

Computación Gráfica

Eduardo Fernández

Mauricio Eguía

Iluminación y Sombreado

Basado en: **Capítulo 14**

Del Libro: **Introducción a la Graficación
por Computador**

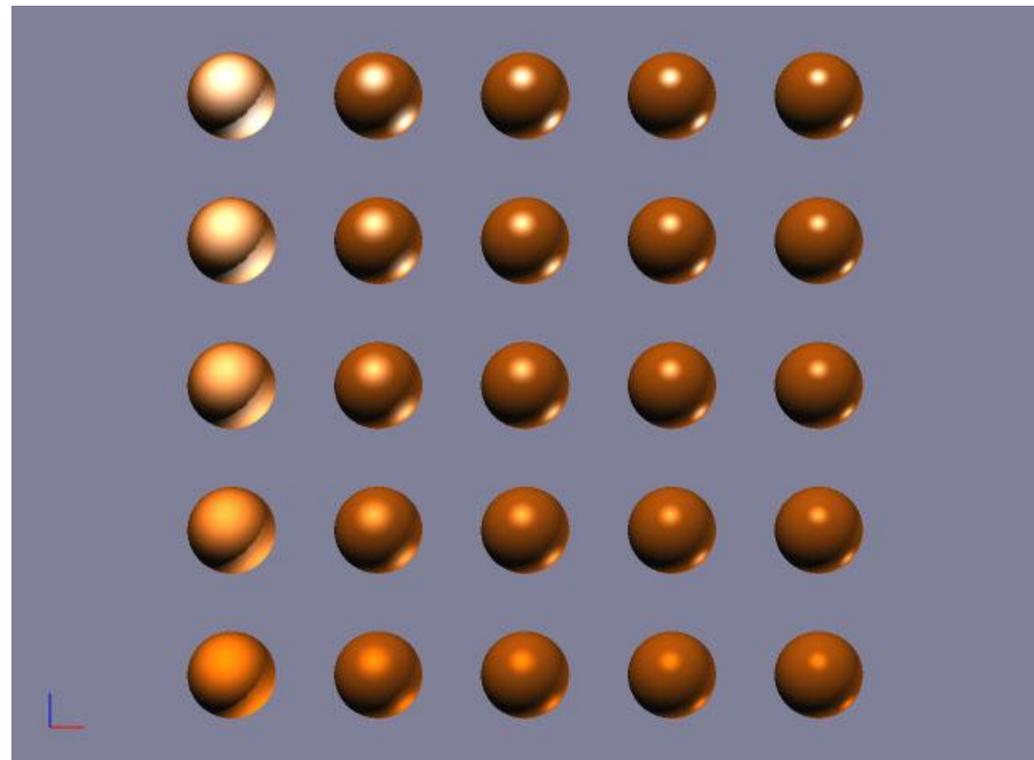
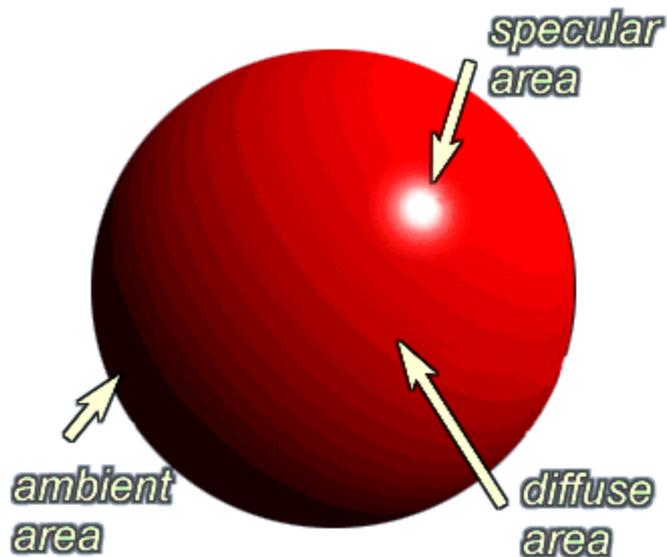
Foley – Van Dam – Feiner – Hughes - Phillips

Resumen del capítulo

- Modelos simples de iluminación.
- Modelos de sombreado más comunes.
- Modelos de sombreado aplicados a texturas.
- Efectos de transparencia, refracción, reflexión, sombras.
- Modelos de iluminación global.
- Ductos Gráficos.

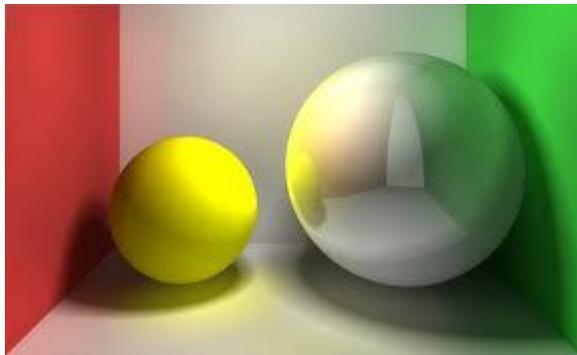
Modelo de iluminación simple

Sólo influyen la superficie en cuestión, las luces y la posición del observador (cálculos simples).



Modelos de iluminación complejos

Las sombras, las interreflexiones y la difusión de la luz en otros medios se calculan por otros algoritmos (implican cálculos complejos).



Modelo de iluminación simple

Pero igualmente, con un modelo de iluminación simple + trucos se puede dar ilusión de realismo.



Ecuación de Iluminación

Luz reflejada por un objeto en un punto de su superficie hacia una dirección particular (radiancia: (W/sr·m²)).

$$I_{(\text{intensidad})} = \text{LUZ}_{\text{ambiental}} + \text{LUZ}_{\text{difusa}} + \text{LUZ}_{\text{especular}}$$

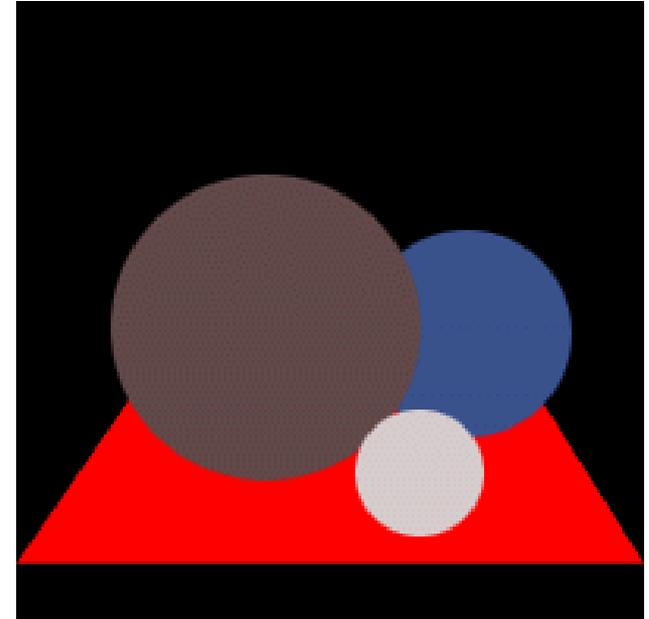
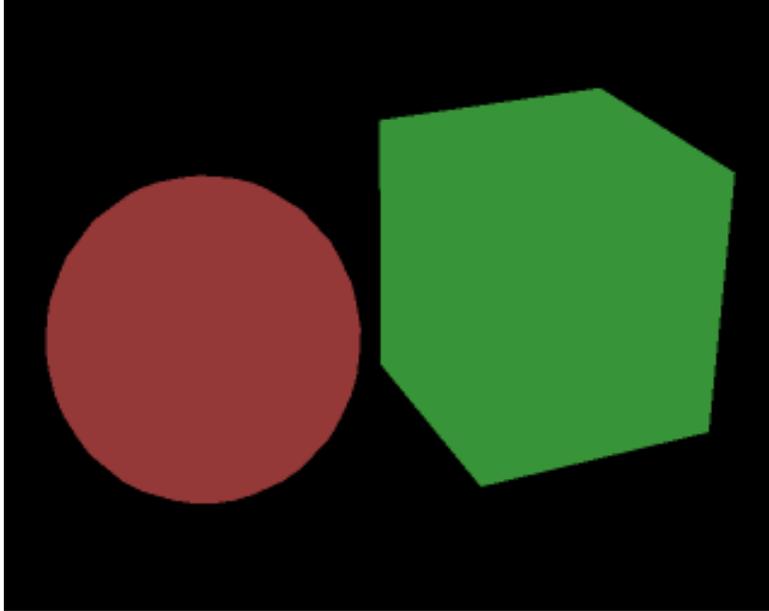
luz y superficie coloreada.

$$I_{(\text{intensidad},\lambda)} = \text{LUZ}_{\text{ambiental},\lambda} + \text{LUZ}_{\text{difusa},\lambda} + \text{LUZ}_{\text{especular},\lambda}$$

múltiples fuentes de luz.

$$I_{(\text{intensidad},\lambda)} = \text{LUZ}_{\text{ambiental},\lambda} + \sum_L (\text{LUZ}_{\text{difusa},\lambda,L} + \text{LUZ}_{\text{especular},\lambda,L})$$

Luz Ambiental



Los objetos no tienen fuente de luz externa.

Cada objeto tiene una silueta monocromática.

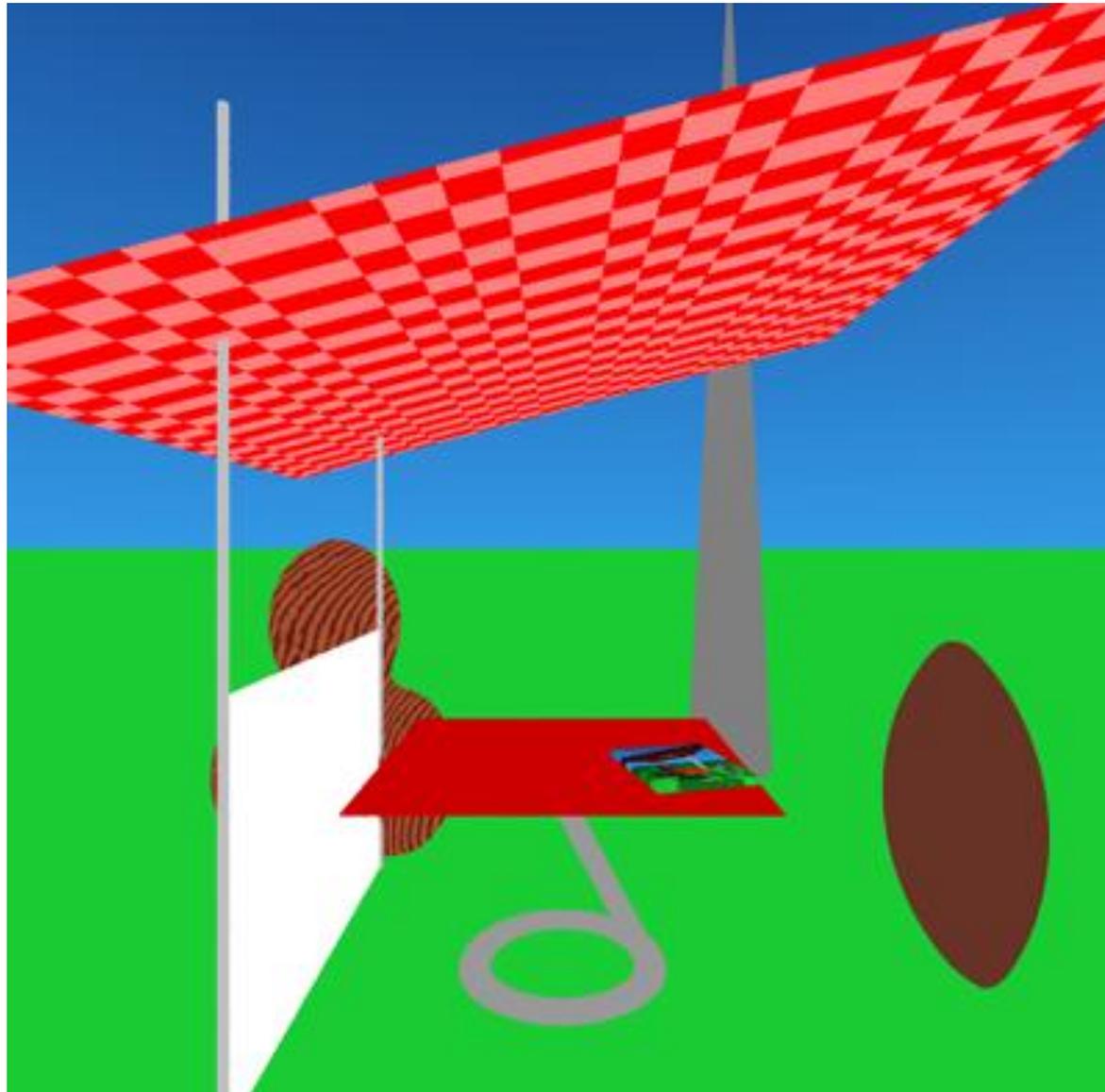
$I = k_j$, valor entre 0 y 1

$I = I_a k_a$,

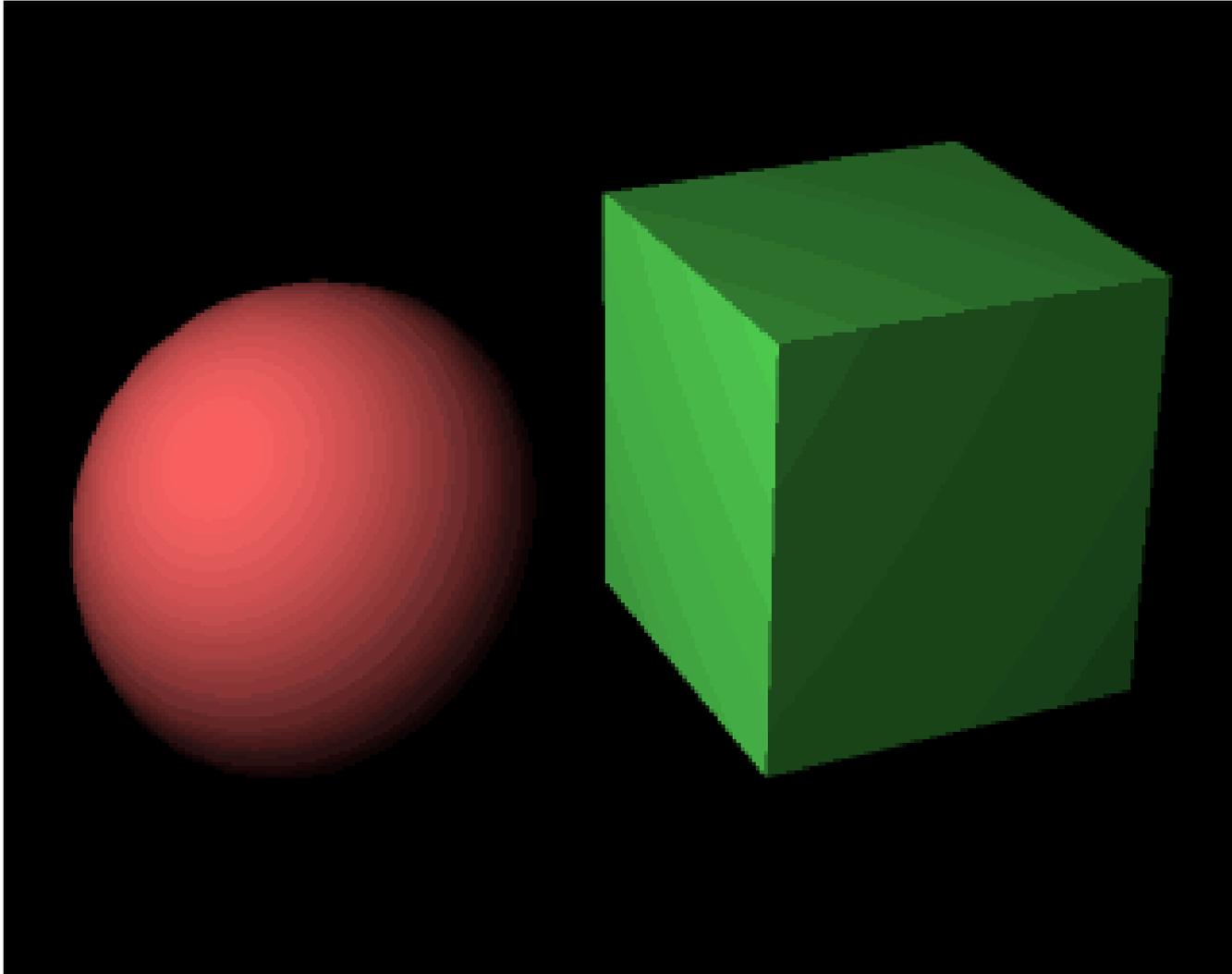
I_a = *Intensidad de la luz ambiental,*

k_a = *cantidad de luz ambiental de un objeto.*

Luz Ambiental



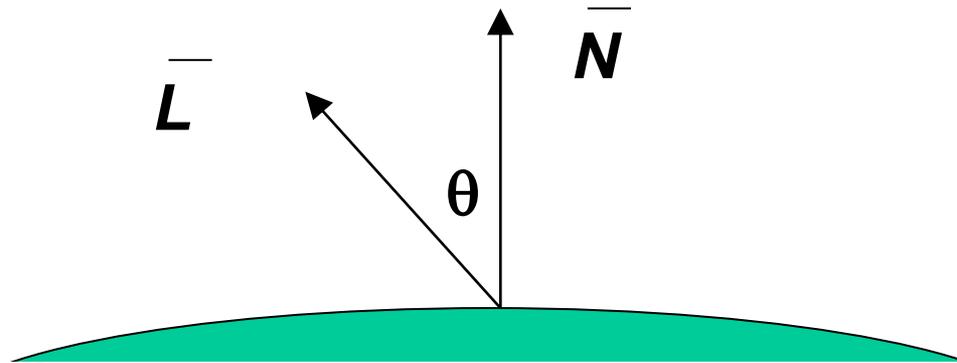
Reflexión difusa



Reflexión difusa

Fuente luminosa puntual: la luz emana en todas direcciones a partir de un solo punto.

Objetos de brillantez variable: la misma depende de la dirección y la distancia respecto a la fuente luminosa.

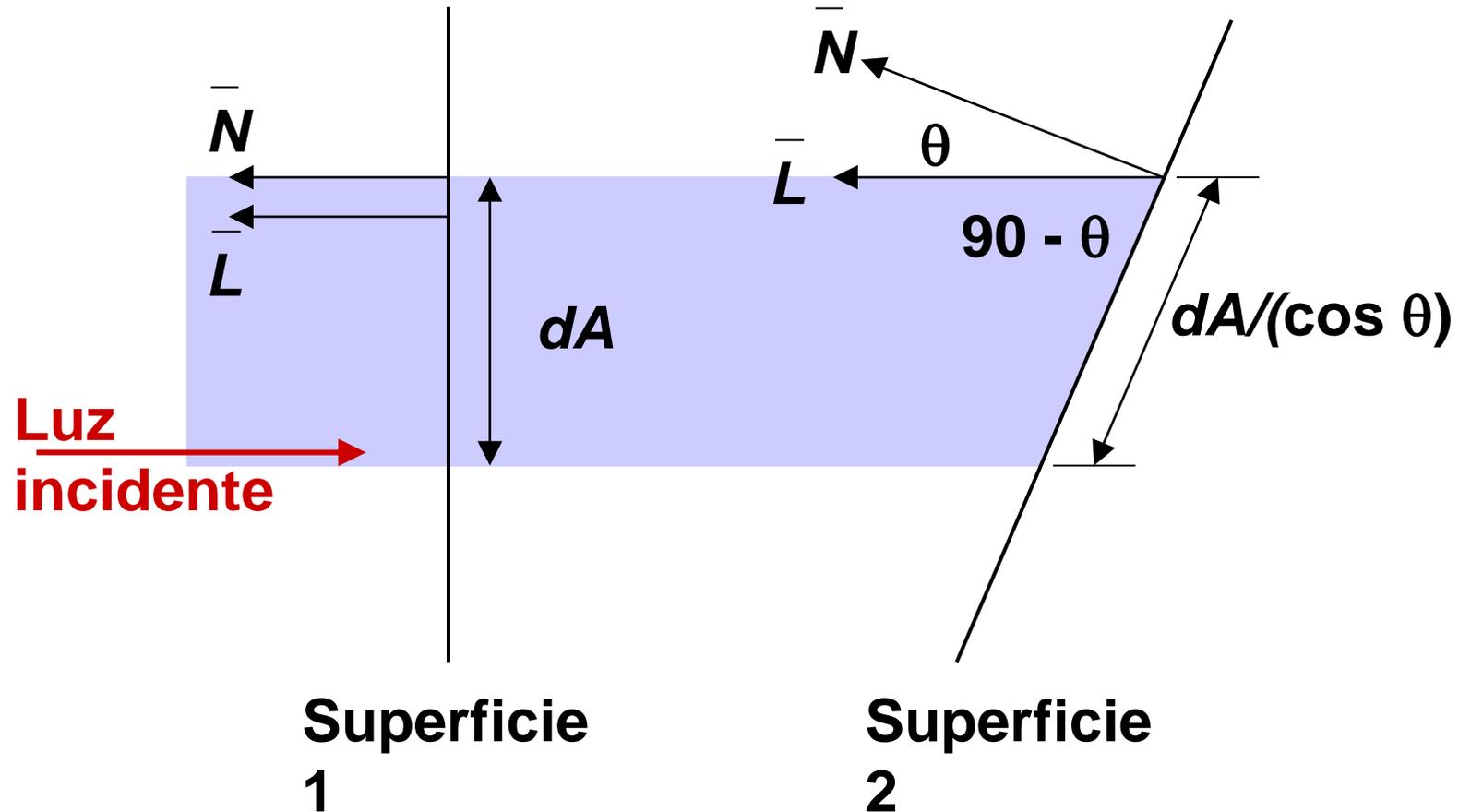


$$I = I_p k_d (N \cdot L)$$

I_p es intensidad de luz difusa.

k_d es cantidad de luz difusa del objeto.

Reflexión difusa (reflexión lambertiana)

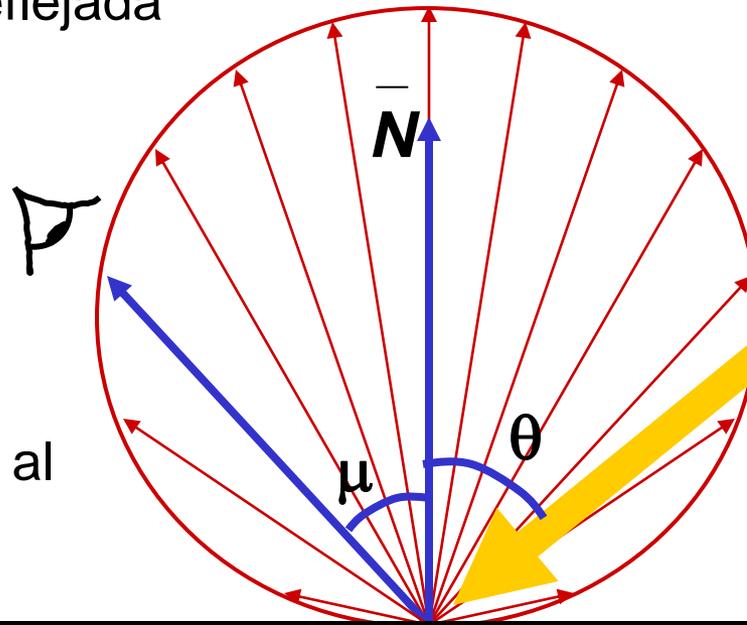


La luz que cae en dA es **directamente** proporcional a $\cos(\theta)$. Esto se aplica a cualquier superficie.

Reflexión difusa (reflexión lambertiana)

Distribución de intensidades
de la Luz reflejada

Luz incidente

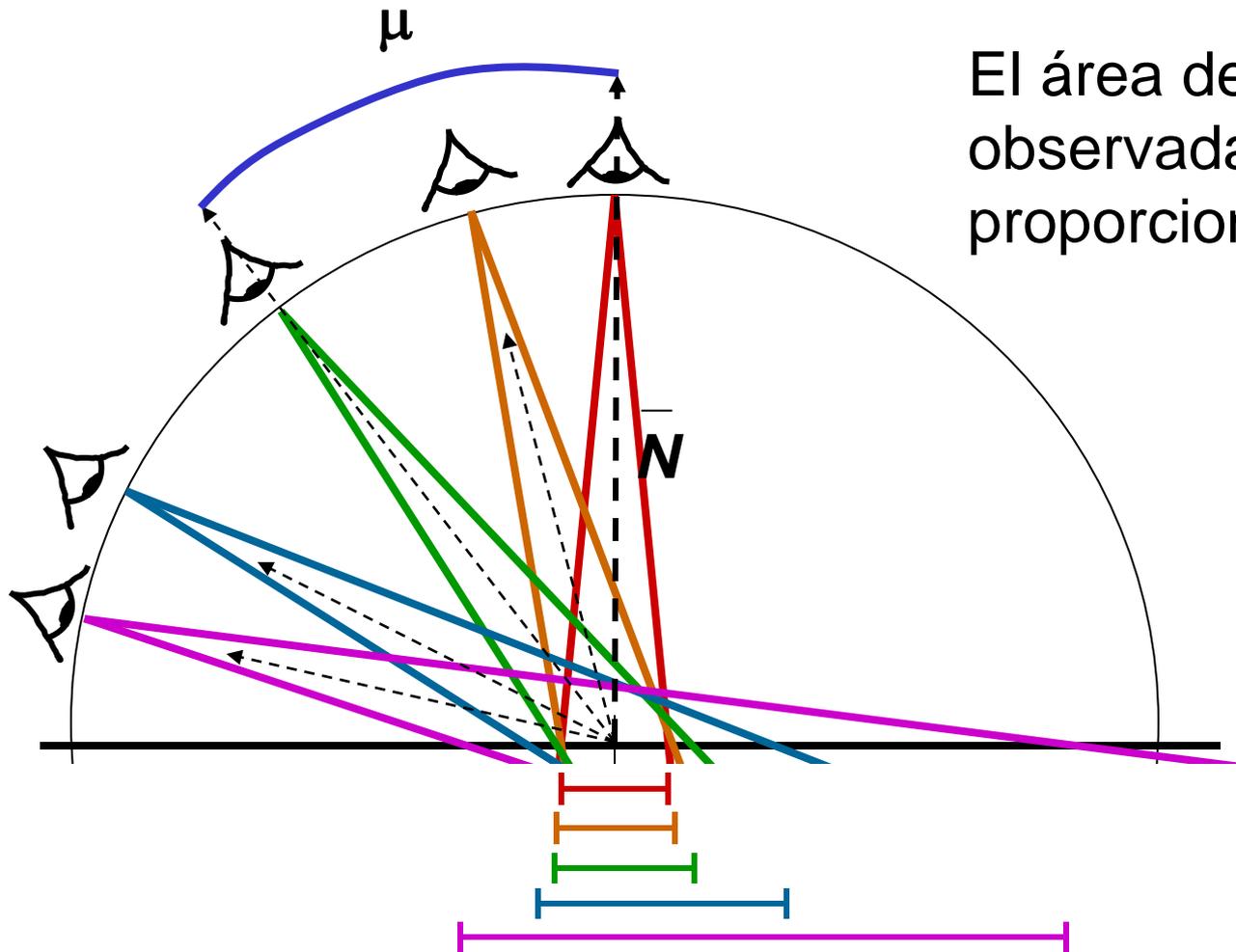


La intensidad de luz
observada es
directamente propor. al
 $\cos(\mu)$

El ángulo θ de la luz
incidente no influye
en la forma de la
distribución, sí en la
intensidad.

La intensidad que le llega al observador es **directamente** proporcional a $\cos(\theta)$. Esto es particular de las superficies Lambertianas

Reflexión difusa (reflexión lambertiana)



El área de superficie observada es **inversamente** proporcional a $\cos(\mu)$

Reflexión difusa (reflexión lambertiana)

Tenemos 2 efectos contrapuestos:

1) La intensidad de luz observada es **directamente** proporcional a $\cos(\mu)$.

2) La cantidad de área de superficie observada es **inversamente** proporcional a $\cos(\mu)$.

=>Ambas cantidades se compensan.

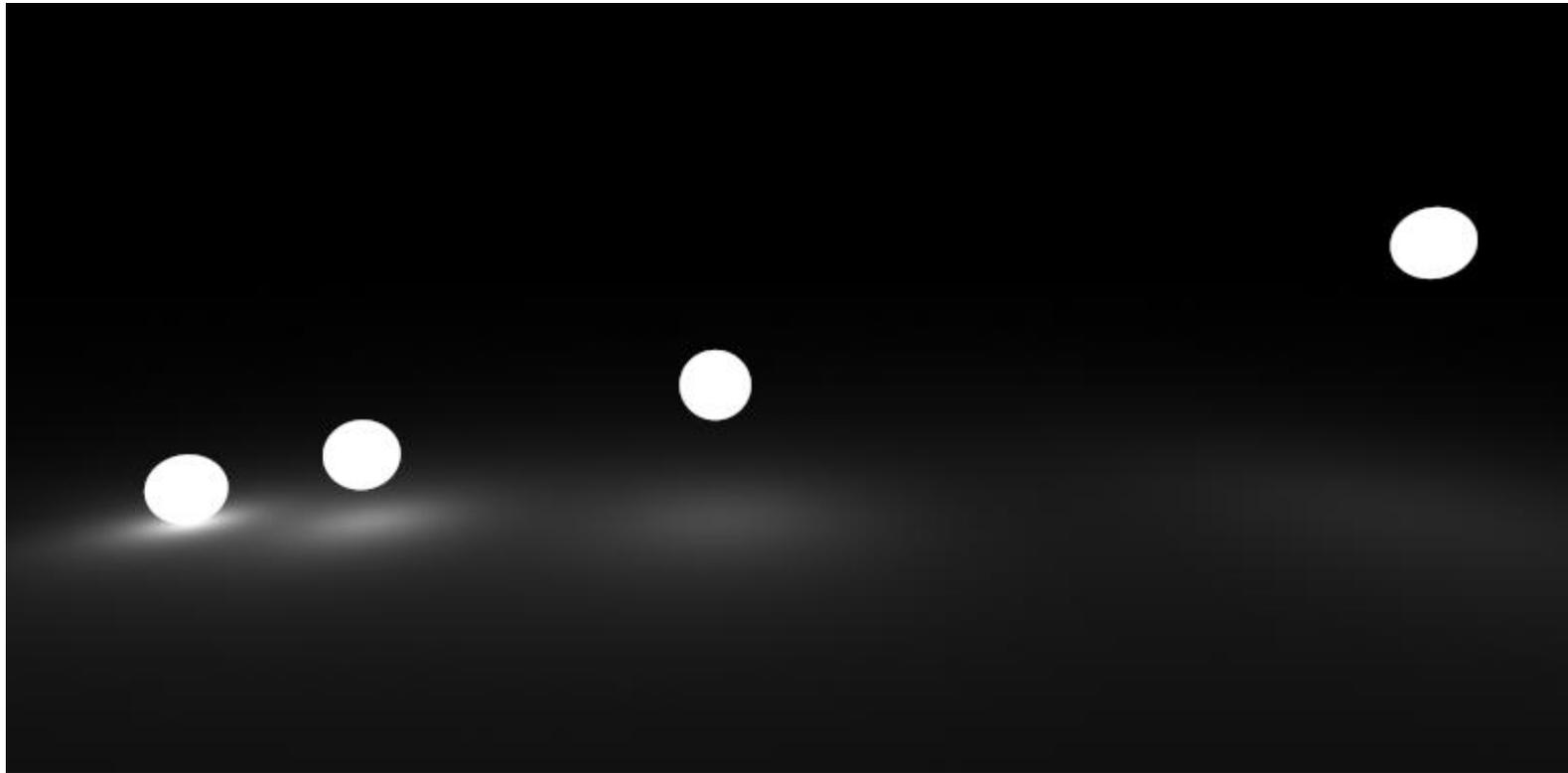
Por tanto, **para las superficies lambertianas, la cantidad de luz que ve el observador es independiente de la dirección de éste** y sólo es proporcional a $\cos(\theta)$, donde θ es el ángulo de incidencia de la luz.

Reflexión ambiental + difusa

$$I = I_a k_a + I_p k_d (N \cdot L)$$

Atenuación de la fuente luminosa

Factor de Atenuación debido a la distancia entre la fuente puntual y la superficie.



Atenuación de la fuente luminosa

Factor de Atenuación debido a la distancia entre la fuente puntual y la superficie.

$$I = I_a k_a + f_{att} I_p k_d (N \cdot L)$$

Fórmulas de f_{att}

$f_{att} = 1/d_L^2$ (donde d_L^2 es la distancia a la fuente luminosa)

$$f_{att} = \min \left(\frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1 \right)$$

Luces y superficies coloreadas

La intensidad de la luz puntual se descompone (de forma simple) en 3 valores: I_{pR} , I_{pG} , I_{pB}

Idem con la luz ambiente: I_{aR} , I_{aG} , I_{aB}

Idem con el color del objeto: O_{pR} , O_{pG} , O_{pB}

⇒

$$I_R = I_{aR}k_a O_{dR} + f_{att} I_{pR}k_d O_{dR}(N \cdot L)$$

$$I_G = I_{aG}k_a O_{dG} + f_{att} I_{pG}k_d O_{dG}(N \cdot L)$$

$$I_B = I_{aB}k_a O_{dB} + f_{att} I_{pB}k_d O_{dB}(N \cdot L)$$

Luces y superficies coloreadas

Modelo más realista, que trabaja directamente con λ .

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} k_d O_{d\lambda} (N \cdot L)$$

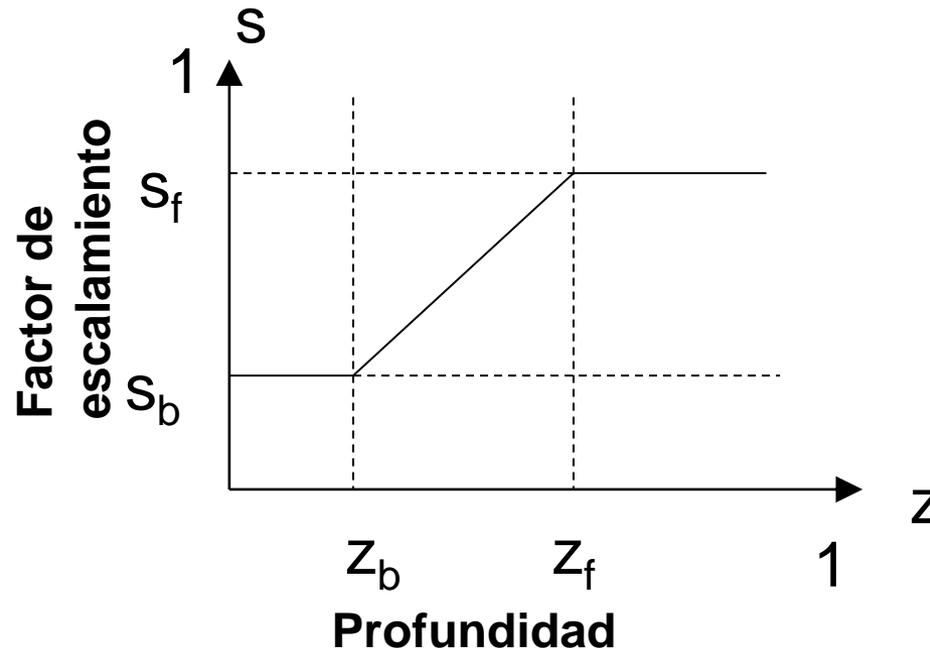
Atenuación Atmosférica

Los objetos más lejanos al observador se generan con menor intensidad que los más cercanos.

s_o = factor de escalamiento del objeto.

I_λ = intensidad inicial del objeto.

$I_{dc\lambda}$ = intensidad indicadora de profundidad.



Atenuación Atmosférica

$$\text{Si } z_o < z_b \quad \Rightarrow \quad s_o = s_b$$

$$\text{Si } z_o > z_f \quad \Rightarrow \quad s_o = s_f$$

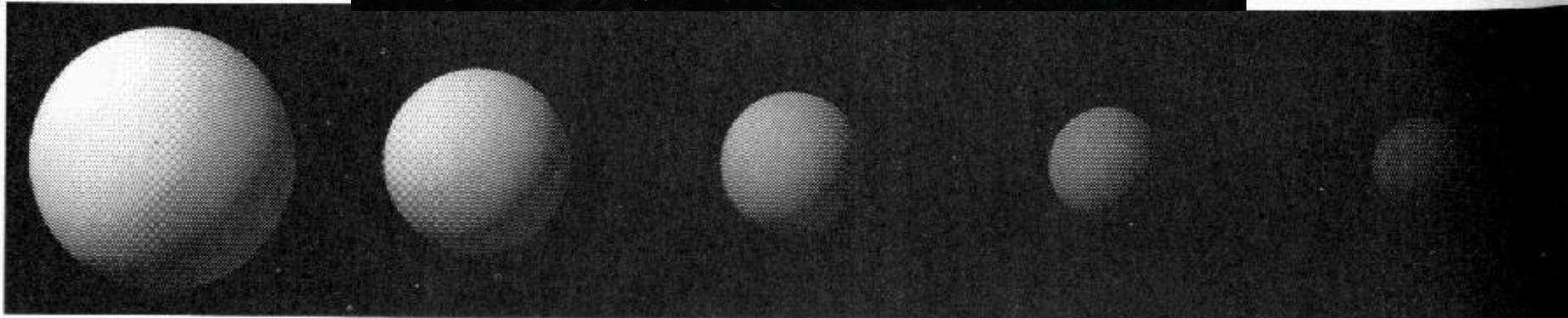
En otros casos:

$$s_o = s_b + (z_o - z_b)(s_f - s_b)/(z_f - z_b)$$

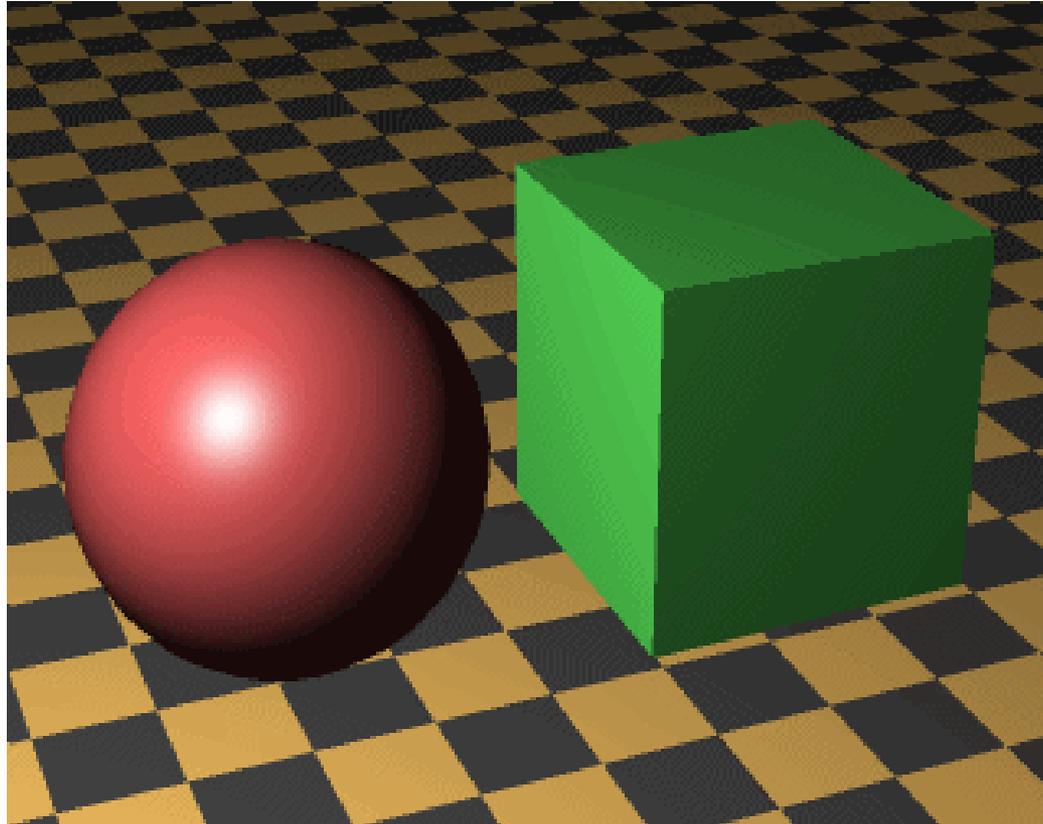
Por tanto, la intensidad del objeto es:

$$I'_\lambda = s_o I_\lambda + (1 - s_o) I_{dc\lambda}$$

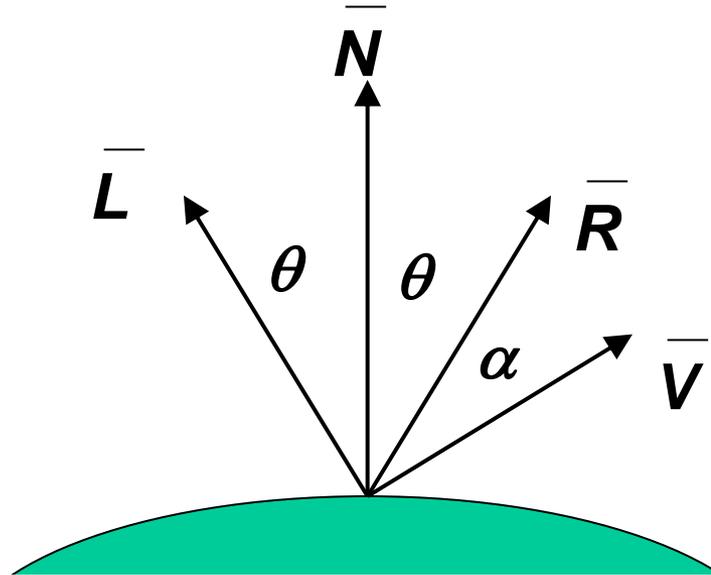
Atenuación Atmosférica



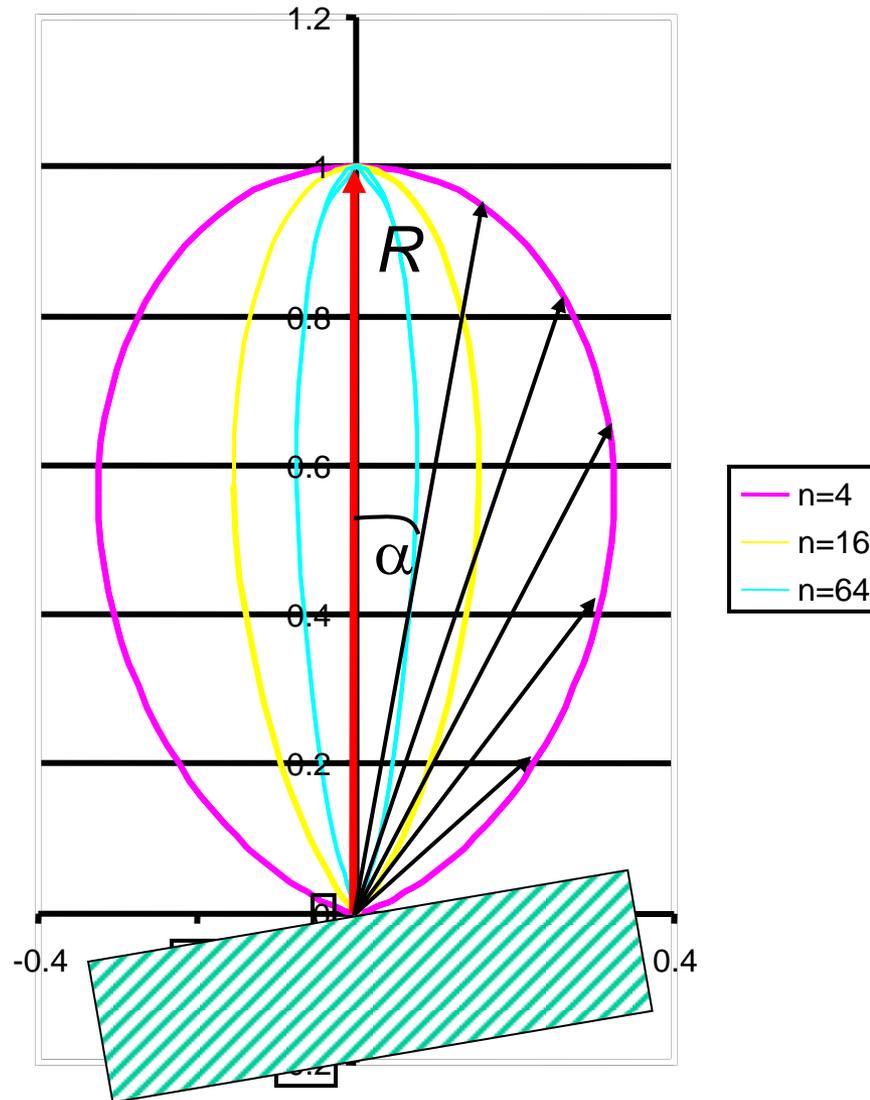
Reflexión Especular



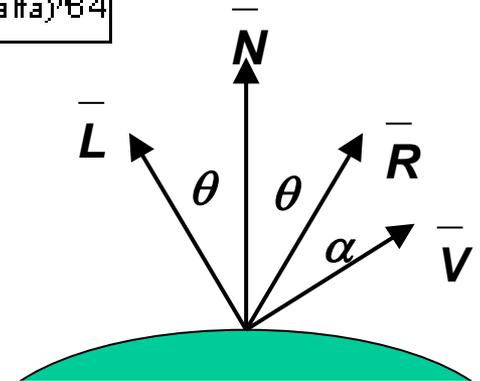
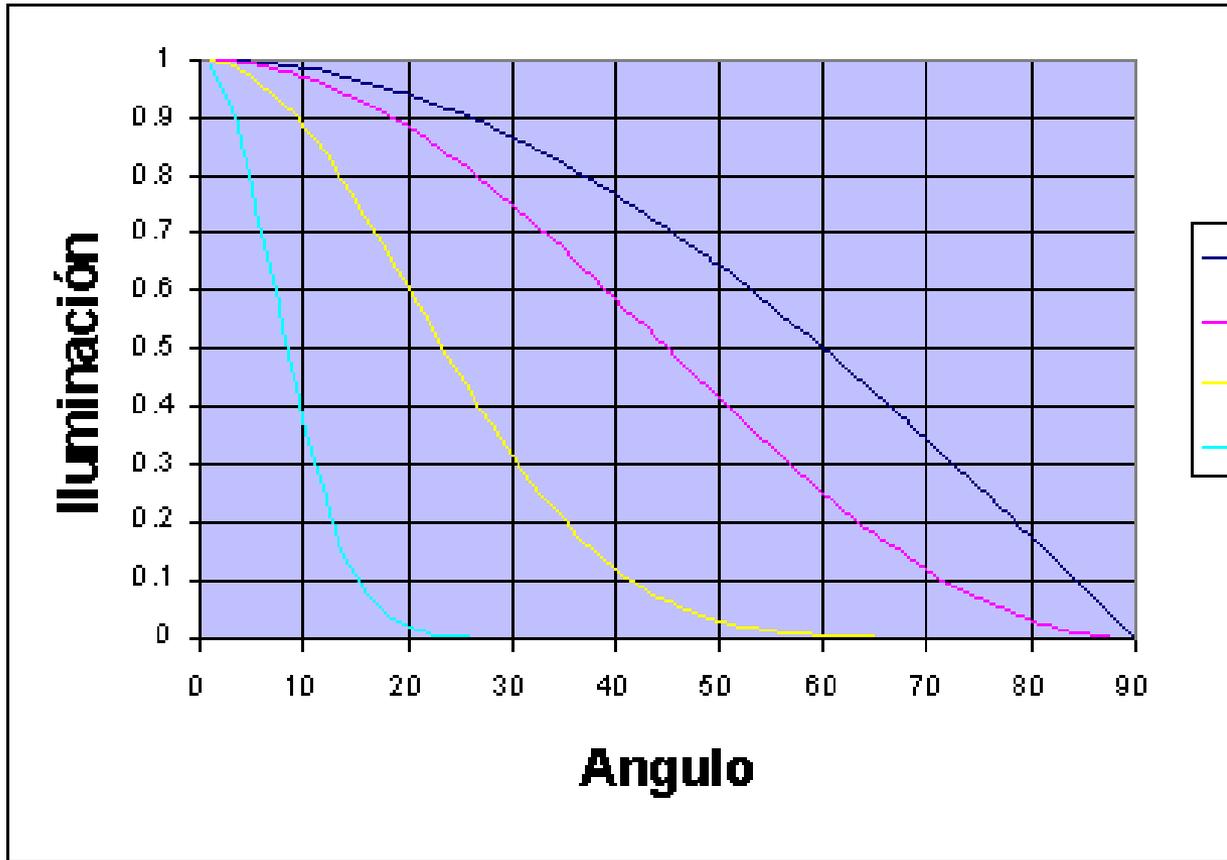
Reflexión Especular



Reflexión Especular (modelo de Phong)



Reflexión Especular (modelo de Phong)



A la ecuación de intensidad se le suma el término

$$(\cos \alpha)^n = (\vec{R} \cdot \vec{V})^n$$

Reflexión Especular (modelo de Phong)

Es un modelo para reflectores imperfectos.

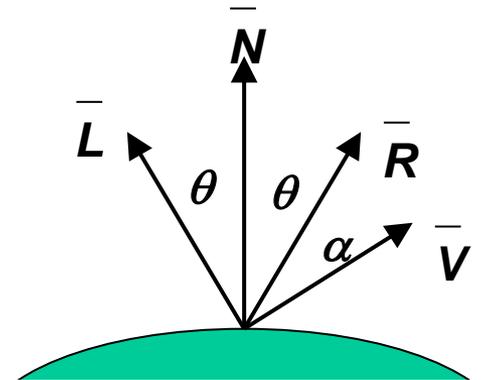
Hay un componente n , llamado exponente de reflexión especular.

$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} (N \cdot L) + W(\theta) (R \cdot V)^n]$$

donde $W(\theta)$ es la fracción de luz reflejada especularmente.

Si considero $W(\theta)$ constante = k_s

Si $O_{s\lambda}$ es el color especular del objeto



$$I_\lambda = I_{a\lambda} k_a O_{d\lambda} + f_{att} I_{p\lambda} [k_d O_{d\lambda} (N \cdot L) + k_s O_{s\lambda} (R \cdot V)^n]$$

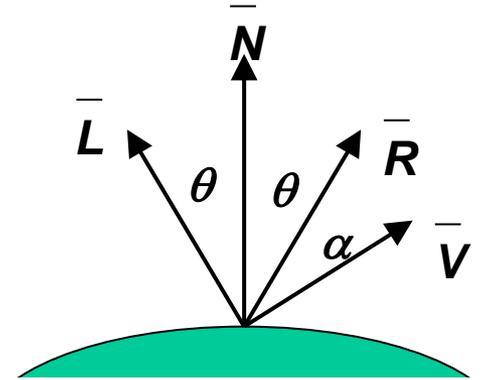
Reflexión Especular (modelo de Phong)

Cálculo de R

$$R = 2N(N \cdot L) - L$$

=>

$$R \cdot V = (2N(N \cdot L) - L) \cdot V$$



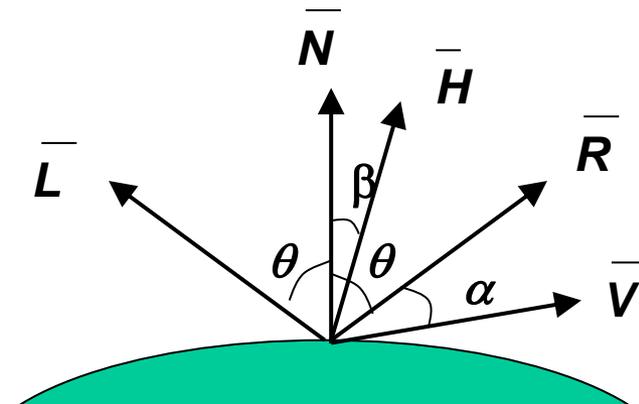
alternativa: $(N \cdot H)^n$ en lugar de $(R \cdot V)^n$

$H =$ vector intermedio entre L y V

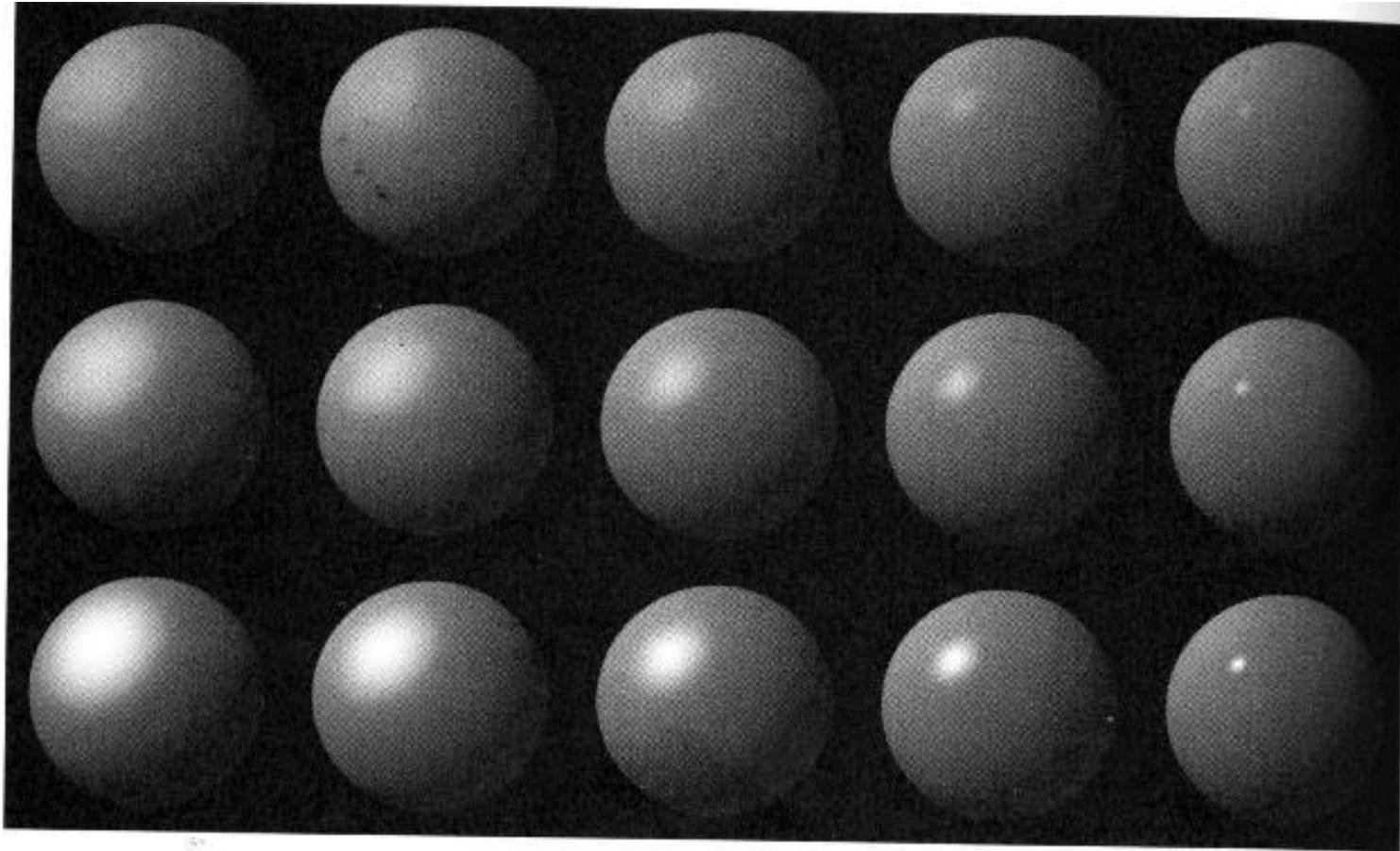
$$H = (L + V) / |L + V|$$

=>

$$(N \cdot H)^n \neq (R \cdot V)^n$$



Reflexión Especular (modelo de Phong)

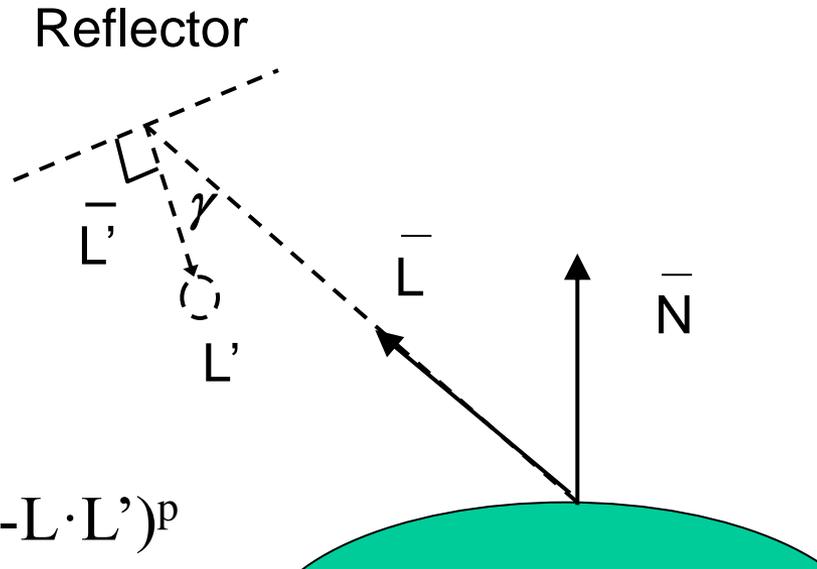


Todas las esferas tienen $l_a = l_p=1$, $k_a= 0.1$, $k_d=0.45$.

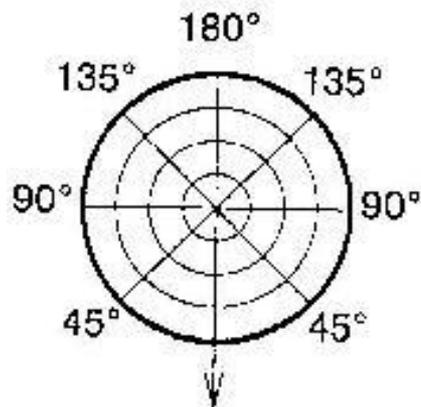
De izquierda a derecha, $n = 3.0, 5.0, 10.0, 27.0, 200.0$

De arriba a abajo, $k_s = 0.1, 0.25, 0.5$

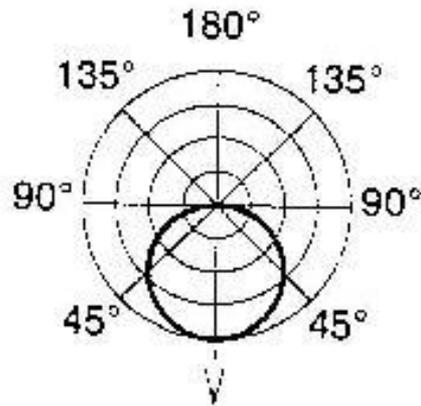
Modelo de Warn



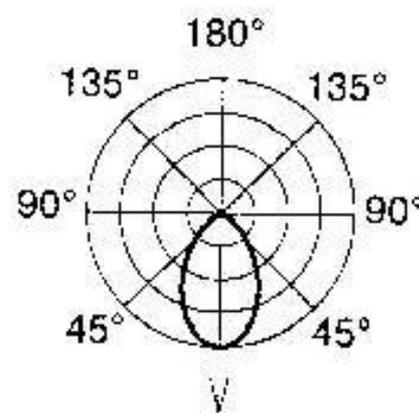
$$I_{L,\gamma} (\cos \gamma)^p = I_{L,\gamma} (-\bar{L} \cdot \bar{L}')^p$$



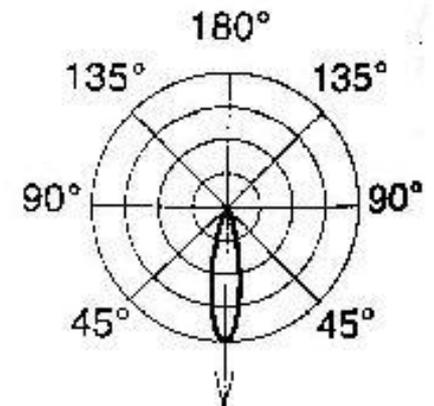
Uniformly radiating point source



$\cos \gamma$



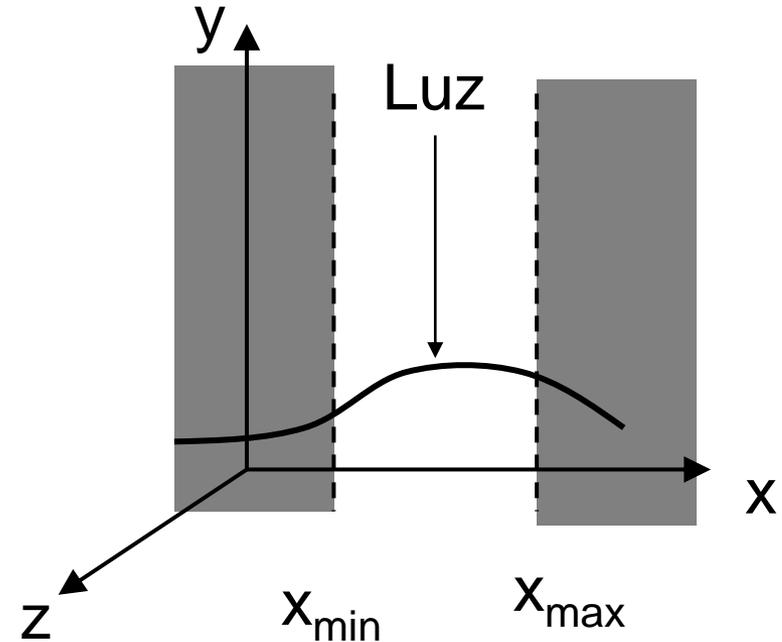
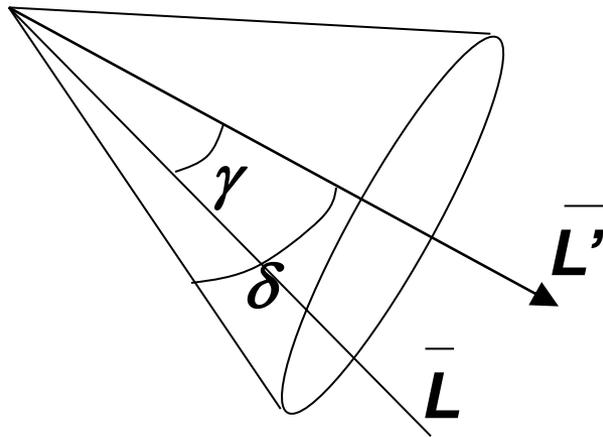
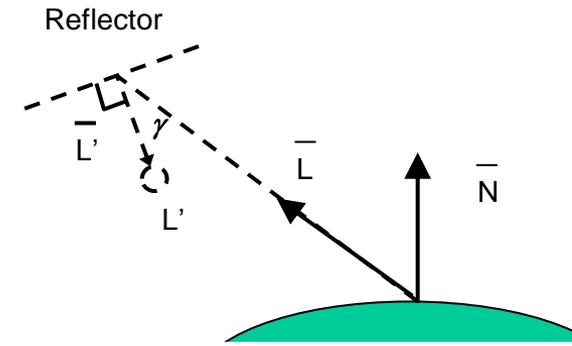
$\cos^4 \gamma$



$\cos^{32} \gamma$

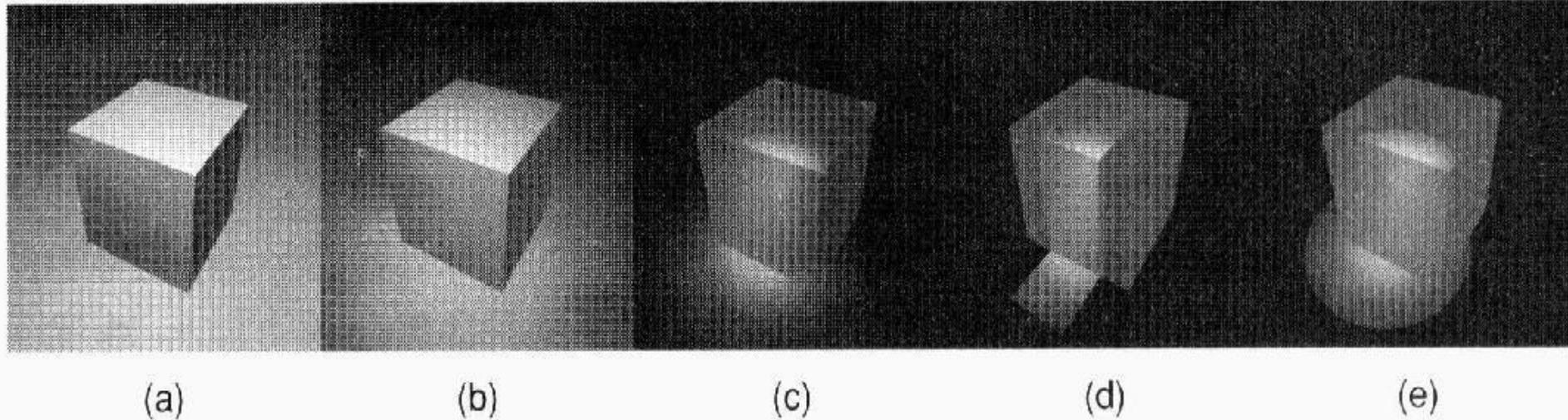
Modelo de Warn

Utilización de aletas y conos.



Modelo de Warn

Utilización de aletas y conos.



Fuentes luminosas múltiples

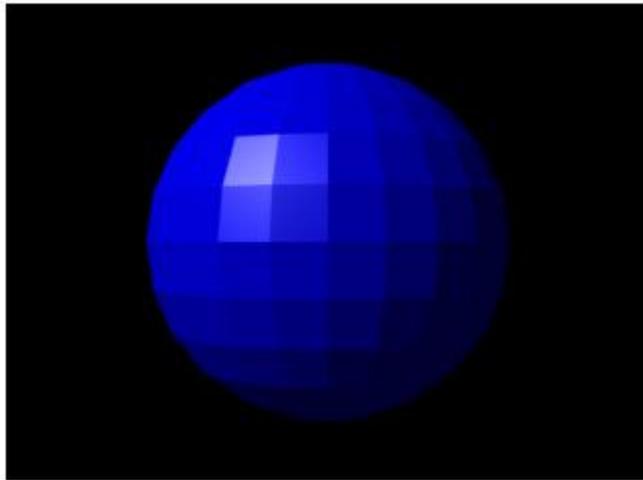
$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + \sum_{1 \leq i \leq m} f_{att_i} I_{p\lambda_i} \left[k_d O_{d\lambda} (\bar{N} \cdot \bar{L}_i) + k_s O_{s\lambda} (\bar{R}_i \cdot \bar{V})^n \right]$$

Se incrementa la posibilidad de problemas

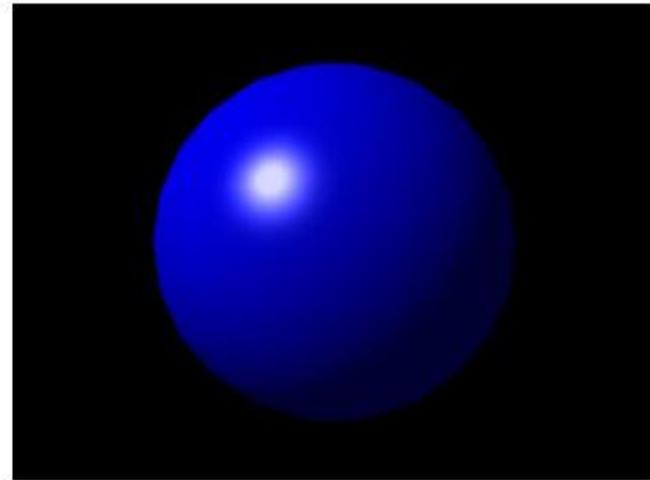
Es probable que I_{λ} supere el máximo permitido (1, o 255).

El control se puede hacer pixel a pixel o considerando toda la imagen.

Modelos de sombreado para polígonos



FLAT SHADING



PHONG SHADING

Modelos de sombreado para polígonos

Sombreado constante:

- aplica una sola vez un modelo de iluminación para todo el polígono.
- Esta simplificación sirve si:
 - La fuente luminosa está en el infinito, por tanto $N \cdot L$ es constante
 - El observador está en el infinito, por tanto $N \cdot V$ es constante en toda la cara del polígono.
 - El polígono representa la superficie real que se modela y no es una aproximación a una superficie curva.
- Si las suposiciones son incorrectas, entonces hay un método para determinar L y V

Sombreado Interpolado

En lugar de evaluar la ecuación de iluminación para cada pixel, esta se interpola linealmente sobre un triángulo a partir de los valores determinados para sus vértices.

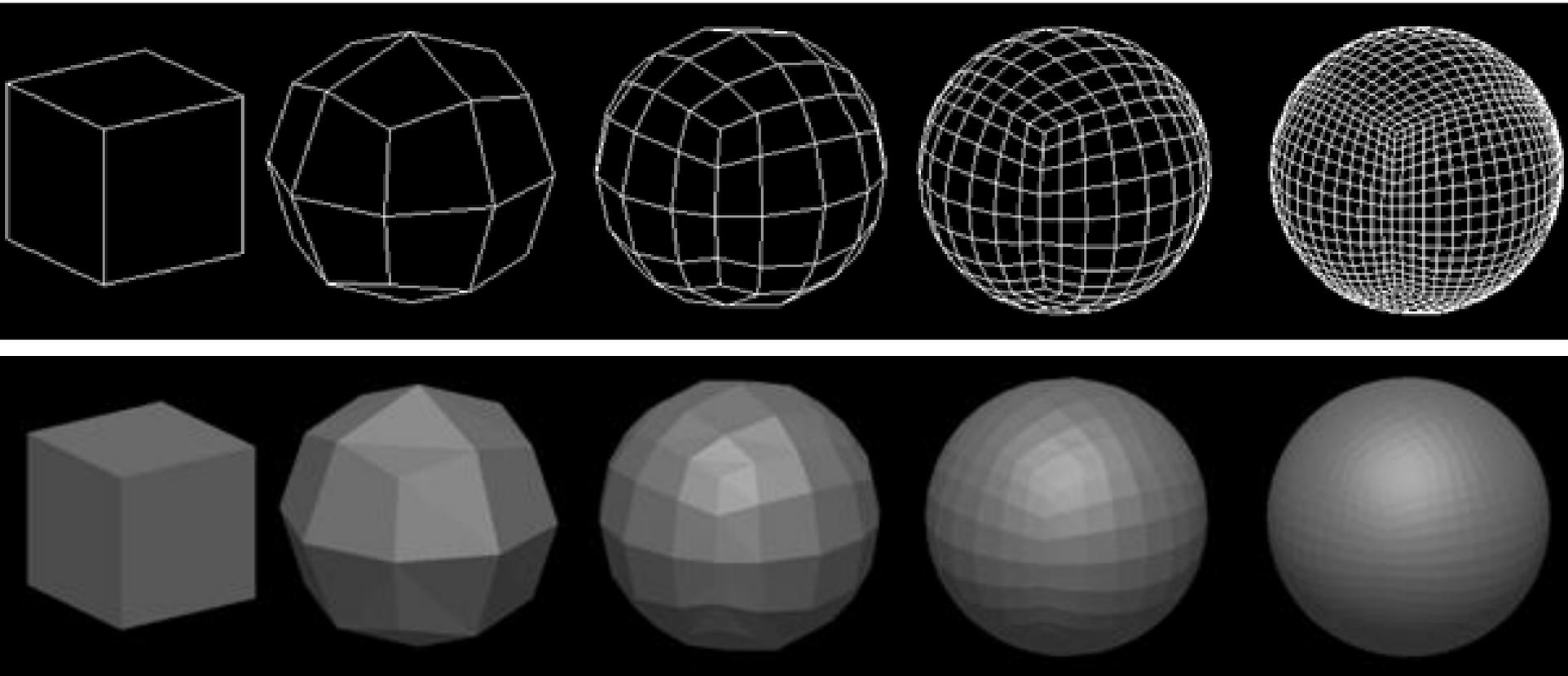
Se puede generalizar para otro tipo de polígonos.

A su vez, en lugar de realizar la interpolación para cada píxel, se puede hallar una ecuación de diferencia.

Esta interpolación no evita la apariencia facetada. Según el objeto a modelar, esto es positivo o no.

Sombreado de malla poligonal

Una superficie curva se puede aproximar a otra facetada (malla poligonal)
No se logran buenos resultados en la interpolación, aunque se trabaje con una densidad alta de polígonos.



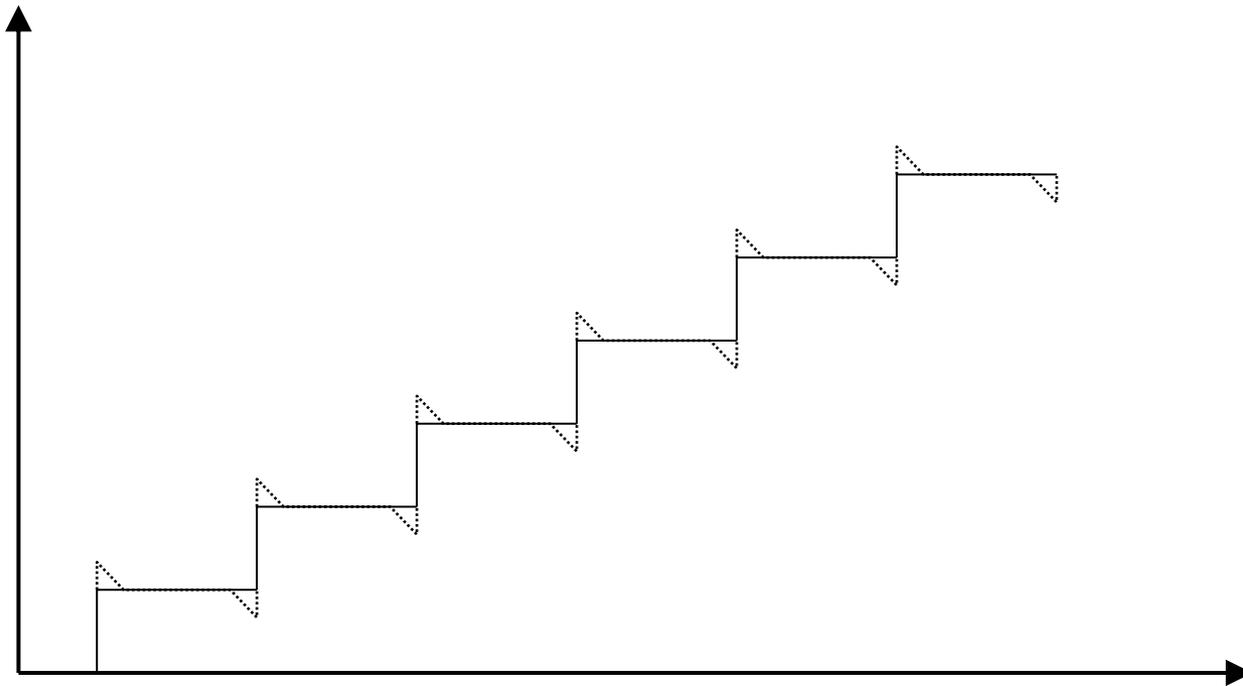
Sombreado de malla poligonal

Los problemas en la visualización de una superficie curva a través de una aproximación facetada, tienen su origen en el efecto de banda de Mach.



Sombreado de malla poligonal

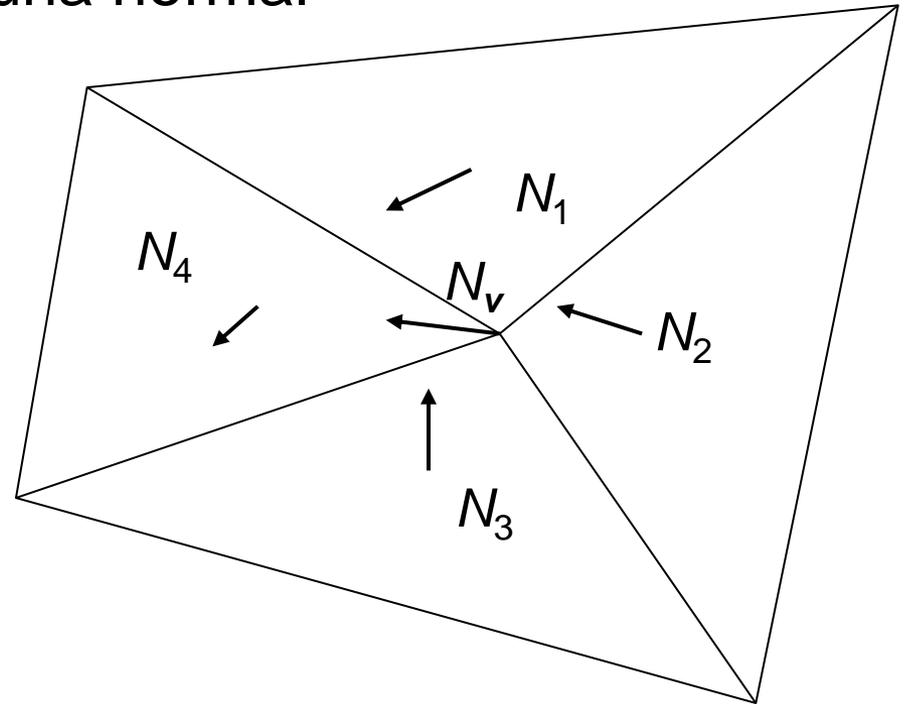
Esquema de las intensidades reales y las percibidas.



Sombreado de Gouraud

1) A cada vértice se le asigna una normal

$$\bar{N}_v = \frac{\sum_{1 \leq i \leq n} \bar{N}_i}{\left| \sum_{1 \leq i \leq n} \bar{N}_i \right|}$$

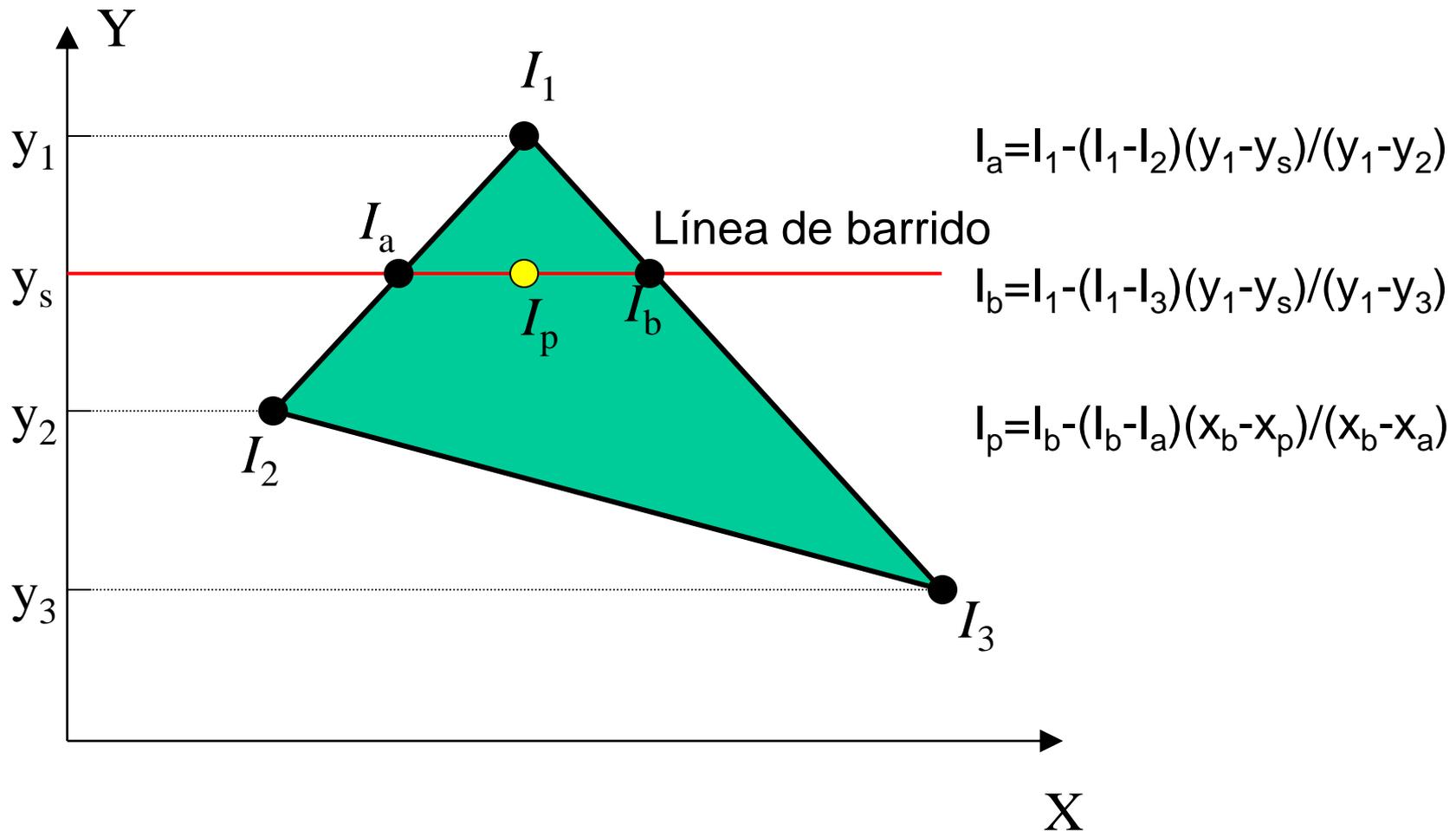


2) Se calculan las intensidades de los vértices usando algún modelo de iluminación ya visto.

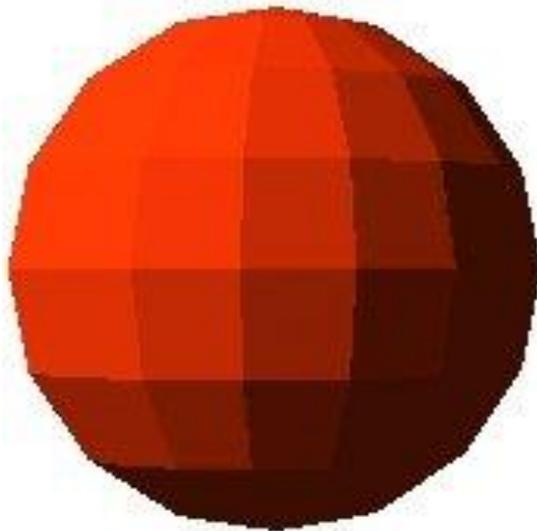
3) Se interpola la intensidad en cada píxel del polígono.

Sombreado de Gouraud

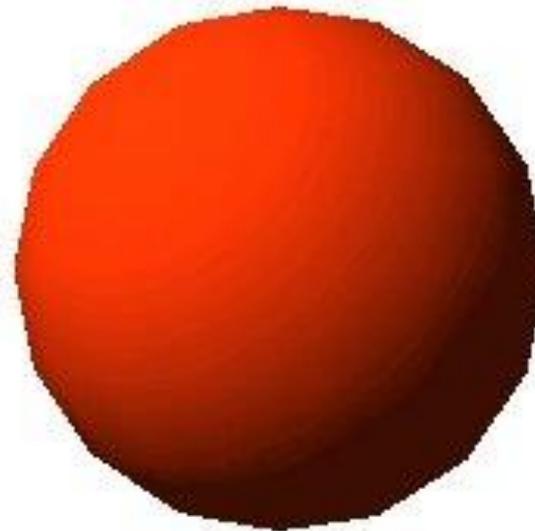
3) Se interpola la intensidad en cada píxel del polígono.



Sombreado de Gouraud



Flat



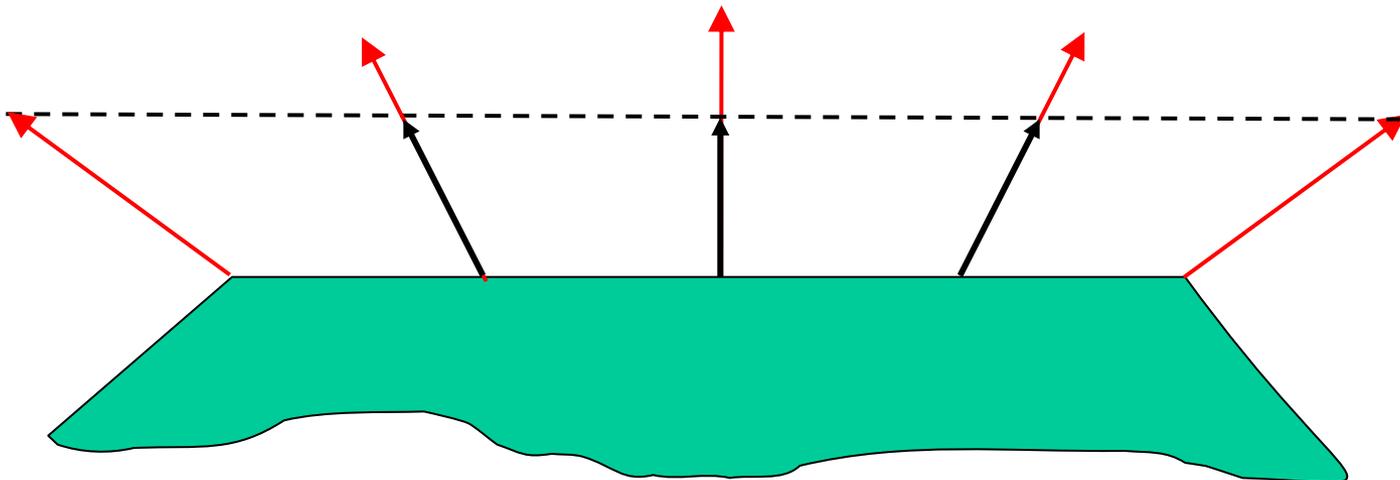
Gouraud

Sombreado de Phong

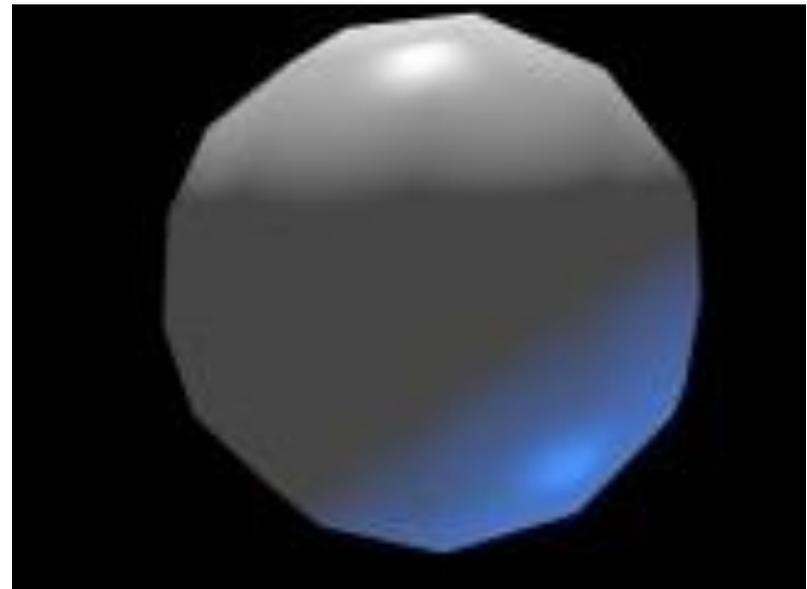
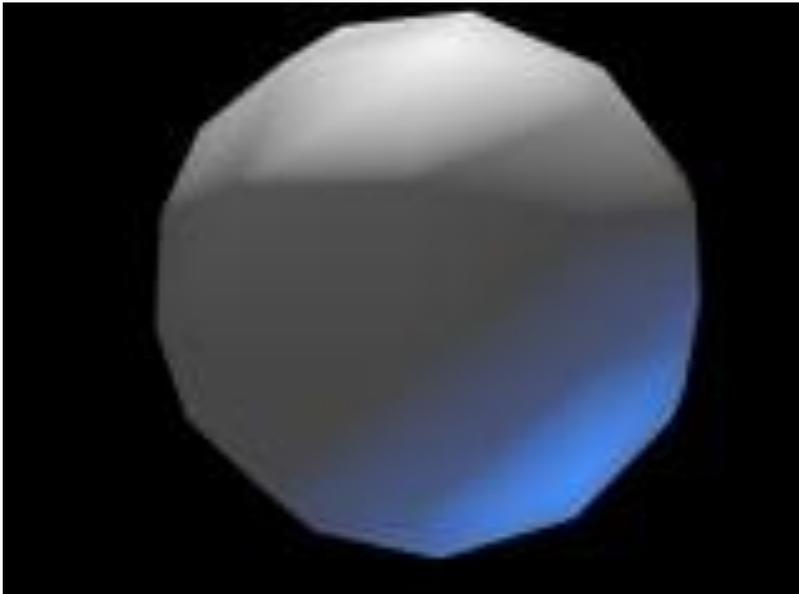
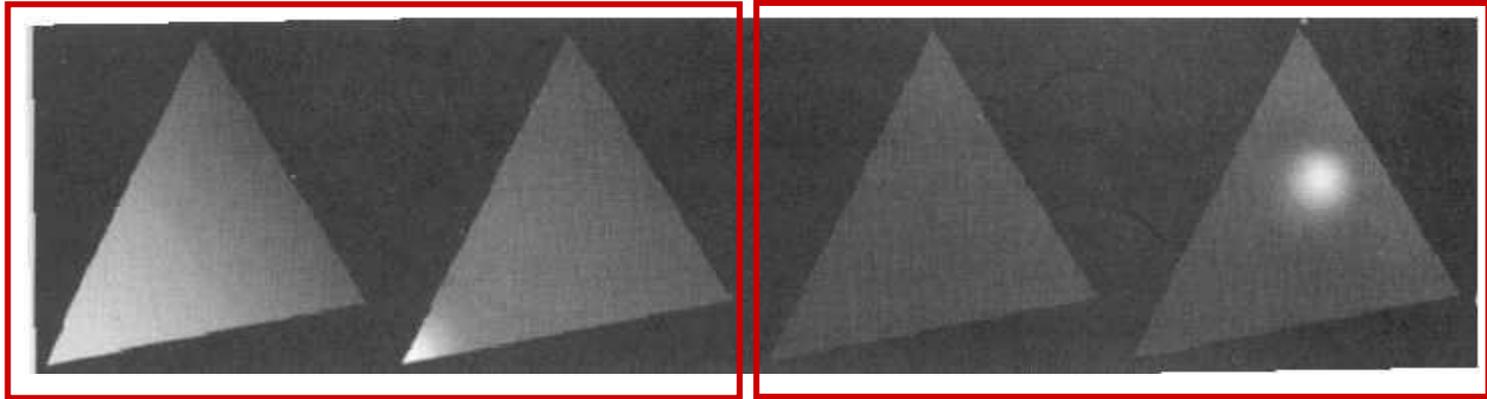
En lugar de interpolar la intensidad del vertice (Gouraud), se interpola y normaliza la normal a los vértices.

Si se utiliza sombreado de Phong con n alto, la diferencia entre Phong y Gouraud puede llegar a ser notable.

Normalizar un vector es costoso, y aplicar un modelo de iluminación a cada pixel también puede serlo.

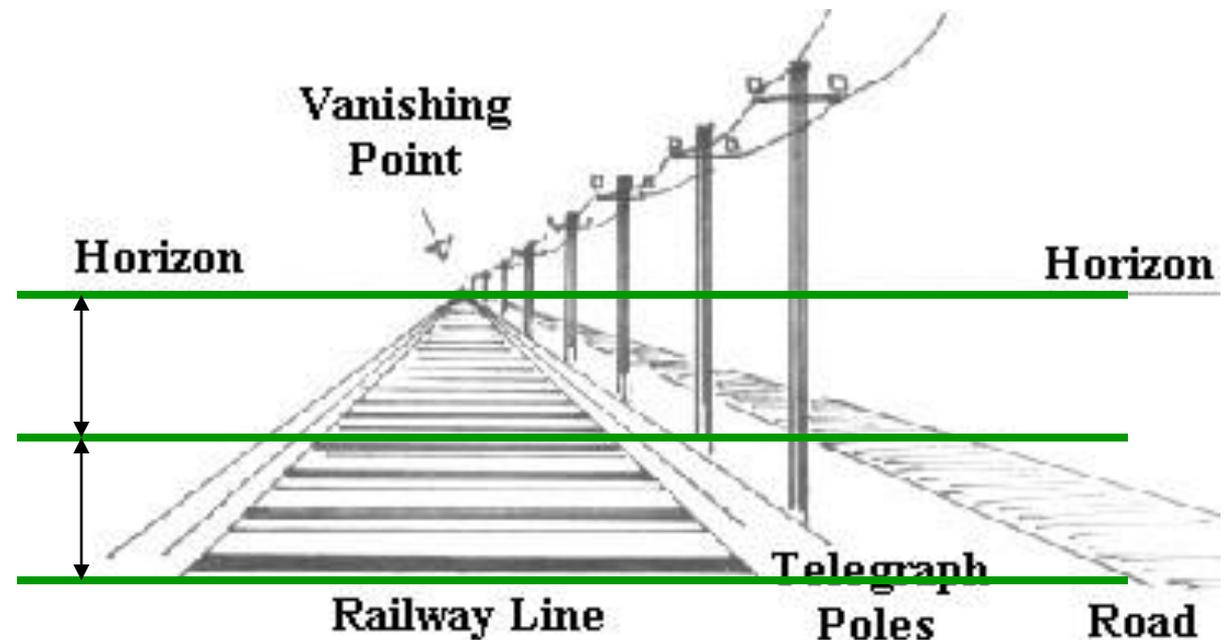


Sombreado de Phong



Problemas con sombreado interpolado

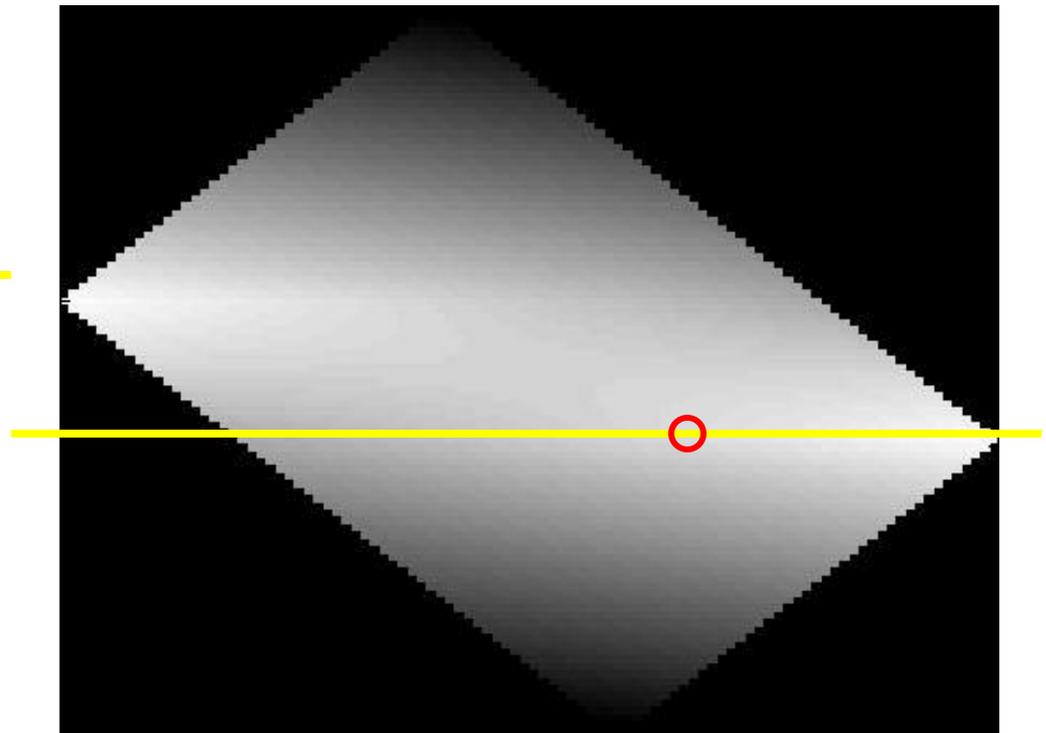
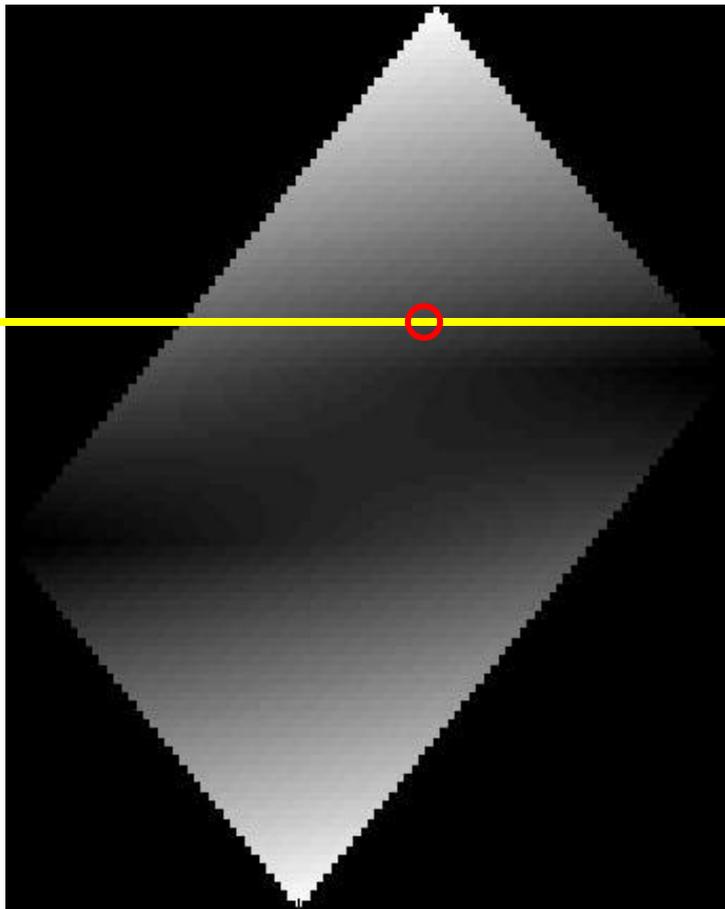
- 1) ***Silueta poligonal***
- 2) ***Distorsión de perspectiva.*** (las interpolaciones se hacen sobre la proyección en pantalla, por tanto no consideran la perspectiva).



Distancias iguales en la **y** no se corresponde con distancias iguales en la **z**.

Problemas con sombreado interpolado

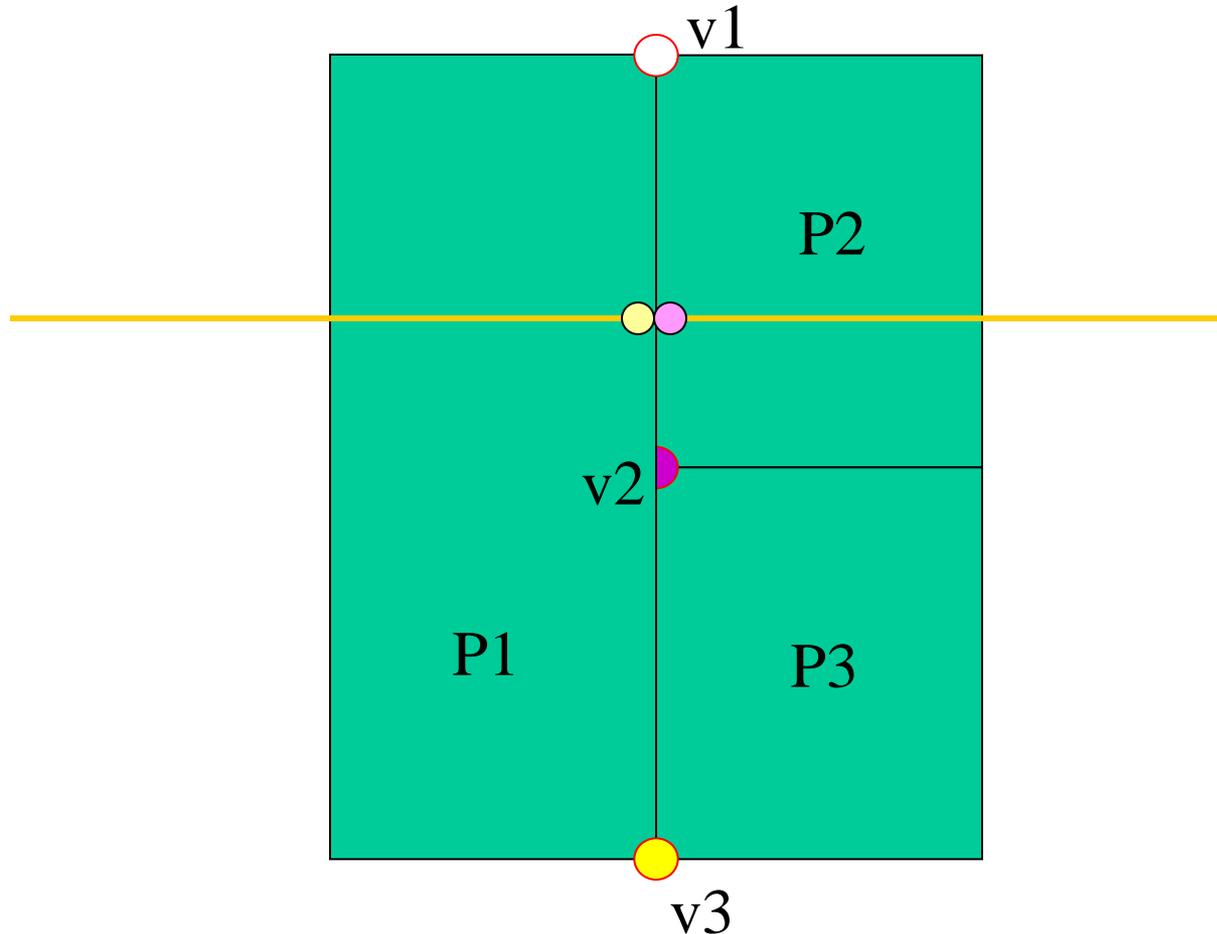
3) ***Dependencia de la orientación.*** Las líneas de rastreo son siempre horizontales, pero el polígono interpolado puede cambiar su orientación.



Problemas con sombreado interpolado

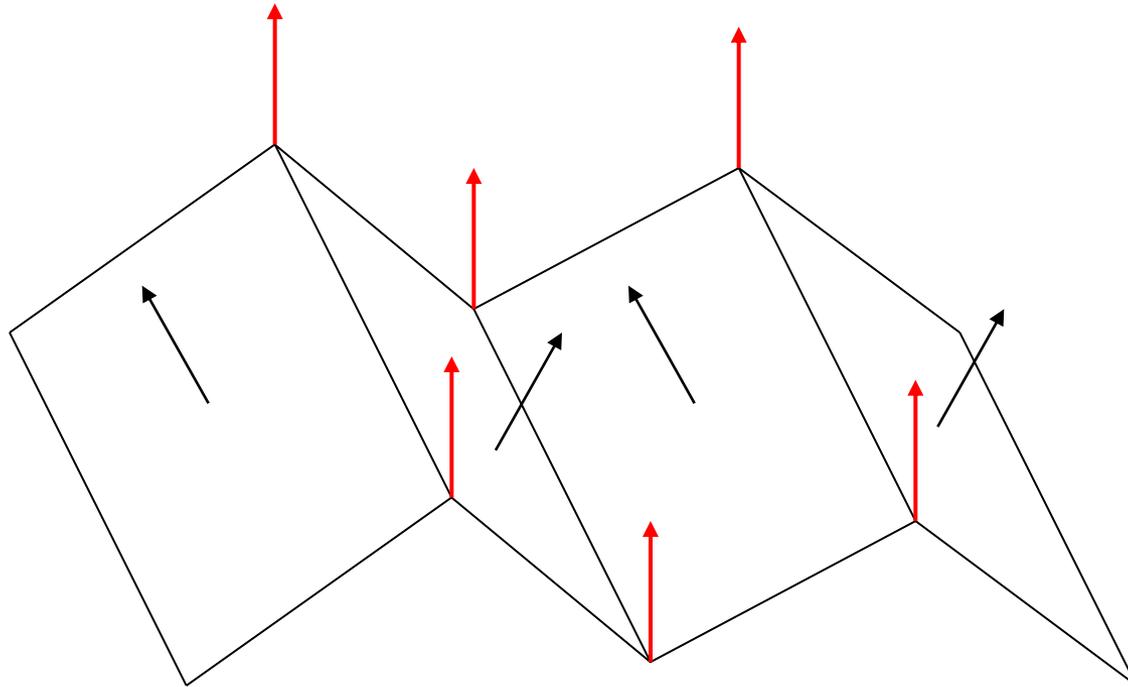
4) *Problemas en vértices compartidos.*

Se genera discontinuidad entre píxeles adyacentes.



Problemas con sombreado interpolado

5) *Normales a vértices que no son representativas.*



Detalle de superficie

Las superficies vistas son planas o bicúbicas. Las mismas son suaves y uniformes, lo cual no se ajusta a la mayoría de las superficies reales.

Hay algunos métodos para salvar estos detalles faltantes.

Polígonos de detalle de superficie

Correspondencia de Texturas

Correspondencia de Protuberancias

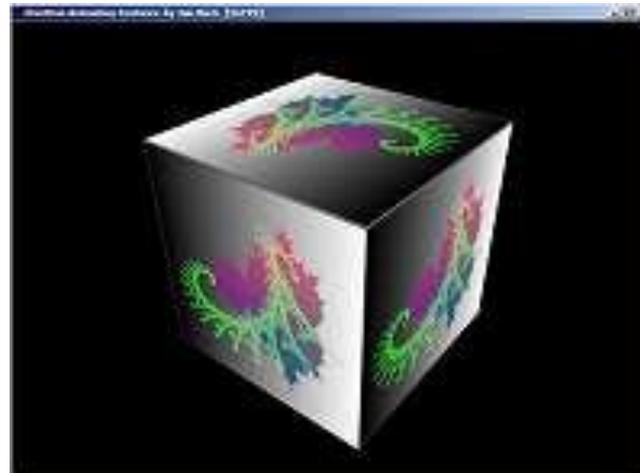
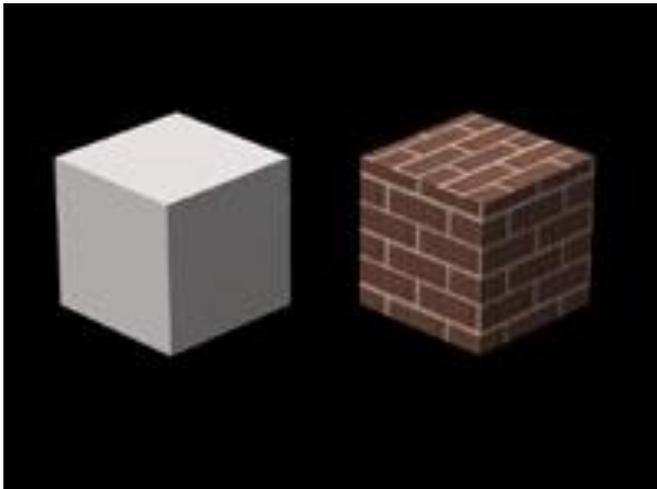
Otros métodos

Polígonos de detalle de superficie

- A la casa del capítulo 6, se le pueden agregar objetos, como ser: ventanas, puertas, letras, etc.
- Esto se hace a través de “polígonos de detalle” asociados a los “polígonos base” (paredes, techo, piso).
- Los “polígonos de detalle” son coplanares con los base.
- Las propiedades de los “polígonos de detalle” tienen prioridad sobre las de los “polígonos base”.
- Al estar asociados al polígono base, se aplica *jerarquía* para el cálculo de las superficies visibles.

Correspondencia de Texturas (texture mapping o pattern mapping)

- Se establece una correspondencia entre una imagen y una superficie.
- Como alternativa, en lugar de imagen se puede utilizar un procedimiento (o algoritmo).
- Con frecuencia un pixel de pantalla puede estar cubierto por varios elementos de textura (píxeles de la imagen). Hay que trabajar con todos para evitar artefactos de discretización.





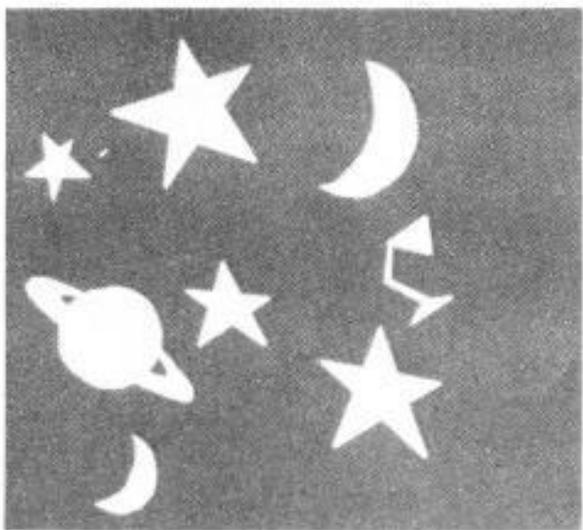
(a)



(b)



(c)



(d)



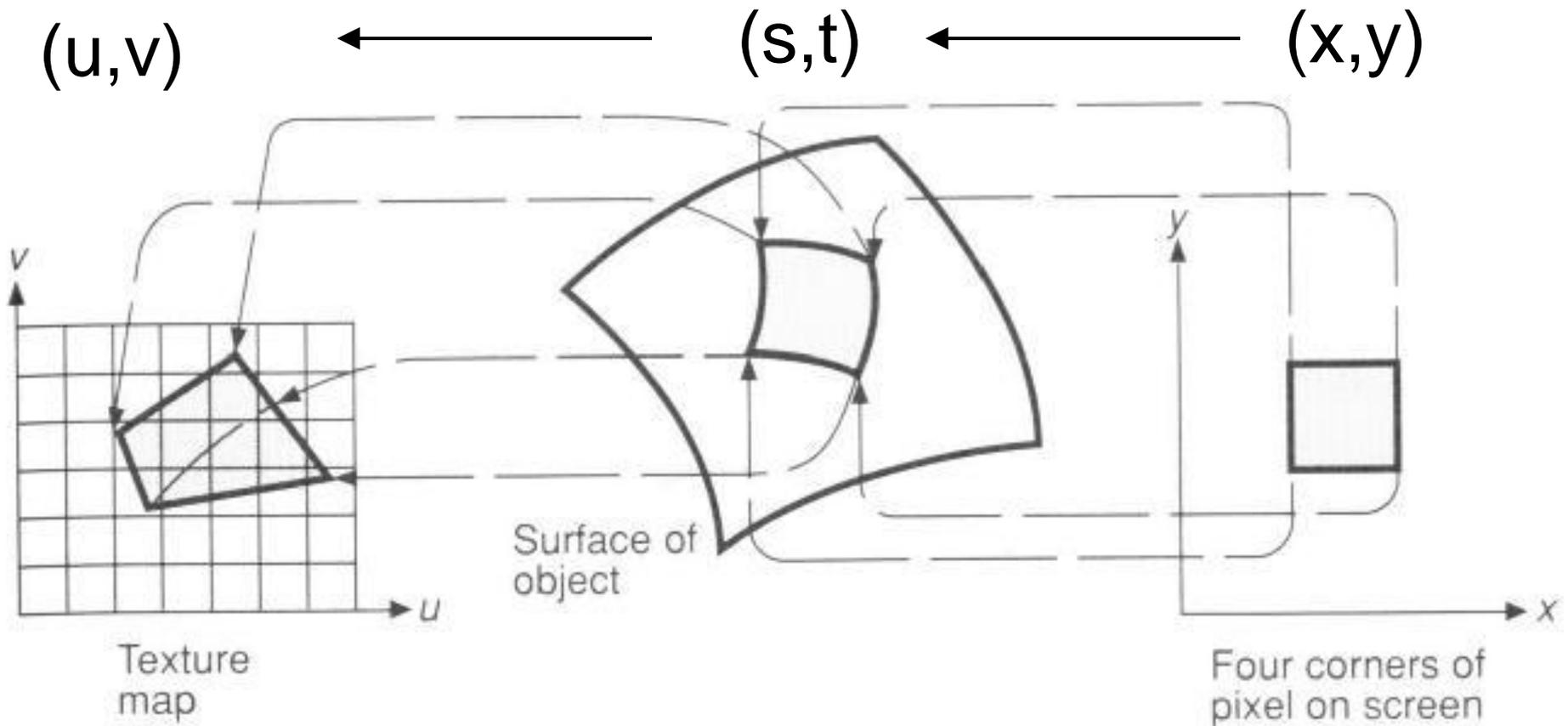
(e)



(f)



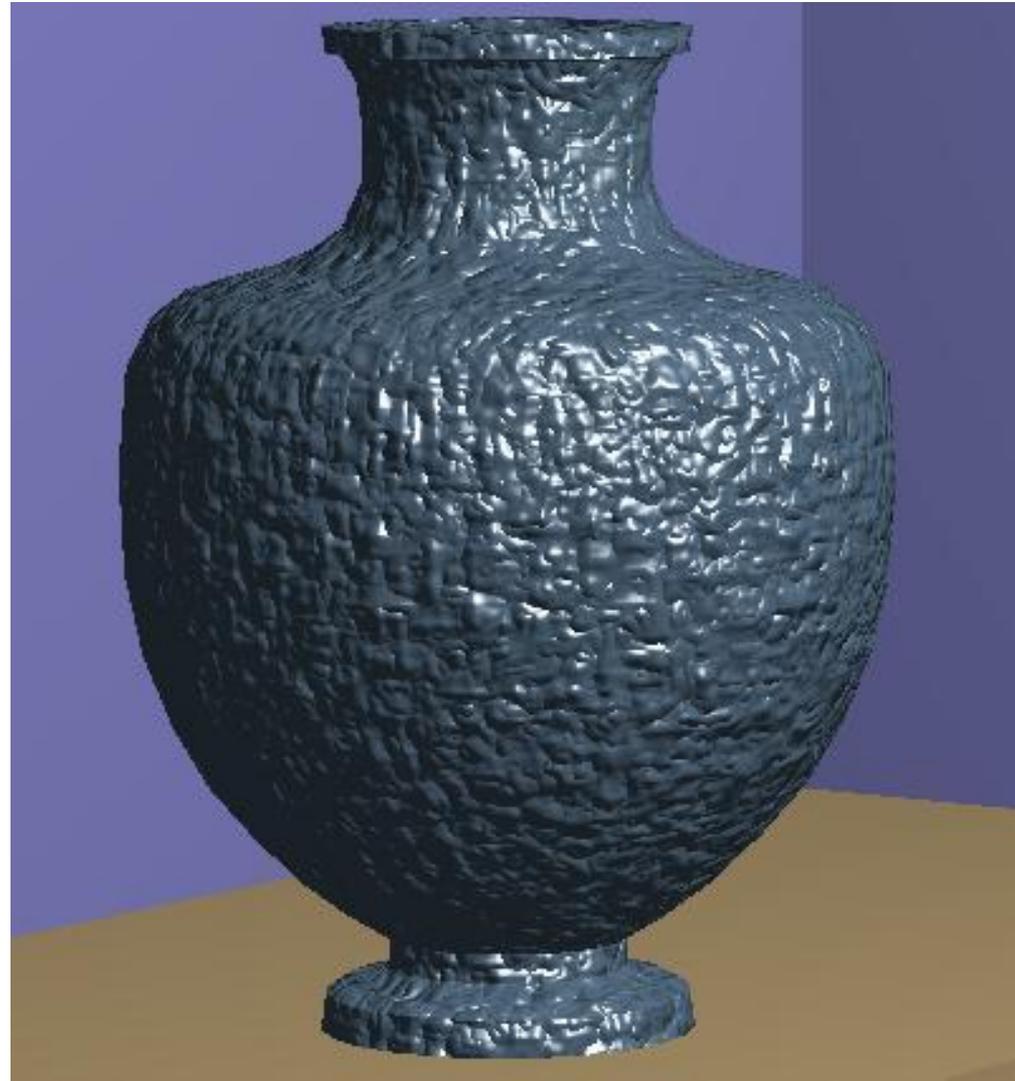
Correspondencia de Texturas (texture mapping o pattern mapping)



Correspondencia de Texturas (texture mapping o pattern mapping)

- El valor de cada pixel se halla haciendo promedio ponderado, según la porción del elemento de textura que está dentro del cuadrilátero del mapa de textura.
- Si las coordenadas (u,v) caen fuera de la imagen, se puede duplicar la imagen.
- Se puede hacer corresponder las 4 esquinas del rectángulo de (s,t) con un cuadrilátero en (u,v) .
- Si la superficie es un polígono, se asignan coordenadas (u,v) de mapa de textura directamente a sus vértices. Los valores internos se interpolan de la forma ya vista.
 - Problema: la interpolación causa distorsión en caso de perspectiva.

Correspondencia de Protuberancias (bump mapping)



Correspondencia de Protuberancias (bump mapping)



Correspondencia de Protuberancias (bump mapping)

- Se define un array de desplazamientos, usado para simular el desplazamiento de un punto de la superficie un poco encima o debajo de su actual posición.

Dado un punto de la superficie $P=[x(u,v), y(u,v), z(u,v)]$, se cumple que, si P_u y P_v son las derivadas parciales de P :

$$N = P_u \times P_v$$

Desplazo P un valor B a lo largo de Normal:

$$P' = P + BN/|N|$$

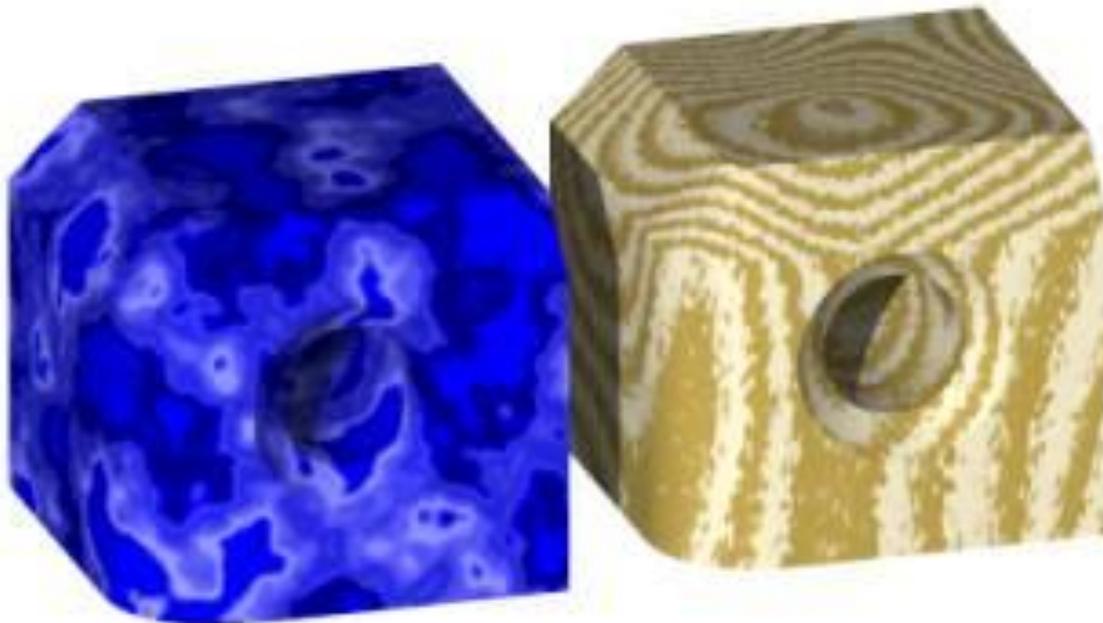
Una aproximación a la nueva normal N' es:

$$N' = N + (B_u(N \times P_v) - B_v(N \times P_u)) / |N|$$

B_u y B_v son derivadas de B respecto de u y v

Otros métodos

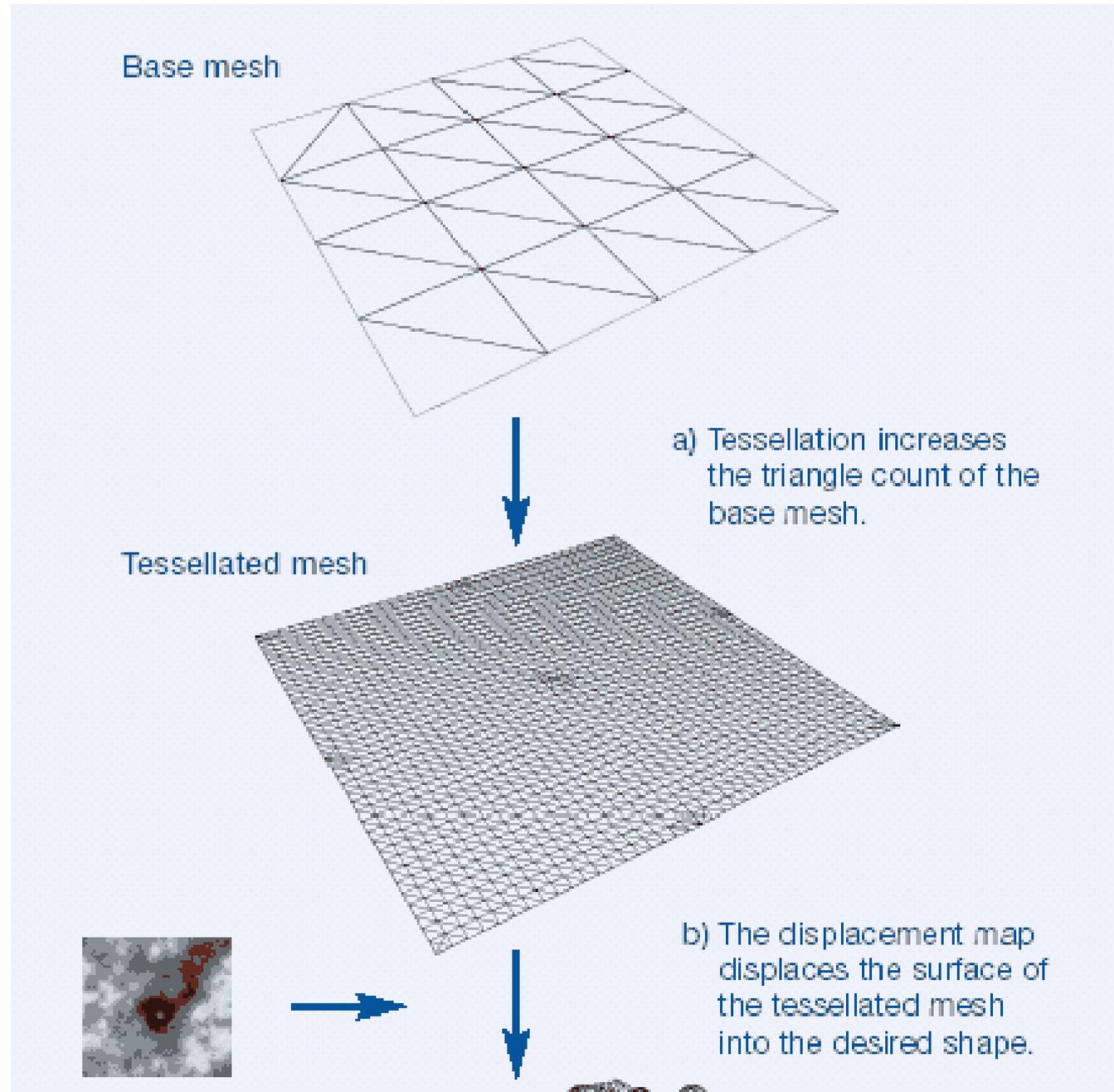
- Texturas tridimensionales permiten mejor simulación de objetos tallados en madera u otros materiales.



Otros métodos

Detalle de superficie

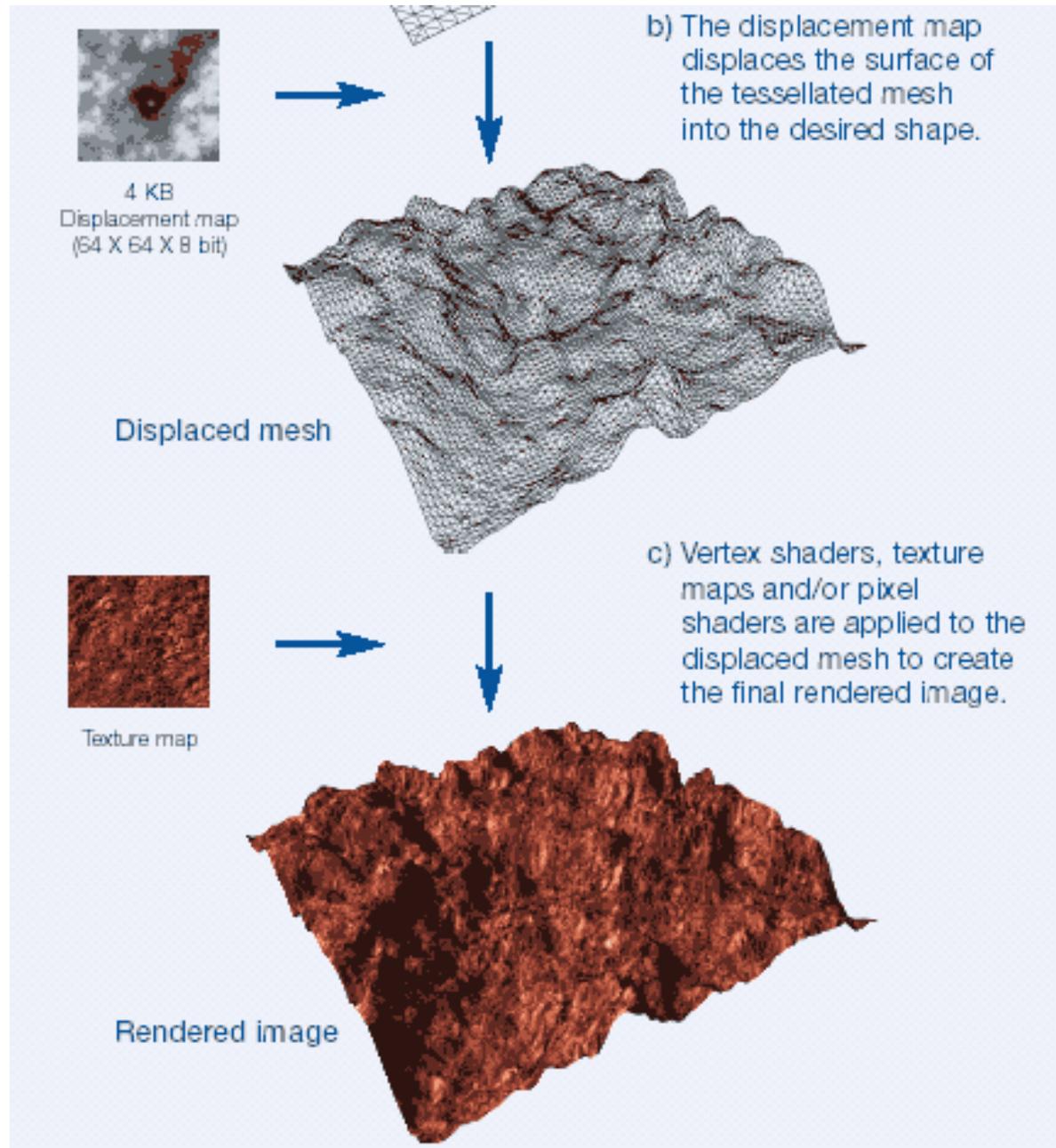
- Correspondencia con otras propiedades de las superficies. Por ejemplo, correspondencia de desplazamiento.



Otros métodos

Detalle de superficie

- Correspondencia con otras propiedades de las superficies. Por ejemplo, correspondencia de desplazamiento.



Sombras

Si la fuente luminosa es puntual, no hay vistas parciales a ella.

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + \sum_{1 \leq i \leq m} S_i f_{att_i} I_{p\lambda_i} \left[k_d O_{d\lambda} (\bar{N} \cdot \bar{L}_i) + k_s O_{s\lambda} (\bar{R}_i \cdot \bar{V})^n \right]$$

$S_i = 0$, si la luz i está bolqueada en este punto.

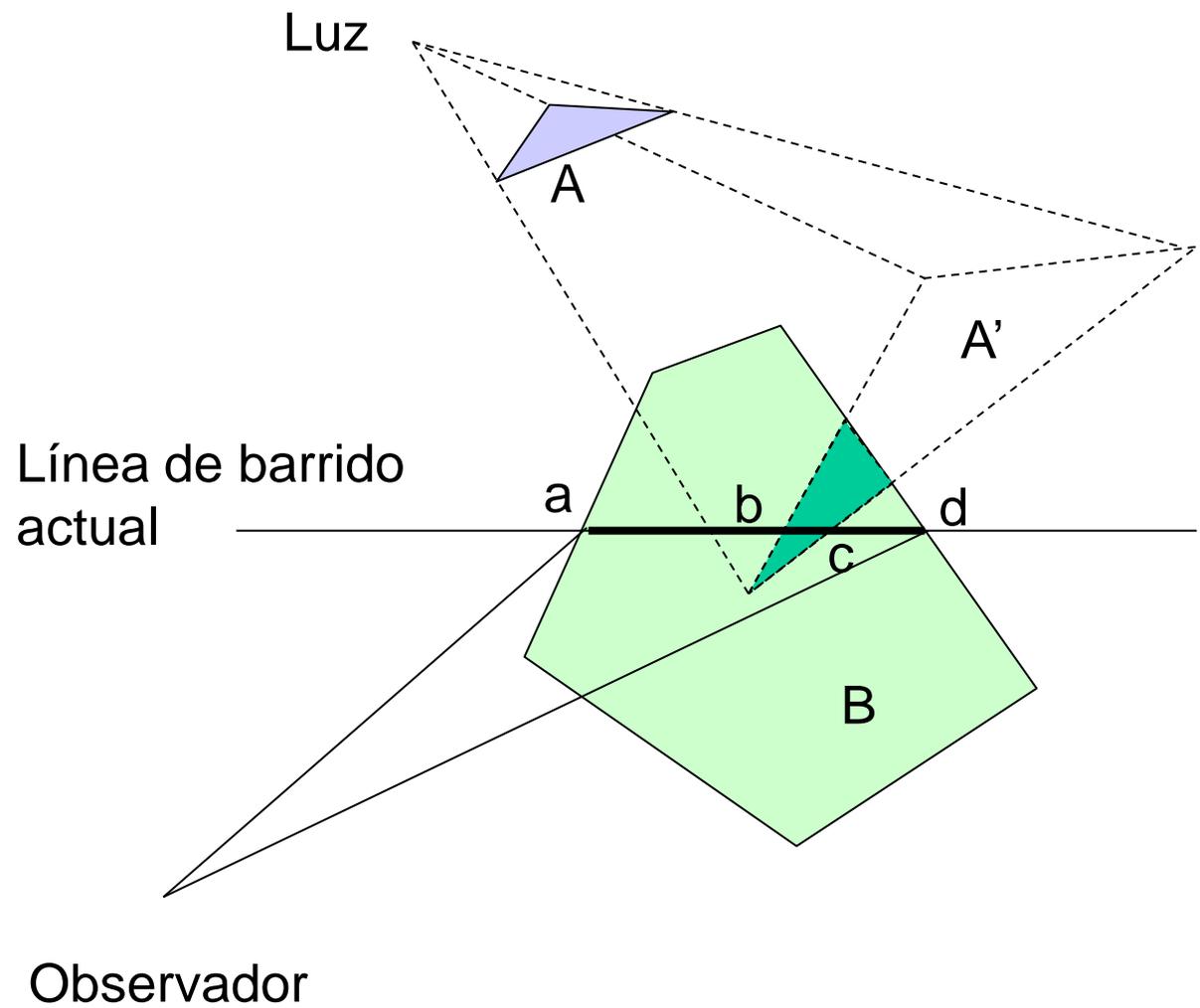
$S_i = 1$, si la luz i no está bloqueada en este punto.

Sombras por línea de barrido

Se pueden combinar los procesamientos de sombras y de superficies visibles.

- La fuente luminosa es el centro de proyección.
- Las aristas de los polígonos se proyectan sobre otros polígonos que intersecten la línea de barrido actual.
- Cuando el barrido cruza una arista de sombra, se cambian los colores de los píxeles de la imagen.

Sombras por línea de barrido



Sombras por línea de barrido

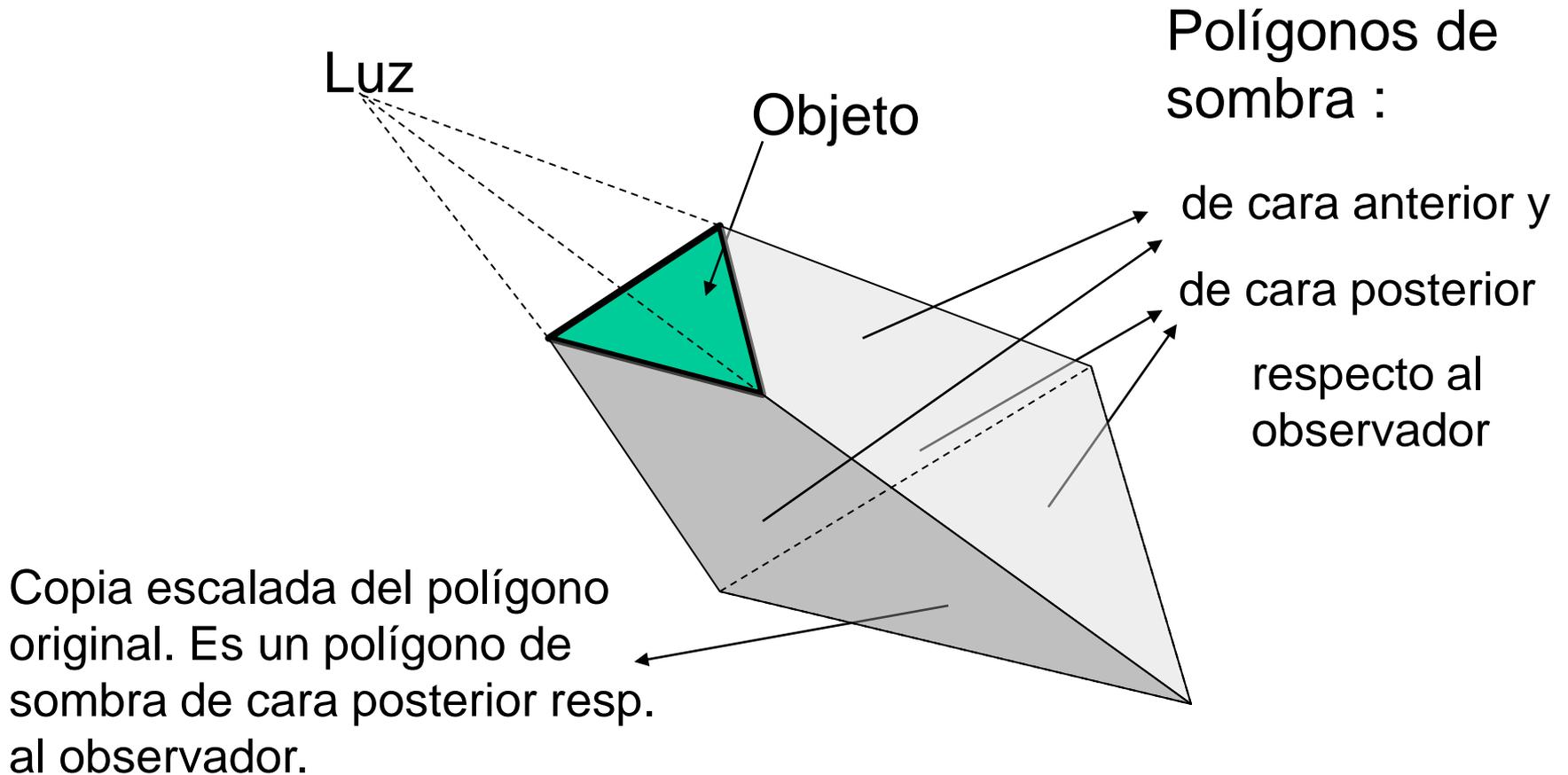
Problema: Si hay n polígonos \Rightarrow hay que calcular $n(n-1)$ proyecciones.

Solución parcial:

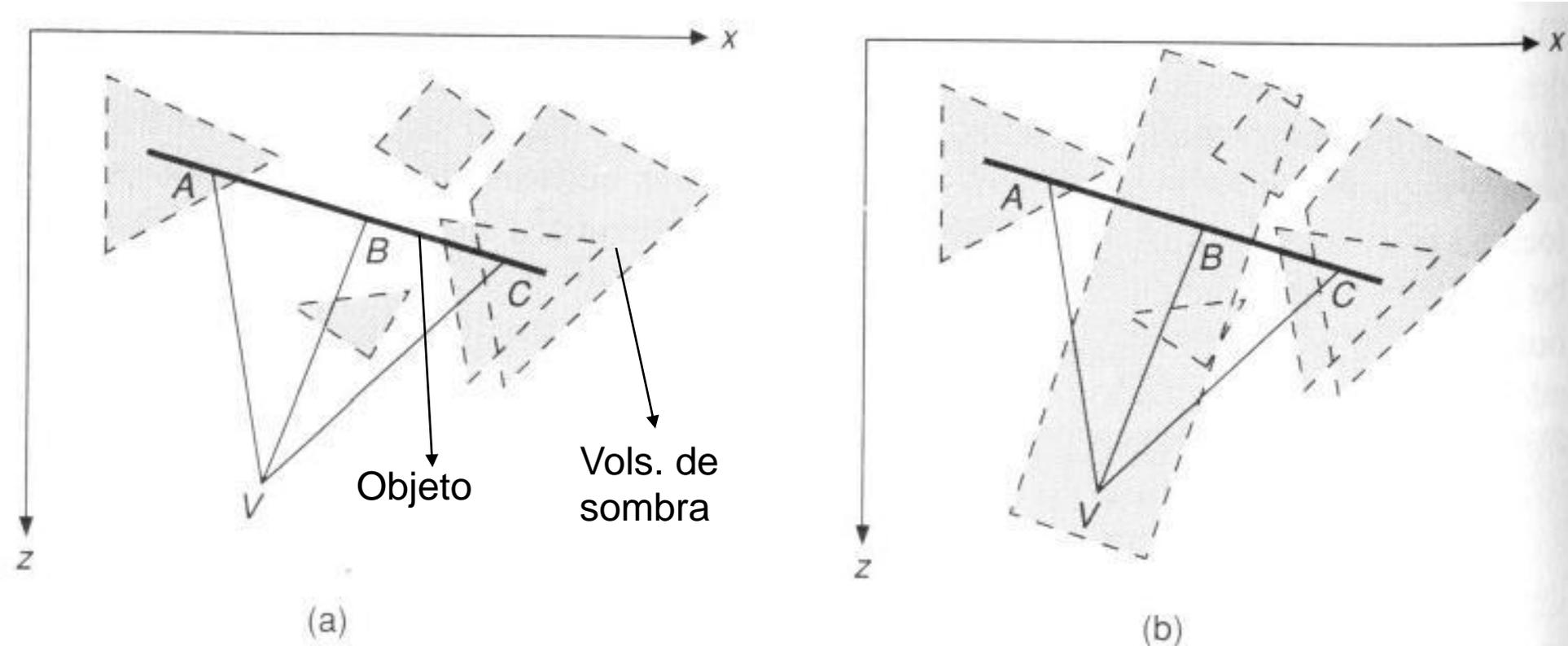
Proyectar los n polígonos en una esfera de centro la fuente luminosa.

No considerar proyecciones cuyas extensiones no se sobrepongan + otros casos particulares.

Volúmenes de sombra



Volúmenes de sombra



V = punto de observación.

Se trazan rayos desde V. Por cada rayo, hay un contador que suma 1 a cada p. de s. de cara anterior y resta 1 a cada p. de s. de cara posterior. Para muchas fuentes luminosas, hay contadores diferentes.

Un punto está bajo sombra si el contador es positivo en él.

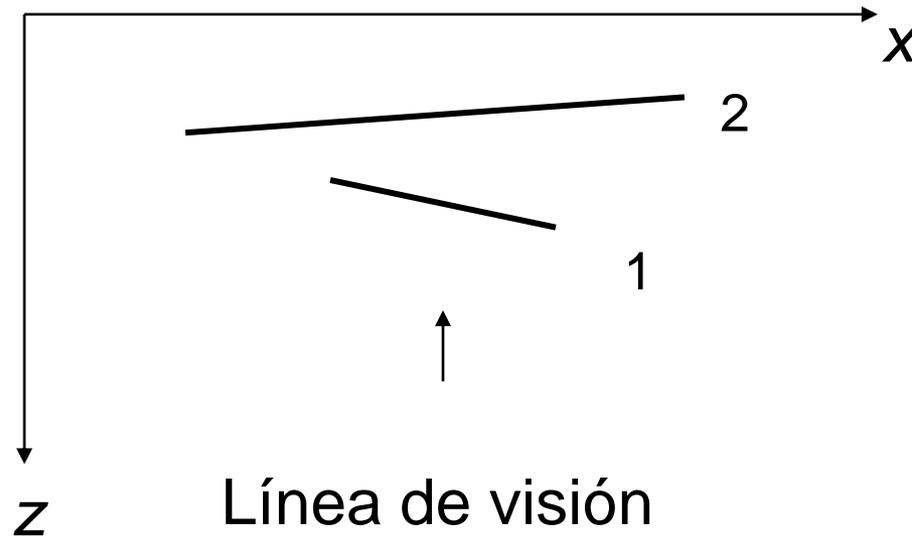
Transparencia

No refractiva

Refractiva

Transparencia no refractiva

Transparencia interpolada



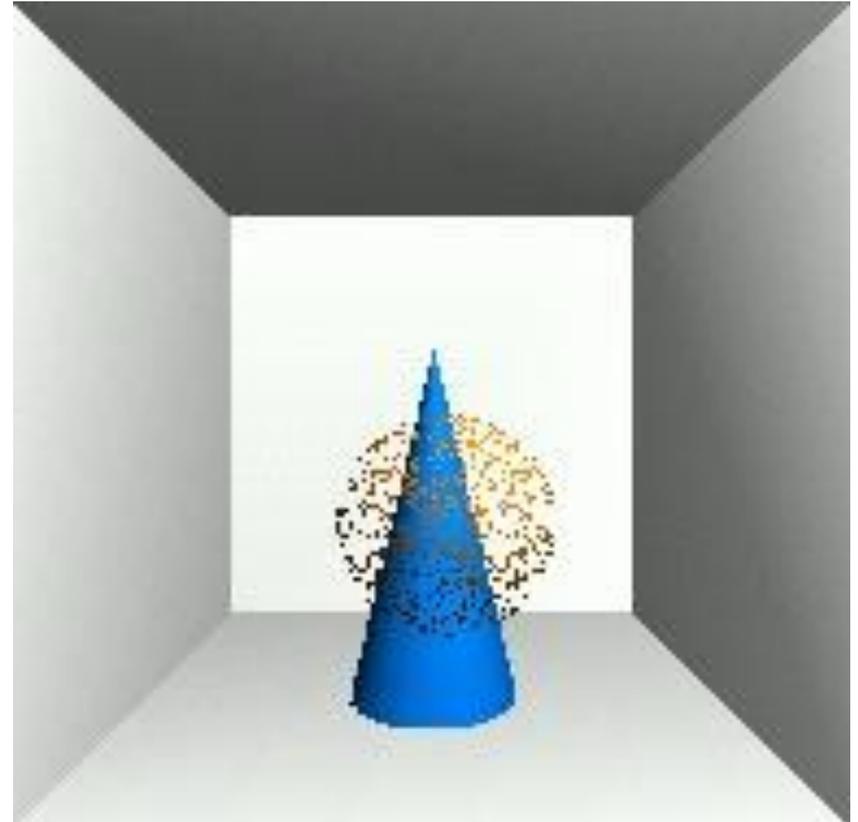
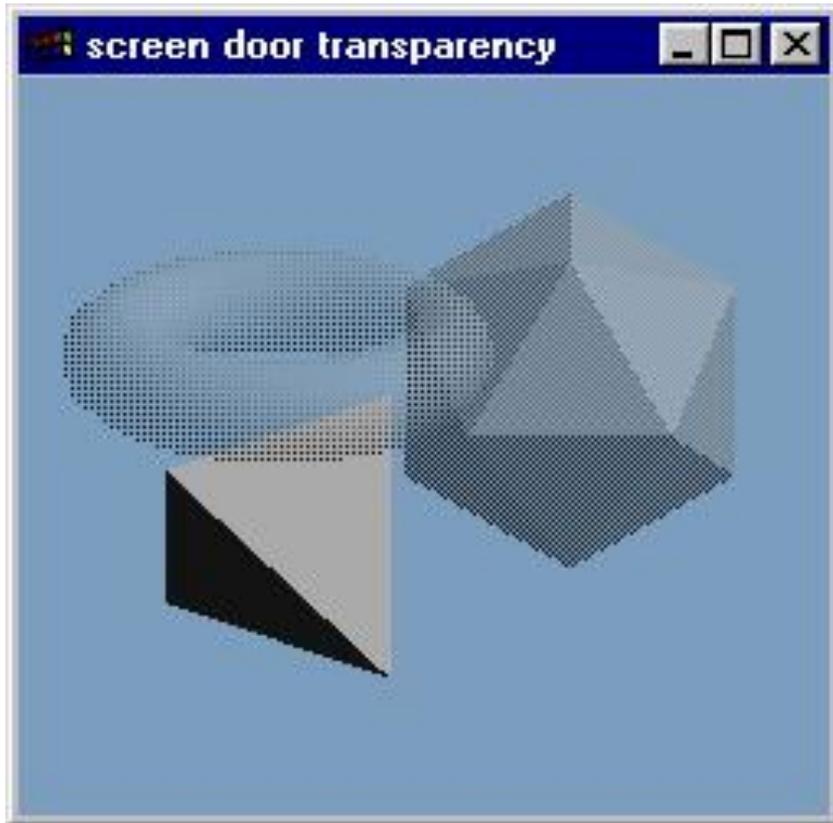
$$I_{\lambda} = (1 - k_{t1})I_{\lambda 1} + k_{t1}I_{\lambda 2}$$

k_{ti} (coeficiente de transmisión) mide la transparencia del pol. i

Transparencia no refractiva

Transparencia de mosquitero (screen-door)

Se implanta una malla que genera solo algunos píxeles relacionados con la proyección del objeto transparente.



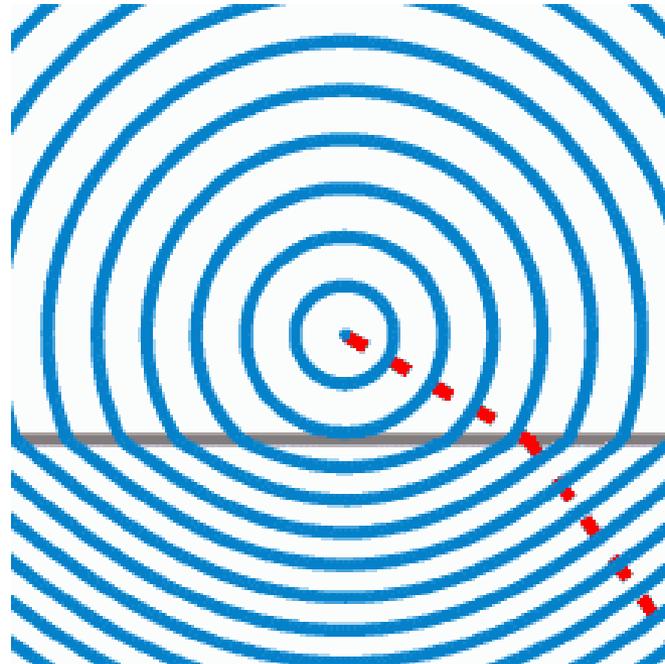
Transparencia filtrada

$$I_{\lambda} = I_{\lambda 1} + k_{t1} O_{t\lambda} I_{\lambda 2}$$

$O_{t\lambda}$ es el color de transparencia del polígono 1.

Esta función se puede invocar recursivamente, si hay varios polígonos transparentes superpuestos.

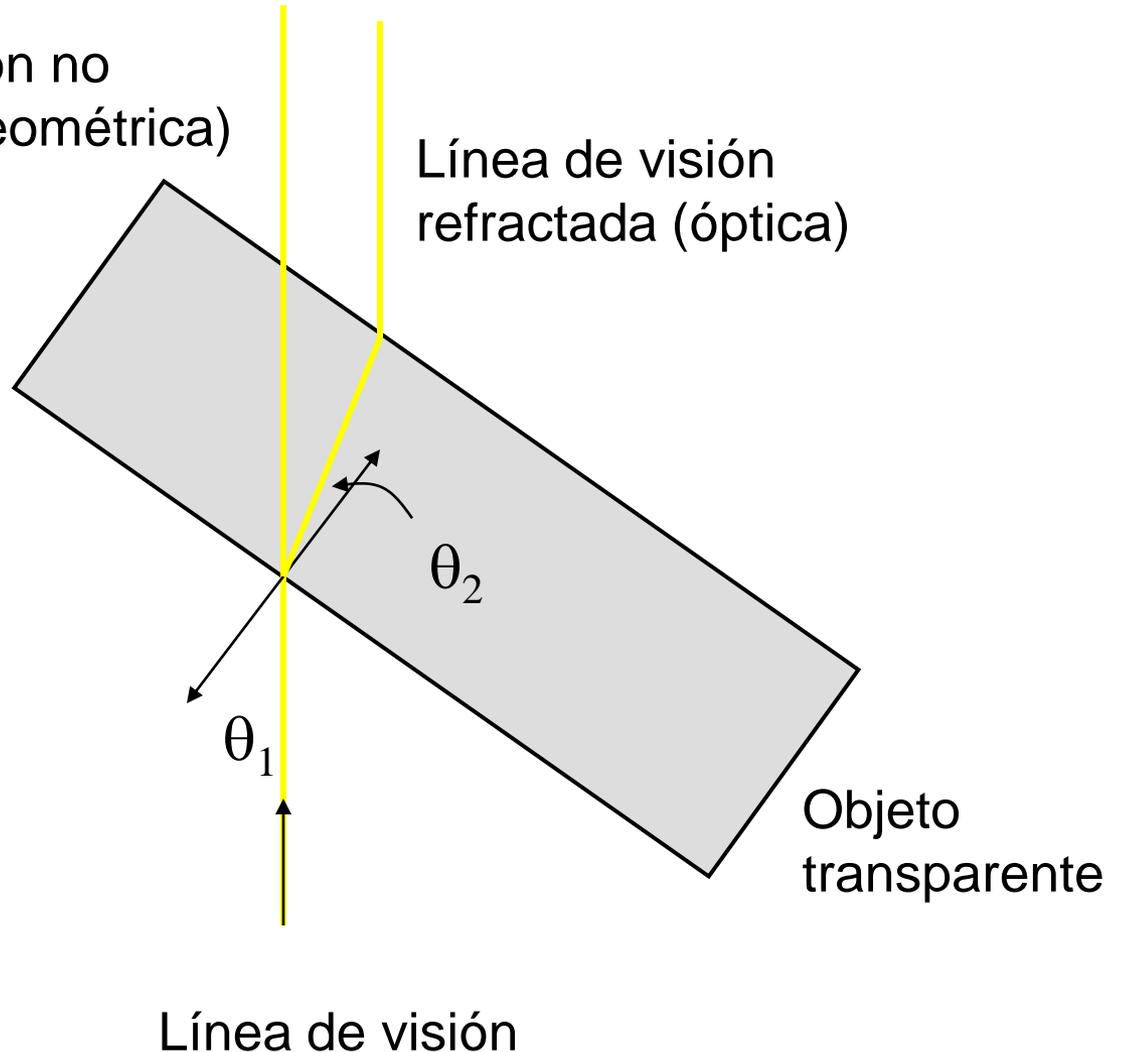
Transparencia refractiva



Transparencia refractiva

Línea de visión no refractada (geométrica)

Línea de visión refractada (óptica)



Ley de Snell:

$$\text{sen } \theta_1 / \text{sen } \theta_2 = \eta_2 / \eta_1 ;$$

η_1 y η_2 son los índices de refracción de los materiales.

Los η dependen también de la longitud de onda

Transparencia refractiva

(índices de refracción)

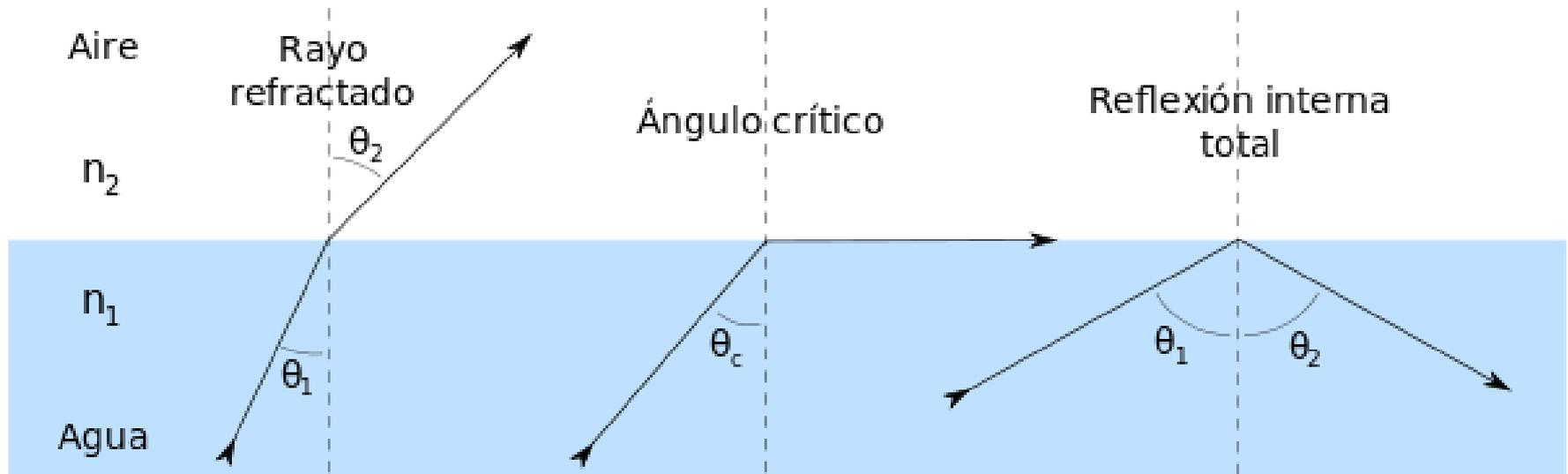
- Vacuum ...1.00000 (exactly)
- Air (STP) ... 1.00029
- Acetone ... 1.36
- Alcohol ... 1.329
- Amorphous Selenium ... 2.92
- Calspar1 ...1.66
- Calspar2 ...1.486
- Carbon Disulfide ...1.63
- Chromium Oxide ...2.705
- Copper Oxide ...2.705
- Crown Glass ...1.52
- Crystal ...2.00
- Diamond ...2.417
- Emerald ...1.57
- Ethyl Alcohol ...1.36
- Flourite ...1.434
- Fused Quartz ...1.46
- Heaviest Flint Glass... 1.89
- Heavy Flint Glass ...1.65
- Glass ...1.5
- Ice ...1.309
- Iodine Crystal ...3.34
- Lapis Lazuli ...1.61
- Light Flint Glass ...1.575
- Liquid Carbon Dioxide ...1.20
- Polystyrene ...1.55
- Quartz 1 ...1.644
- Quartz 2 ...1.553
- Ruby ...1.77
- Sapphire ...1.77
- Sodium Chloride (Salt)1 . 1.544
- Sodium Chloride (Salt)2 . 1.644
- Sugar Solution (30%)... 1.38
- Sugar Solution (80%)... 1.49
- Topaz ...1.61
- Water (20 C) ...1.333
- Zinc Crown Glass ...1.517

Transparencia refractiva

Reflexión interna total:

Ocurre cuando la luz pasa de un medio (1) a otro (2) con menor índice de refracción ($n_1 > n_2$).

Si el ángulo de incidencia es mayor que $\theta_c = \arcsin(n_2 / n_1)$, entonces en lugar de refracción ocurre una reflexión.

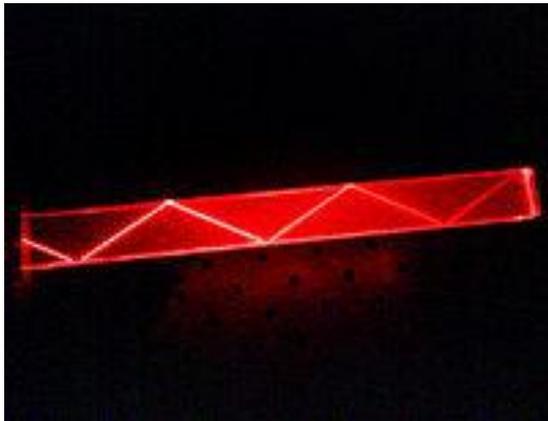


Transparencia refractiva

Reflexión interna total:

Ocurre cuando la luz pasa de un medio (1) a otro (2) con menor índice de refracción ($n_1 > n_2$).

Si el ángulo de incidencia es mayor que $\theta_c = \arcsin(n_2 / n_1)$, entonces en lugar de refracción ocurre una reflexión.



Algoritmos de iluminación global

Gran parte de la luz en el mundo real no proviene de fuentes directas.

Algoritmos (vistos aquí) para abordar este problema:

- Traza de rayos recursiva (ray tracing)

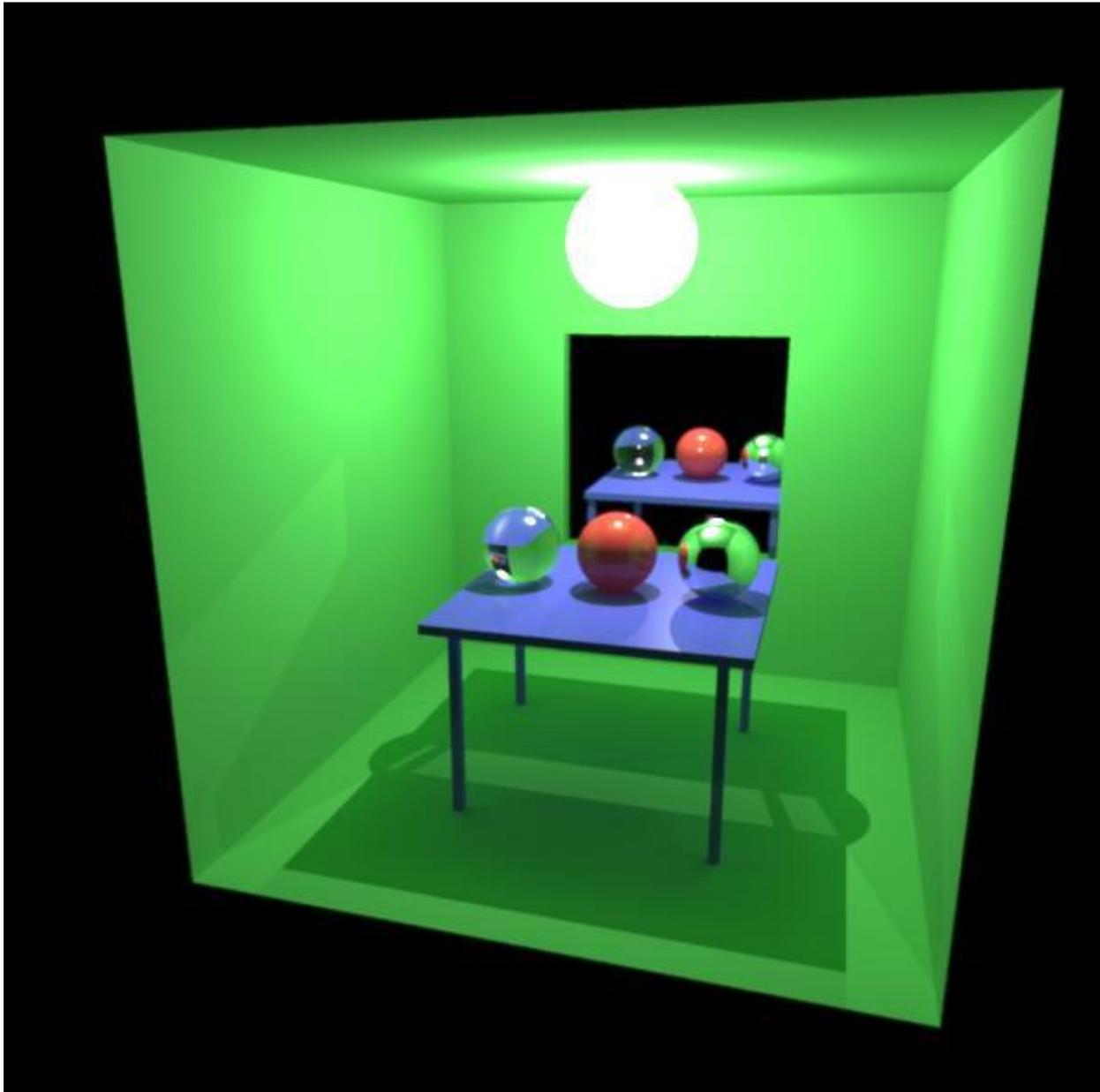
Es dependiente de la ubicación del observador

- Radiosidad (Radiosity)

Es independiente de la ubicación del observador

Algoritmos de iluminación global

Algoritmos de iluminación global



Ecuación de rendering (Kajiya 1986)

$$I(x, x') = g(x, x') \left[\varepsilon(x, x') + \int_s \rho(x, x', x'') I(x', x'') dx'' \right]$$

x, x', x'' son puntos en el ambiente

$I(x, x')$ es la intensidad que pasa de x' a x

$g(x, x')$ es un término geométrico.

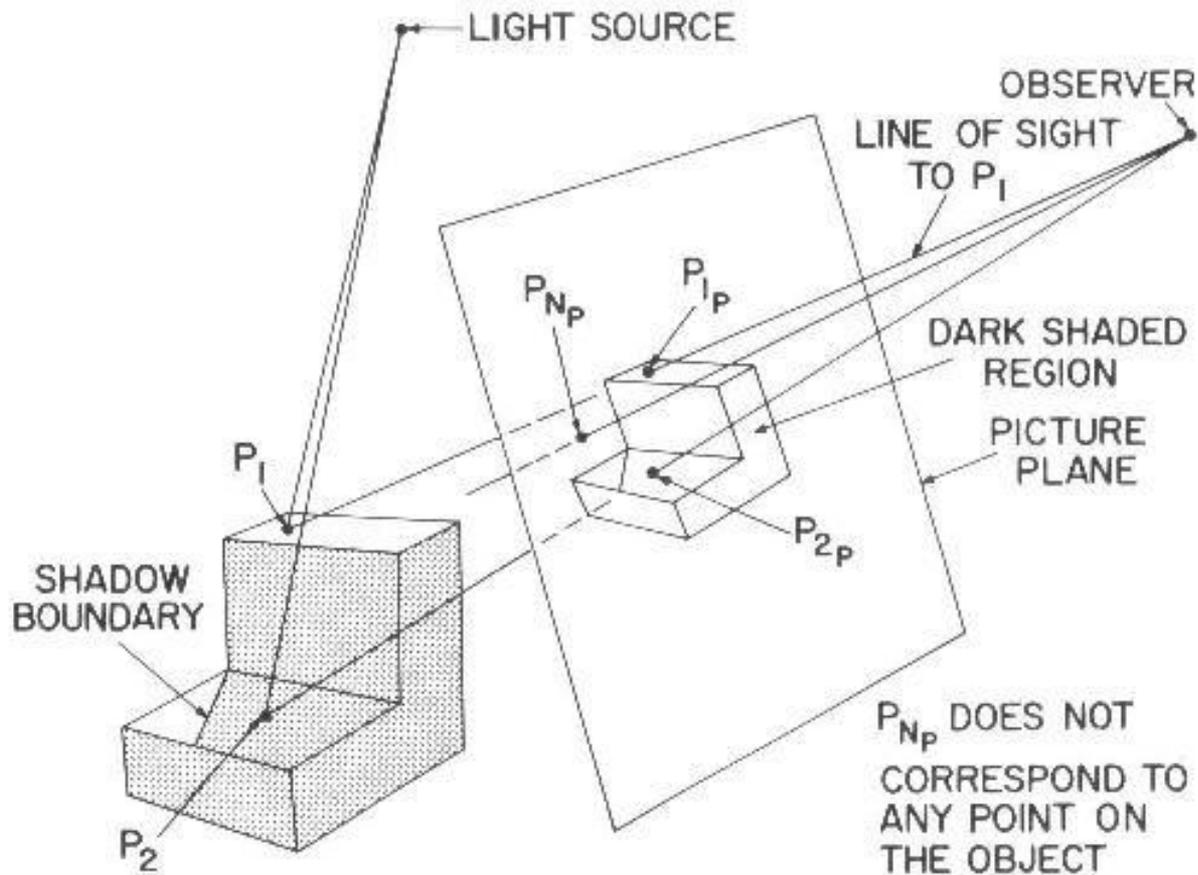
Vale 0 cuando x y x' están ocultos entre si.

Vale $1/r^2$ cuando son visibles (r =distancia (x, x'))

$\varepsilon(x, x')$ intensidad de luz que se emite de x' a x

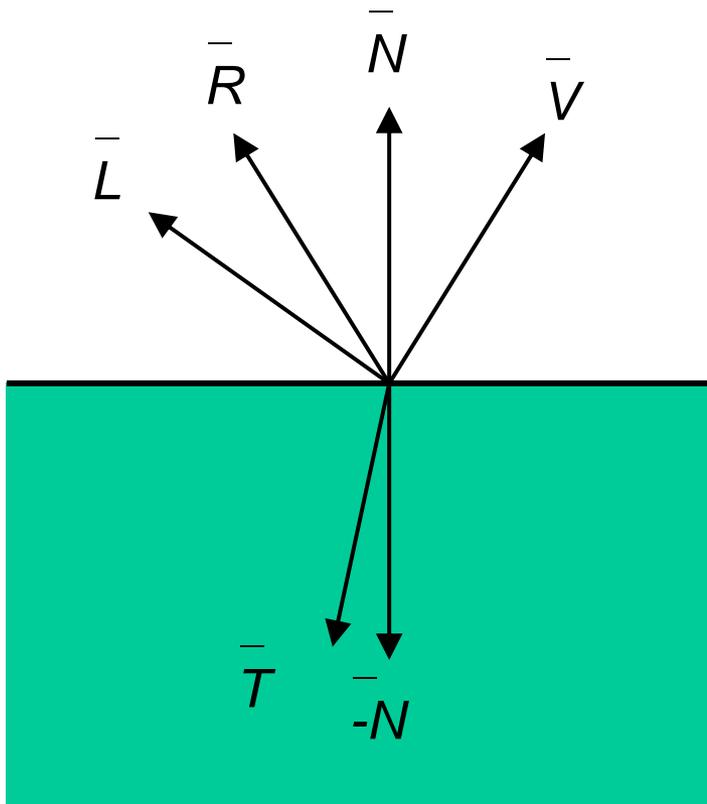
$\rho(x, x', x'')$ se relaciona con la intensidad de luz reflejada en x' , que sale de x'' y llega a x

Traza de rayos (ray-tracing)



Determinando si un punto de un objeto está en sombra.

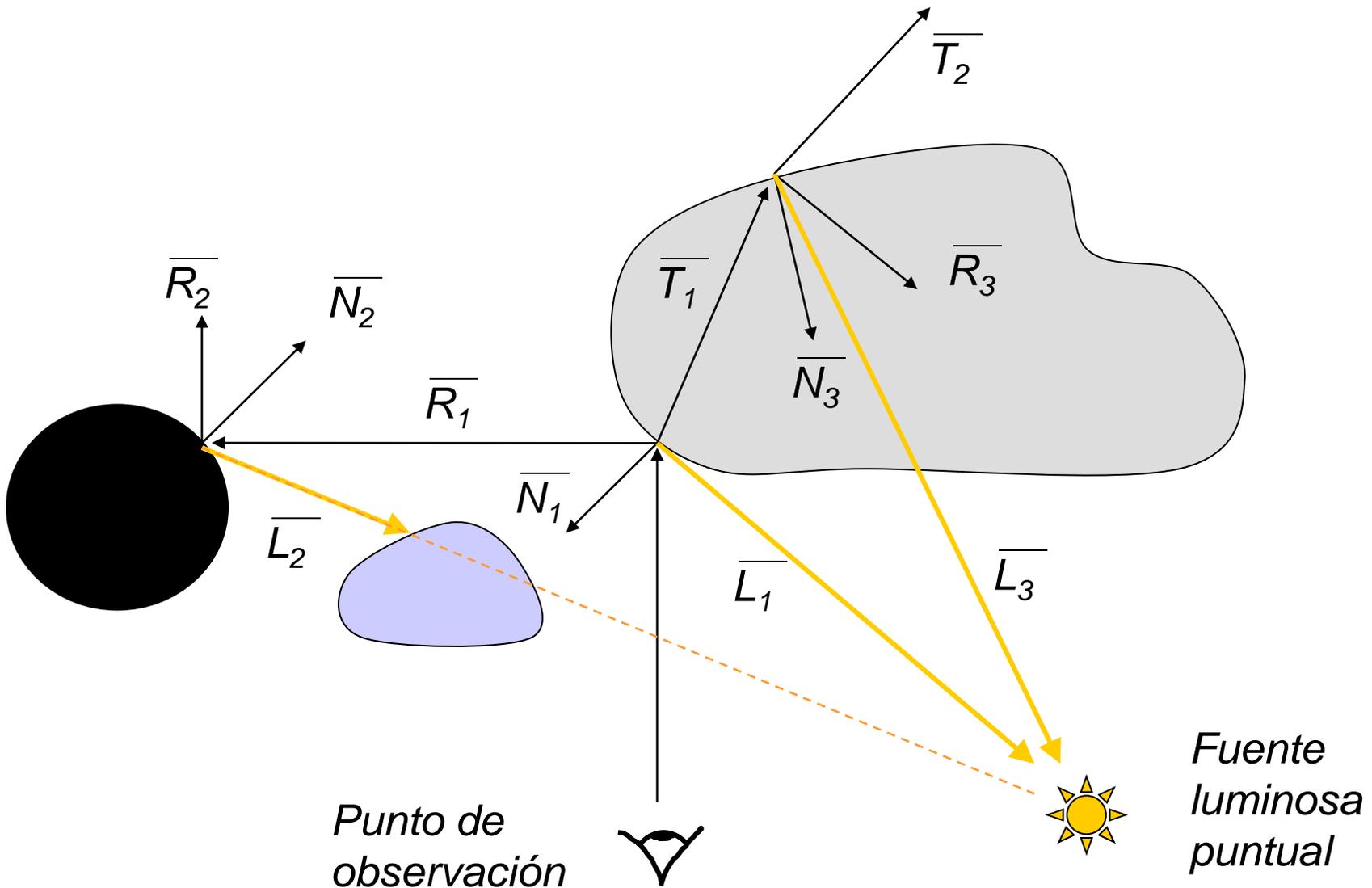
Traza de rayos recursiva (Whitted)



Se generan rayos de sombra, reflexión y de refracción en el punto de intersección del rayo con la superficie. Se llaman **rayos secundarios**.

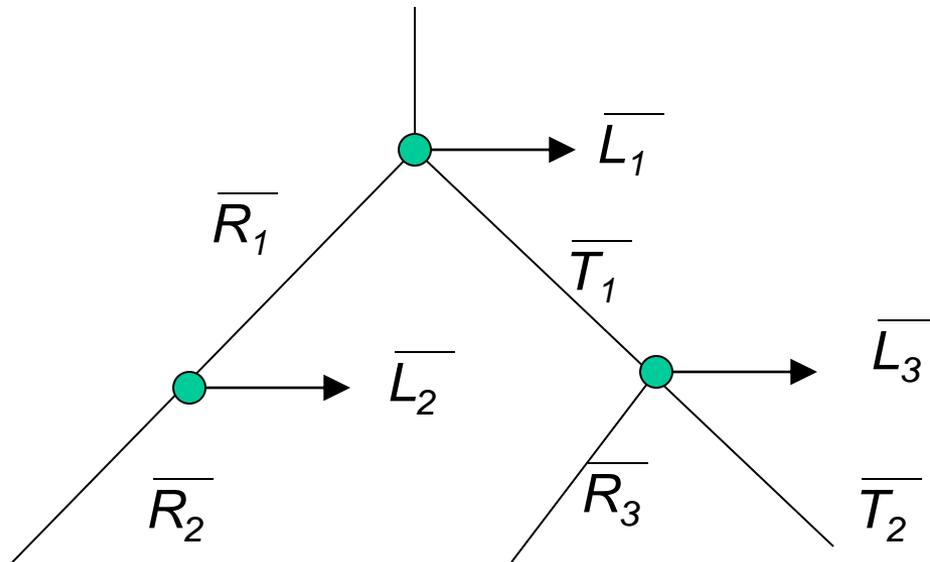
Rayos primarios son los que parten del observador.

Traza de rayos recursiva (Whitted)

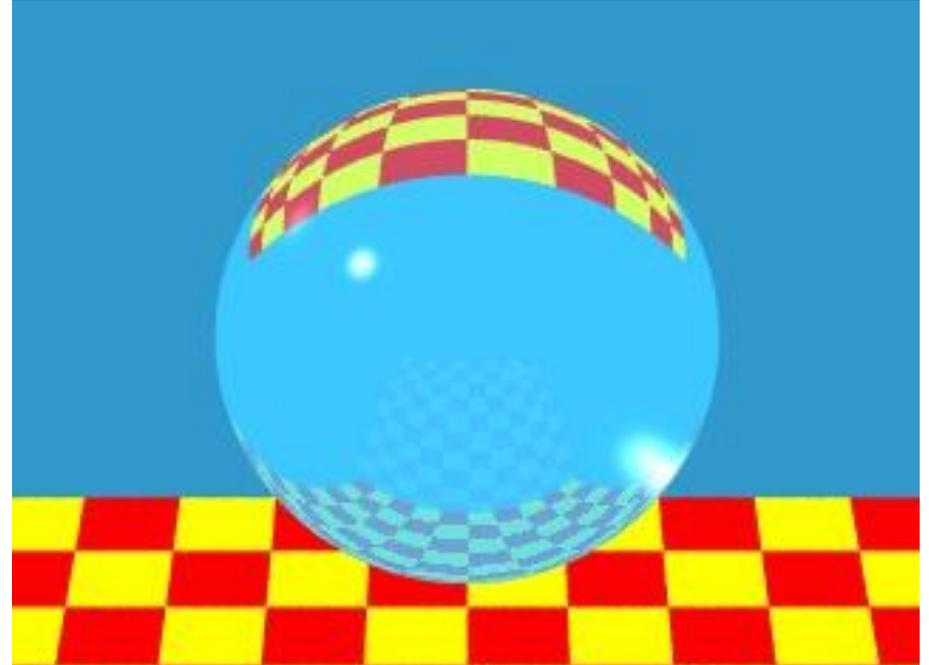
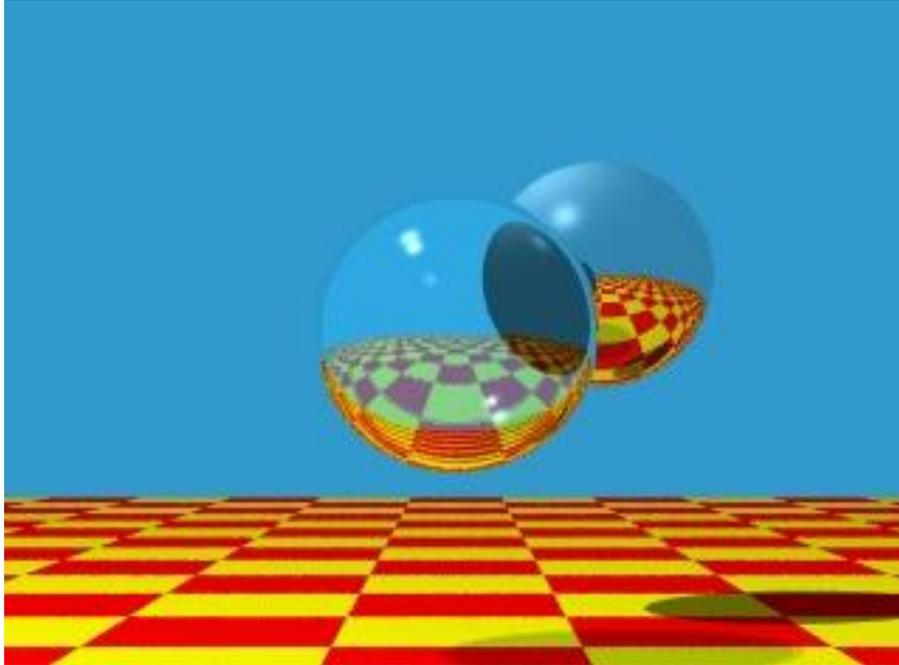


Traza de rayos recursiva (Whitted)

Punto de observación



Traza de rayos recursiva (Whitted)



Traza de rayos recursiva (Whitted)

$$I_{\lambda} = I_{a\lambda} k_a O_{d\lambda} + \sum_{1 \leq i \leq m} S_i f_{att_i} I_{p\lambda_i} \left[k_d O_{d\lambda} (\bar{N} \cdot \bar{L}_i) + k_s O_{s\lambda} (\bar{R}_i \cdot \bar{V})^n \right] + k_s I_{r\lambda} + k_t I_{t\lambda}$$

$I_{r\lambda}$ Intensidad del rayo reflejado.

$I_{t\lambda}$ Intensidad del rayo transmitido refractado.

k_t coeficiente de transmisión (entre 0 y 1).

$I_{r\lambda}$ e $I_{t\lambda}$ se calculan recursivamente y se multiplican por el inverso de la distancia.

S_i es una función continua. Varía según la transparencia de los objetos intersecados.

Traza de rayos recursiva (Whitted)

seudocódigo

(1)

```
Seleccionar el centro de proyección y la ventana en el plano de vista;  
for (cada línea de barrido en la imagen){  
    for (cada pixel en la línea de barrido){  
        determinar rayo por centro de proyección y pixel;  
        pixel=traza_RR(rayo, 1);  
    }  
}
```

Traza de rayos recursiva (Whitted)

seudocódigo

(2)

```
/* Intersecar rayo con los objetos y calcular la sombra en la intersección
   más cercana. */
/* La profundidad es la profundidad actual en el árbol de rayos */
color_RR traza_RR (rayo_RR, int profundidad)
{
    determinar la intersección más cercana de rayo con un objeto;
    if (Hay objeto intersecado) {
        calcular la normal en la intersección;
        return sombra_RR(obj. intersecado más cercano, rayo,
                        intersección, normal, profundidad);
    }
    else
        return VALOR_FONDO
}
```

Traza de rayos recursiva (Whitted)

seudocódigo

(3)

```
/* Calcular sombra en un punto, con rayos para sombras, reflex. y refrac. */  
color_RR sombra_RR (objeto, rayo, punto, normal, int profundidad)  
{  
    color = término del ambiente;  
    for (cada luz) {  
        rayo_s = rayo desde punto a la luz;  
        if (el producto punto de normal y direc. de luz es positivo) {  
            Calcular cuánta luz es bloqueada por sup. Opacas y transp., y  
            usarlo para escalar los términos difusos y especulares antes de  
            añadirlos a color;  
        }  
    }  
}
```

Traza de rayos recursiva (Whitted)

seudocódigo

(4)

```

if (profundidad < profundidad_max) {      /*Regresar si la prof. es excesiva */
    if (objeto es reflejante) {
        rayo_r = rayo en la dirección de reflexión desde punto;
        color_r = traza_RR (rayo_r, profundidad + 1);
        escalar color_r por el coeficiente especular y añadir a color;
    }
    if (objeto es transparente) {
        if (no ocurre la reflexión interna total) {
            rayo_t = rayo en la dirección de refracción desde
                                                                punto;
            color_t = traza_RR (rayo_t, profundidad + 1);
            escalar color_t por el coeficiente de transmisión y
                añadir a color;
        }
    }
}
return color;      /* Devolver color del rayo. */

```

Traza de rayos recursiva (Whitted)

El algoritmo es susceptible a problemas de precisión numérica.

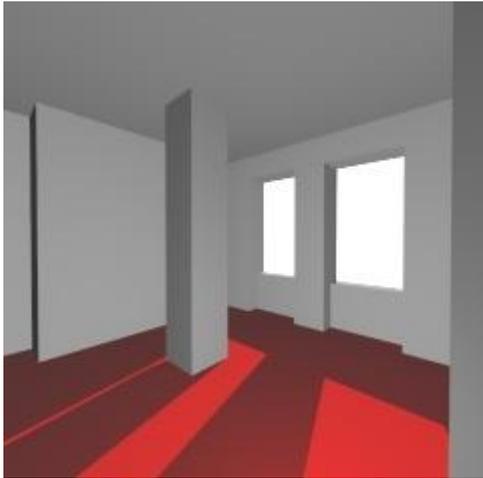
Los rayos generados pueden intersectar los objetos de donde salen.

Los rayos de luz **L** no se refractan en su trayectoria hacia la luz.

Métodos de Radiosidad

Ray tracing precisa un término de iluminación ambiente (para contemplar la luz indirecta difusa).

El método de radiosidad calcula la iluminación ambiente.



Radiosidad: tasa con la que la energía parte de una superficie. Se compone de energía emitida + la reflejada.

$$\text{Energía/unidad de tiempo/unidad de área} = \text{W/m}^2$$

Métodos de Radiosidad

El método de radiosidad trabaja exclusivamente con superficies difusas.

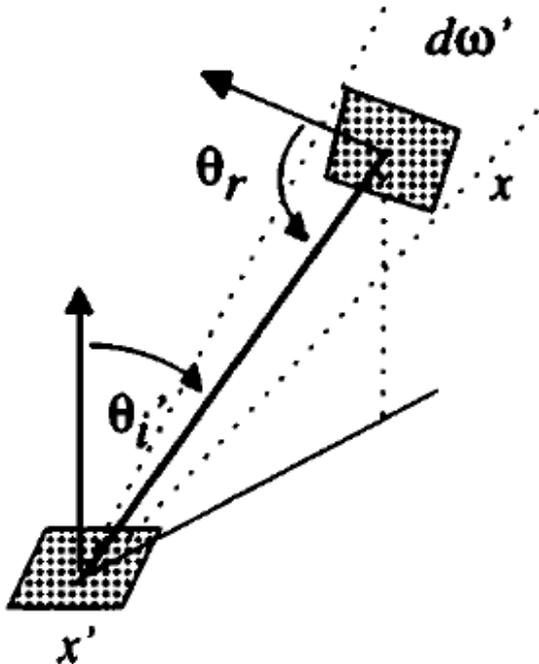
El color de las superficies difusas es independiente del ángulo de vista.

Esto simplifica la ecuación de rendering (Kajiya).

Ecuación de Radiosidad (continua)

La ecuación de rendering se transforma en:

$$B(x) = \varepsilon(x) + \rho(x) \int_s G(x, x') B(x') dx'$$



$\varepsilon(x)$ = Emisión de x

$\rho(x)$ = Reflectividad en x

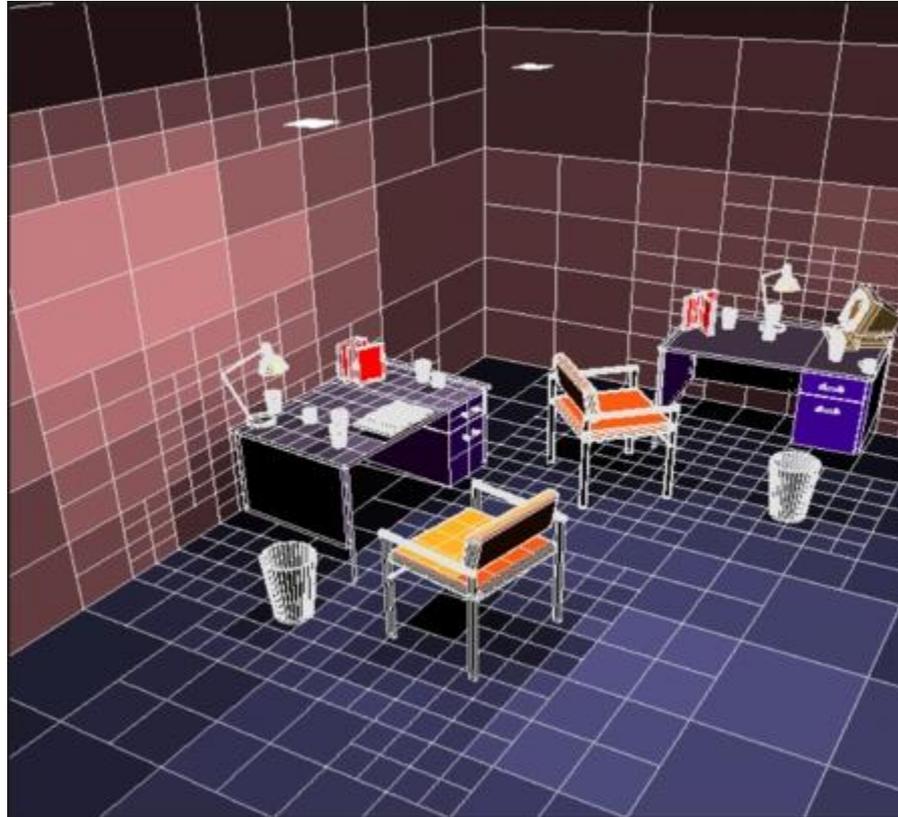
$$G(x, x') = \frac{V(x, x') \cos \theta_i' \cos \theta_r}{|x - x'|^2}$$

$$V(x, x') = \begin{cases} 0 & \text{si } x, x' \text{ no se ven entre sí} \\ 1 & \text{si } x, x' \text{ se ven entre sí} \end{cases}$$

Ecuación de Radiosidad (continua)



Ecuación de Radiosidad (discreta)



Las superficies se discretizan en n polígonos (llamados parches)

Ecuación de Radiosidad (discreta)

$$B_i = E_i + \rho_i \sum_{j=1}^n \left(\frac{F_{j-i} A_j}{A_i} \right) B_j$$

B_i = radiosidad de los parche i

E_i = tasa con que el parche i emite luz.

ρ_i = reflectividad del parche i (sin dimensiones).

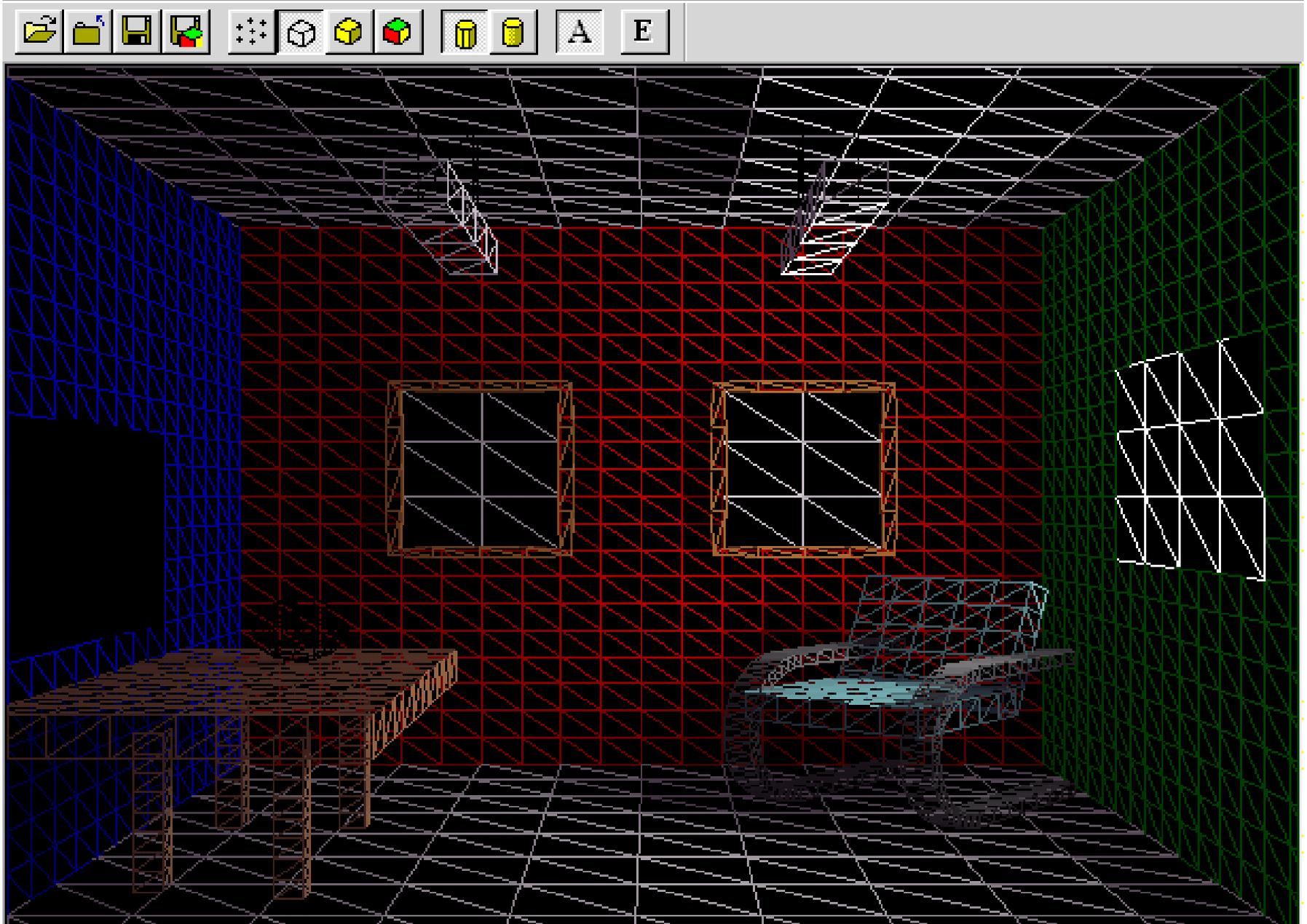
F_{j-i} = Factor de forma, fracción de energía que va del parche j al i (sin dimensiones)

A_i = Area del parche i

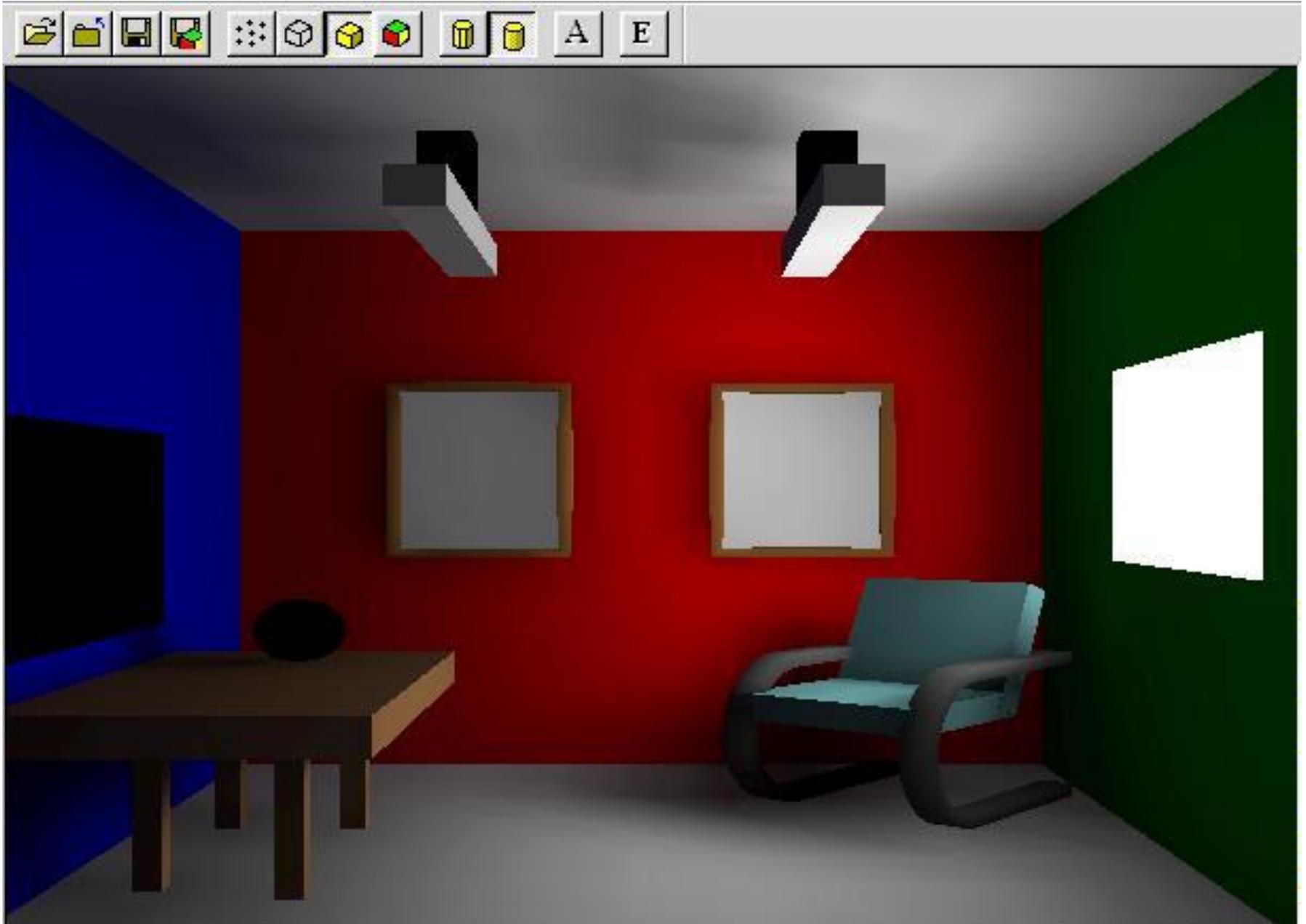
Métodos de Radiosidad

Las interacciones entre superficies se calculan de forma independiente del punto de vista.

Luego de este cálculo, se generan las vistas, solo determinando cuáles son las superficies visibles y el sombreado por interpolación (por ejemplo, con Gouraud).







Métodos de Radiosidad

- Todas las superficies pueden emitir y reflejar luz.
- Por tanto, las fuentes luminosas son áreas.
- Al ambiente se lo divide en un número finito de parches discretos.
- En cada parche, la luz es emitida y reflejada de forma uniforme.

Ecuación de Radiosidad

$$B_i = E_i + \rho_i \sum_{1 \leq j \leq n} B_j F_{j-i} (A_j/A_i)$$

B_i = radiosidad de los parche i

E_i = tasa con que el parche i emite luz.

ρ_i = reflectividad del parche i (sin dimensiones).

F_{j-i} = Factor de forma, fracción de energía que va del parche j al i (sin dimensiones)

A_i = Area del parche i

Ecuación de Radiosidad

$$B_i = E_i + \rho_i \sum_{1 \leq j \leq n} B_j F_{j-i} (A_j/A_i)$$

- **La energía que parte de un parche** = suma de la energía emitida + la reflejada.
- **Luz reflejada** = luz incidente x reflectividad ρ .
- **Luz incidente** = Suma de luz que parte de toda el área de cada parche en el ambiente, escalada a la fracción de luz que llega a un área unidad del parche receptor.
- $B_j F_{j-i}$ = cantidad de luz que parte de un área unidad de A_j y llega a toda el área A_i .
- $B_j F_{j-i} (A_j/A_i)$ = cantidad de luz que parte de toda el área A_j y llega al área unidad de A_i .

Ecuación de Radiosidad

En los ambientes difusos, los factores de forma cumplen una regla útil:

$$A_i F_{i-j} = A_j F_{j-i}$$

Por tanto, la ecuación anterior se simplifica:

$$B_i = E_i + \rho_i \sum_{1 \leq j \leq n} F_{i-j} B_j$$

O, de forma equivalente:

$$B_i - \rho_i \sum_{1 \leq j \leq n} F_{i-j} B_j = E_i$$

Ecuación de Radiosidad

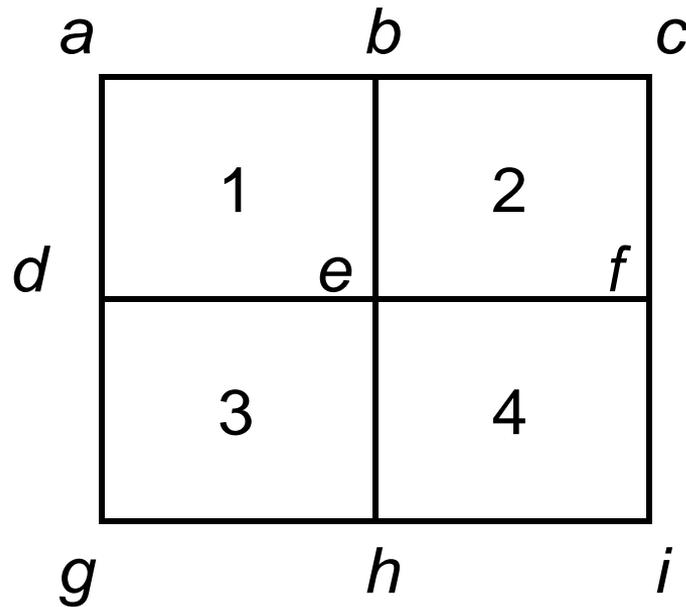
$$B_i - \rho_i \sum_{1 \leq j \leq n} F_{i-j} B_j = E_i$$

Esto permite generar un sistema de ecuaciones lineales:

$$\begin{bmatrix}
 1 - \rho_1 F_{1-1} & -\rho_1 F_{1-2} & \dots & -\rho_1 F_{1-n} \\
 -\rho_2 F_{2-1} & 1 - \rho_2 F_{2-2} & \dots & -\rho_2 F_{2-n} \\
 \cdot & \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot & \cdot \\
 -\rho_n F_{n-1} & -\rho_n F_{n-2} & \dots & -\rho_n F_{n-n}
 \end{bmatrix}
 *
 \begin{bmatrix}
 B_1 \\
 B_2 \\
 \cdot \\
 \cdot \\
 \cdot \\
 B_n
 \end{bmatrix}
 =
 \begin{bmatrix}
 E_1 \\
 E_2 \\
 \cdot \\
 \cdot \\
 \cdot \\
 E_n
 \end{bmatrix}$$

Se puede resolver por Gauss-Seidel

Determinación de la radiosidad de los vértices



$$B_e = (B_1 + B_2 + B_3 + B_4)/4$$

$$(B_b + B_e)/2 = (B_1 + B_2)/2 \quad \Rightarrow \quad B_b = (B_1 + B_2 - B_e)$$

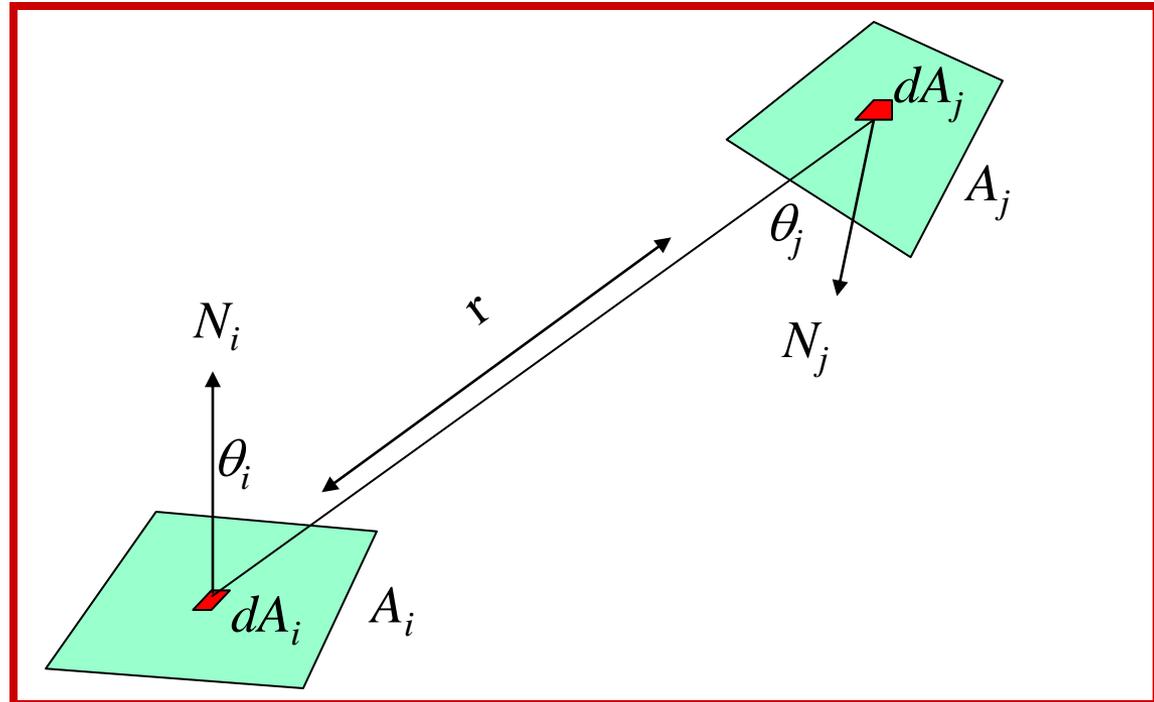
$$(B_a + B_e)/2 = B_1 \quad \Rightarrow \quad B_a = 2B_1 - B_e$$

Cálculo de los Factores de Forma

El factor de forma de un área diferencial dA_i a un área diferencial dA_j es:

$$dF_{di-dj} = \cos\theta_i \cos\theta_j H_{ij} dA_j / \pi r^2$$

H_{ij} es 1 o 0 dependiendo de si dA_j es visible desde dA_i



Cálculo de los *F*actores de *F*orma

$$dF_{di-dj} = \cos\theta_i \cos\theta_j H_{ij} dA_j / \pi r^2$$

$$F_{di-j} = \int_{A_j} \cos\theta_i \cos\theta_j H_{ij} / \pi r^2 dA_j$$

$$F_{i-j} = (1/A_i) \int_{A_i} \int_{A_j} \cos\theta_i \cos\theta_j H_{ij} / \pi r^2 dA_j dA_i$$

Suponemos que el punto central de un parche tipifica los demás puntos del parche.

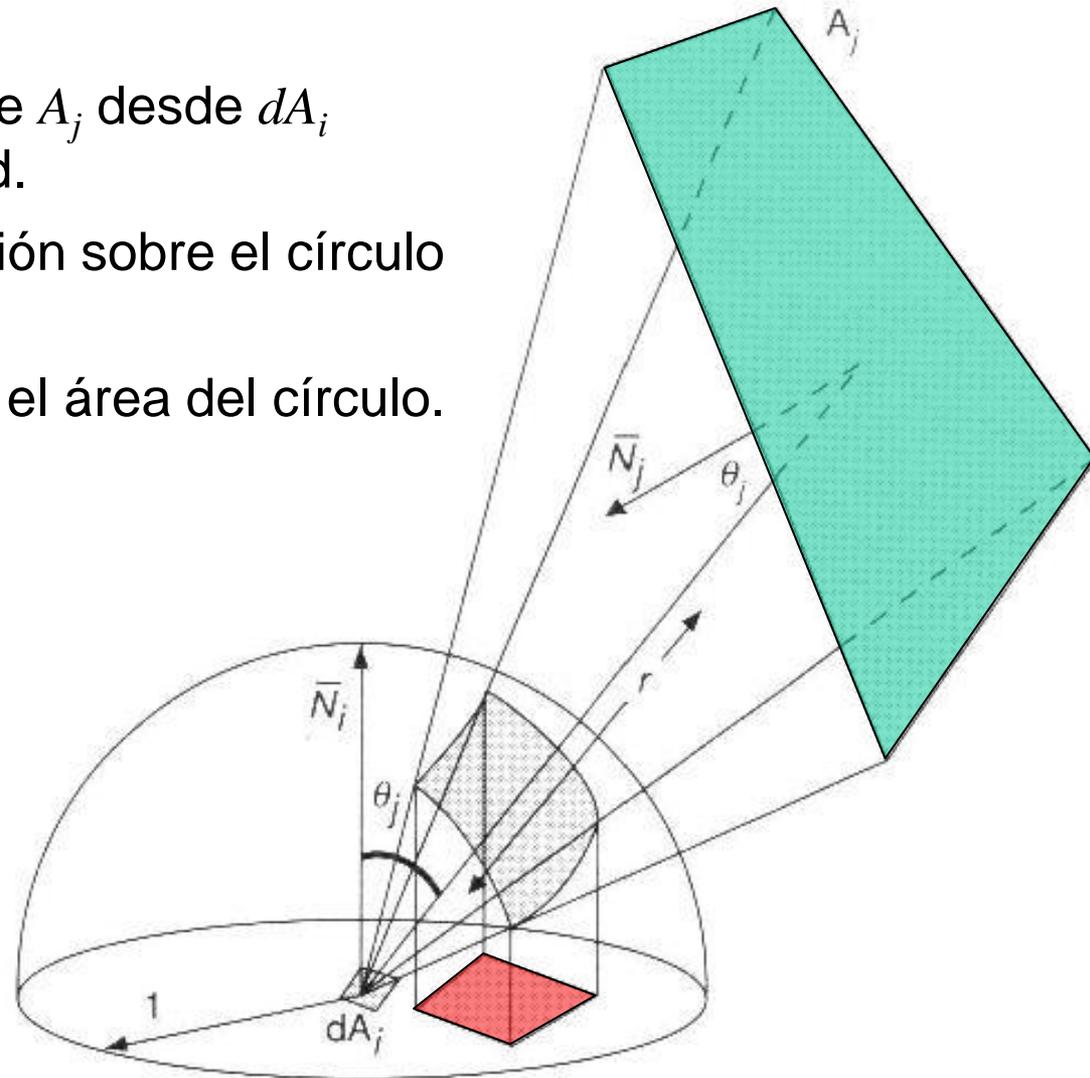
=>

F_{i-j} se puede aproximar con F_{di-j} calculado para un dA_i en el centro del parche i .

Cálculo de los *F*actores de *F*orma

Calcular F_{di-j} equivale a:

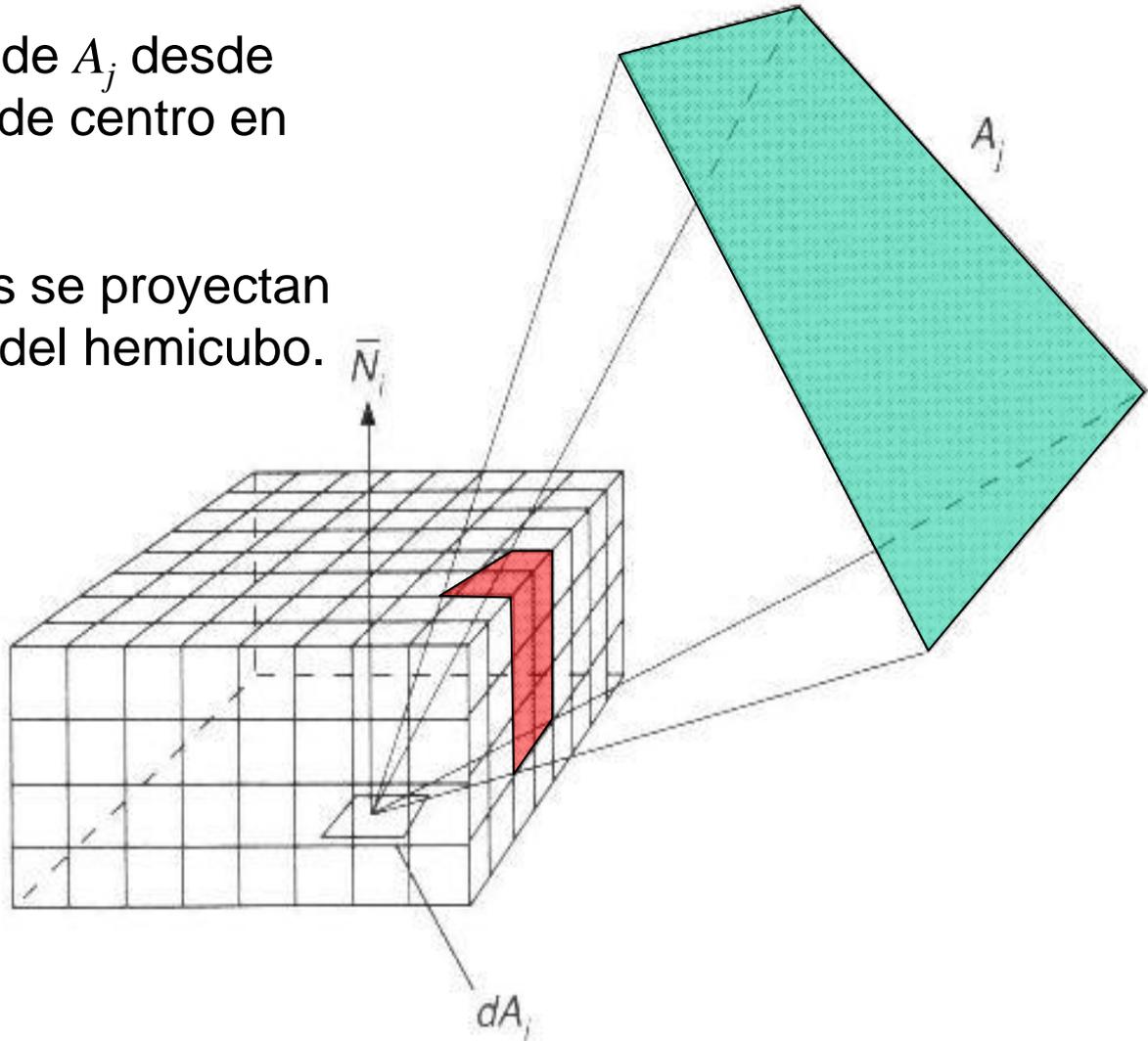
- 1) proyectar el área visible de A_j desde dA_i sobre el hemisferio unidad.
- 2) Hacer una nueva proyección sobre el círculo unidad.
- 3) Dividir el área hallada por el área del círculo.



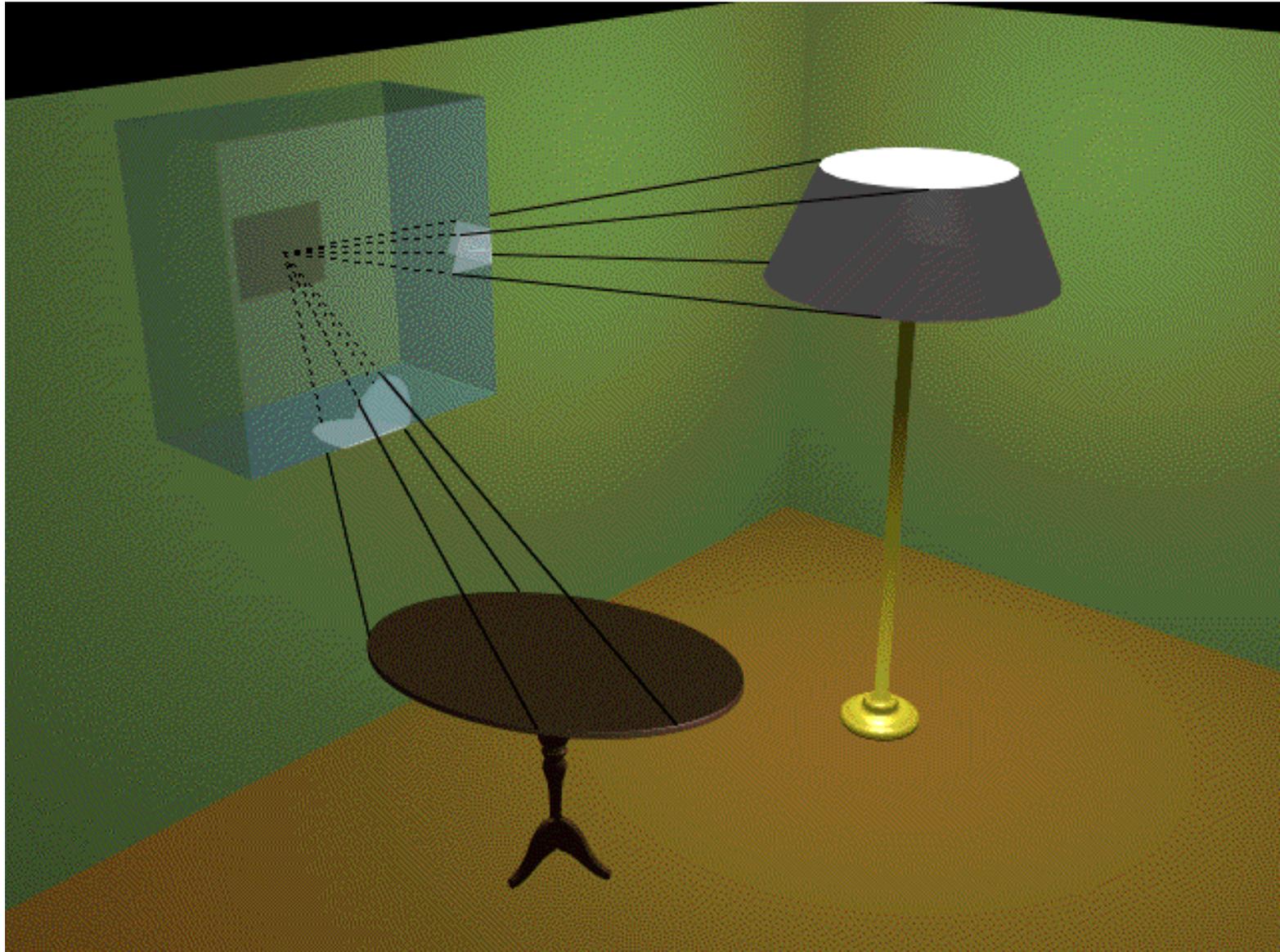
Cálculo de los Factores de Forma

Calcular F_{di-j} equivale a:

- 1) proyectar el área visible de A_j desde dA_i sobre un **hemicubo** de centro en dA_i .
- 2) Cada uno de los parches se proyectan sobre la cara apropiada del hemicubo.
- 3) Cada celda del hemicubo tiene un $FdeF$ propio, debido a su posición. F_{di-j} se calcula sumando los $FdeF$ de todas las celdas que contengan el identificador del parche j .



Cálculo de los *F*actores de *F*orma

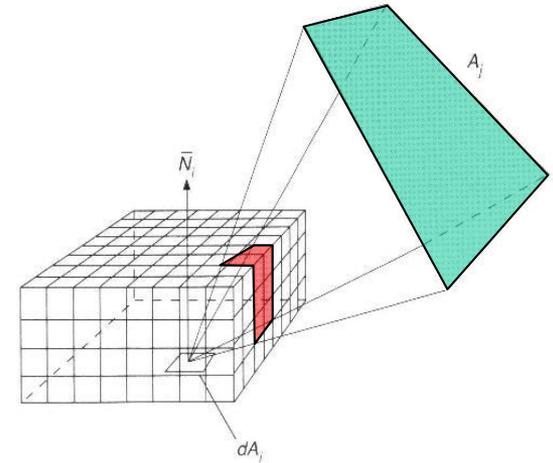


Cálculo de los *F*actores de *F*orma

Este algoritmo se puede realizar aplicando el algoritmo de z-buffer a cada lado del hemisucubo, y registrando el código del parche en lugar del color.

Se puede aprovechar el hardware existente para la memoria de profundidad z.

Pueden haber artefactos de discretización, por usar operaciones de precisión de la imagen.



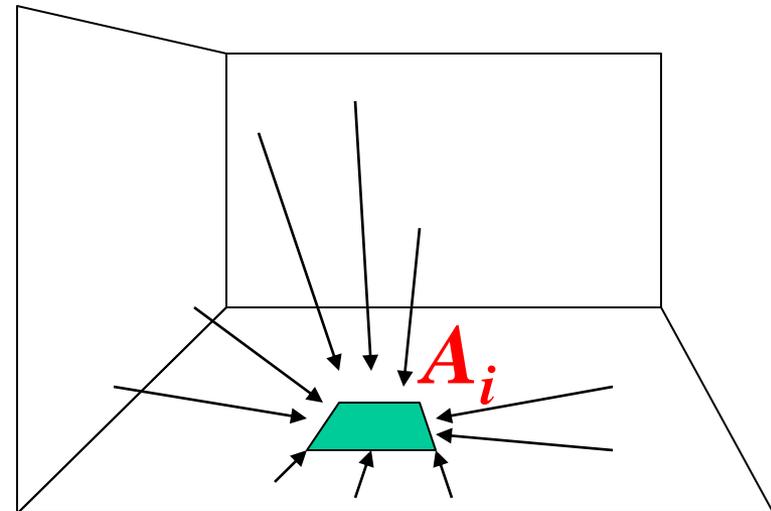
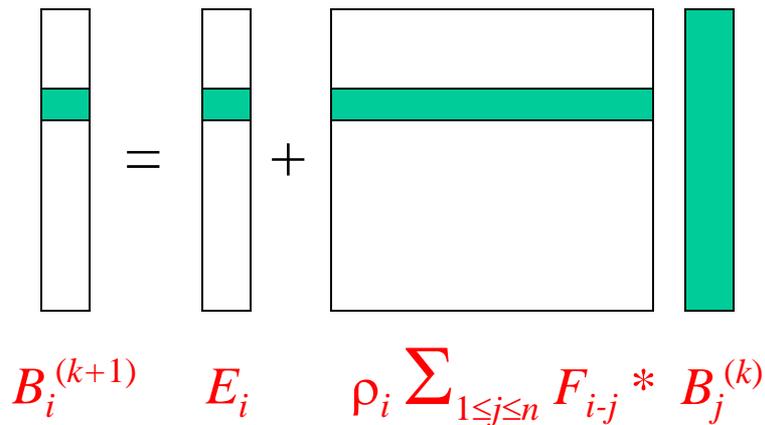
Soluciones parciales = Refinamiento progresivo

En el método anterior, hay n filas del tipo:

$$B_i^{(k+1)} = E_i + \rho_i \sum_{1 \leq j \leq n} F_{i-j} B_j^{(k)}$$

se **colecta** la energía del ambiente sobre el parche i .

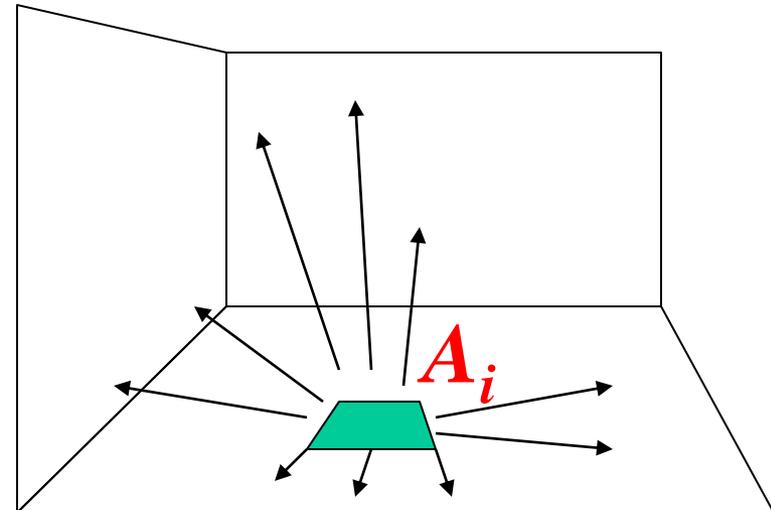
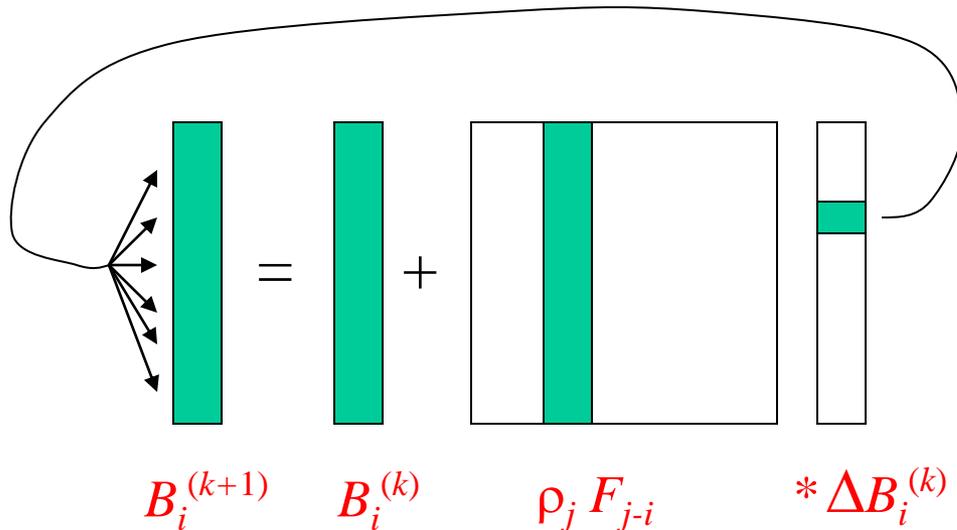
Por cada fila i procesada, se obtiene un nuevo valor para el parche i .



Soluciones parciales = Refinamiento progresivo

En este nuevo método, se **irradia** la energía del parche i a todo el ambiente.

Por cada columna i procesada, se obtiene un nuevo valor para todos los parches.



Soluciones parciales = Refinamiento progresivo

$\Delta B_i = B_i = E_i$ para todos los parches.

While (no hay convergencia)

Hallar i tal que $\Delta B_i A_j$ sea máximo.

Calcular F_{i-j} para todo j

For ($j = 1:n$)

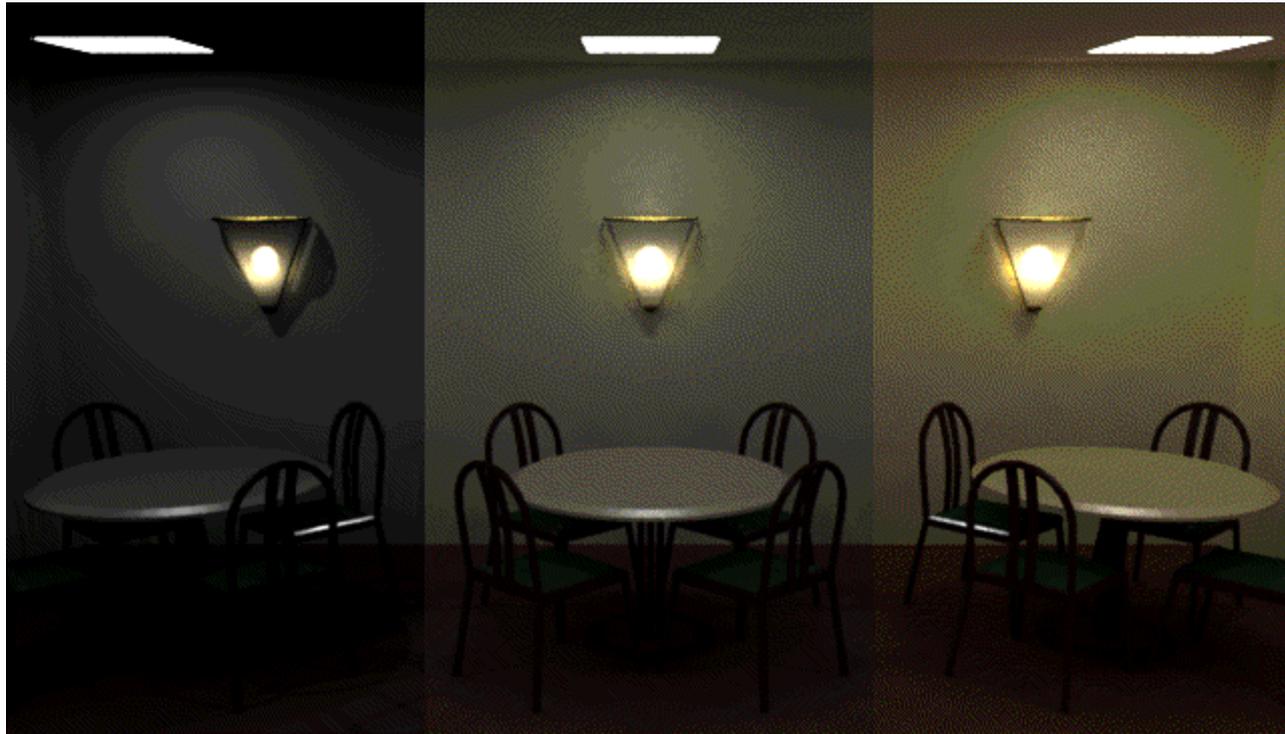
$$\Delta Rad = \rho_j (F_{i-j} A_j / A_i) \Delta B_i ;$$

$$B_j += \Delta Rad ;$$

$$\Delta B_j += \Delta Rad ;$$

$$\Delta B_i = 0 ;$$

Soluciones parciales = Refinamiento progresivo



PROGRESSIVE SOLUTION

The above images show increasing levels of global diffuse illumination. From left to right: 0 bounces, 1 bounce, 3 bounces.

Soluciones parciales = Refinamiento progresivo

- El cálculo de F_{i-j} precisa de un hemisferio diferente por i .
- Cuando se irradia por primera vez del parche i , este irradia $E_i = \Delta B_i$
- Las siguientes veces que se irradie del parche i , solo se considera el incremento ΔB_i logrado desde la última vez que se irradizó.
- Conviene elegir los parches que al irradiar den el mayor impacto, que son los que tienen más energía para irradiar.

Ductos de generación (Rendering)

DUCTOS DE ILUMINACIÓN LOCAL

Z-buffer y sombreado de Gouraud.

Z-buffer y sombreado de Phong.

Algoritmo de prioridad de listas y sombreado de Phong.

DUCTOS DE ILUMINACIÓN GLOBAL

Radiosidad.

Traza de rayos.

Z-buffer y sombreado de Gouraud

- 1) **Recorrido de base de datos.** No importa el orden de lectura de las primitivas, dado que Z-buffer no lo precisa.
- 2) **Transformación de modelado.** Pasar las primitivas al sistema de W.C.
- 3) **Aceptación/rechazo trivial.** Cohen-Sutherland por ejemplo, pero sin recorte de primitivas. Se hace aquí para evitar cálculos innecesarios en la iluminación.
- 4) **Iluminación.** Si las primitivas vienen con sus normales, aplicarles también la transformación de modelado.
Se evalúa la ecuación de iluminación en cada vértice en W.C., para que de bien.

Z-buffer y sombreado de Gouraud

- 5) **Transformación de vista.** Transformar los objetos a NPC (coordenadas de proyección normalizada).
- 6) **Recortes.** Recortar las primitivas respecto al volumen de vista canónico. Si hay recortes parciales, entonces calcular las intensidades de iluminación para los nuevos vértices.
- 7) **División entre W , correspondencia al área de vista.**
- 8) **Generación de barrido.** Se utiliza Z-buffer, que hace la generación de barrido, discretizando e interpolando para hallar los z y las intensidades. La intensidad se puede alterar de acuerdo a la atenuación atmosférica.
- 9) **Dibujo.**

Z-buffer y sombreado de Gouraud

Temas a tratar:

Las superficies curvas (parches bicúbicos, esferas, etc.) deben teselarse (generar mallas poligonales que las aproximen).

El proceso de teselado debe contemplar el tamaño del parche en la pantalla.

Z-buffer y sombreado de Phong

- 1) **Recorrido de base de datos.** No importa el orden de lectura de las primitivas, dado que Z-buffer no lo precisa.
- 2) **Transformación de modelado.** Pasar las primitivas al sistema de W.C.
- 3) **Aceptación/rechazo trivial.** Cohen-Sutherland por ejemplo, pero sin recorte de primitivas. Se hace aquí para evitar cálculos innecesarios en la iluminación.
- 4) **Transformación de vista.** Transformar los objetos a NPC (coordenadas de proyección normalizada).
- 5) **Recortes.** Recortar las primitivas respecto al volumen de vista. Si hay recortes parciales, entonces calcular las intensidades de iluminación para los nuevos vértices.

Z-buffer y sombreado de Phong

- 6) **División entre W, correspondencia al área de vista.**
- 7) **Generación de barrido.** Se utiliza Z-buffer, que hace la generación de barrido, discretizando e interpolando para hallar los z y las intensidades. La intensidad se puede alterar de acuerdo a la atenuación atmosférica.
Se aplica interpolación de Phong. Para interpolar correctamente las normales y evaluar la ecuación de iluminación, se debe hacer un mapeo invertido de cada punto y su normal a un sistema de coordenadas isométrico a las coordenadas mundiales.
- 8) **Dibujo.**

Algoritmo de prioridad de listas y sombreado de Phong

Son 2 ductos relacionados.

Primer Ducto

- 1) **Recorrido de la base de datos.**
- 2) **Transformación de modelado.**
- 3) **Determinación preliminar de superficies visibles.** Por ejemplo con BSP (Binary Space Partitioning), dado que es independiente de la vista. Si los polígonos se dividen, hay que calcular información de sombreado en los nuevos vértices
- 4) **Generación de nueva base de datos.**

Algoritmo de prioridad de listas y sombreado de Phong

Segundo Ducto

- 1) Nuevo recorrido de la base de datos.
- 2) Aceptación/rechazo trivial.
- 3) Transformación de vista
- 4) Recortes
- 5) División entre W , correspondencia al área de vista tridimensional.
- 6) Generación de barrido (incluyendo iluminación) Aquí no es necesario Z-buffer, basta dibujar los polígonos en el orden correcto.
- 7) Dibujo.

Radiosidad

Son 2 ductos relacionados.

Primer Ducto

- 1) Recorrido de la base de datos.**
- 2) Transformación de modelado.**
- 3) Cálculo de intensidad de vértices usando el método de radiosidad.**
- 4) Generación de nueva base de datos.**

Radiosidad

Segundo Ducto

- 1) Nuevo recorrido de la base de datos.
- 2) Aceptación/rechazo trivial.
- 3) Transformación de vista
- 4) Recortes
- 5) División entre W , correspondencia al área de vista tridimensional.
- 6) **Generación de barrido** Se puede aplicar una variante del Z-buffer y sombreado de Gouraud que elimina la etapa de iluminación.
- 7) **Dibujo.**

Rastreo de rayos

Los objetos visibles en cada pixel y su iluminación se consideran en coordenadas de mundo.

- 1) **Recorrido de la base de datos.**
- 2) **Transformación de modelado**
- 3) **traza de rayos**
- 4) **Dibujo.**

Refinamiento Progresivo

- Dado que las imágenes muchas veces se ven durante un tiempo finito, puede ser interesante generar primero versiones burdas de la misma.
- Usar primero modelos simples, sombreado sencillo sin eliminación de artefactos de discretización.
- Traza de rayos y radiosidad son particularmente apropiados.