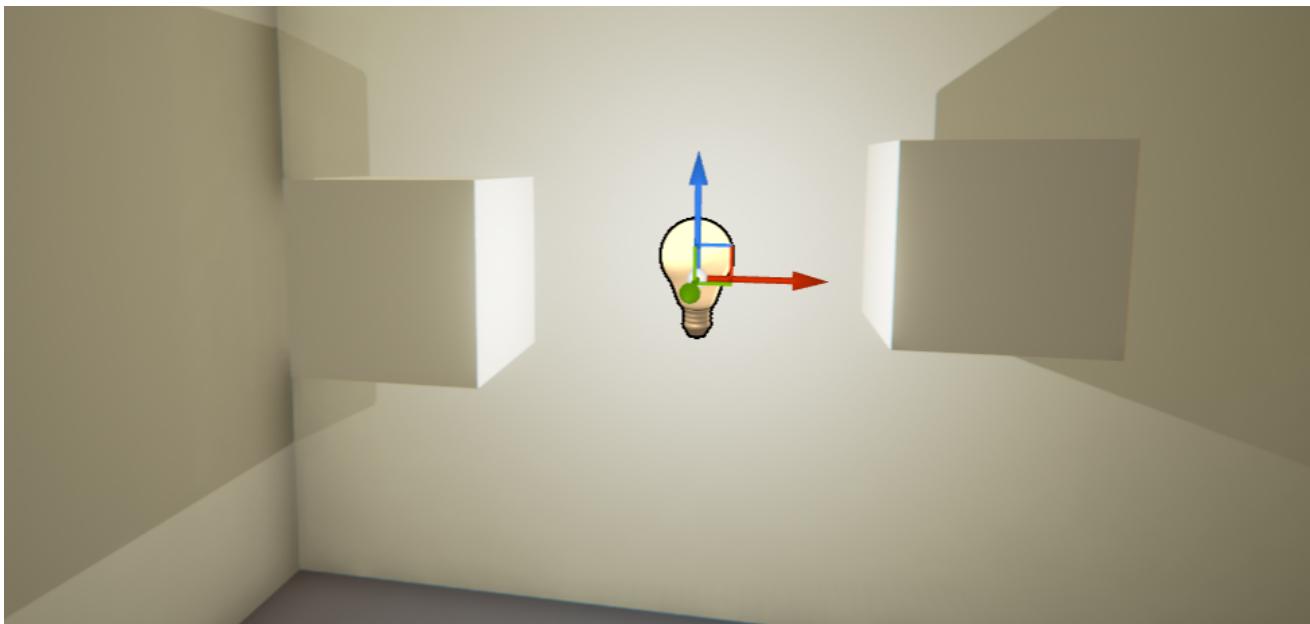


Computación Gráfica

Clase 3 - Luces y sombras simples, Lambert y Phong shading



Luces y sombras simples

Qué se desea saber de una luz con respecto a un punto?

- Recibe iluminación?
- Vector incidente L_i
- Intensidad y color de la luz incidente

Luces y sombras simples

Cómo saber si un punto del espacio recibe luz?

Función de Visibilidad

Define si dos puntos son mutuamente visibles.

$$\forall x, y \in A: V(x, y) = \begin{cases} 1 & \text{if } x \text{ and } y \text{ are mutually visible} \\ 0 & \text{if } x \text{ and } y \text{ are not mutually visible} \end{cases}$$

Luces y sombras simples

Algoritmo simple:

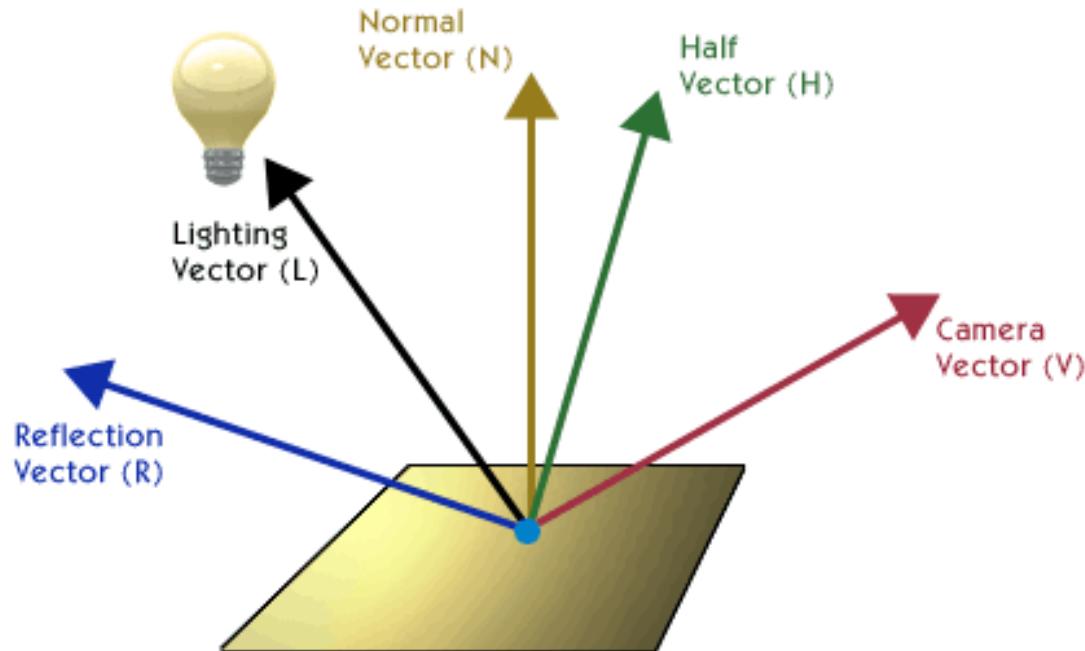
Por cada punto P del espacio muestreado

Para P_L de cada luz

Si $V(P, P_L) == 1$

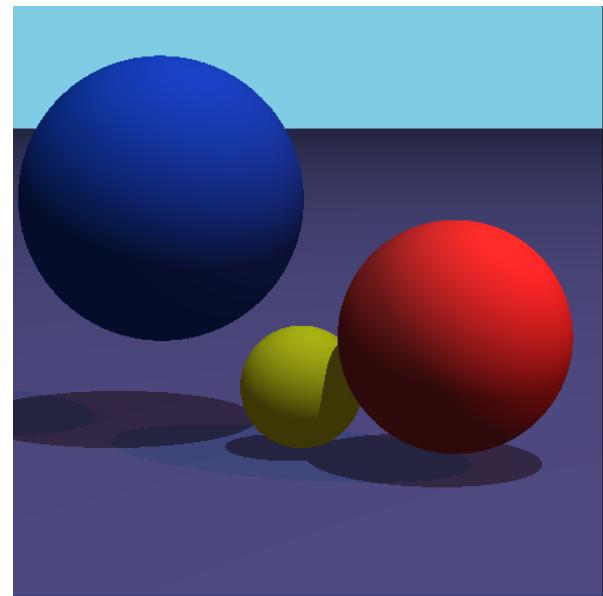
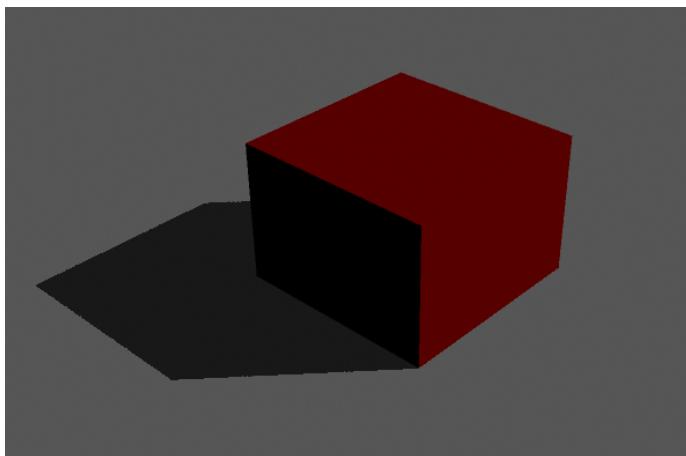
Illumino el pixel con L_i

Luces y sombras simples



Luces y sombras simples

Esto va a resultar en sombras directas
naturalmente:



Luces y sombras simples

Tipos de luces más comunes:

- Direccional
- Ambiente
- Puntual
- Spot lights
- Area lights (más adelante...)

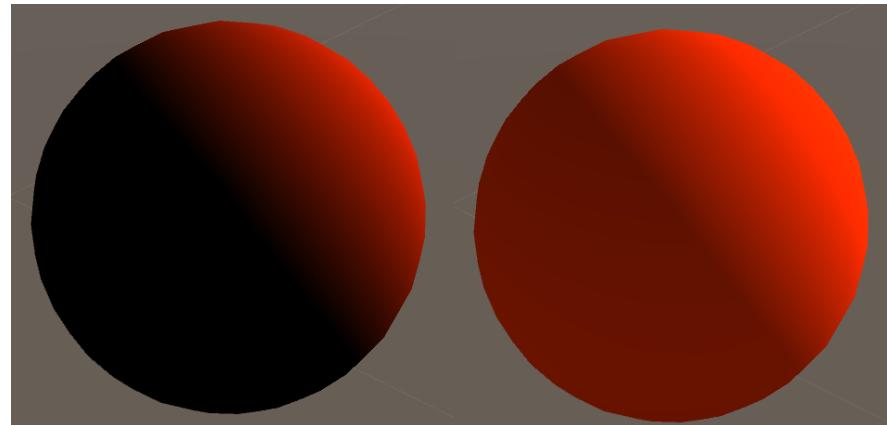
Luces y sombras simples

Tipos de luces más comunes:

- Ambiente

Luz ambiente

- No es físicamente correcta
- No posee sombras
- Solo aporta color



Luces y sombras simples

Tipos de luces más comunes:

- Ambiente
- Direccional

Luz direccional

- Definida por un vector dirección

Luz direccional

- Definida por un vector dirección
- No tiene una posición determinada!

Luz direccional

- Definida por un vector dirección
- No tiene una posición determinada!
- Tampoco es físicamente correcta!

Luz direccional

- Definida por un vector dirección
- No tiene una posición determinada!
- Tampoco es físicamente correcta!
- Cómo definir la función de visibilidad para una luz direccional?

Luz direccional

La función de visibilidad para luces la debe implementar la misma luz

```
public double getVisibility(Sample<Vector3> sample, ShadingContext data);
```

En este caso, retorna 1 si un rayo en la dirección \mathbf{L}_i no interseca ningún objeto de la escena

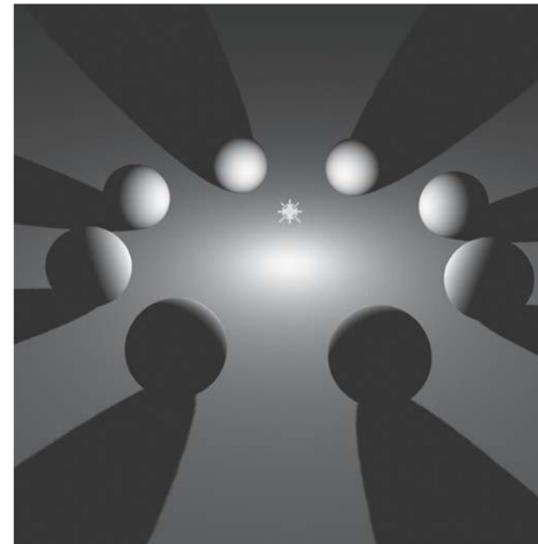
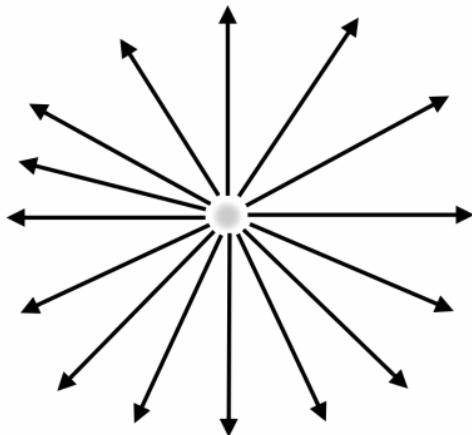
Luces y sombras simples

Tipos de luces más comunes:

- Direccional
- Ambiente
- Puntual

Luz puntual

- Definida por un punto
- Función de visibilidad sencilla
- Sombras simples



Luces y sombras simples

Tipos de luces más comunes:

- Direccional
- Ambiente
- Puntual
- Spot lights

Spotlights

- Definida por un punto, una dirección y un ángulo
- Tampoco son físicamente correctas!
- Función de visibilidad?

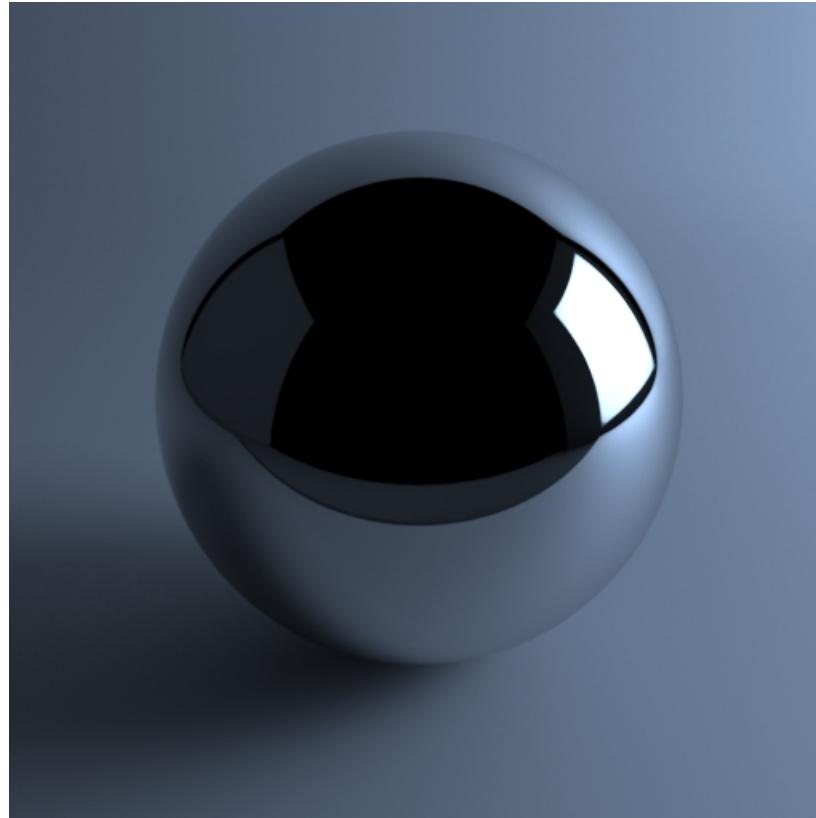


Luces y sombras simples

Tipos de luces más comunes:

- Direccional
- Ambiente
- Puntual
- Spot lights
- Area lights (más adelante...)

Area lights...



Luces simples

De cada luz, nos interesa definir:

- El color
- La intensidad

Luces simples

De cada luz, nos interesa definir:

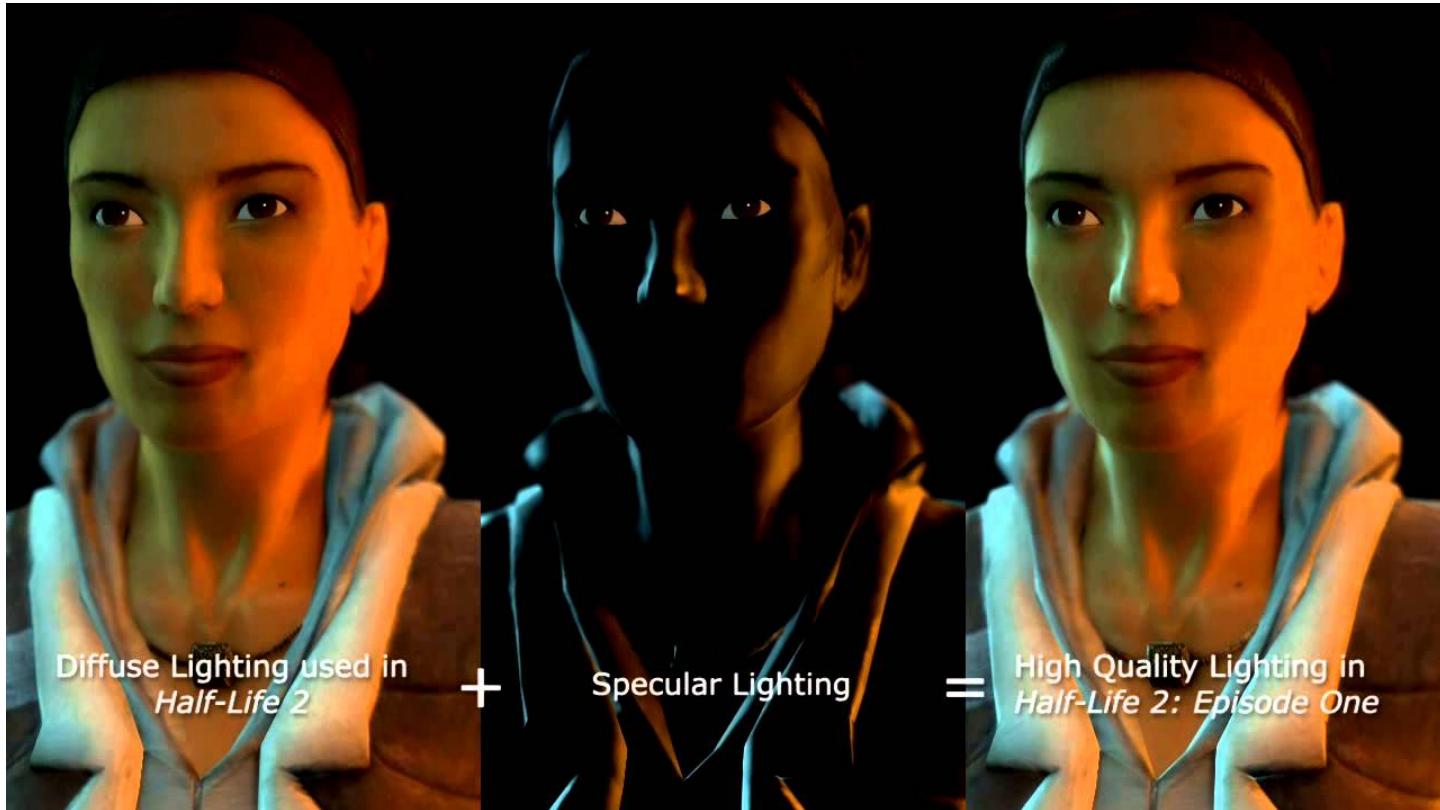
- El color
- La intensidad
- Distancia máxima (físicamente incorrecto)
- Color de sombra (físicamente incorrecto)

Luces simples

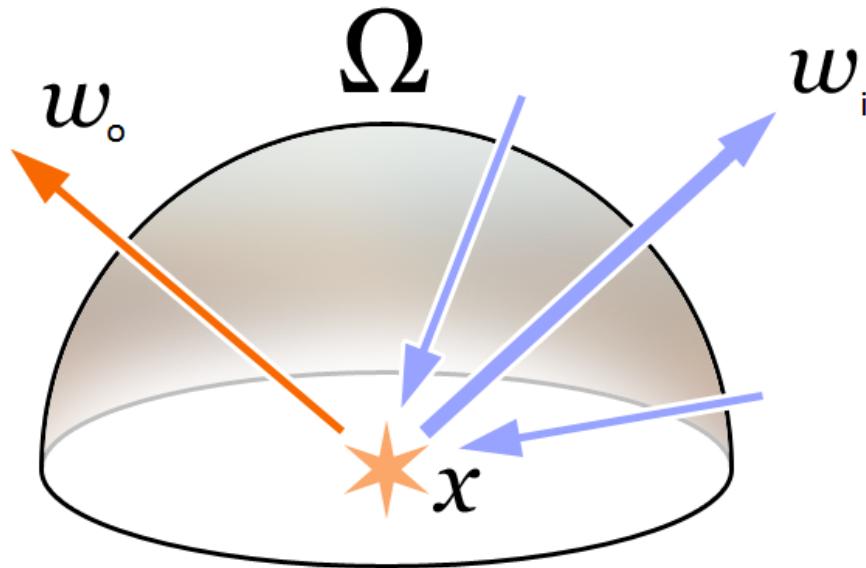
De cada luz, nos interesa definir:

- El color
- La intensidad
- Distancia máxima (físicamente incorrecto)
- Color de sombra (físicamente incorrecto)
- *Decay* (más adelante...)
- La temperatura (más adelante...)

Modelos de iluminación



Modelos de iluminación



$$L_o(\mathbf{x}, \omega_o, \lambda, t) = L_e(\mathbf{x}, \omega_o, \lambda, t) + \int_{\Omega} f_r(\mathbf{x}, \omega_i, \omega_o, \lambda, t) L_i(\mathbf{x}, \omega_i, \lambda, t) (\omega_i \cdot \mathbf{n}) d\omega_i$$

Modelos de iluminación simples

Utilizando iluminación simple, no hay ningún tipo de integración del hemisferio, sino aproximaciones del resultado.

Una vez que se resolvió la luz, se utilizan todos los datos para calcular el color final de la muestra.

Modelos de iluminación simples

Modelos simples de BRDF:

- Lambert (diffuse)
- Phong shading
- Blinn-Phong shading

Iluminación difusa

- La luz se esparce uniformemente a lo largo de todo el hemisferio
- Función de BRDF constante
- Simplificando,

$$I_D = \mathbf{L} \cdot \mathbf{N} C I_L$$

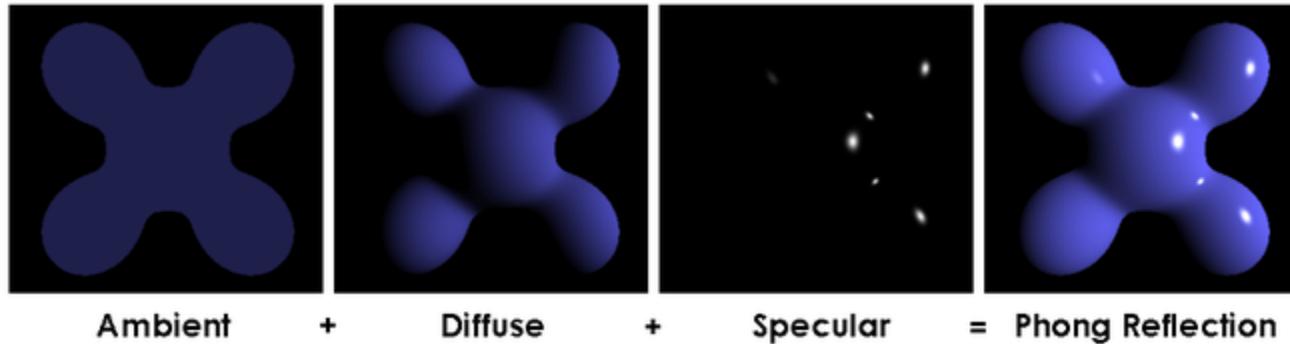
$$\mathbf{L} \cdot \mathbf{N} = |N||L| \cos \alpha = \cos \alpha$$

Illuminación difusa

Ejemplos:



Phong Shading



- Empírico y aproximado
- Razonablemente realista cuando se propuso

$$I_p = k_a i_a + \sum_{m \in \text{lights}} (k_d (\hat{L}_m \cdot \hat{N}) i_{m,d} + k_s (\hat{R}_m \cdot \hat{V})^\alpha i_{m,s}).$$

Phong Shading

Utiliza el vector incidente de luz reflejado,

$$\hat{R}_m = 2(\hat{L}_m \cdot \hat{N})\hat{N} - \hat{L}_m$$

Y luego exponencia el producto escalar de R y V, donde V es el vector incidente de la cámara

Phong Shading

Intenta modelar la reflectancia especular mediante un coeficiente.

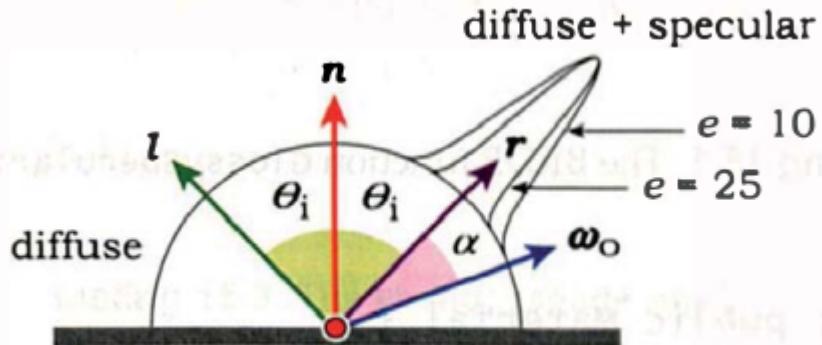


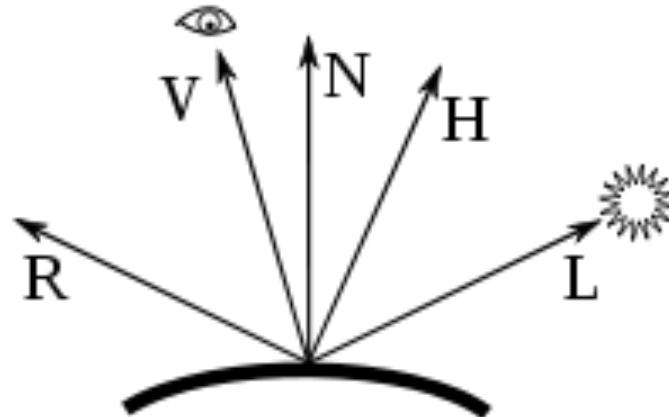
Figure 15.6. The addition of diffuse and specular reflection.

Blinn-Phong Shading

En vez de usar el vector reflejado R , se usa el *half vector* H

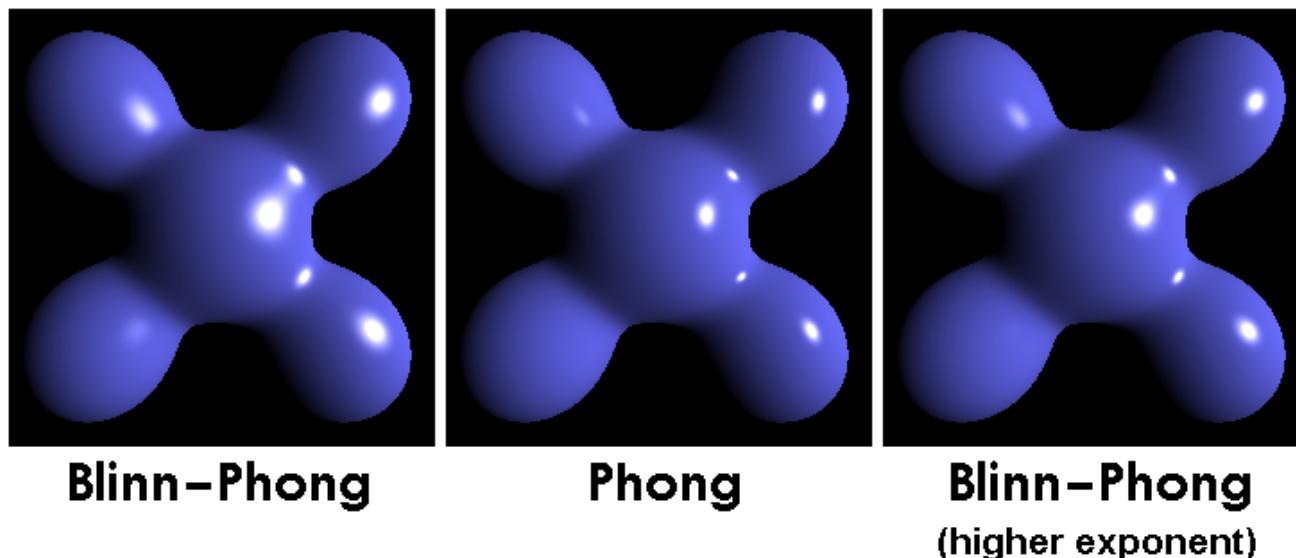
$$H = \frac{L + V}{|L + V|}$$

Se reemplaza
 RV por NH



Blinn-Phong Shading

En vez de usar el vector reflejado R, se usa el *half vector H*



Modelos de iluminación simples

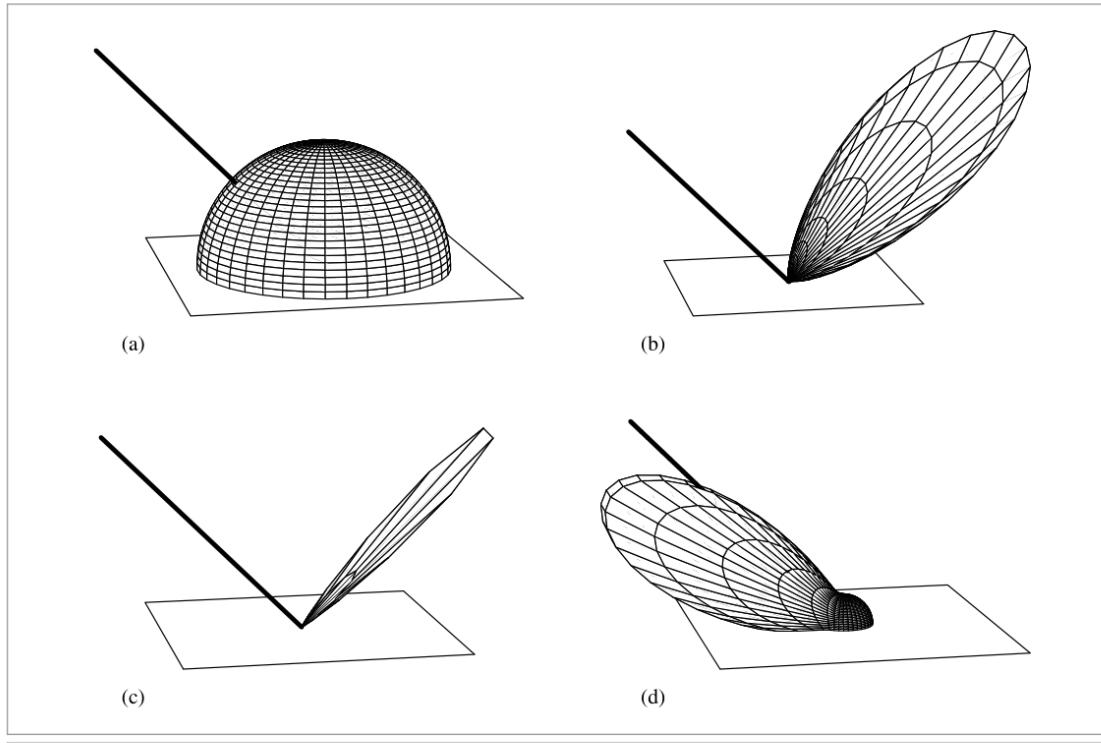


Figure 8.1: Reflection from a surface can be generally categorized by the distribution of reflected light from an incident direction (heavy lines): (a) diffuse, (b) glossy specular, (c) perfect specular, and (d) retro-reflective distributions.

Modelos más complejos...

- Cook-Torrance

$$k_{\text{spec}} = \frac{DFG}{4(V \cdot N)(N \cdot L)}$$

- Ward Anisotropic

$$k_{\text{spec}} = \frac{1}{\sqrt{(N \cdot L)(N \cdot R)}} \frac{N \cdot L}{4\pi\alpha_x\alpha_y} \exp \left[-2 \frac{\left(\frac{H \cdot X}{\alpha_x} \right)^2 + \left(\frac{H \cdot Y}{\alpha_y} \right)^2}{1 + (H \cdot N)} \right]$$

Modelos más complejos...

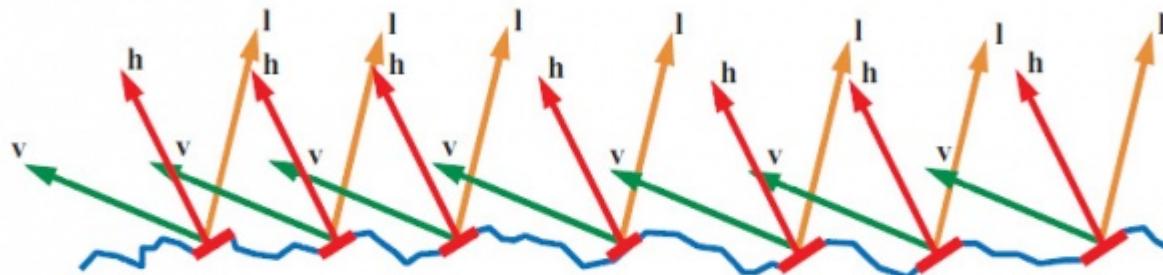


Figure 22: Microfacets with $m = h$ are oriented to reflect l into v —other microfacets do not contribute to the BRDF. (*Image from “Real-Time Rendering, 3rd edition”*)

Modelos más complejos...

