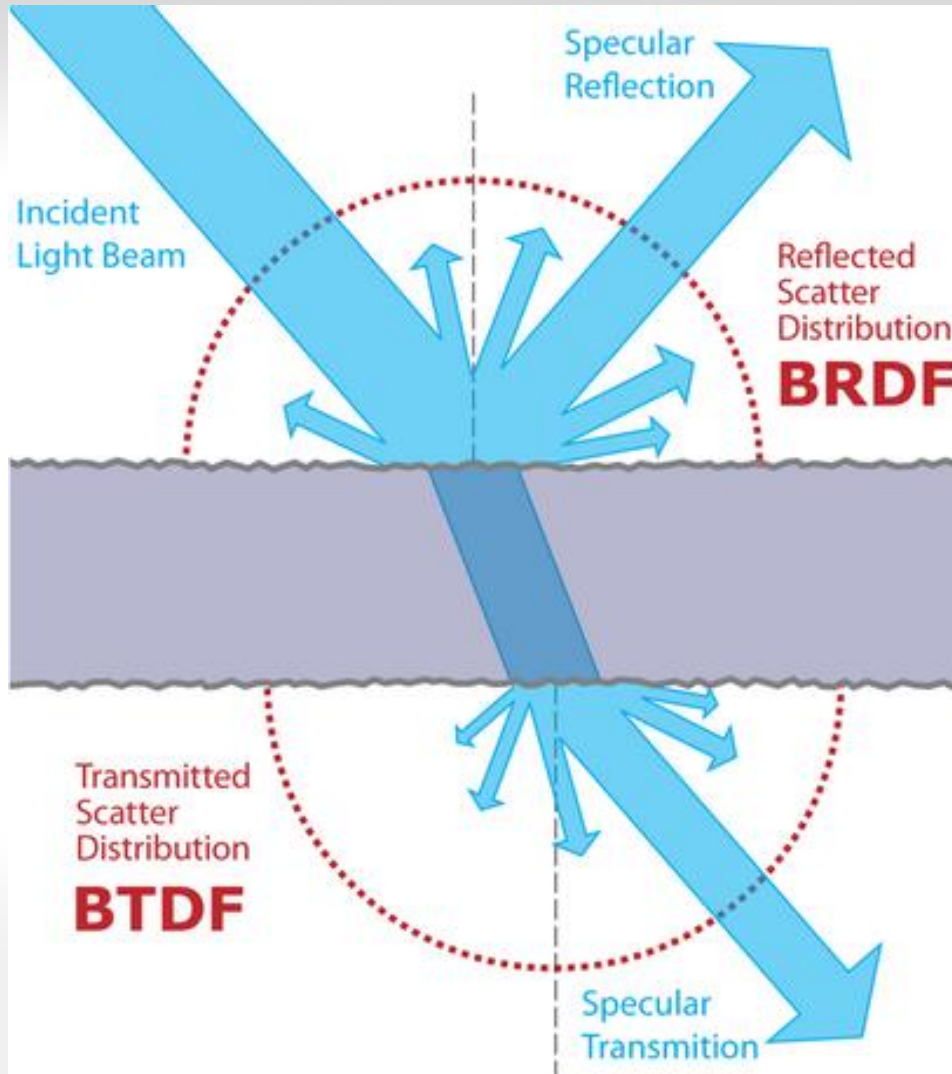
Sub-Surface Scattering



Ambient Occlusion



Función BSDF



**Bidirectional
Scattering
Distribution
Function**

Tipos de algoritmos



Iluminación **directa**



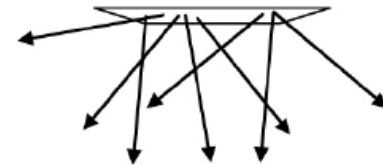
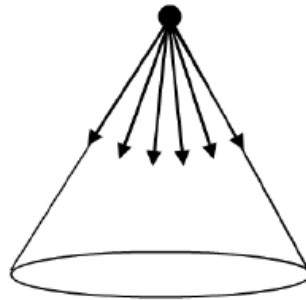
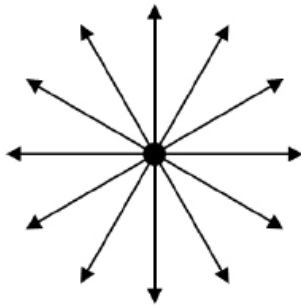
Iluminación **global**

Aproximación de iluminación

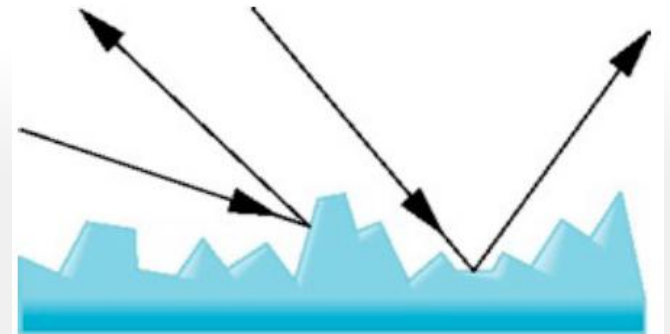
- Considerar **solo** 3 longitudes de onda



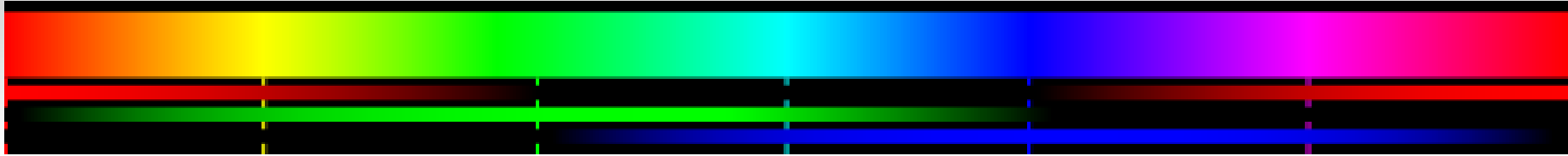
- Fuentes de luz idealizadas



- Aproximación de materiales



Longitudes de onda

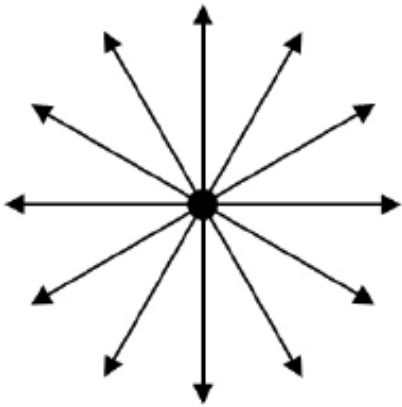


- No se considera el espectro completo. **Solo** se computan las frecuencias: Roja, Verde y Azul

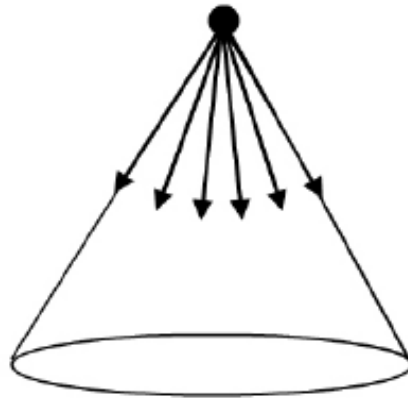


- Cada luz y material tendrá parámetros separados para cada canal (RGB)
- Los 3 canales de color se computan en paralelo

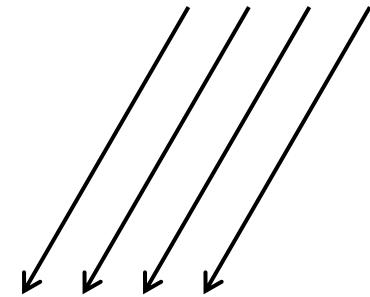
Fuentes de luz idealizadas



Fuente puntual



Fuente spotlight



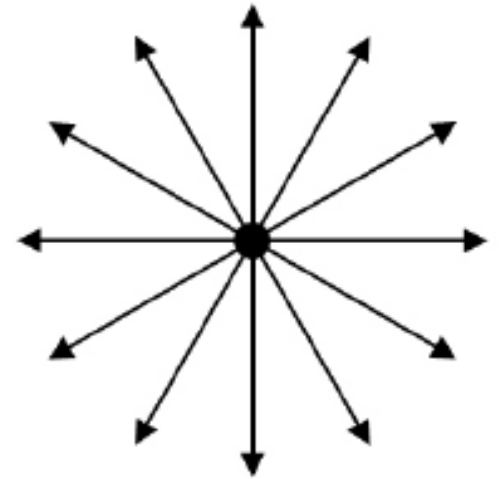
Fuente directa

- En la realidad los fotones son emitidos por objetos de geometrías diversas
- Solo consideramos fuentes simples

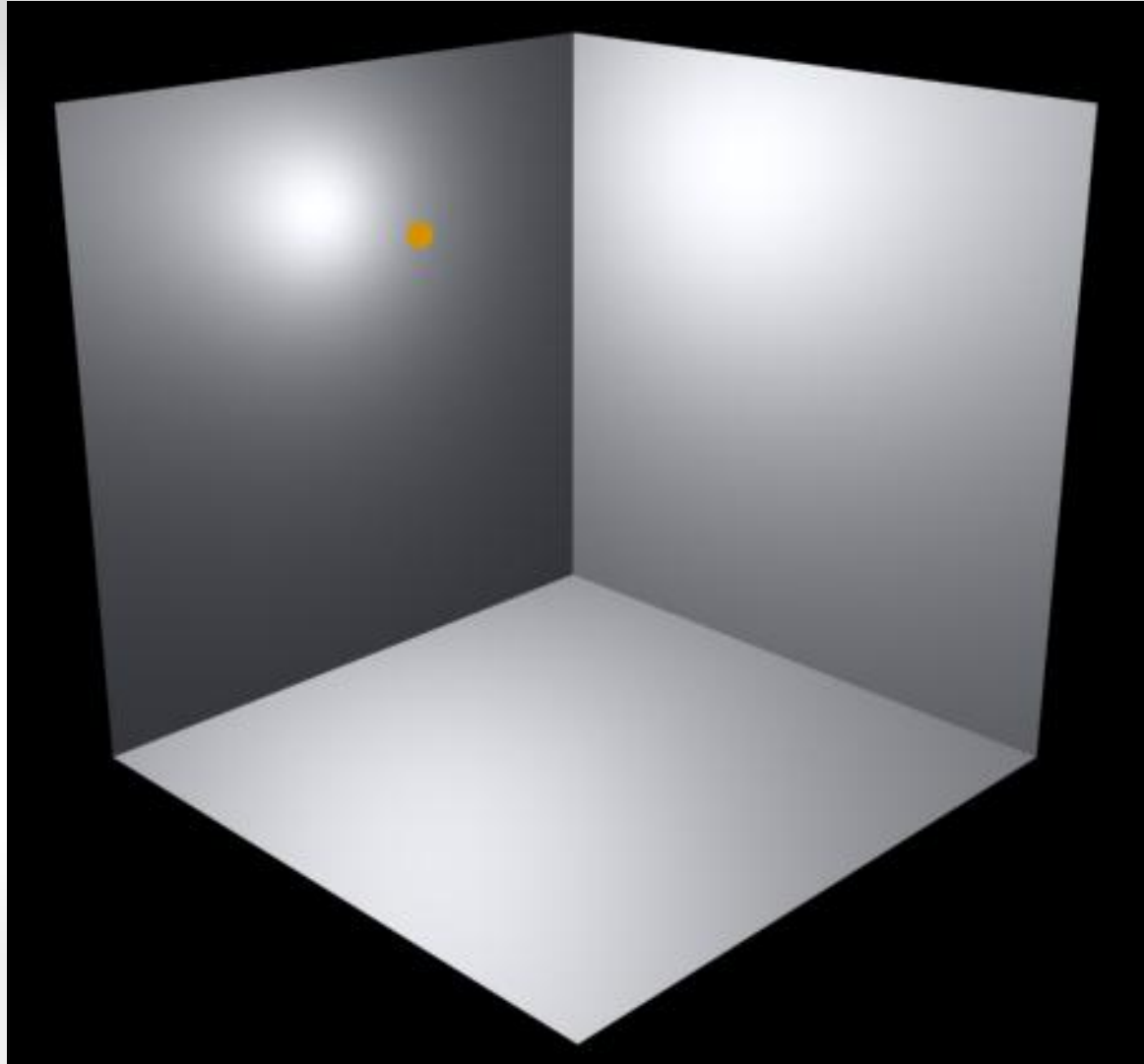
Luces puntuales

- Emite luz de igual intensidad en todas las direcciones
- En la naturaleza, la intensidad decae con cuadrado de la distancia
- En la practica se utiliza en factor de decaimiento:

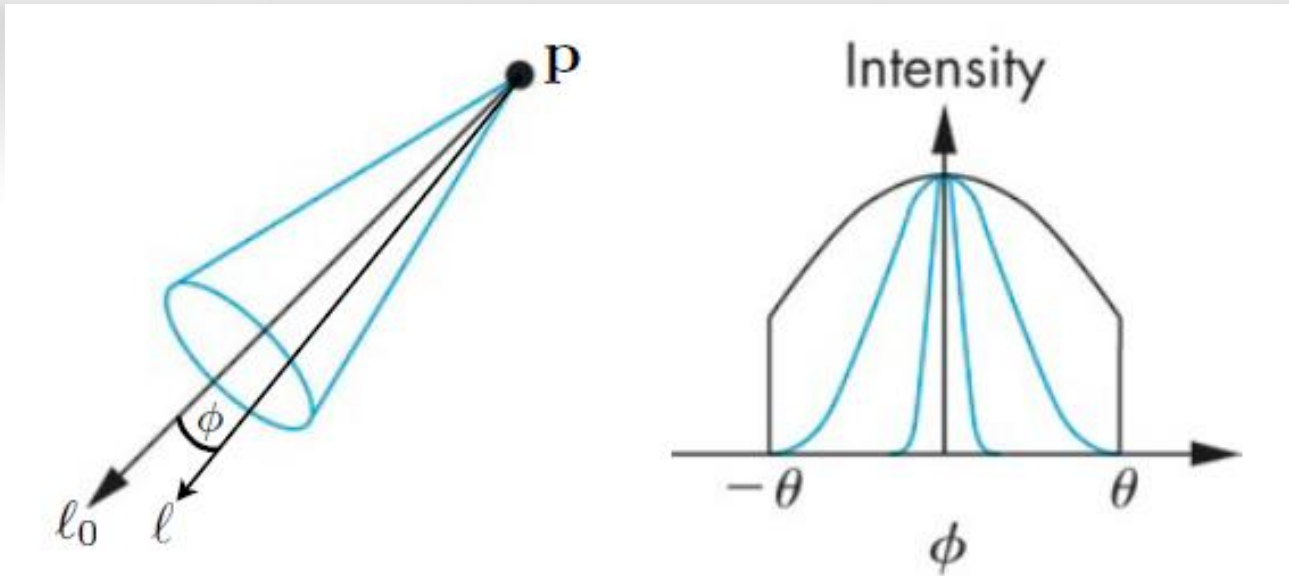
$$\frac{1}{ad^2 + bd + c} L_p$$



Luces puntuales



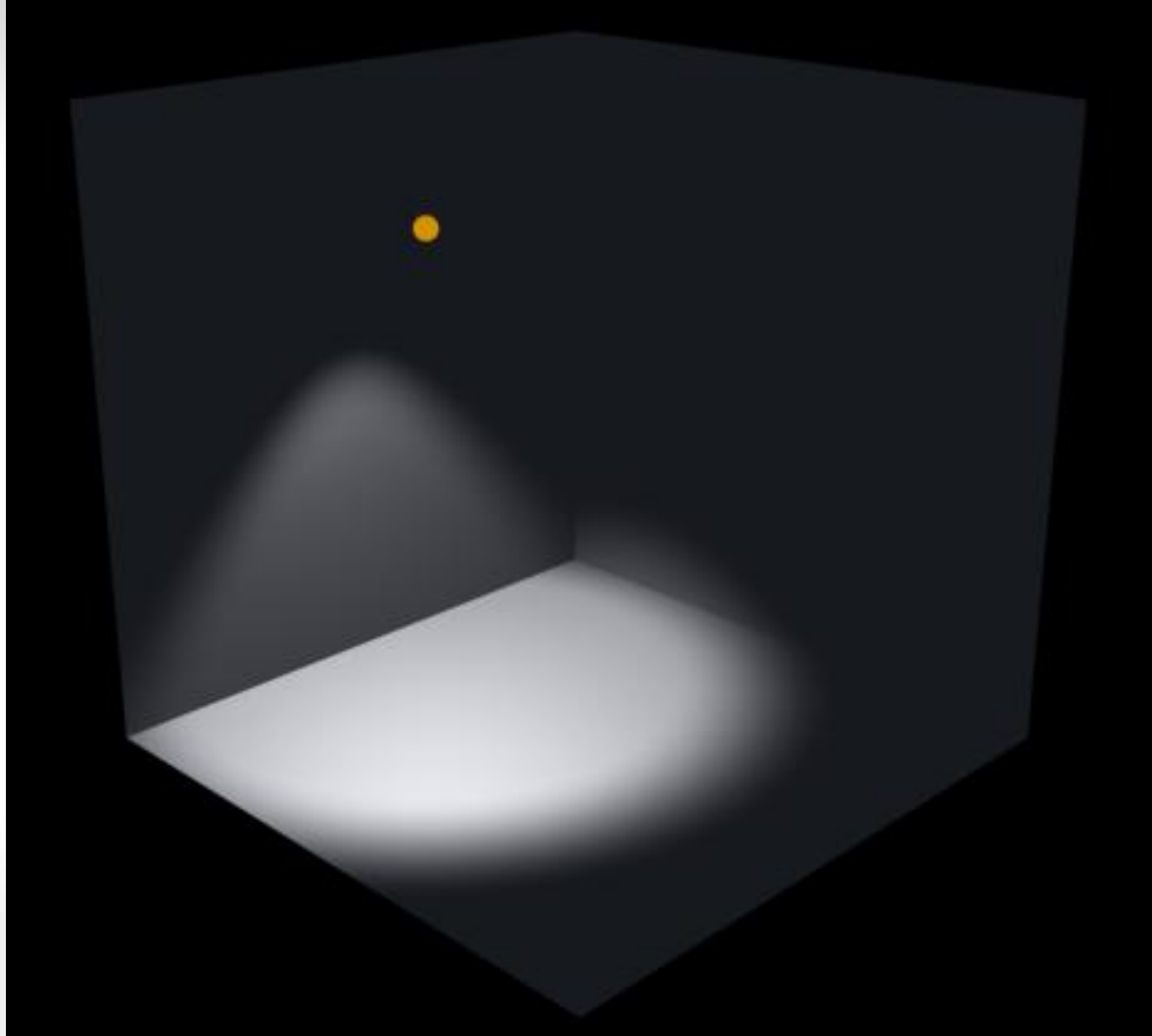
Luces tipo Spot



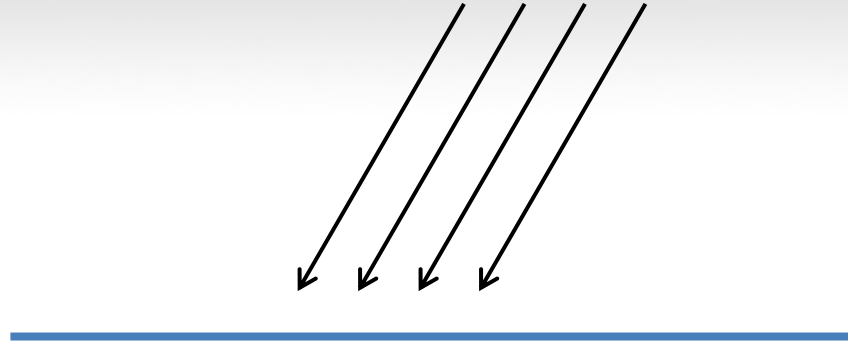
- Propiedades: dirección, ángulo máximo
- Decaimiento angular de Intensidad

$$\cos^a(\phi) = (\ell_0 \cdot |\ell|)^a$$

Luces tipo spot

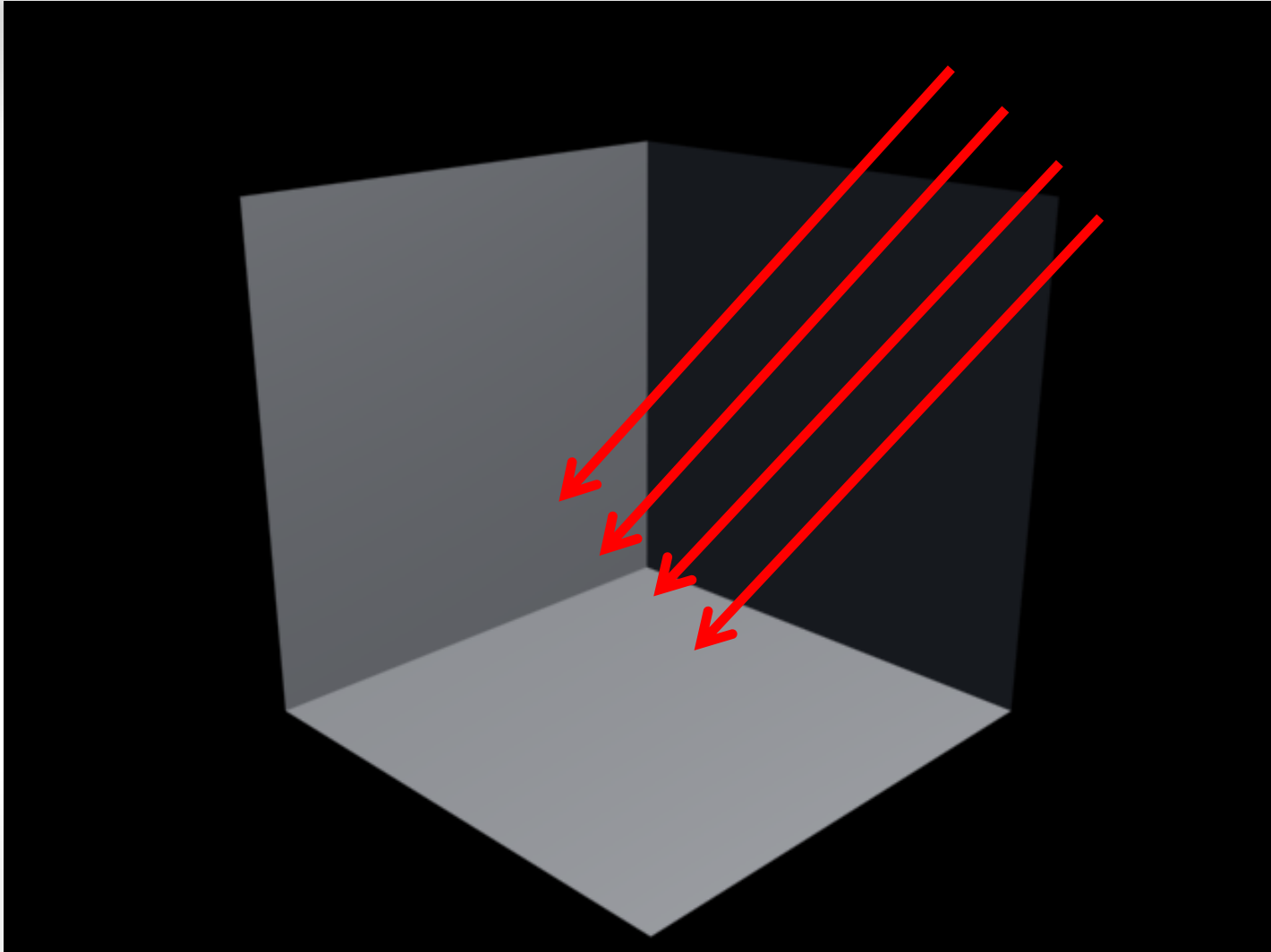


Luces direccionales

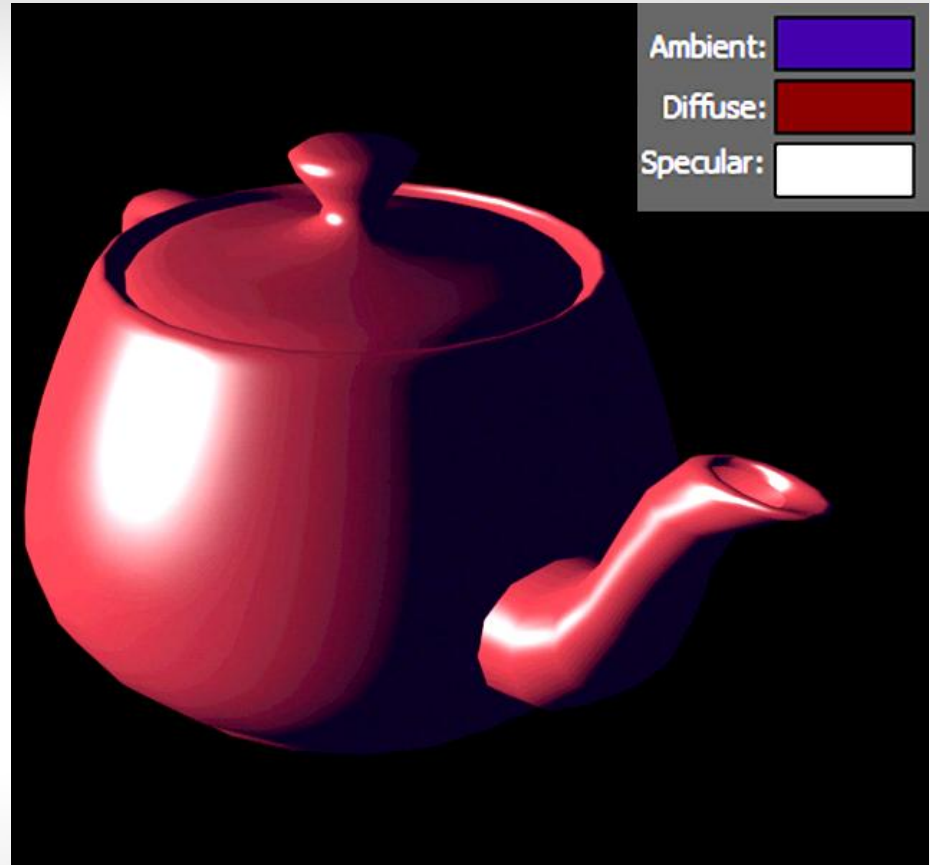
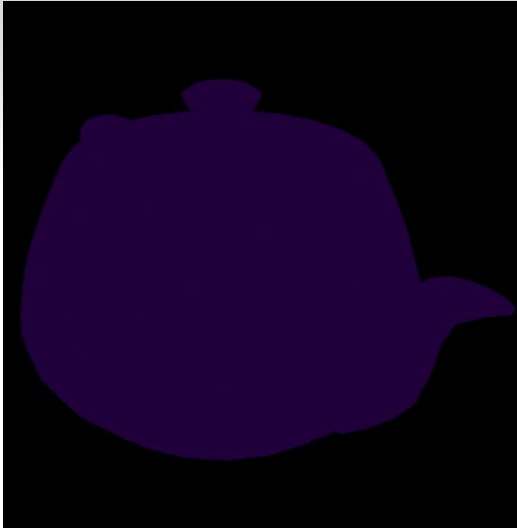


- Propiedades: dirección
- Simula luces distantes (Sol)
- No decae con la distancia
- El sombreado solo depende de la orientación

Luces direccionales

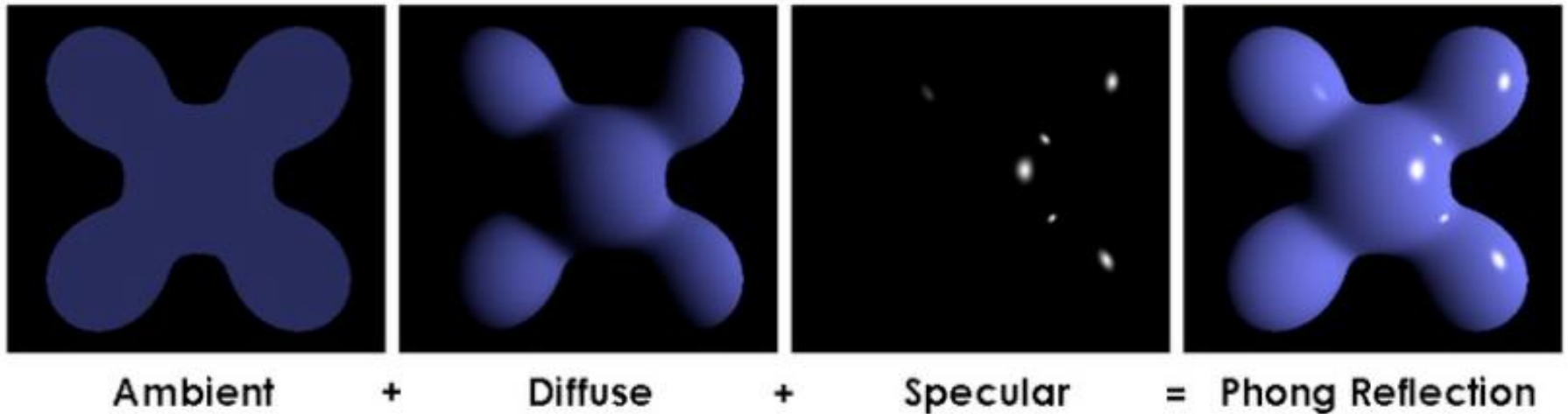


Modelo de Phong



Computa iluminación directa

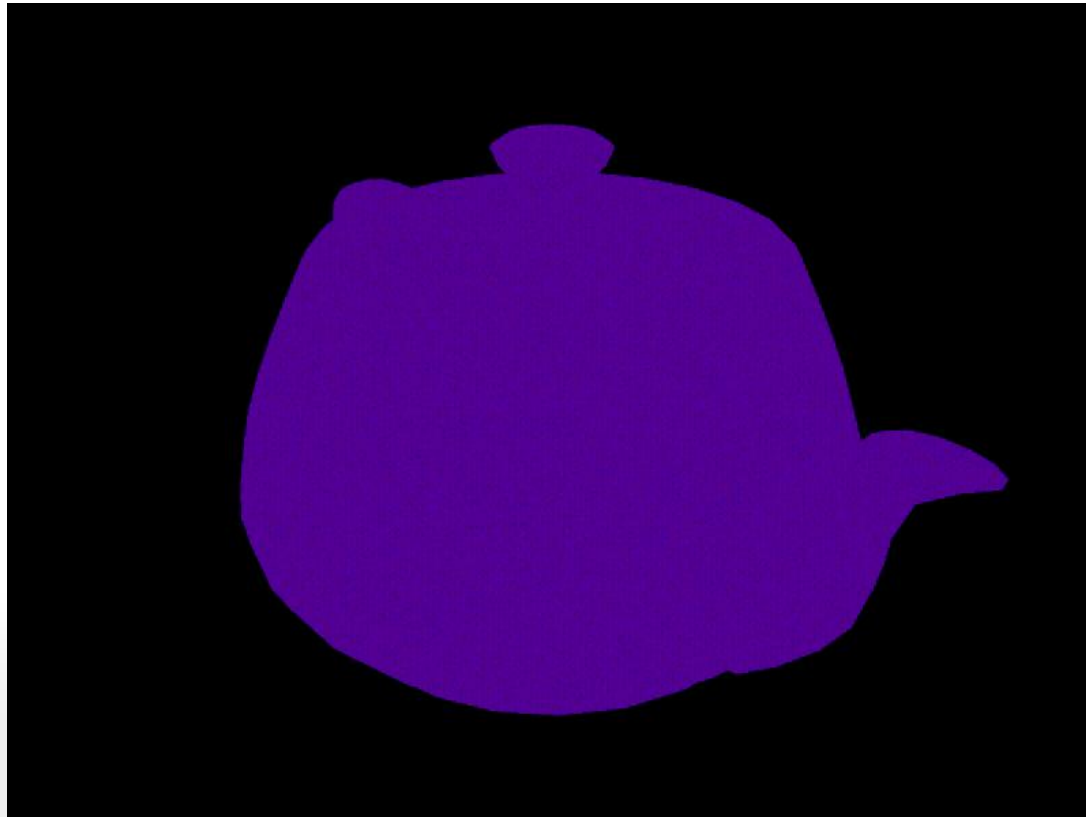
Modelo de Phong



$$I = I_a + I_d + I_s = R_a L_a + R_d L_d + R_s L_s$$

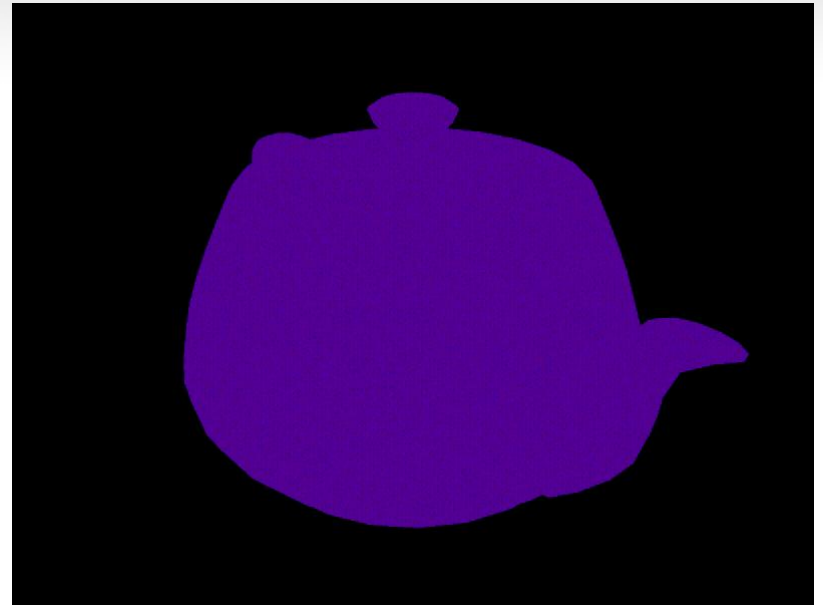
Luz ambiente

- Aproximación gruesa de efecto de dispersión
- Es un escalar que simula iluminación general



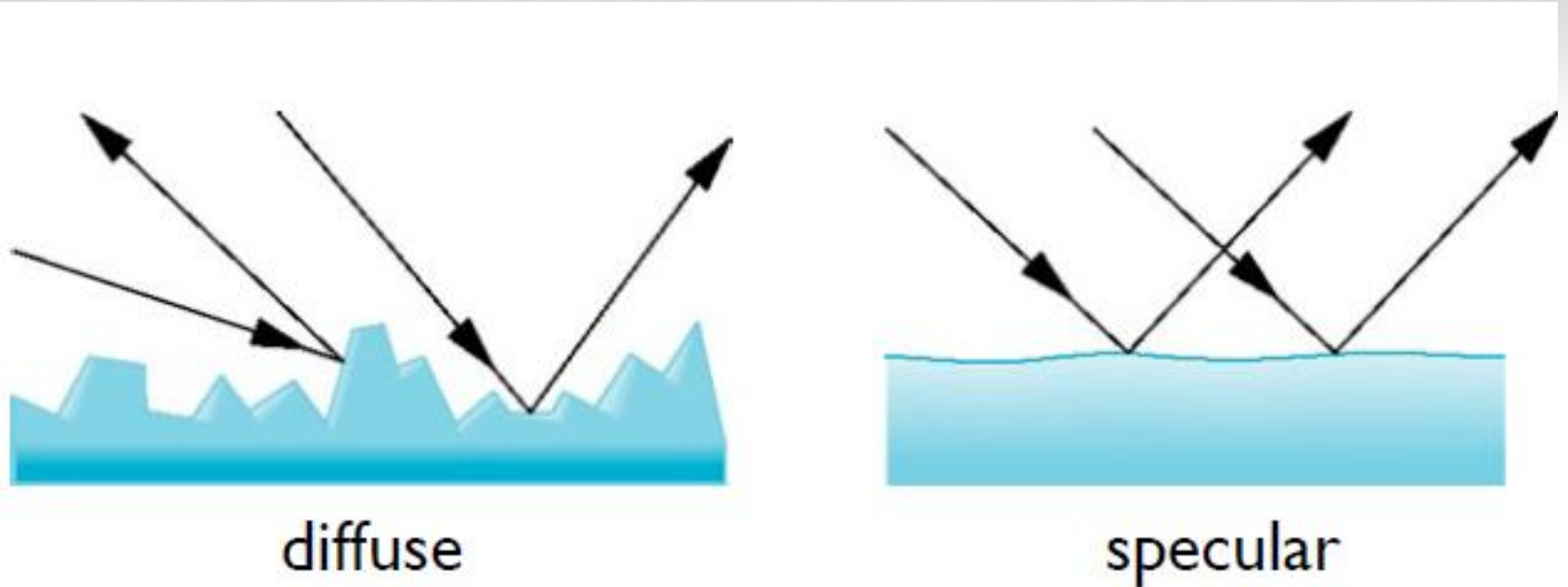
Reflexión de luz ambiente

$$R_a = k_a, \quad 0 \leq k_a \leq 1$$



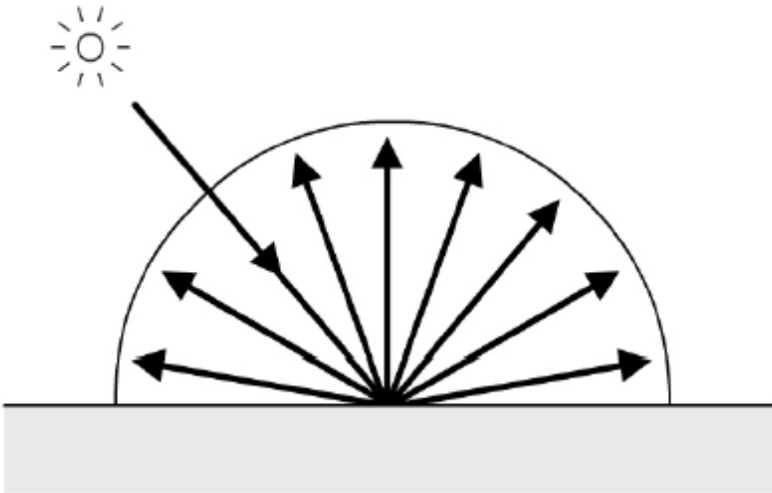
- Es un termino constante que depende del material

Aproximación de Materiales

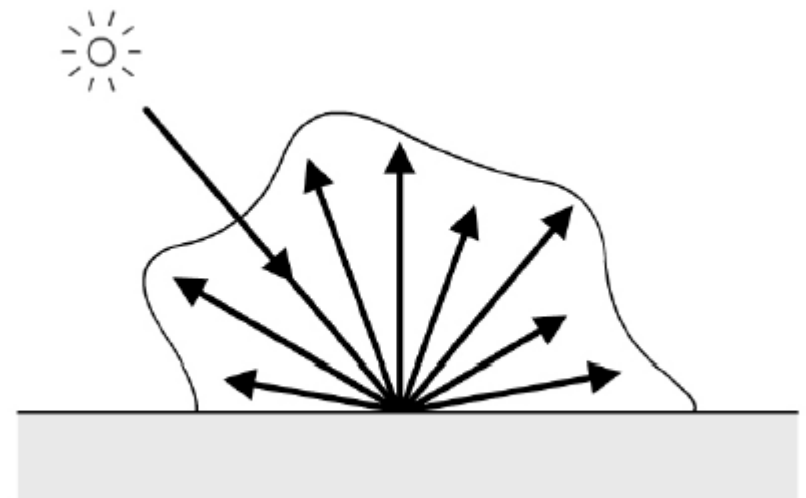


- Comportamiento como combinación de 2 componentes:
 - Difusa
 - Especular

Componente difusa

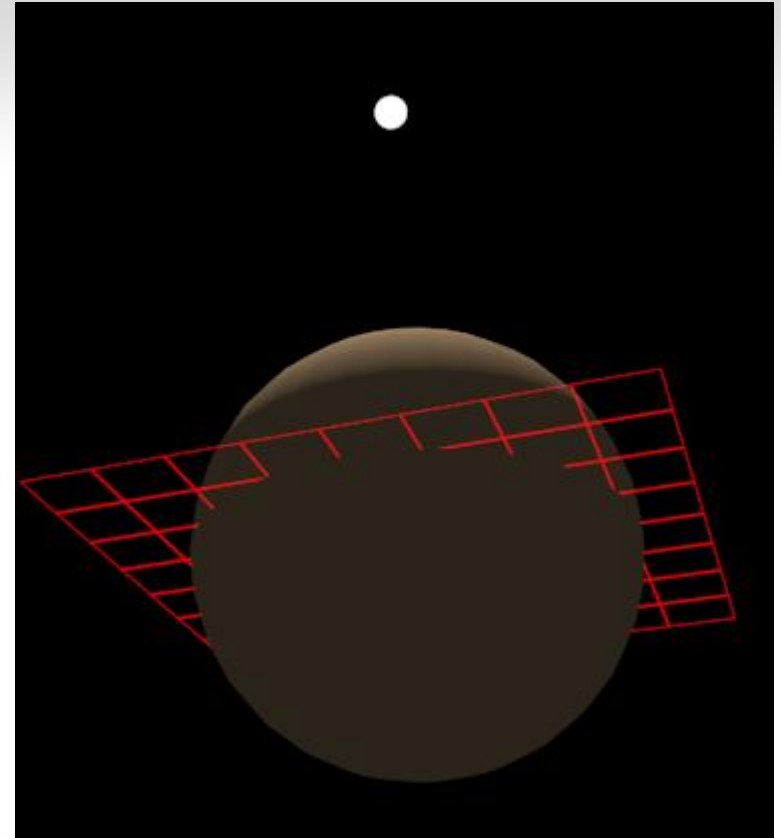
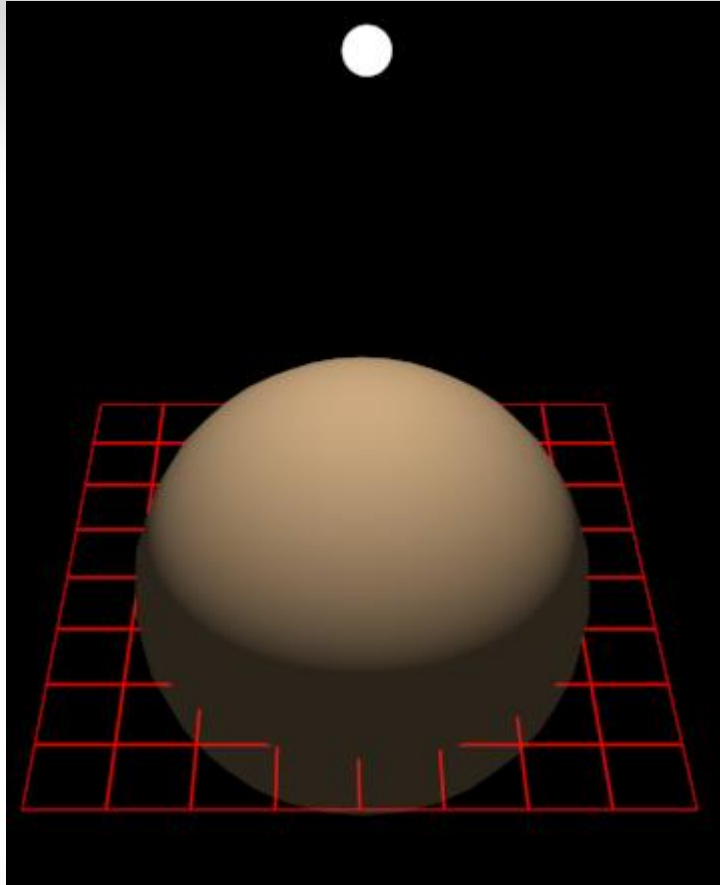


Reflexión difusa ideal
o Lambertiana



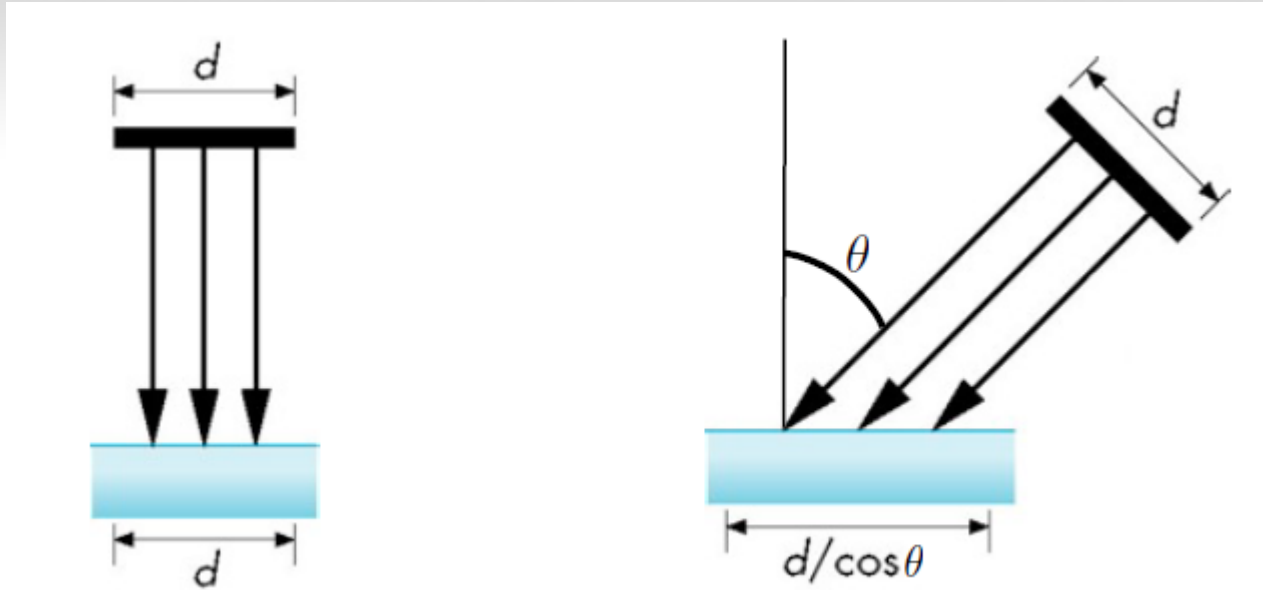
Reflexión difusa real

Reflexión difusa



- Independiente del ángulo de vista

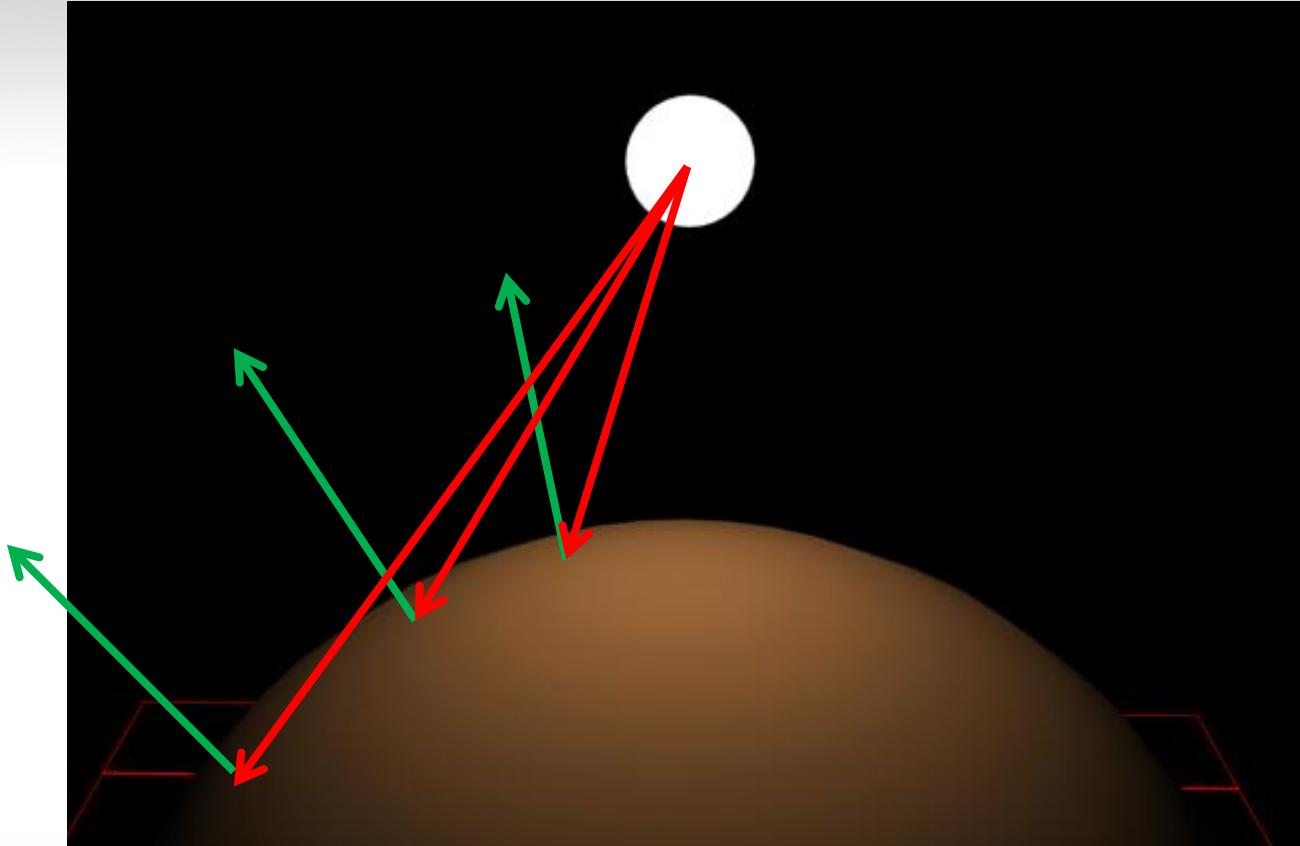
Reflexión difusa



- Ley de Lambert

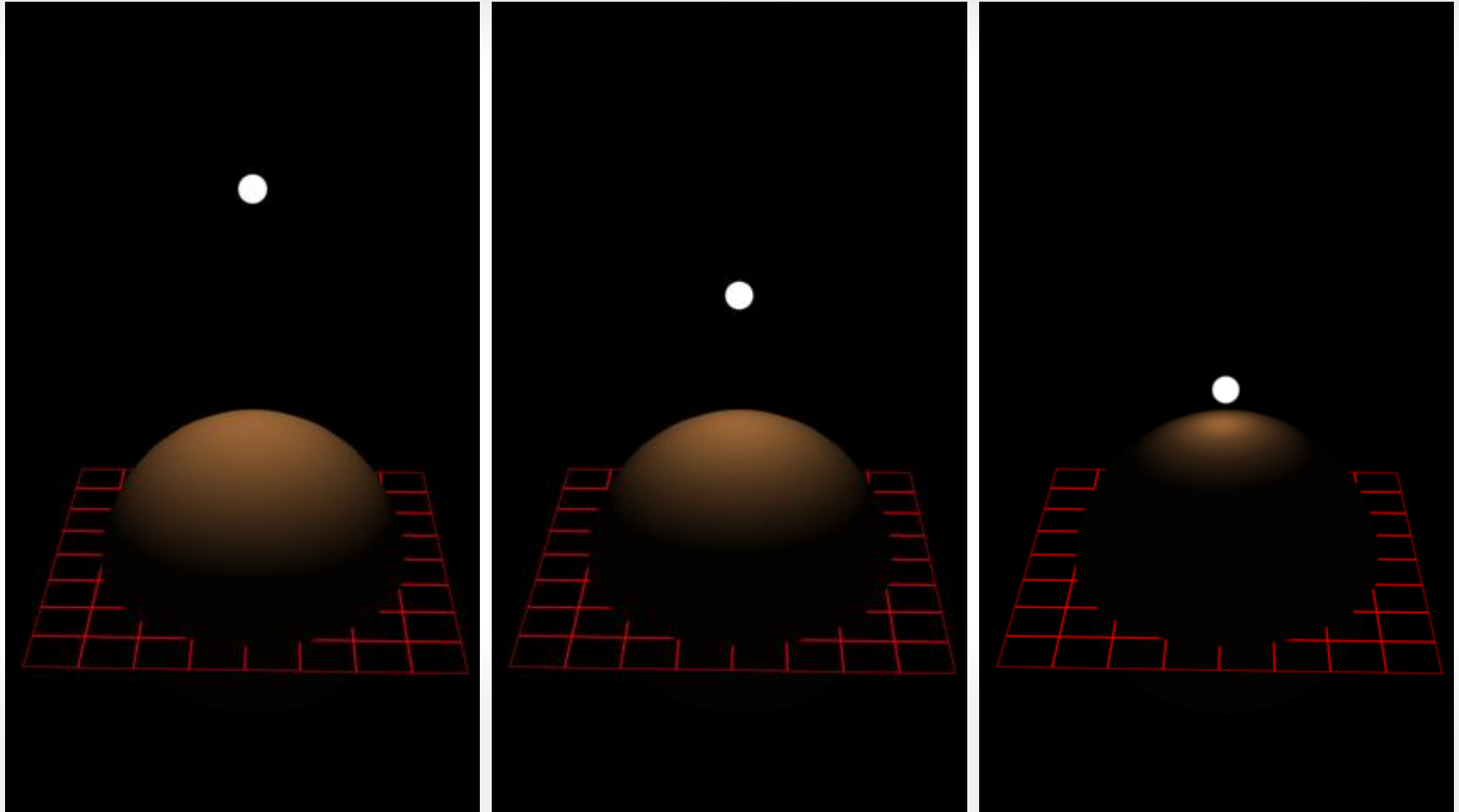
$$R_d = k_d \cos \theta = k_d(\mathbf{l} \cdot \mathbf{n})$$

Reflexión difusa

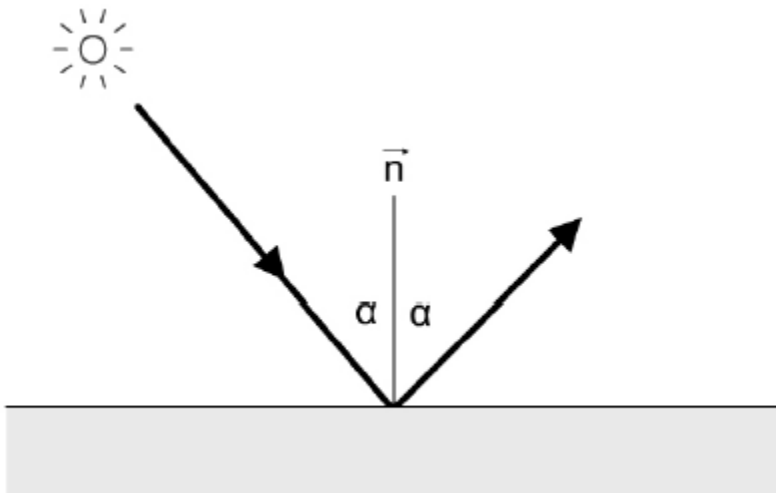


Solo depende del ángulo de incidencia de la luz

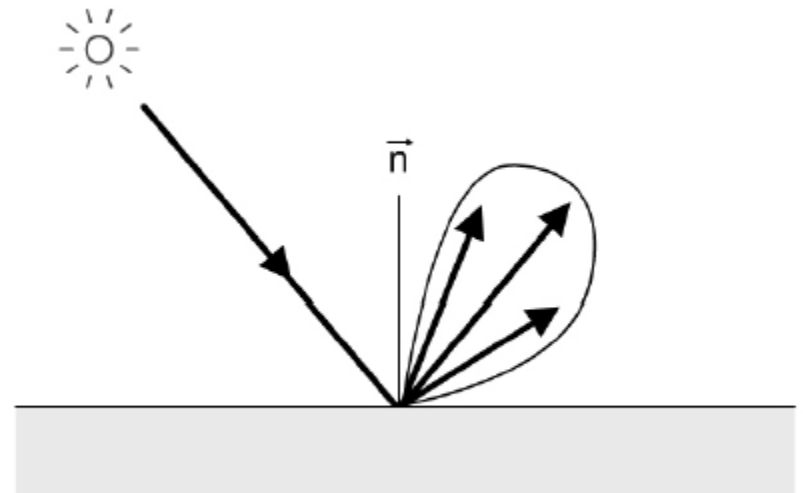
Reflexión difusa



Componente especular

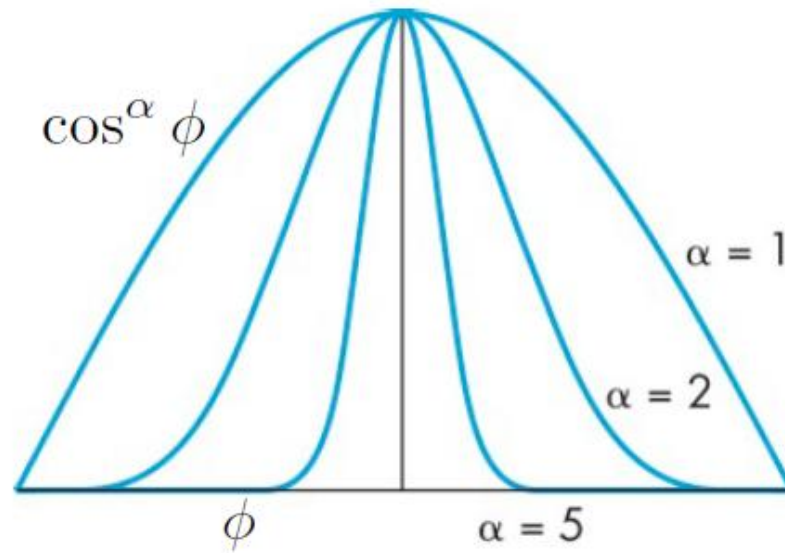


Reflexión especular perfecta
(Ej: espejo)



Reflexión especular “glossy”
(Ej: plástico brillante)

Reflexión especular



α factor “glossiness”

$$R_s = k_s \cos^\alpha \phi = k_s (\mathbf{r} \cdot \mathbf{v})^\alpha$$

- Depende del ángulo de vista
- Depende del ángulo de incidencia de la luz

Reflexión especular - glossiness

$\alpha =$ 1 5 20 100 500



+ glossiness →

Reflexión especular - K_s

$K_s = (1,1,1)$

0.8

0.6

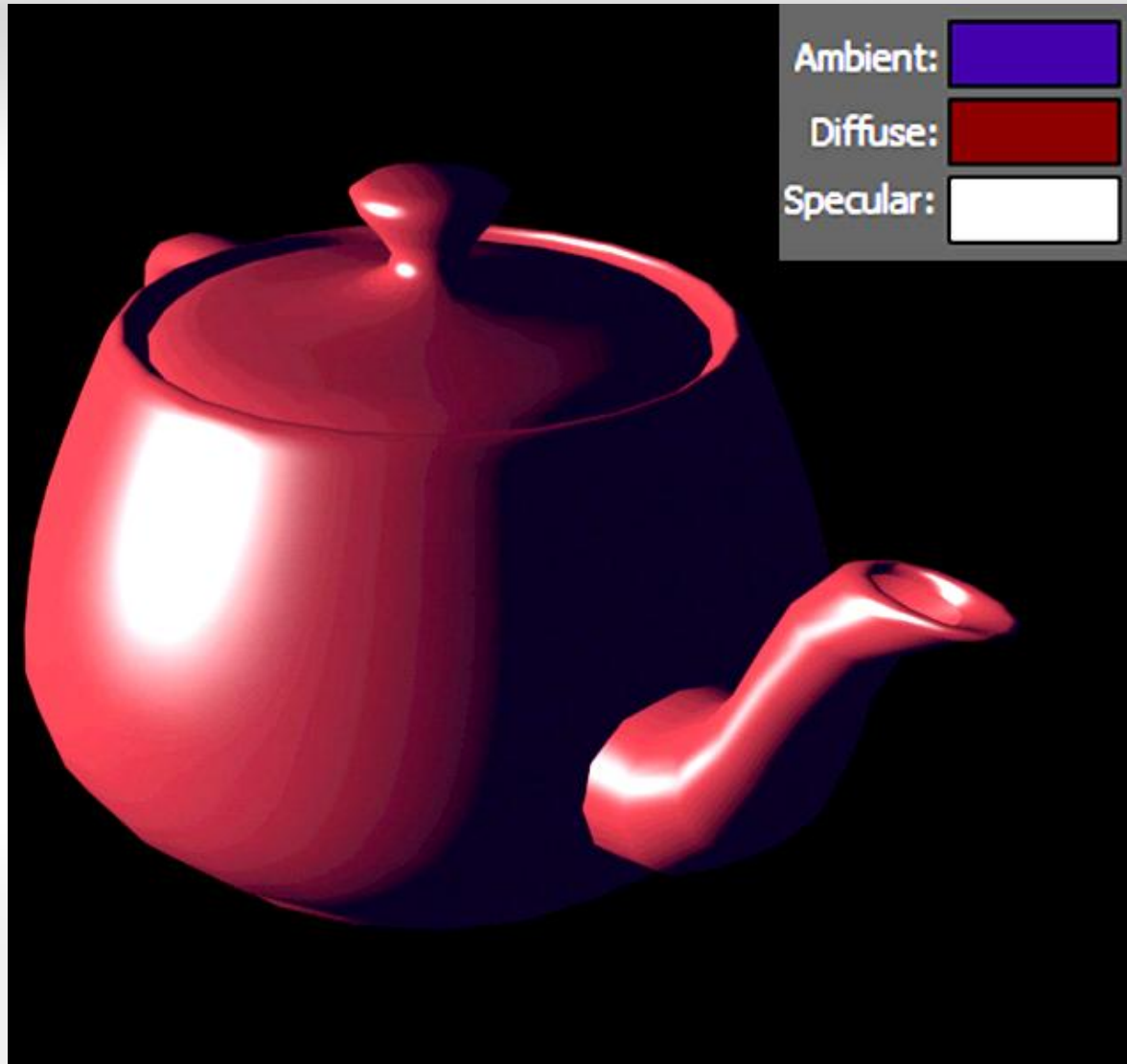
0.4

0.2



- K_s →

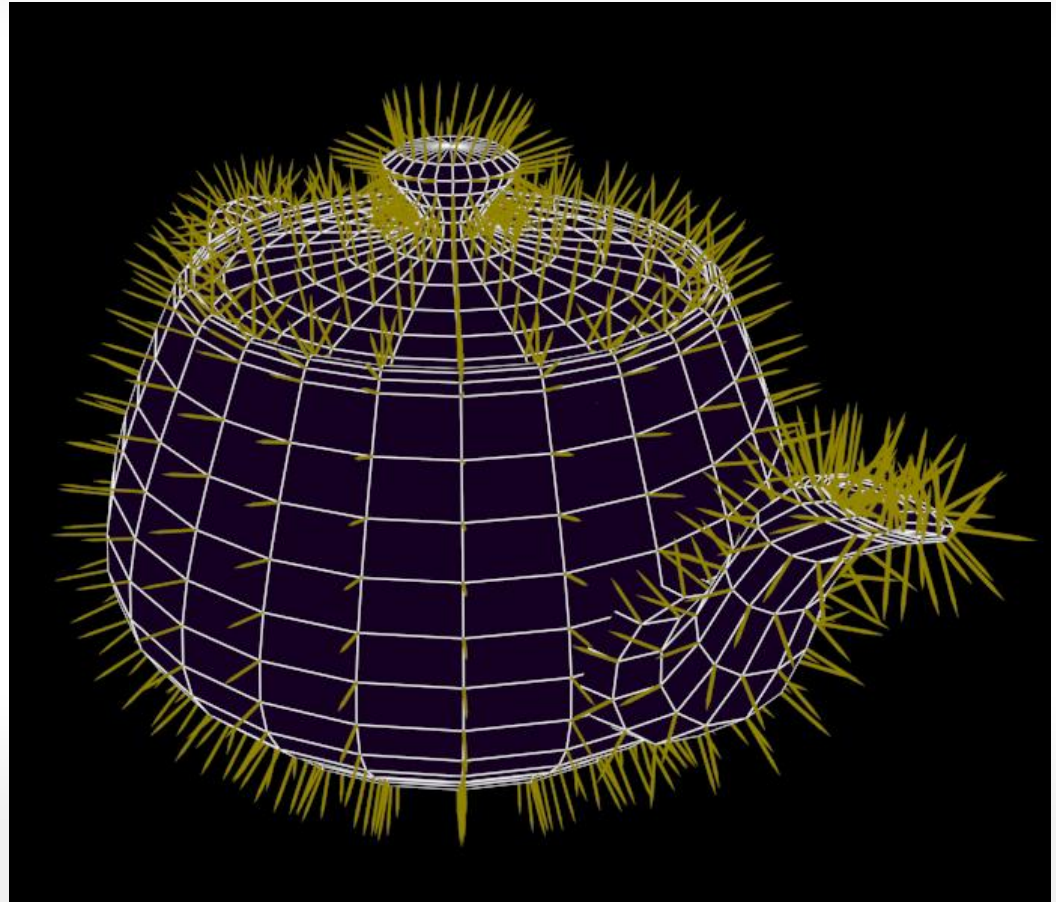
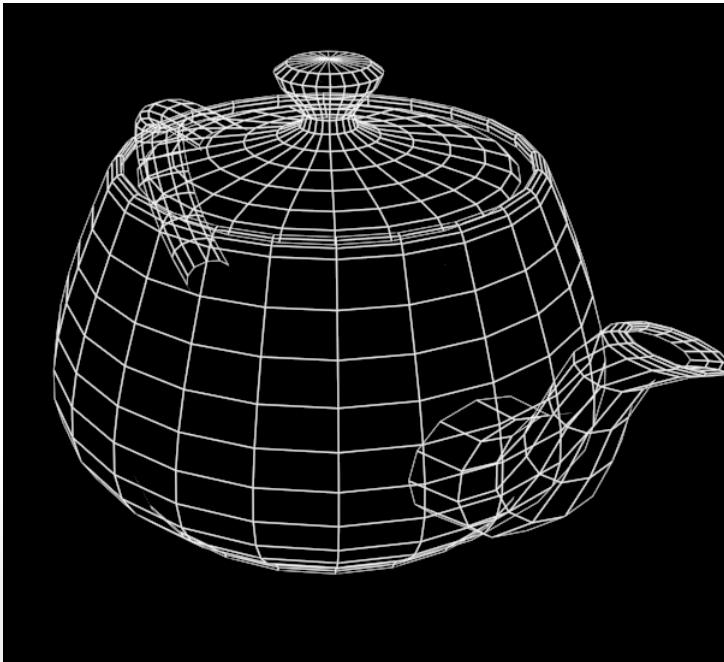
Ambiente + Difusa + Especular



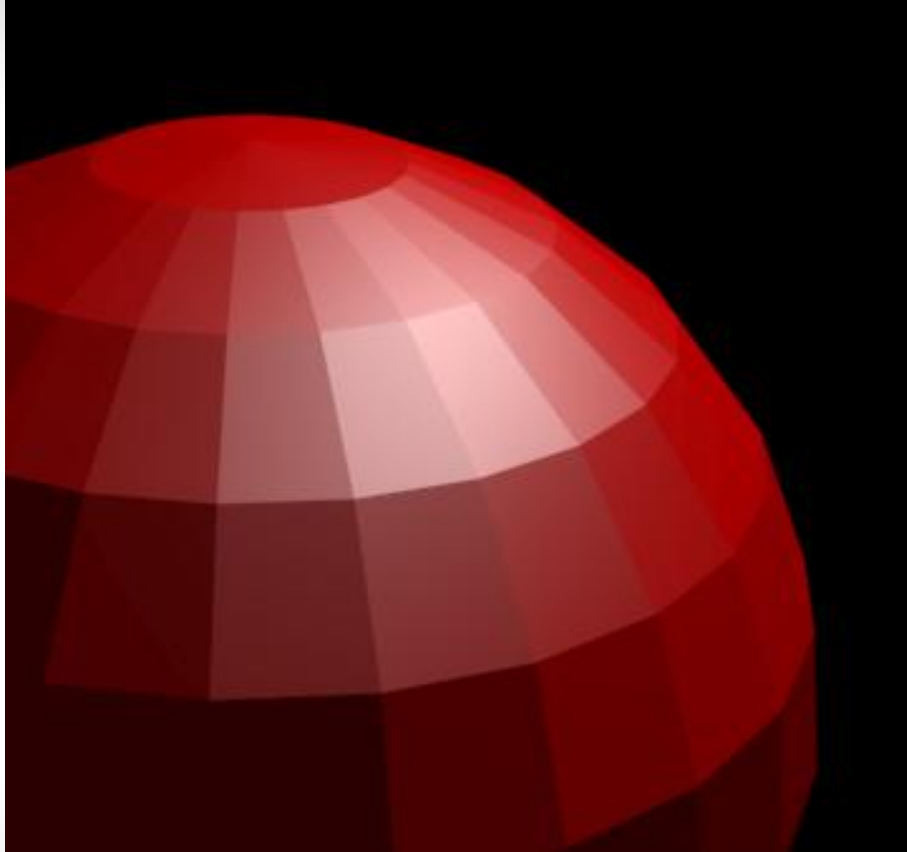
Phong y Normales

Vértice : X,Y,Z

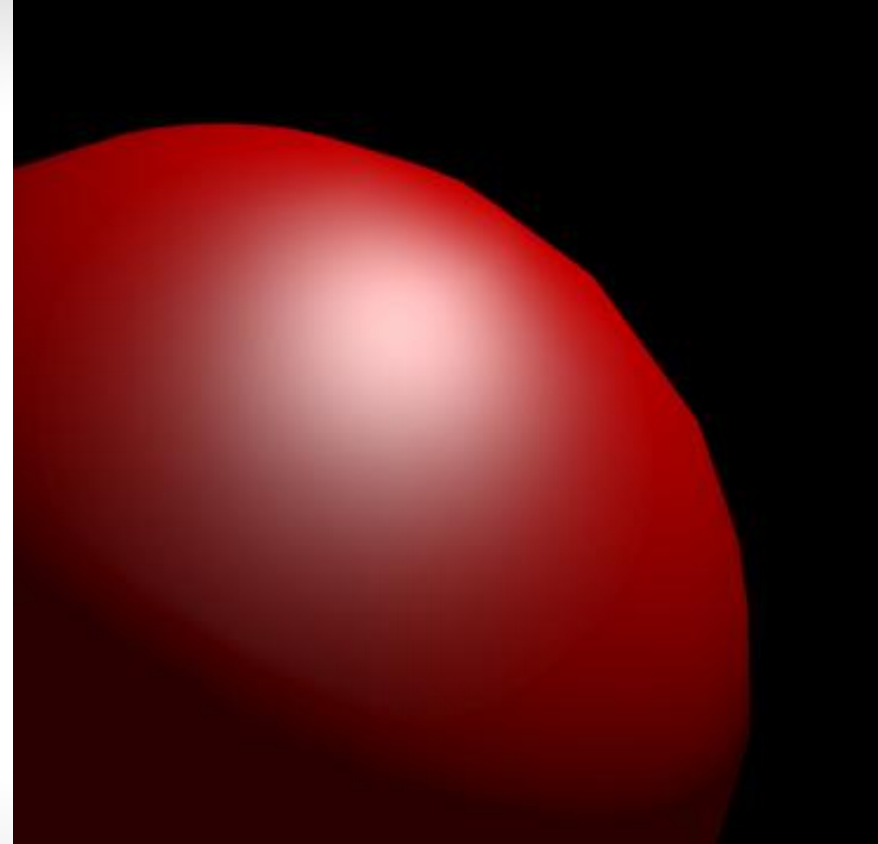
Normal: X,Y,Z



Interpolación de Normales

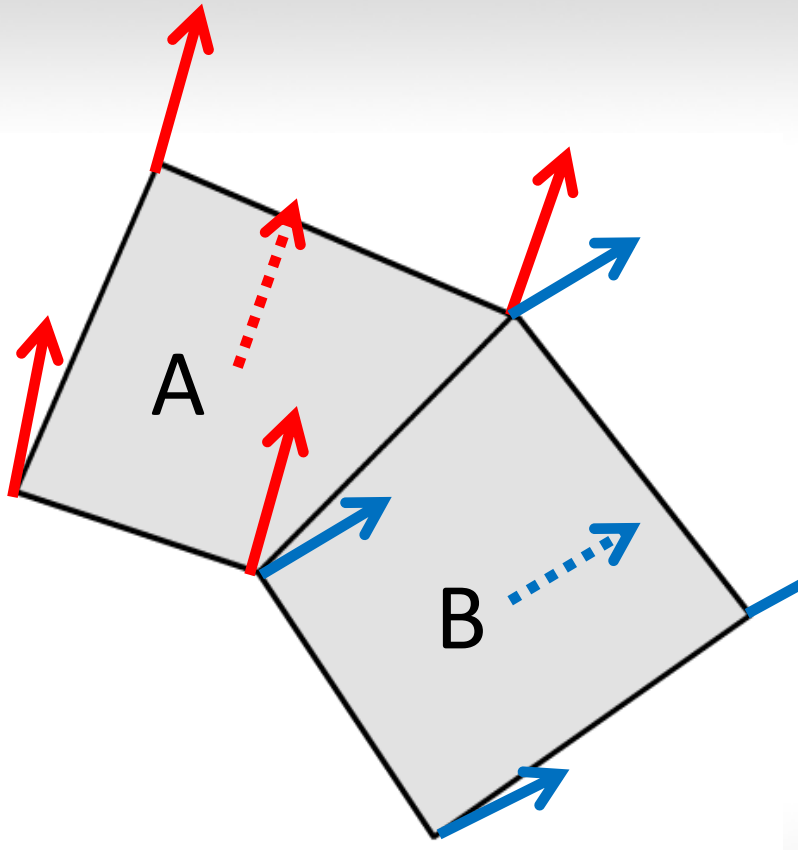


Facetado

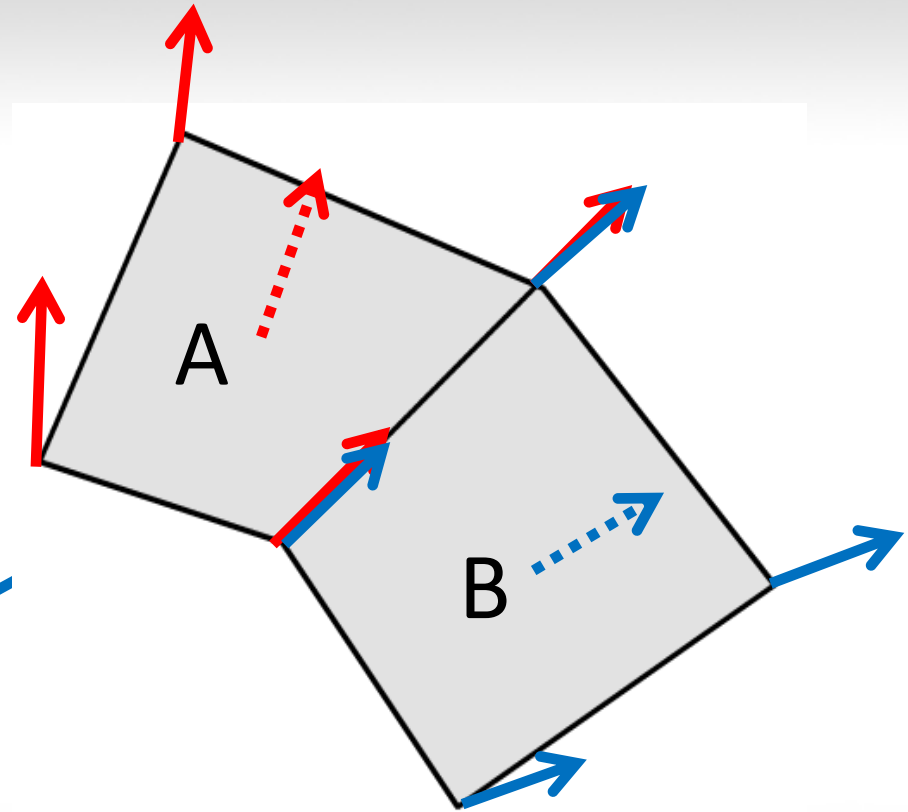


Suave

Interpolación de Normales

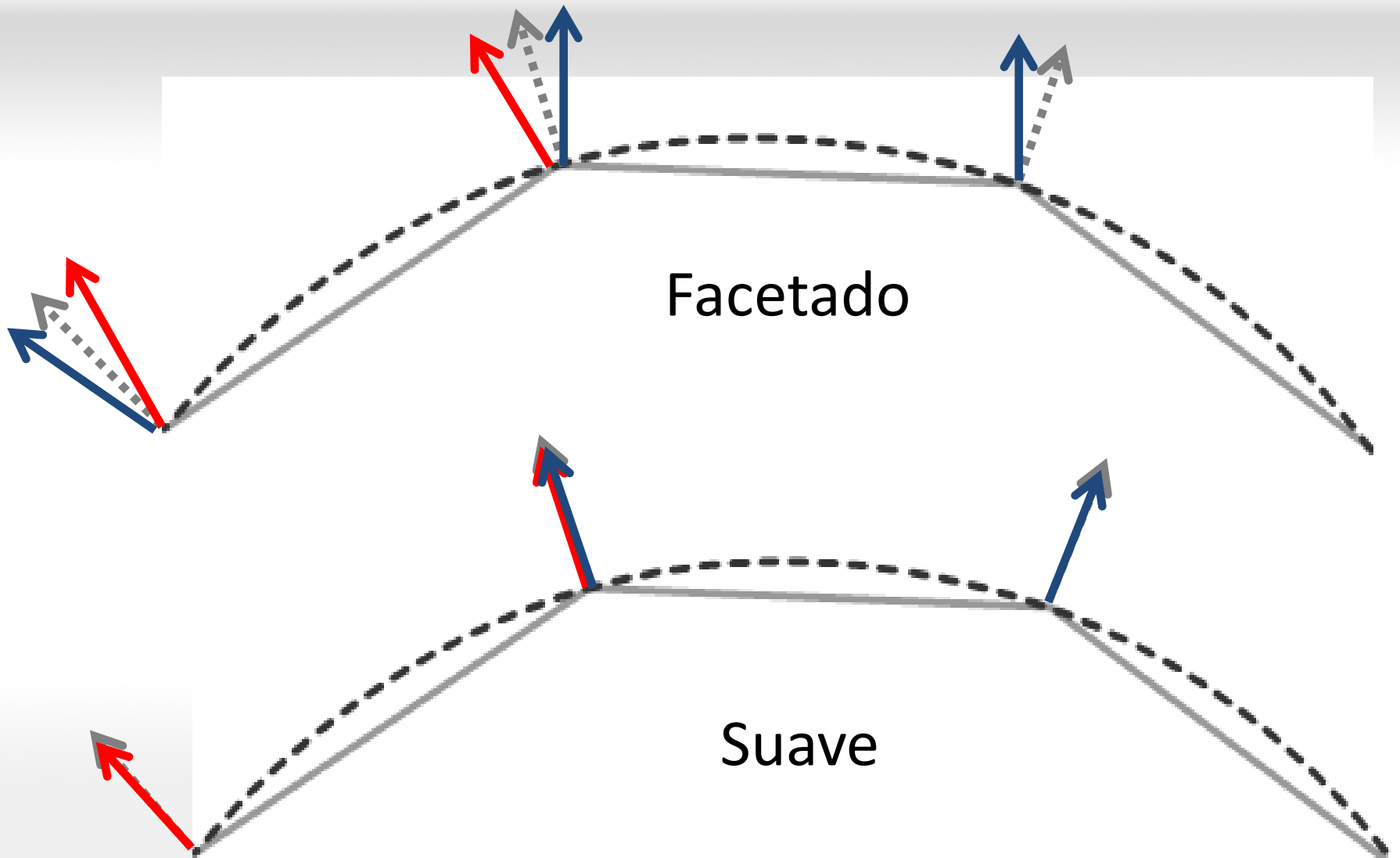


Facetado

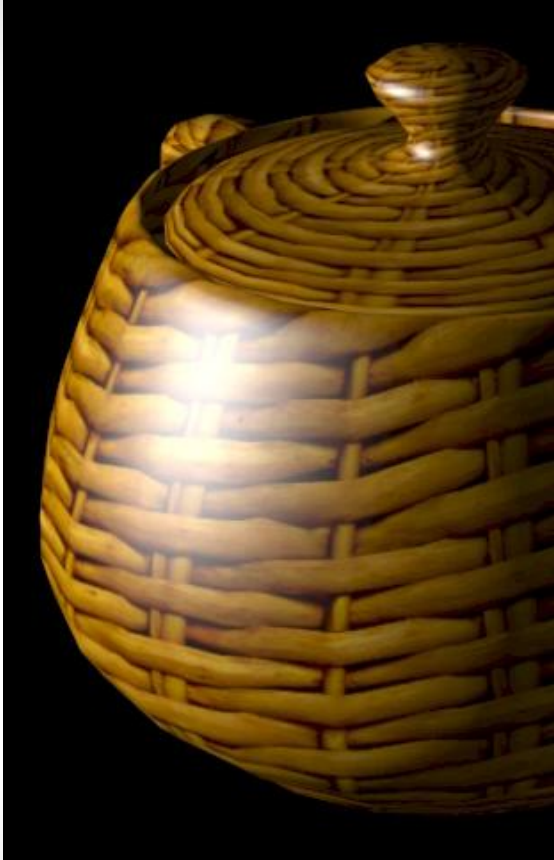


Suave

Interpolación de Normales



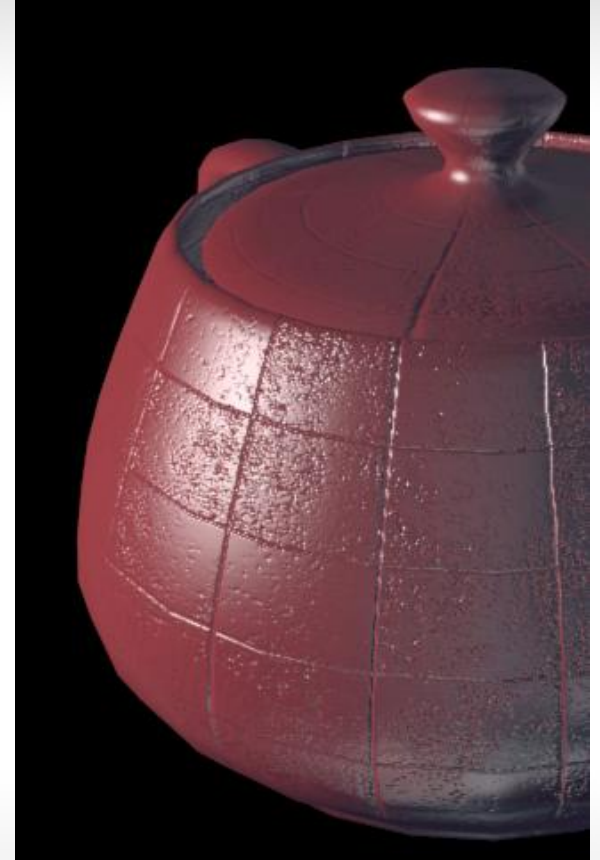
Texturas y Phong



Mapa Difuso

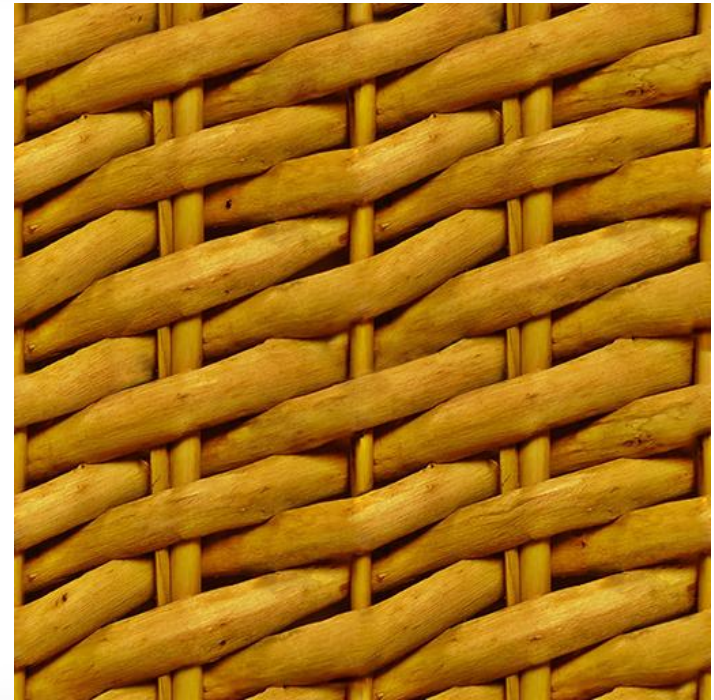


Mapa Reflexión



Mapa Relieve

Mapa Difuso



Mapas de Reflexión



← mapa

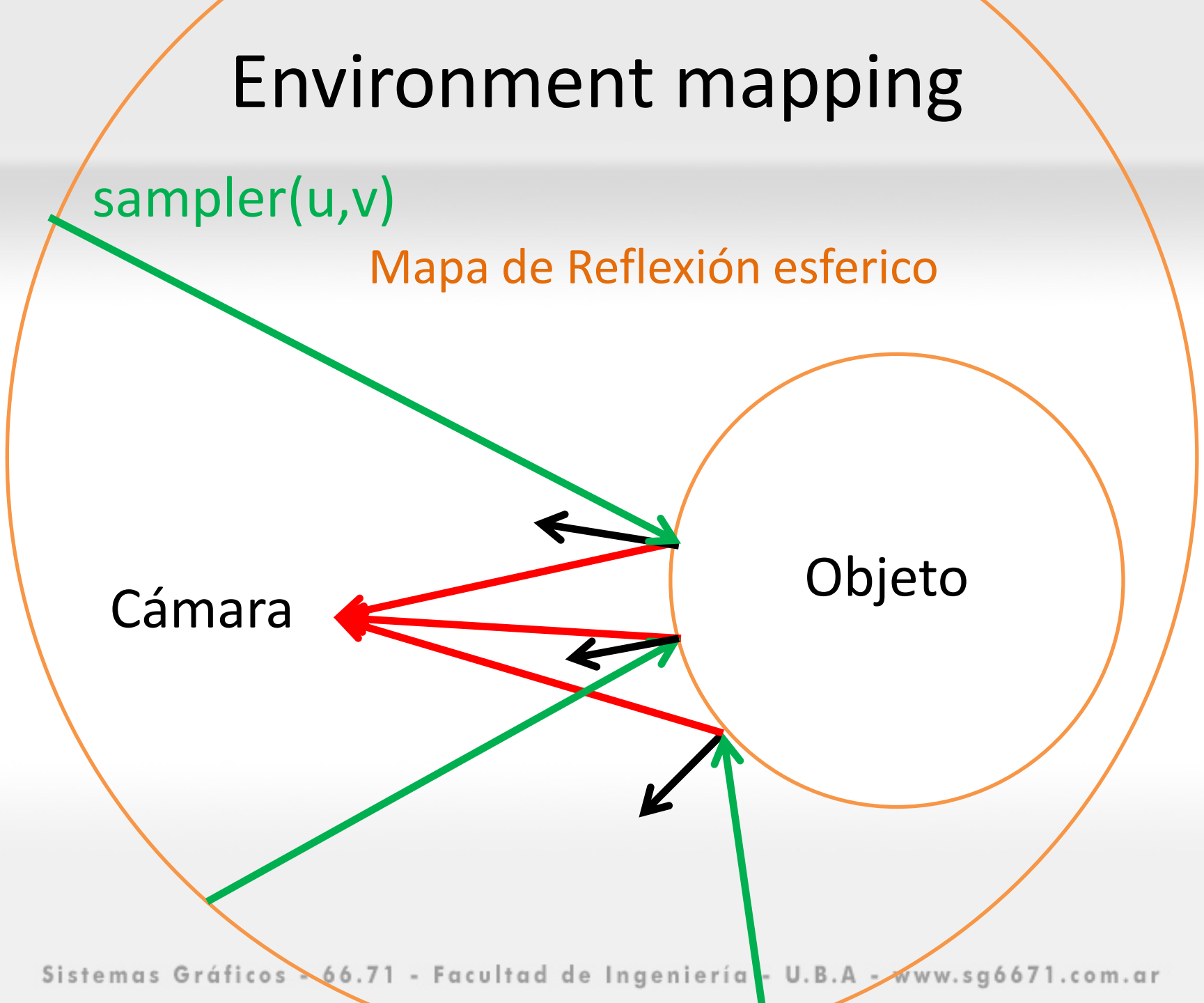
Environment mapping

sampler(u,v)

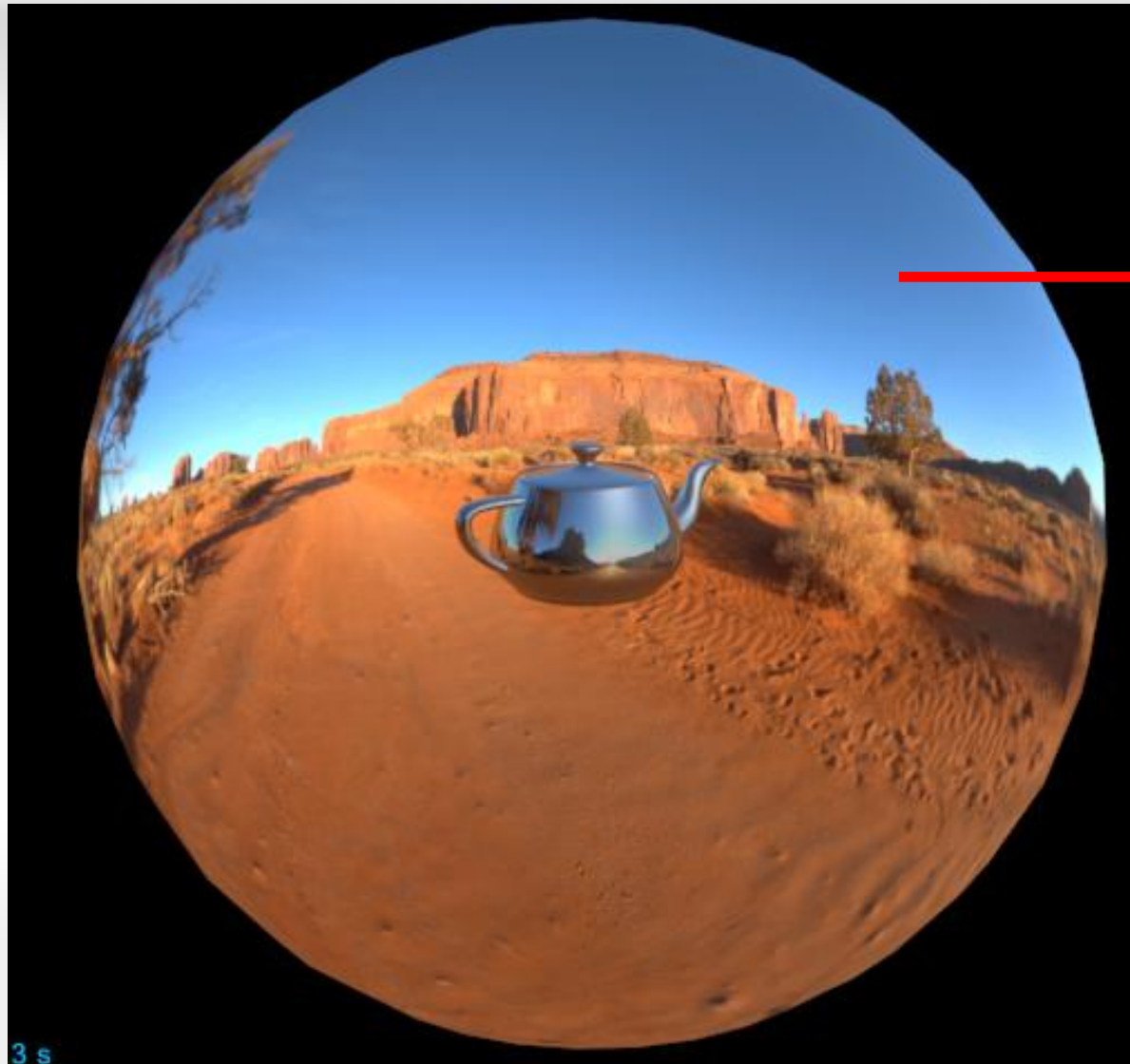
Mapa de Reflexión esferico

Cámara

Objeto



Environment mapping

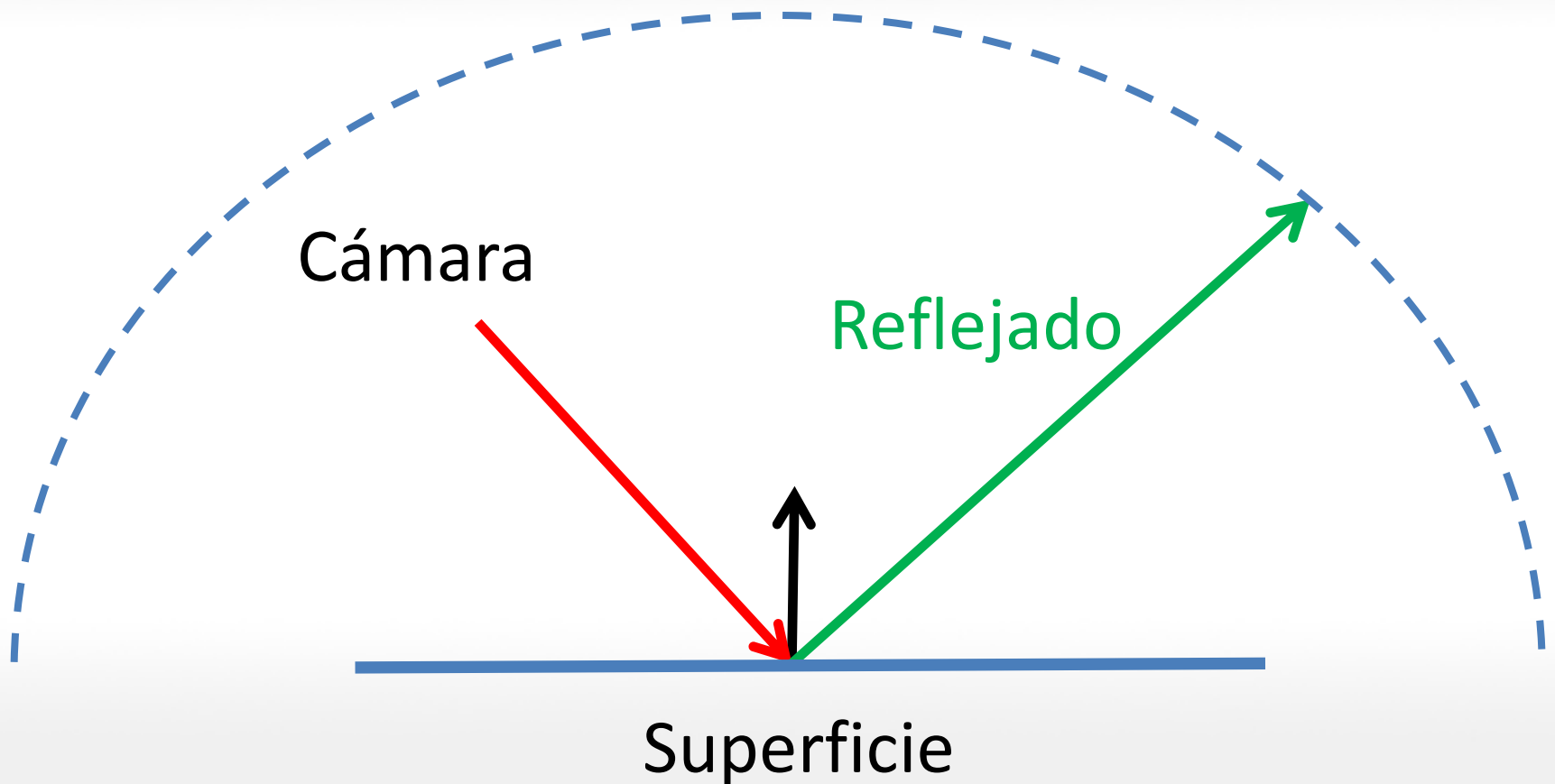


Esfera de
radio
infinito



Environment mapping

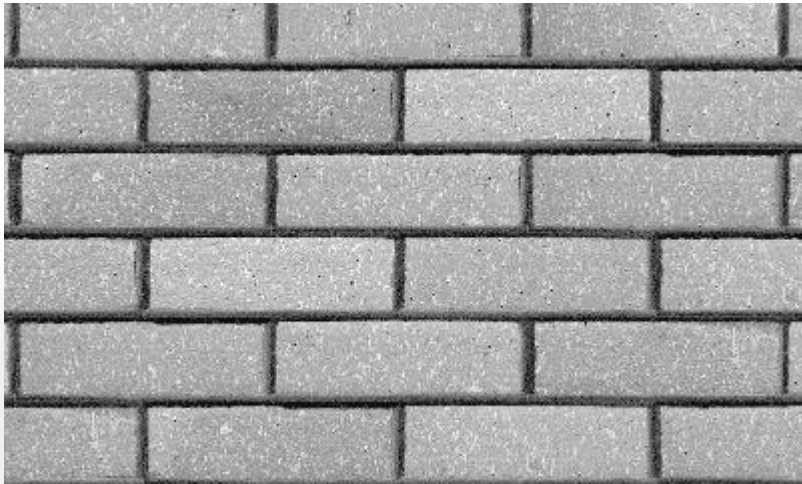
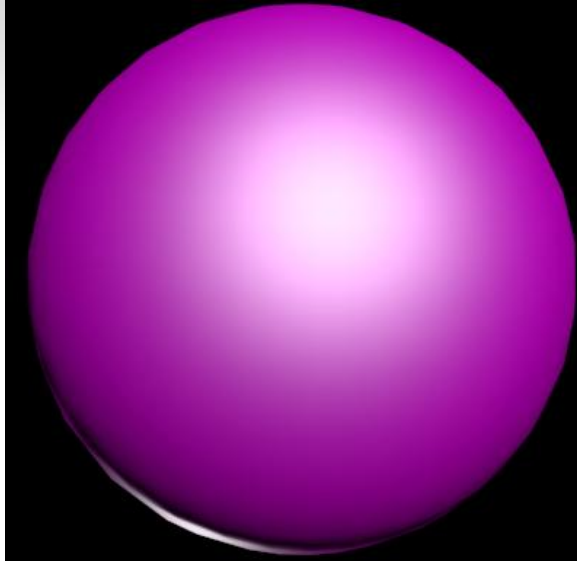
Cada R_x, R_y, R_z mapea en u, v del mapa



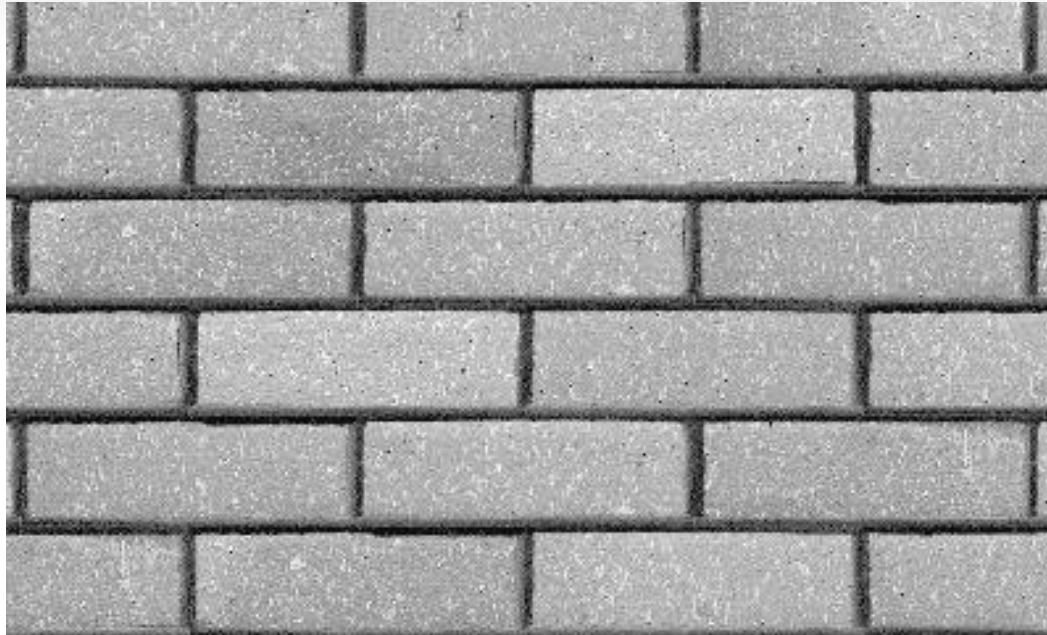
Environment mapping



Mapa de Relieves



Bump Mapping



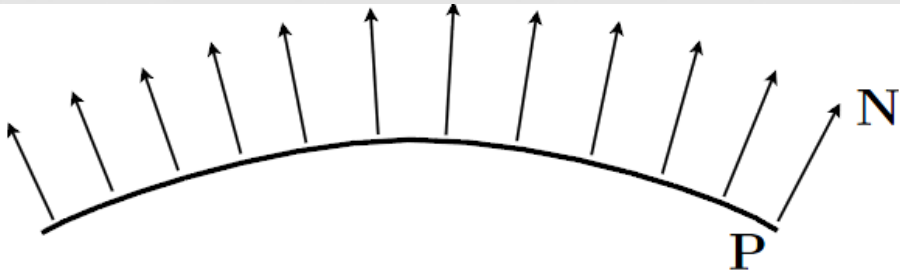
Bump Map: textura monocromática que representa altura local de la superficie

Bump Mapping

La normal se puede aproximar a partir del bumpmap



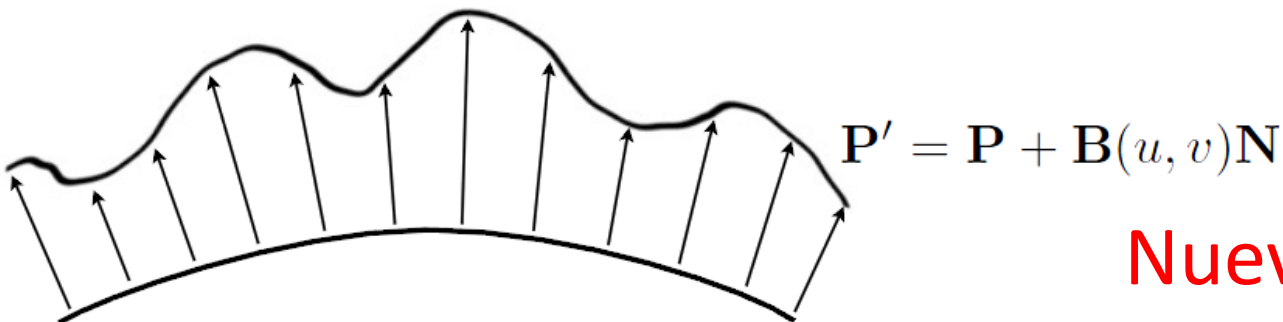
Bump Mapping



Normal Interpolada
(según normales de vértices)



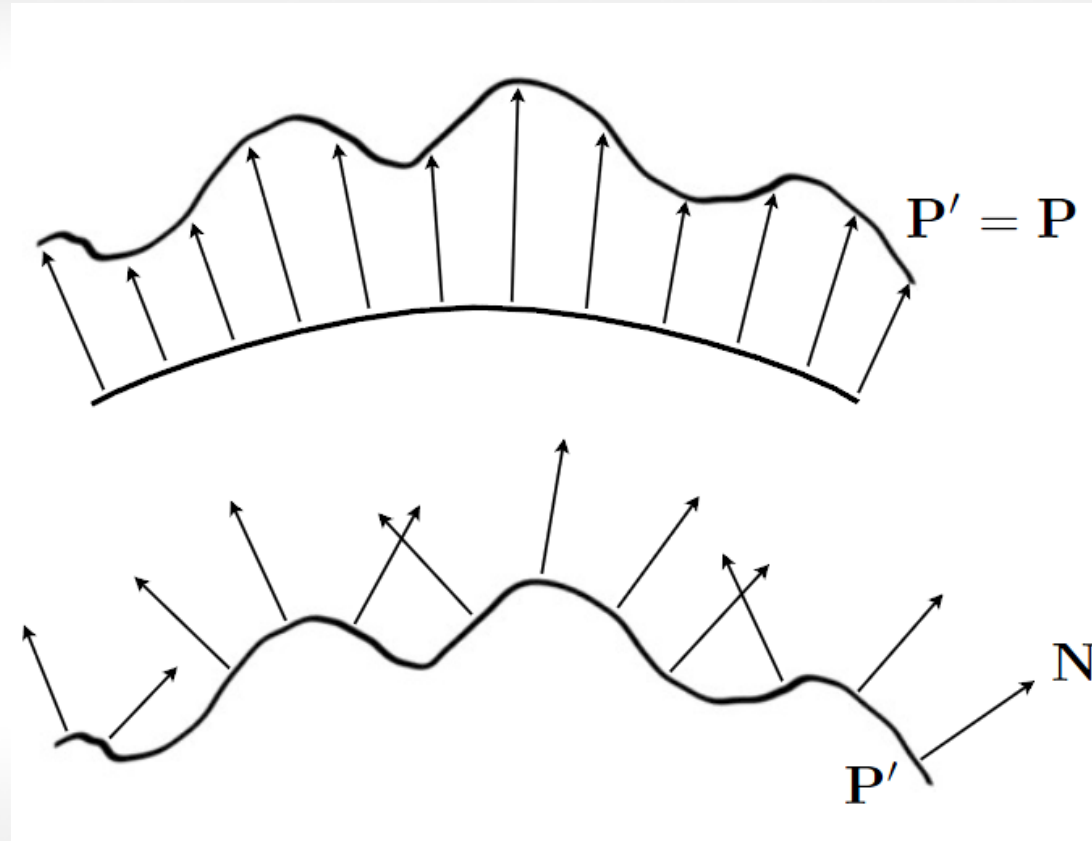
Altura local según
BumpMap



Nuevo Perfil de S

Bump Mapping

Es necesario
computar el
gradiente de la
superficie a partir
del **mapa de
alturas**



Bump Mapping - Gradiente

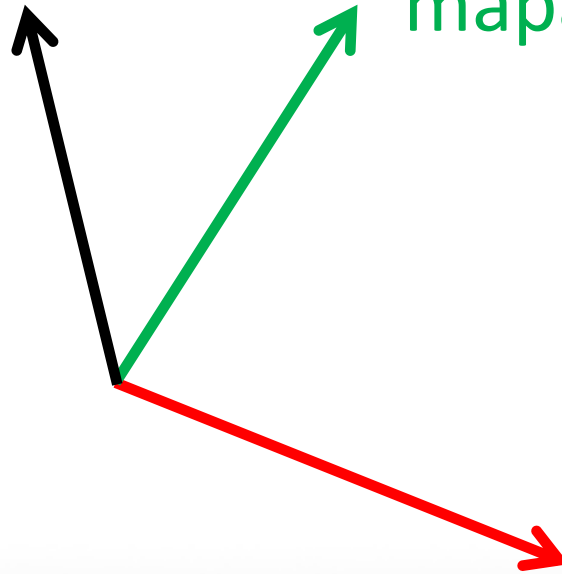
Normal

Derivada en v del
mapa de alturas

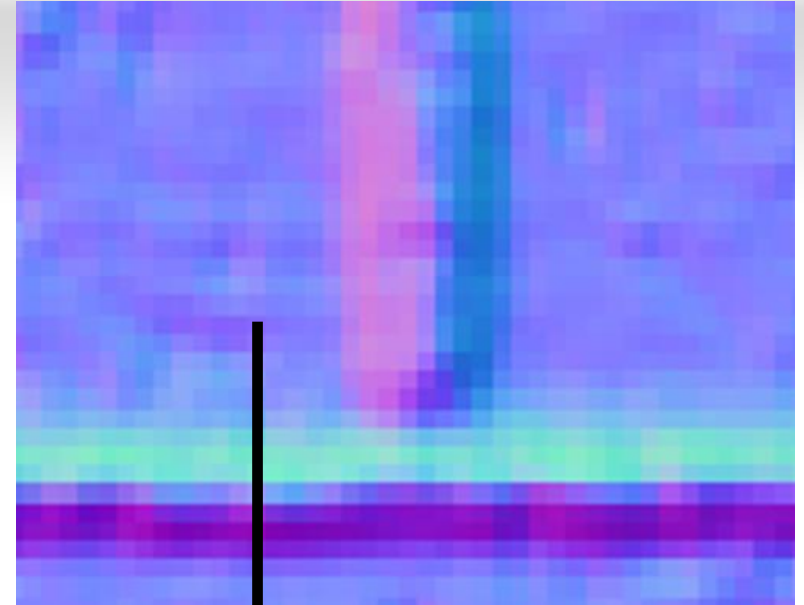
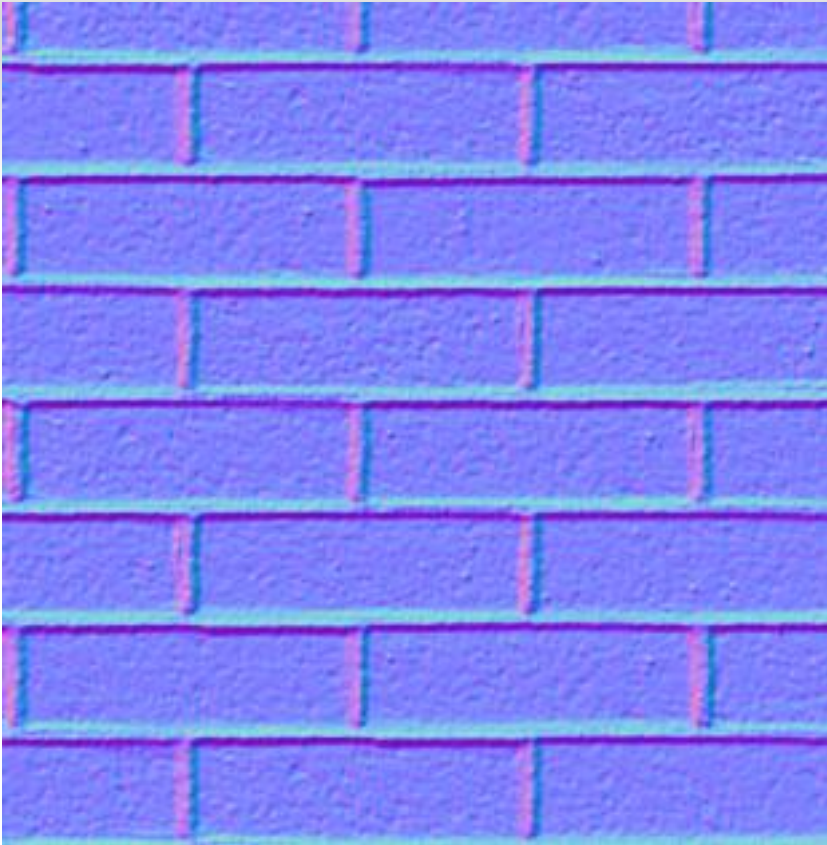
$$N = D_u \times D_v$$

La normal local se
obtiene como
producto vectorial
de las derivadas
del mapa de
alturas

Derivada en u del
mapa de alturas



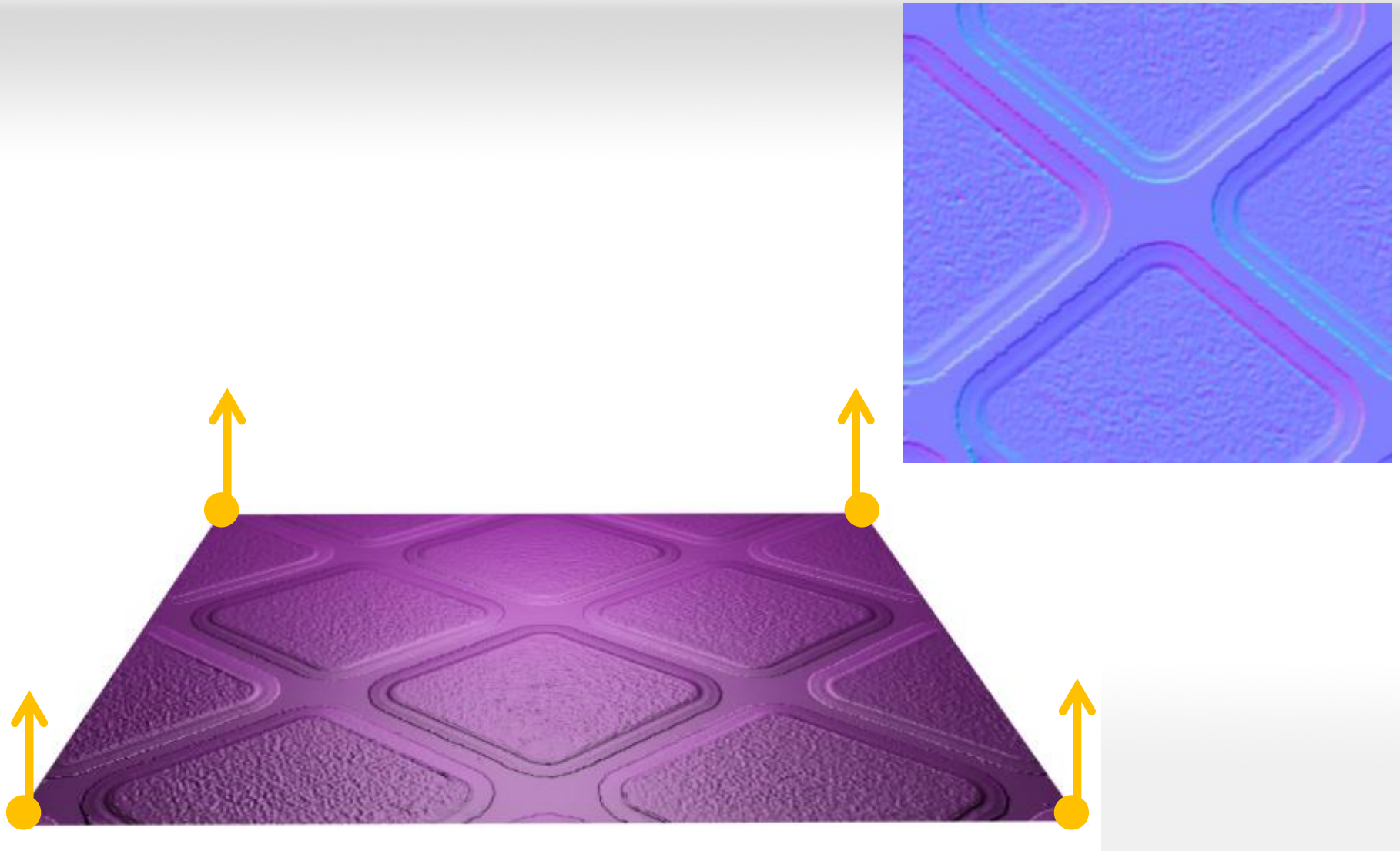
Mapa de normales



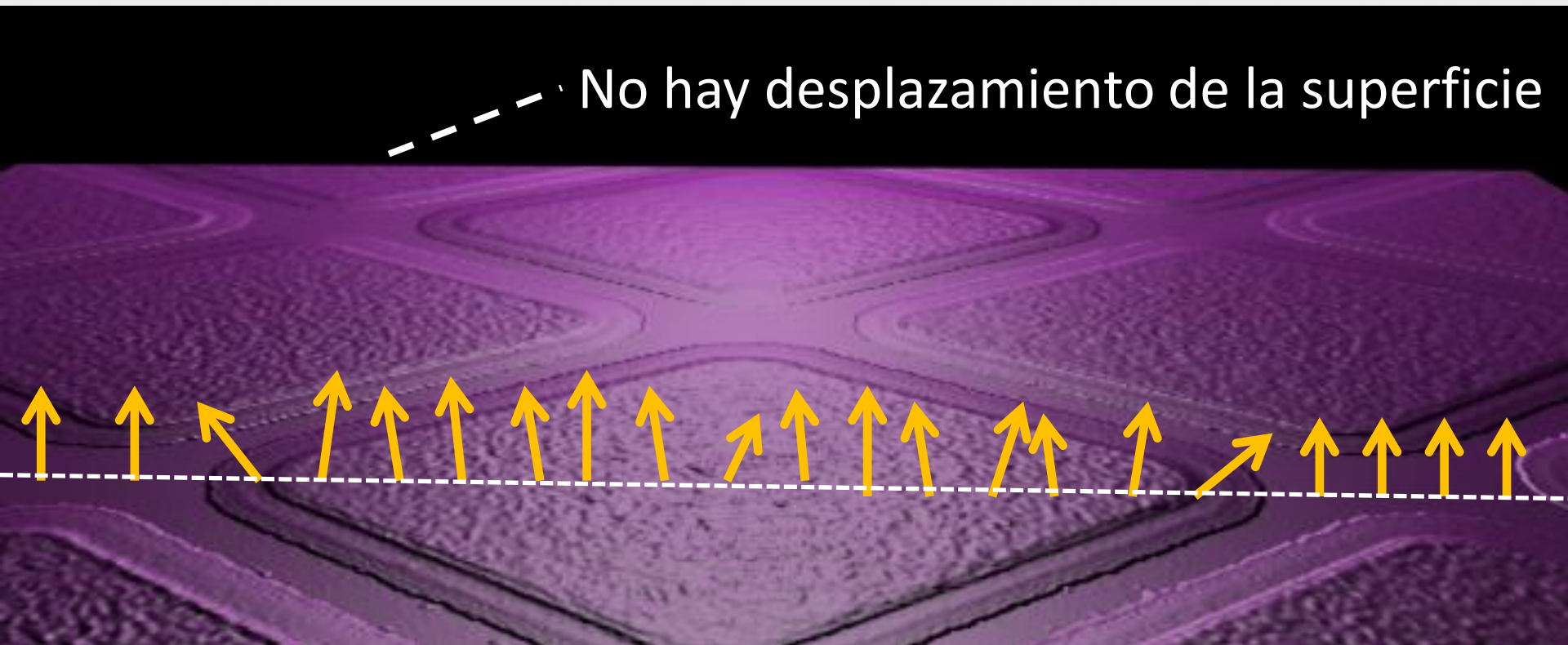
$$\text{RGB} = N_x, N_y, N_z$$

Requiere menos computo ya que el mapa contiene la normal local codificada en RGB

Mapa de normales



Mapa de normales



Variación local de la normal da ilusión de relieve

Mapa de normales – codific. RGB

Red = 0 ... 128 255



$N_x = -1$... 0 1

Green = 0 ... 128 255



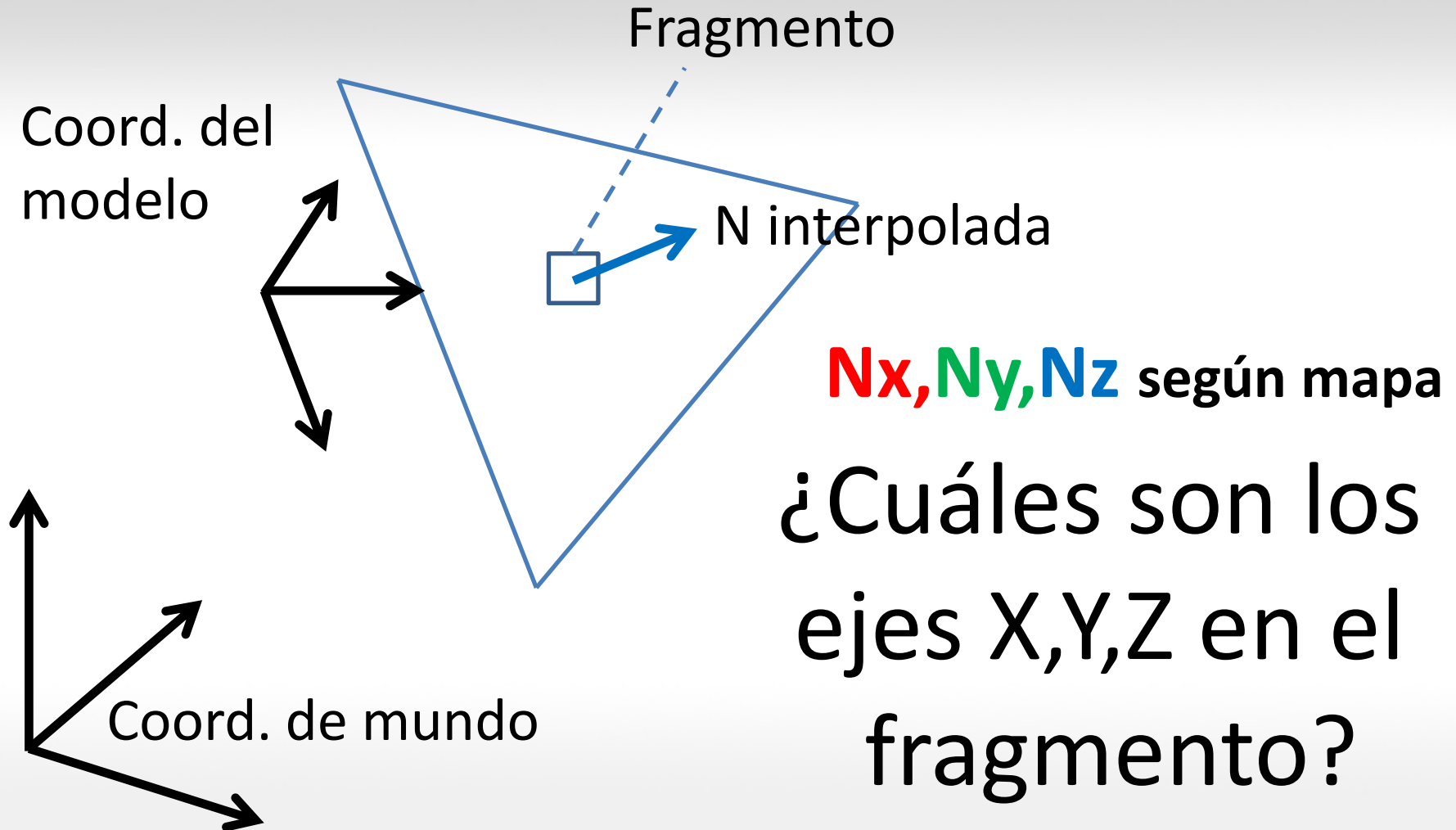
$N_y = -1$... 0 1

Blue = 0 ... 128 255



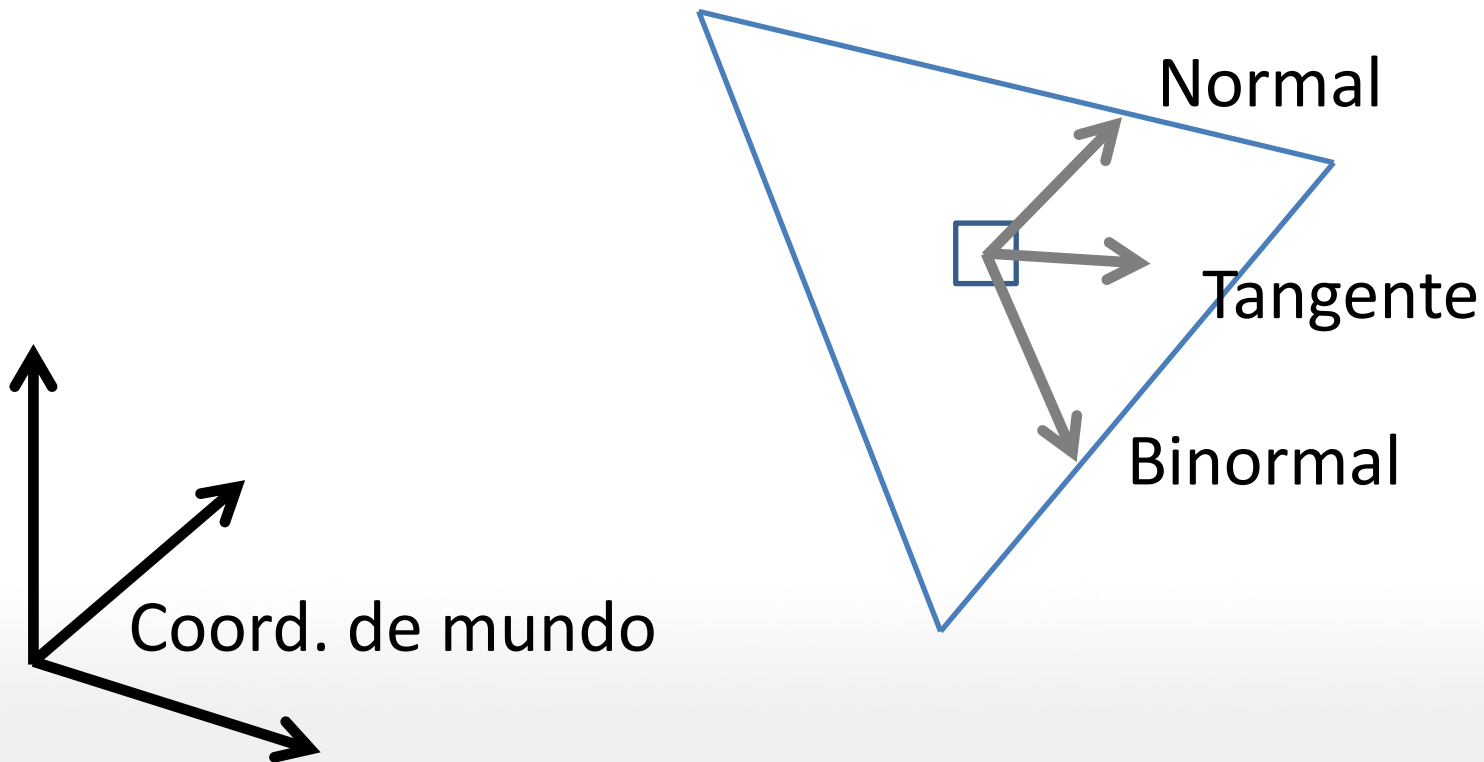
$N_z = -1$... 0 1

Mapa de normales



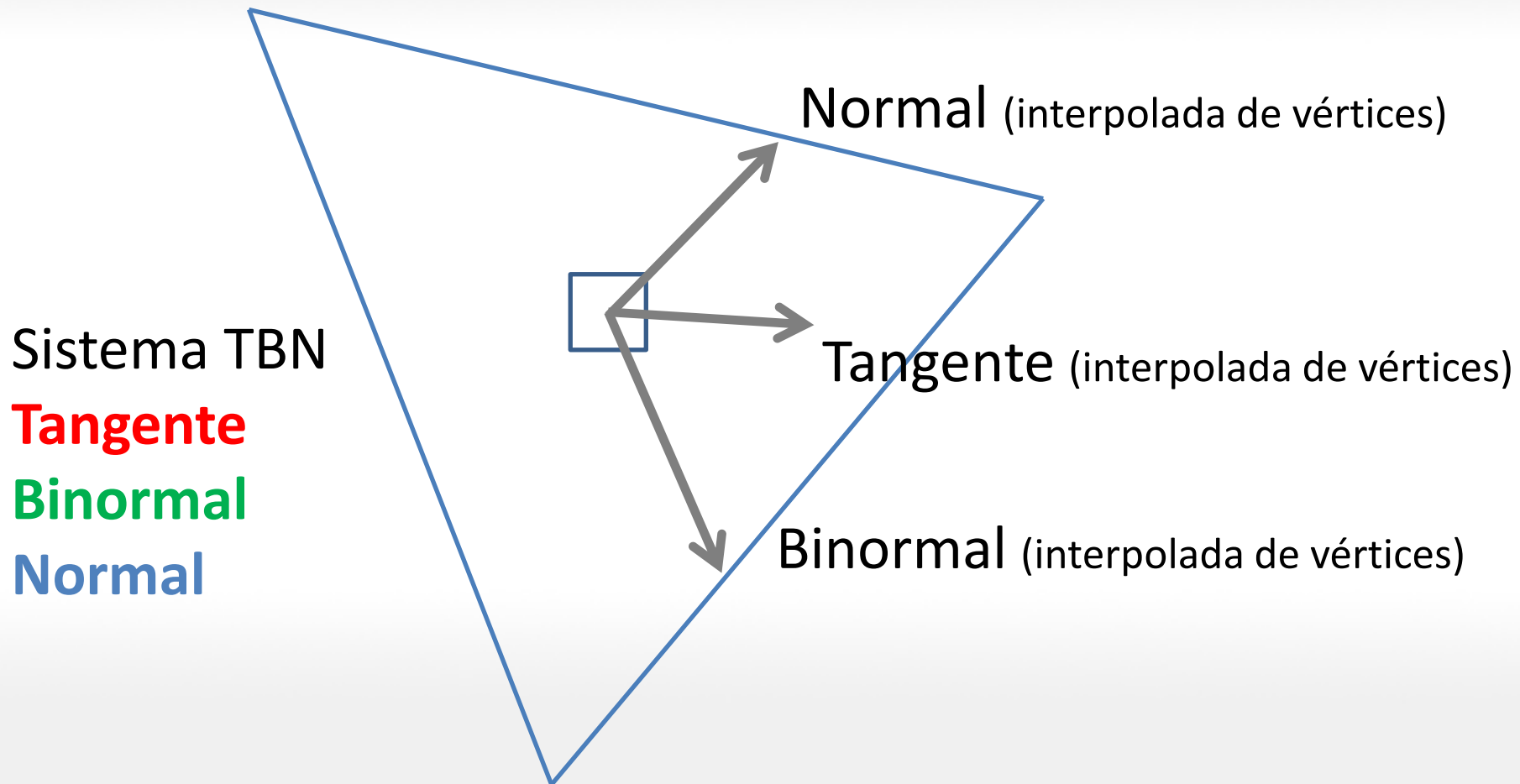
Mapa de normales

Necesitamos un sistema de coord. local fragmento



Mapa de normales

Nx, Ny, Nz computada del mapa



Mapa de normales

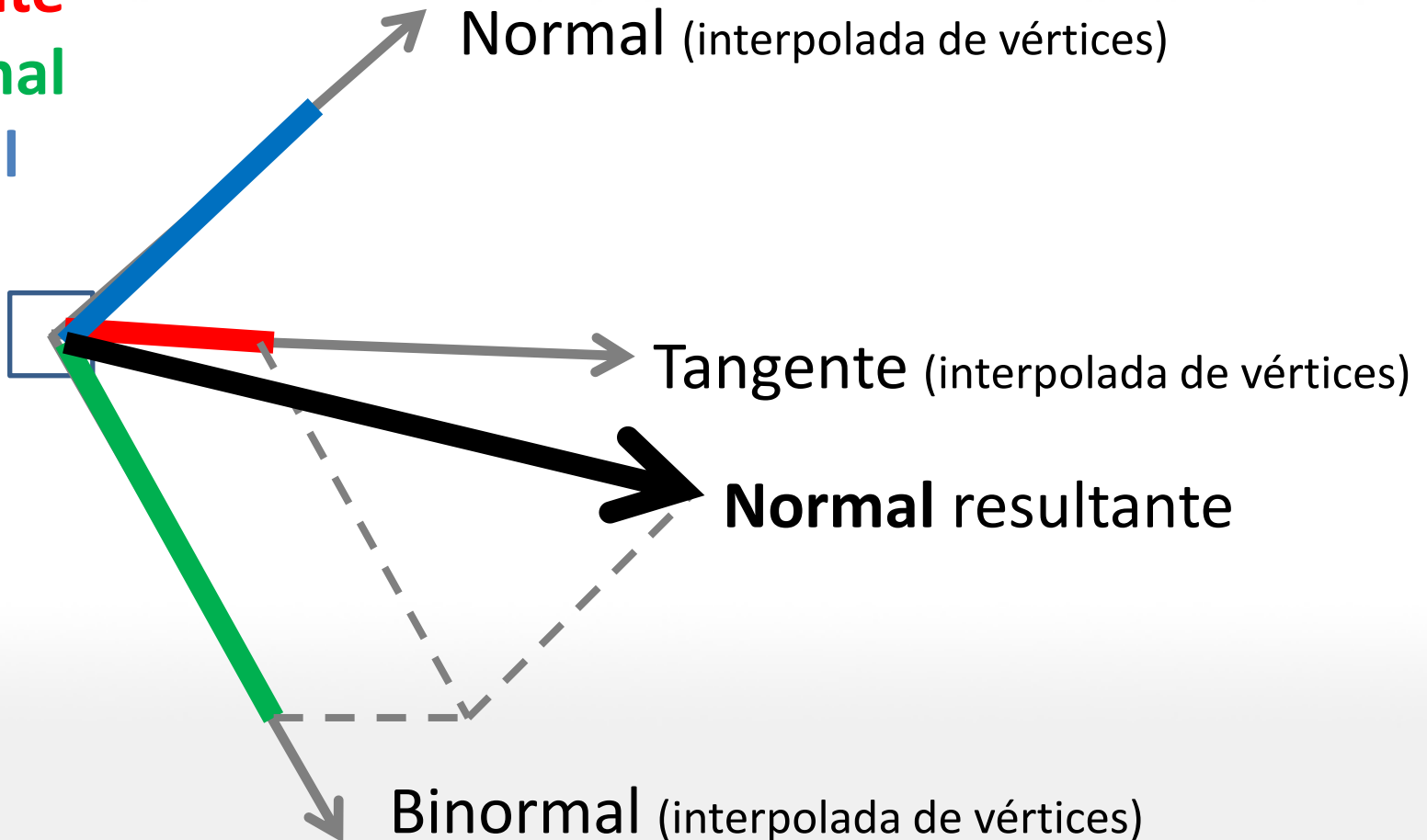
N_x, N_y, N_z computada del mapa

Sistema TBN

Tangente

Binormal

Normal



Mapa de normales

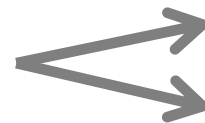
Sistema TBN

Tangente

Binormal

Normal

agregados



**Atributos de
vértices**

P_x, P_y, P_z

N_x, N_y, N_z

T_x, T_y, T_z

B_x, B_y, B_z

u, v

Mapa de normales

