

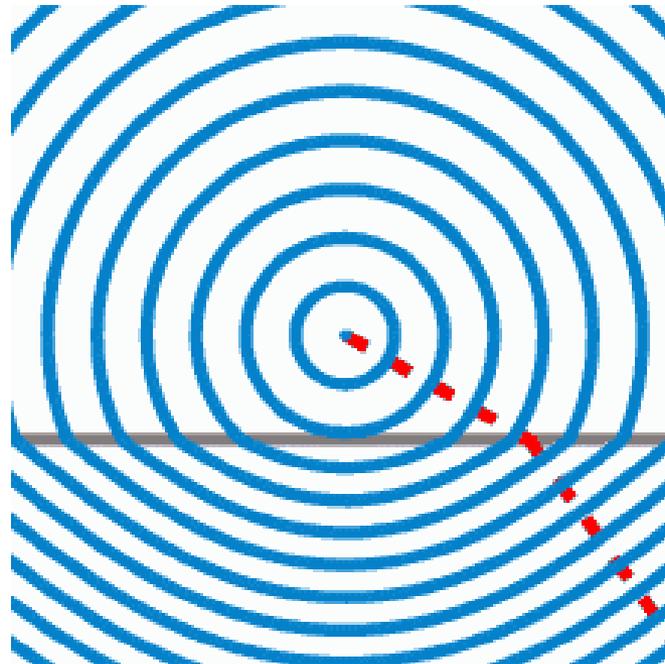
Computación Gráfica

Eduardo Fernández

Resumen del capítulo

- Modelos de iluminación global.
- Ductos Gráficos.

Transparencia refractiva

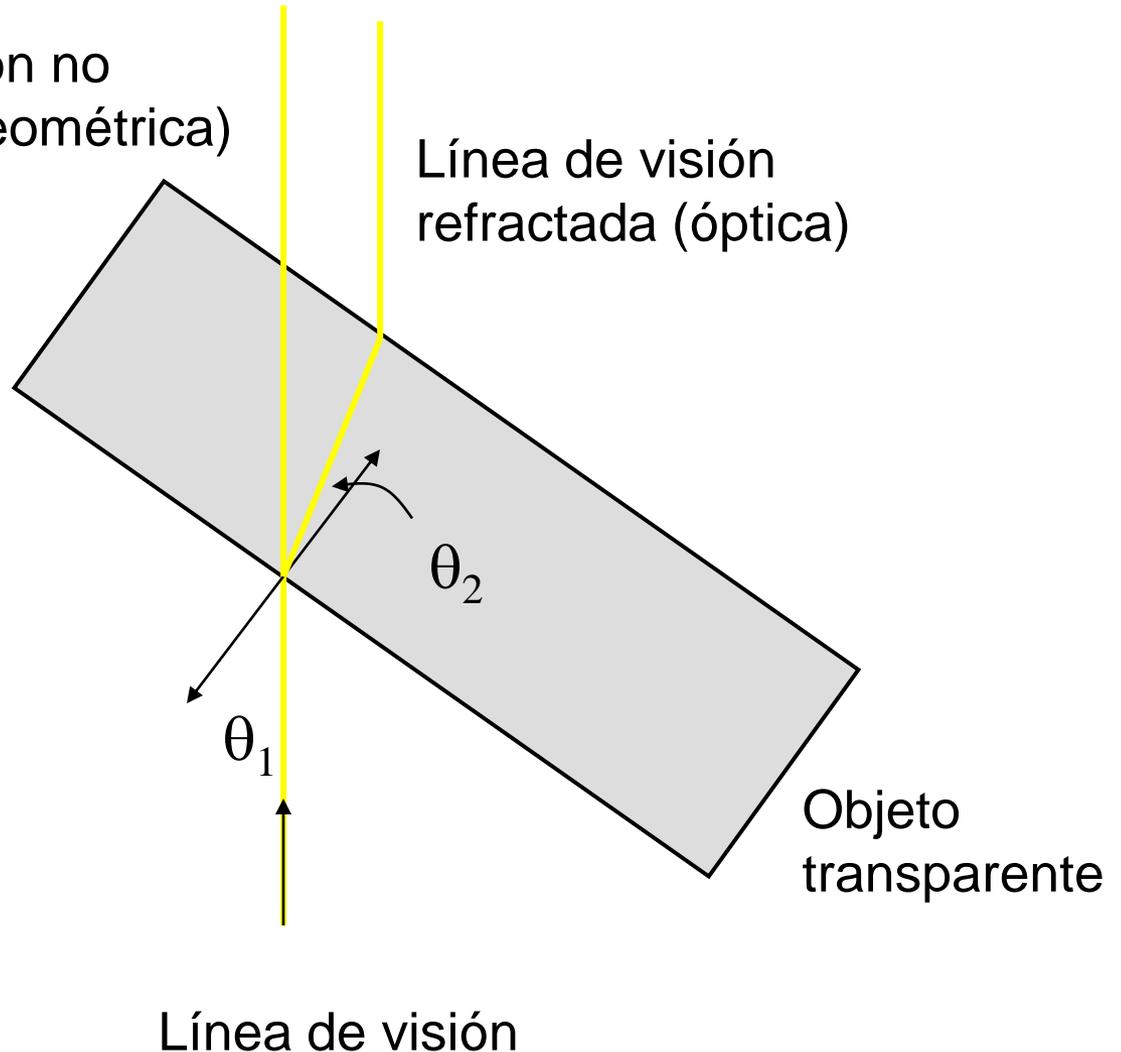


Algoritmos de iluminación global

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)

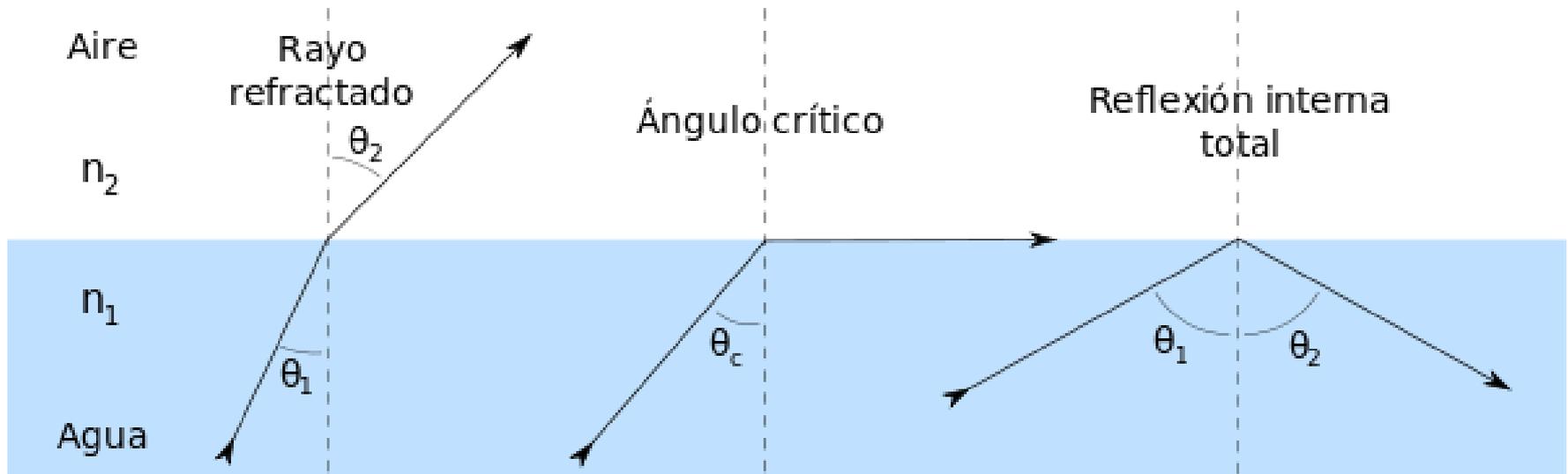
- Vacuum ...1.00000 (exactly)
- Air (STP) ... 1.00029
- Acetone ... 1.36
- Alcohol ... 1.329
- Amorphous Selenium ... 2.92
- Calspar1 ...1.66
- Calspar2 ...1.486
- Carbon Disulfide ...1.63
- Chromium Oxide ...2.705
- Copper Oxide ...2.705
- Crown Glass ...1.52
- Crystal ...2.00
- Diamond ...2.417
- Emerald ...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.



Iluminación y Sombreado

Basado en: **Capítulo 14**

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

Foley – Van Dam – Feiner – Hughes - Phillips

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:

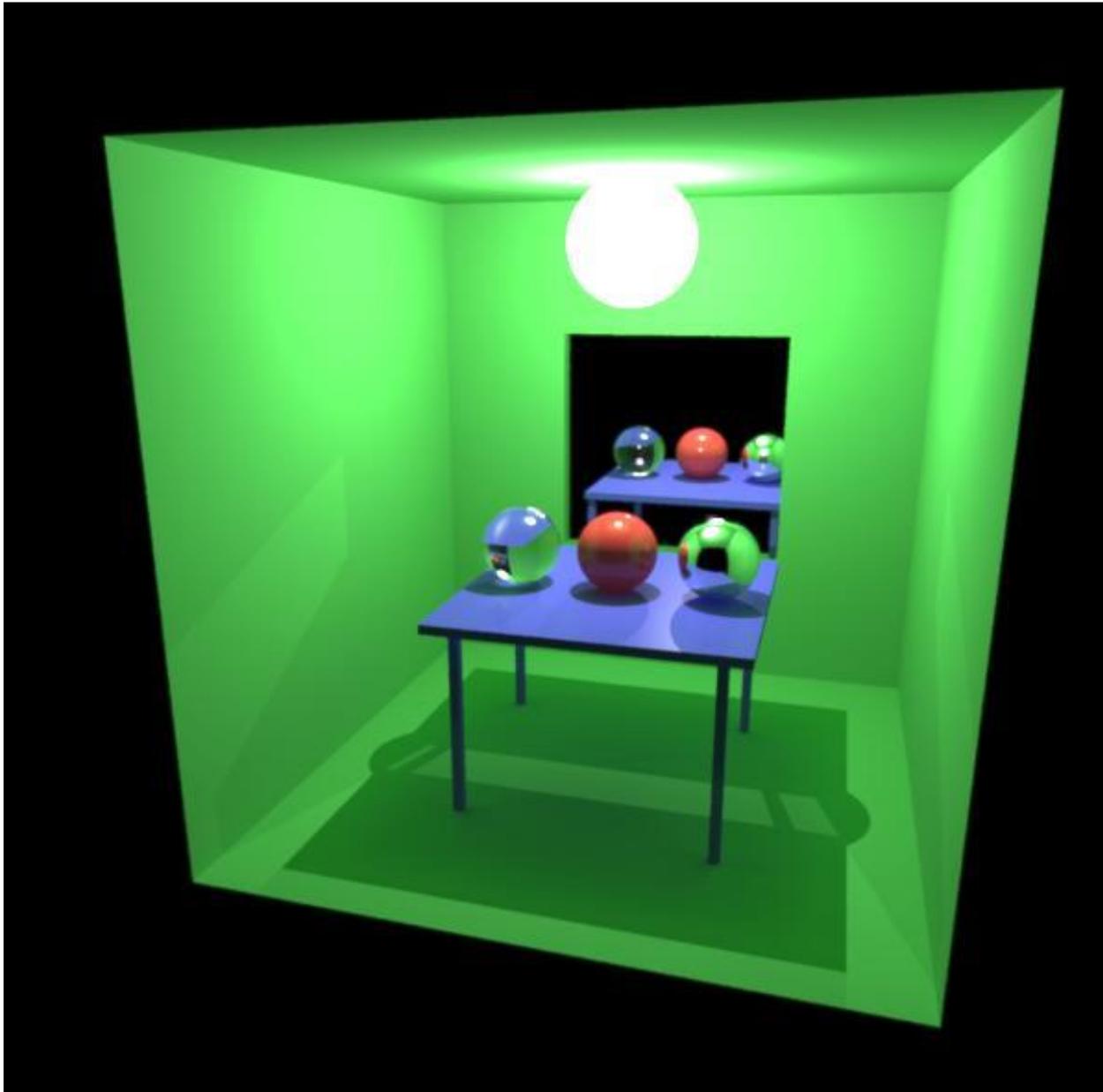
- Traza de rayos recursiva (ray tracing)

Es dependiente de la ubicación del observador

- Radiosidad (Radiosity)

Es independiente de la ubicación del observador

Algoritmos de iluminación global



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.

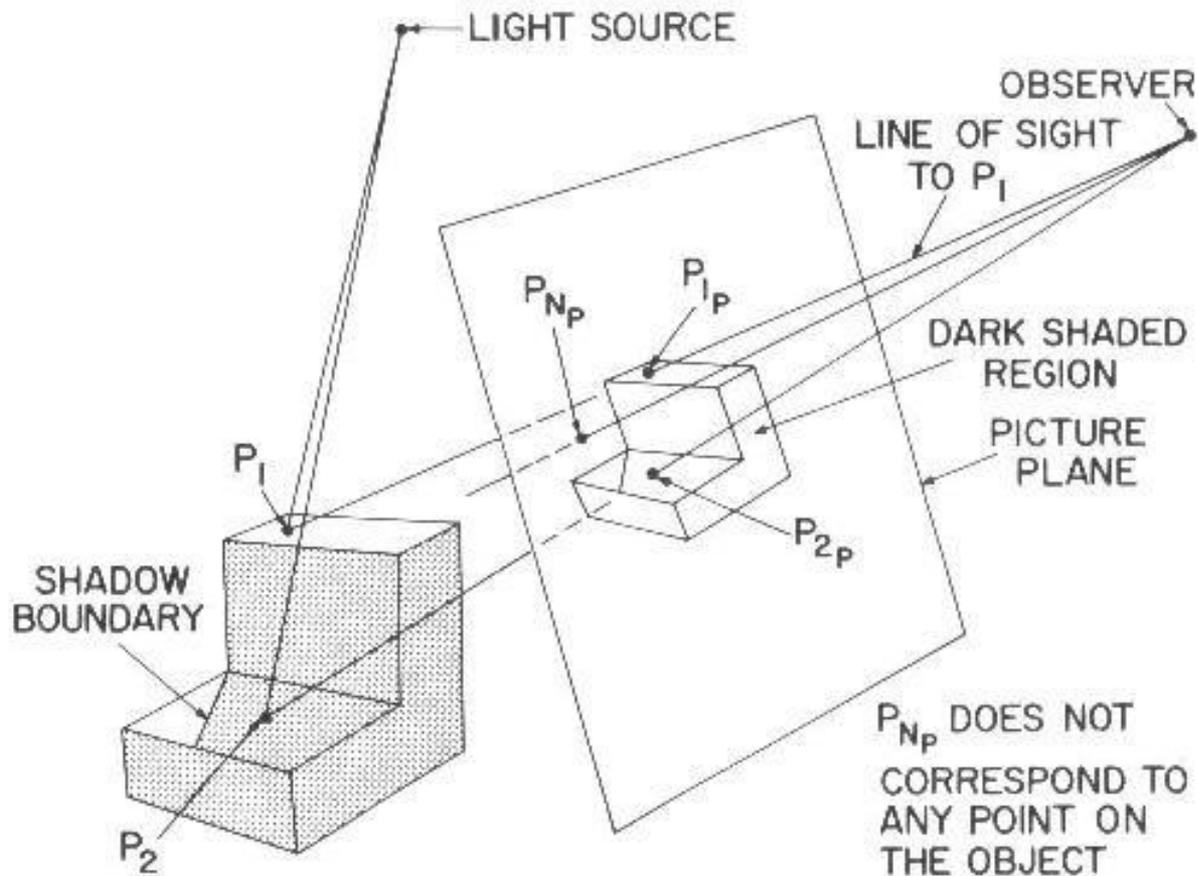
Vale 0 cuando x y x' están ocultos entre si.

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

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

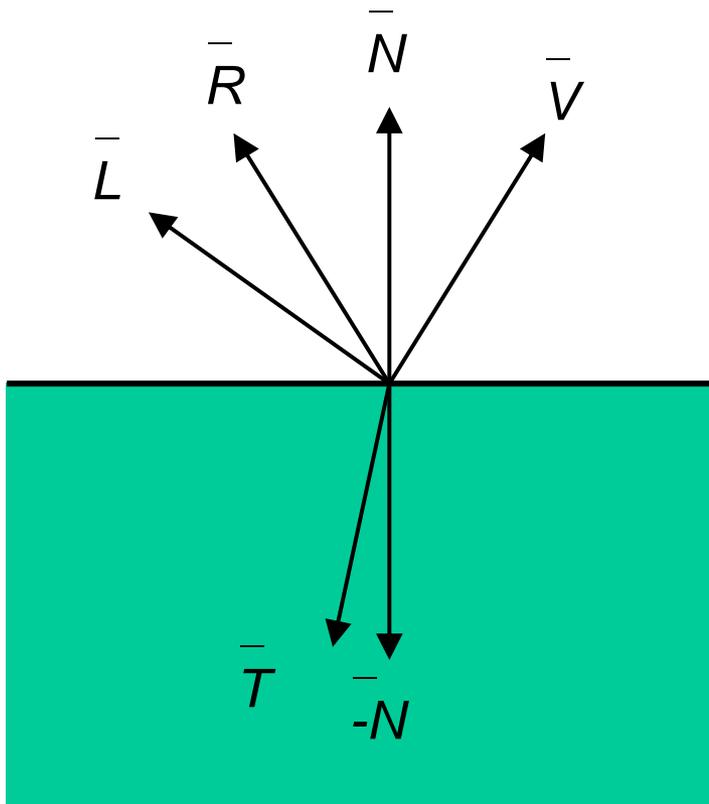
$\rho(x, x', x'')$ se relaciona con la intensidad de luz reflejada en x' , que sale de x'' y llega a x

Traza de rayos (ray-tracing)



Determinando si un punto de un objeto está en sombra.

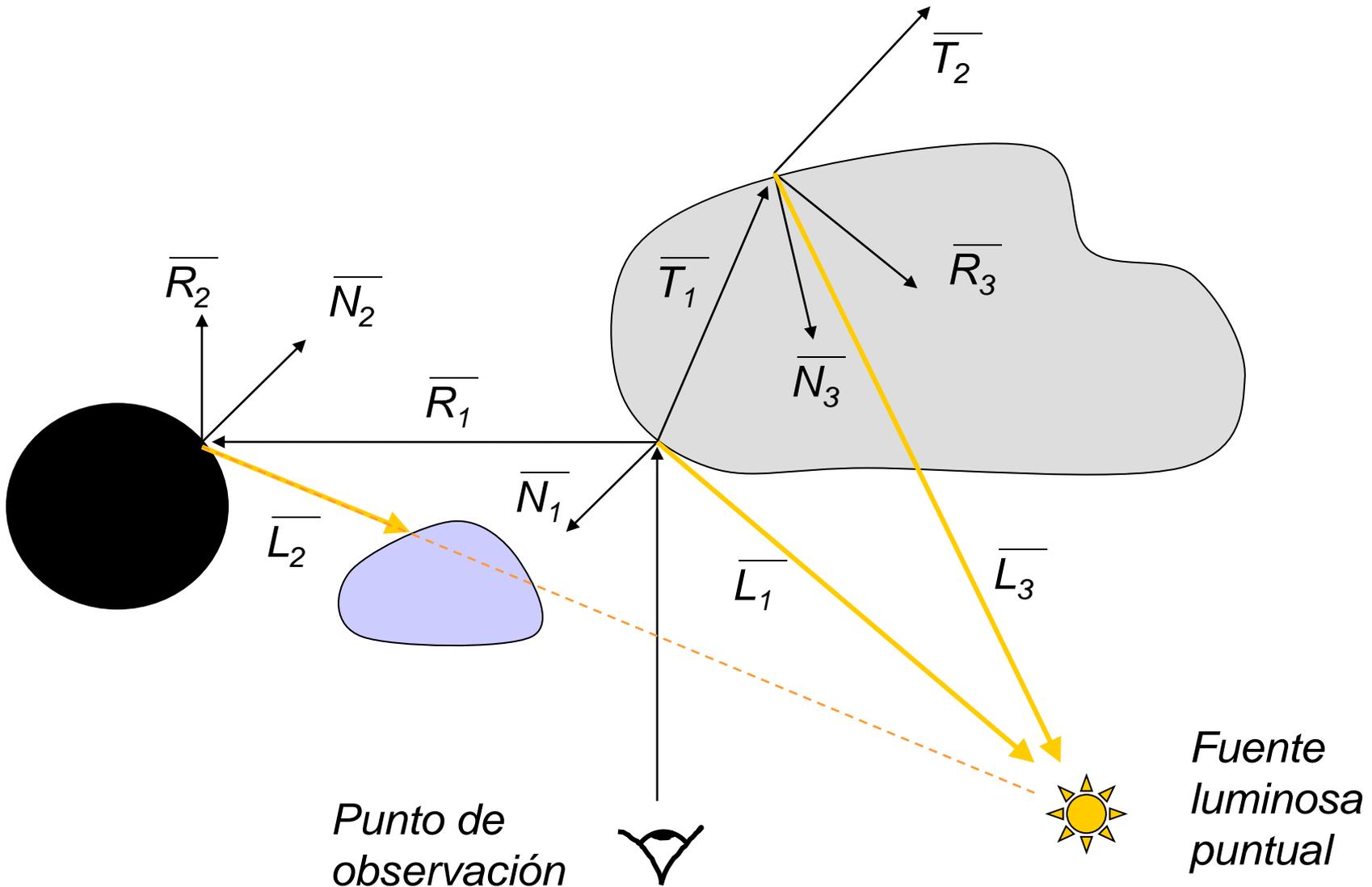
Traza de rayos recursiva (Whitted)



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

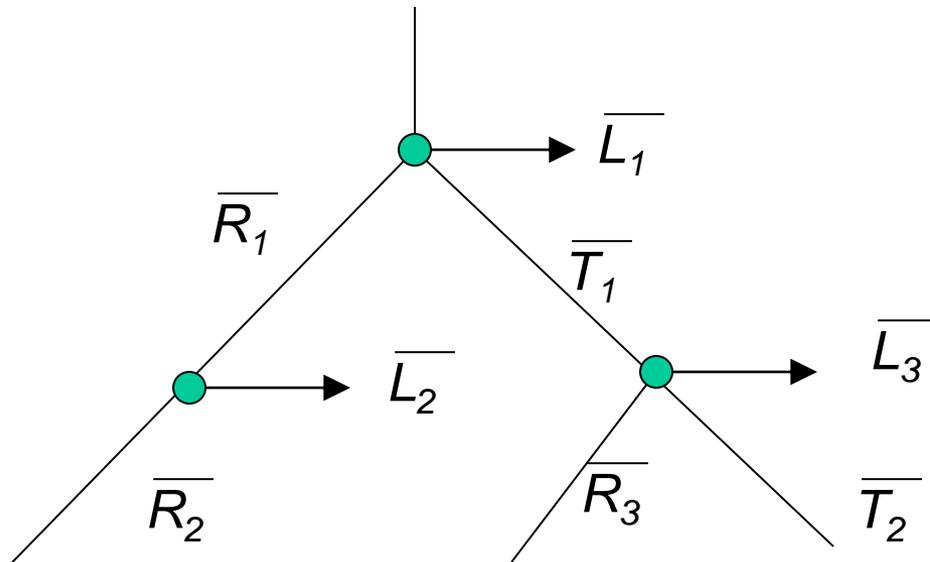
Rayos primarios son los que parten del observador.

Traza de rayos recursiva (Whitted)

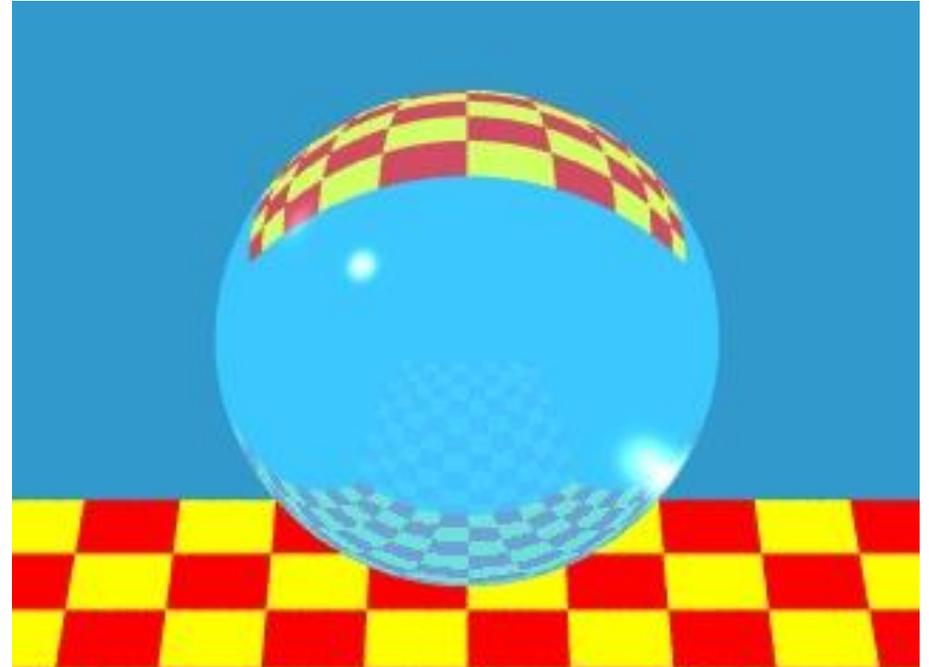
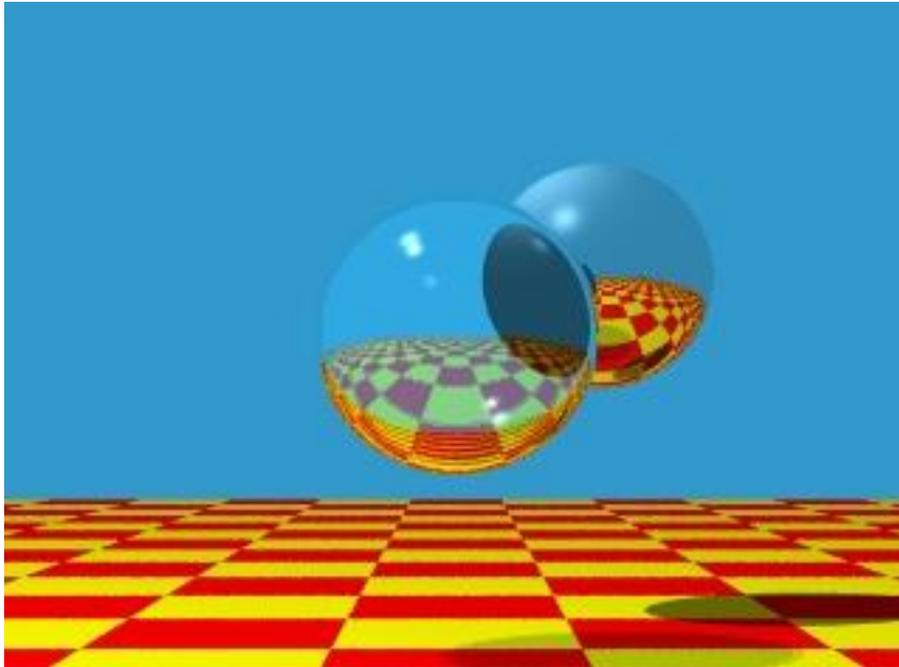


Traza de rayos recursiva (Whitted)

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 píxel en la línea de barrido){  
    determinar rayo por centro de proyección y píxel;  
    píxel=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 el punto a la luz;  
        if (producto punto entre normal y dirección de la 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.

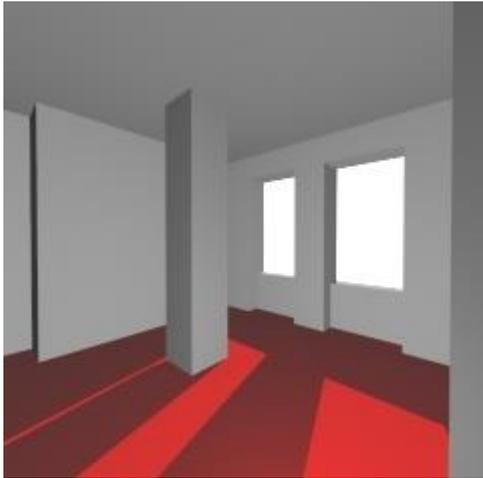
- Los rayos generados pueden intersectar los objetos de donde salen.

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 de reflexión difusa.

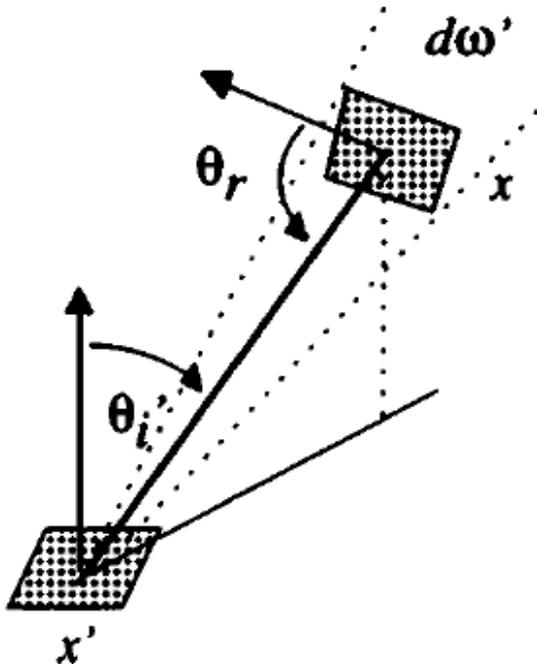
El color de las superficies de reflexión difusa es independiente del ángulo de vista.

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

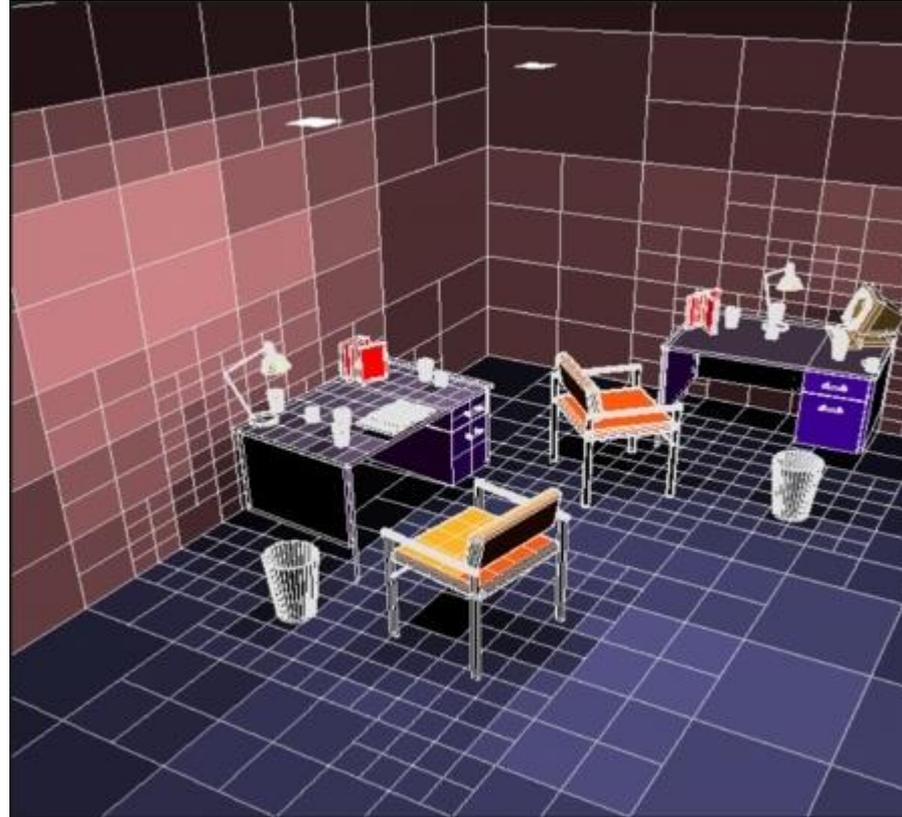
$$G(x, x') = \frac{V(x, x') \cos \theta_i' \cos \theta_r}{|x - x'|^2}$$

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

Ecuación de Radiosidad (continua)

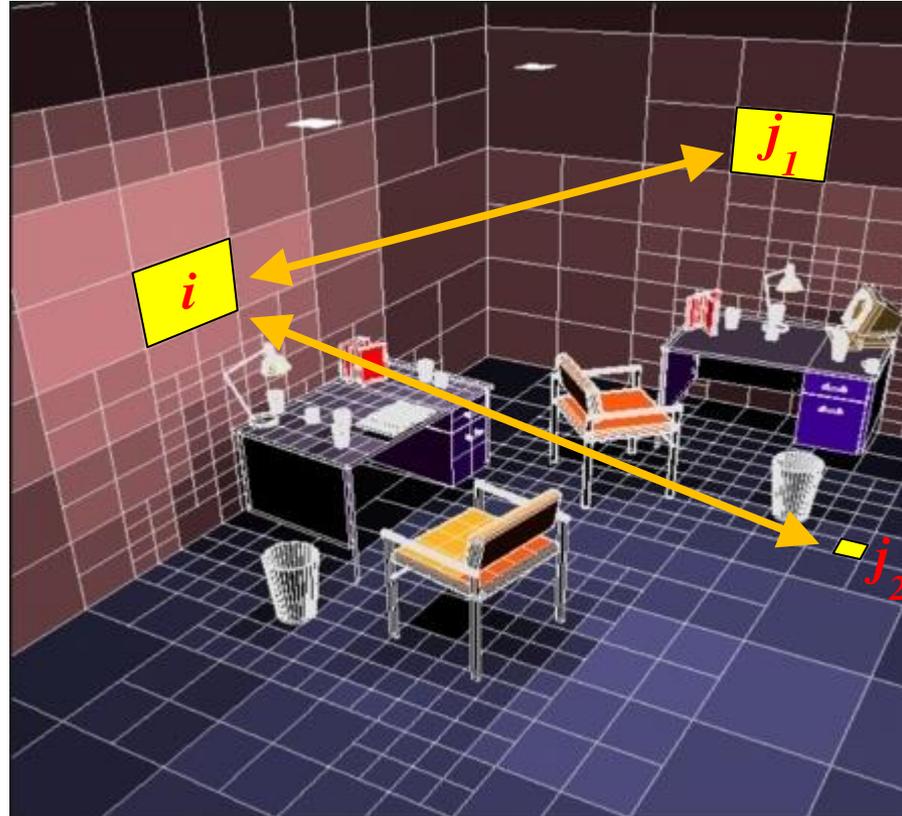


Ecuación de Radiosidad (discreta)



Las superficies se discretizan en n polígonos (llamados parches)

Ecuación de Radiosidad (discreta)



$$B(i) = E(i) + R(i) \sum_{j=1..n} B(j) \frac{\mathbf{F}(j,i)A(j)}{A(i)} \quad , \quad \forall i \in \{1..n\}$$

$\mathbf{F}(j,i)$ Es la proporción de luz de j que incide en i .

Ecuación de Radiosidad (discreta)

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

$B_i = B(i)$ radiosidad total de los parche i

$E_i = E(i)$ radiosidad emitida por el parche i .

$\rho_i = R(i)$ reflectividad del parche i (sin dimensiones).

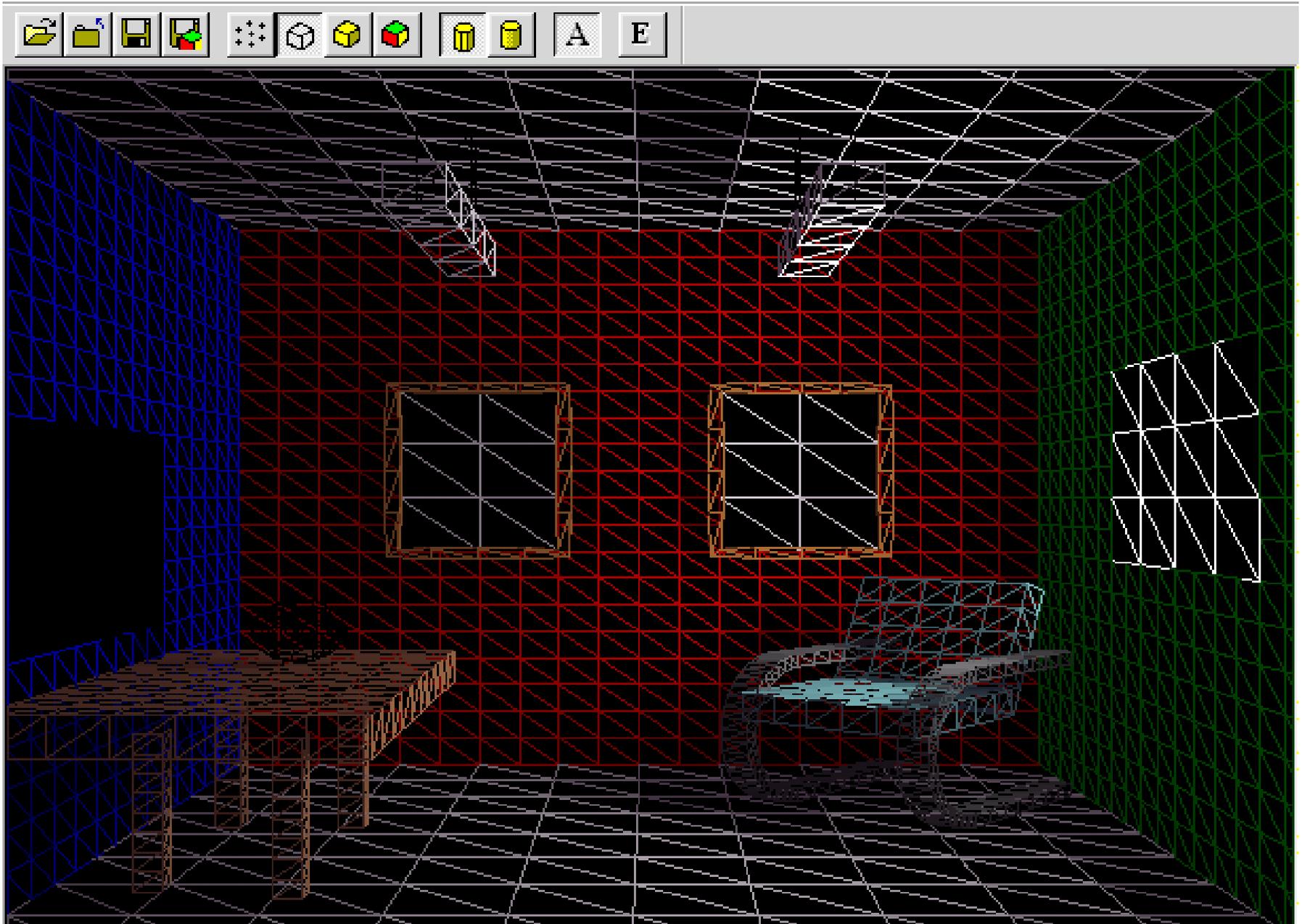
$F_{j-i} = \mathbf{F}(j,i)$ Factor de forma, fracción de energía que va del parche j al i (sin dimensiones)

$A_i = A(i)$ Area del parche i

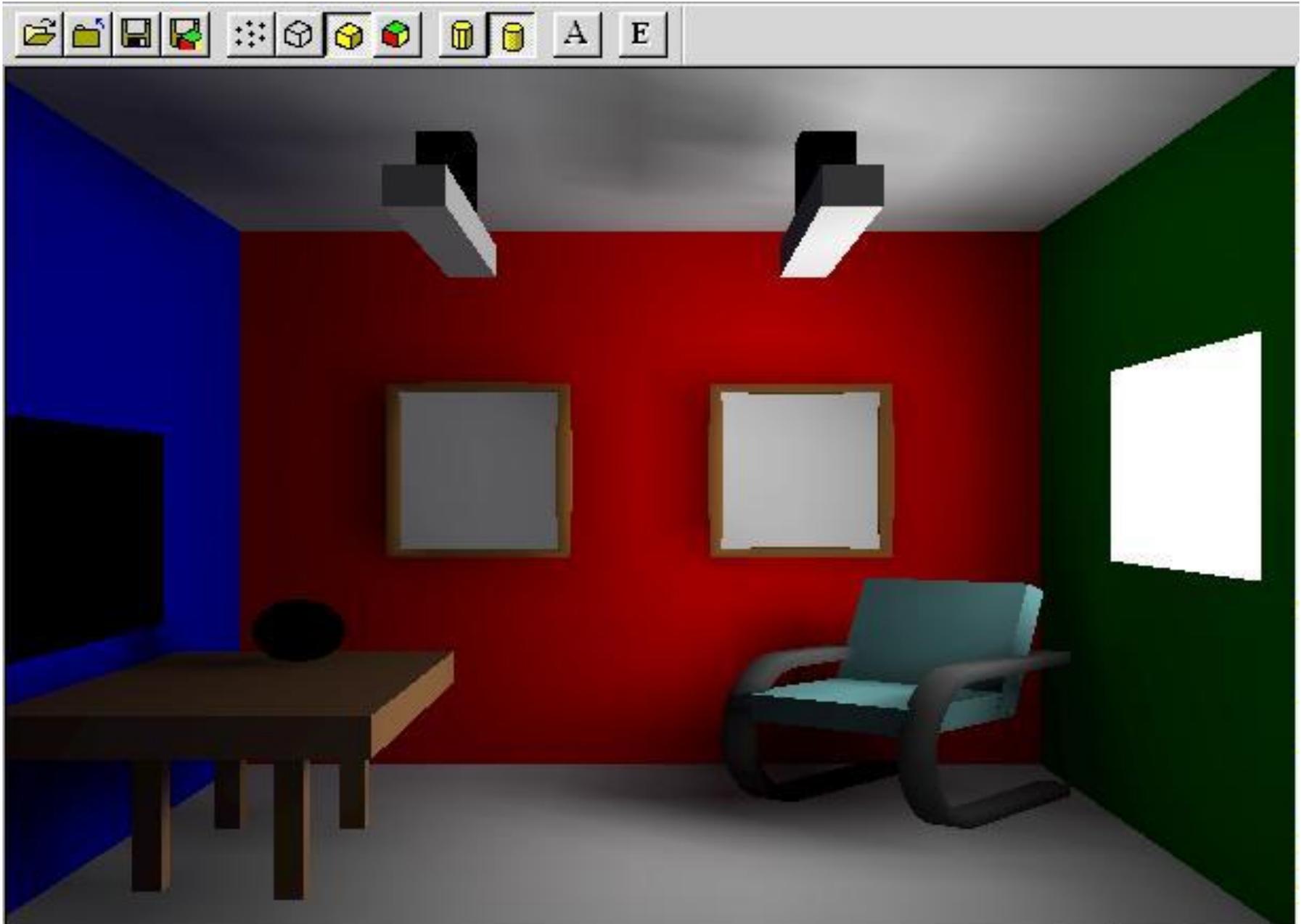
Métodos de Radiosidad

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

Luego de este cálculo, se generan las vistas, sólo 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: $B_i = E_i + \rho_i \sum_{1 \leq j \leq n} B_j F_{j-i} (A_j/A_i)$

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

O, de forma equivalente: $B_i - \rho_i \sum_{1 \leq j \leq n} F_{i-j} B_j = E_i$

Ecuación de Radiosidad

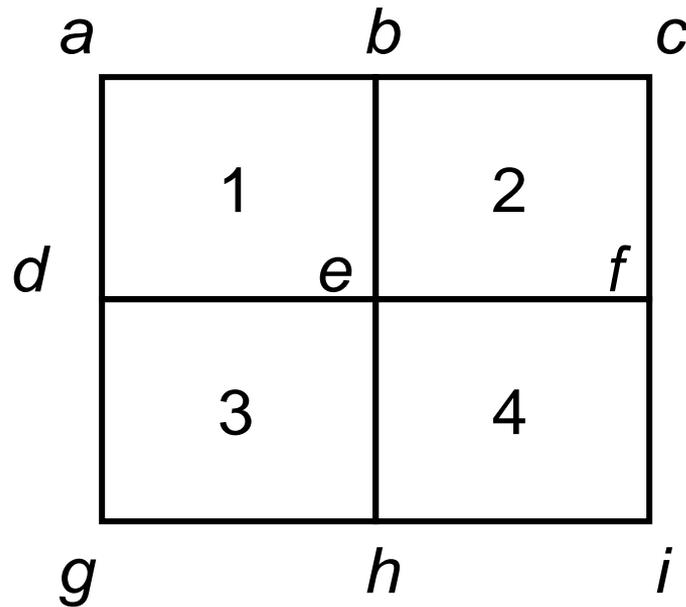
$$B_i - \rho_i \sum_{1 \leq j \leq n} F_{i-j} B_j = E_i \quad \forall i$$

Esto permite generar un sistema de ecuaciones lineales:

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

Se puede resolver por Gauss-Seidel

Determinación de la radiosidad de los vértices



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

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

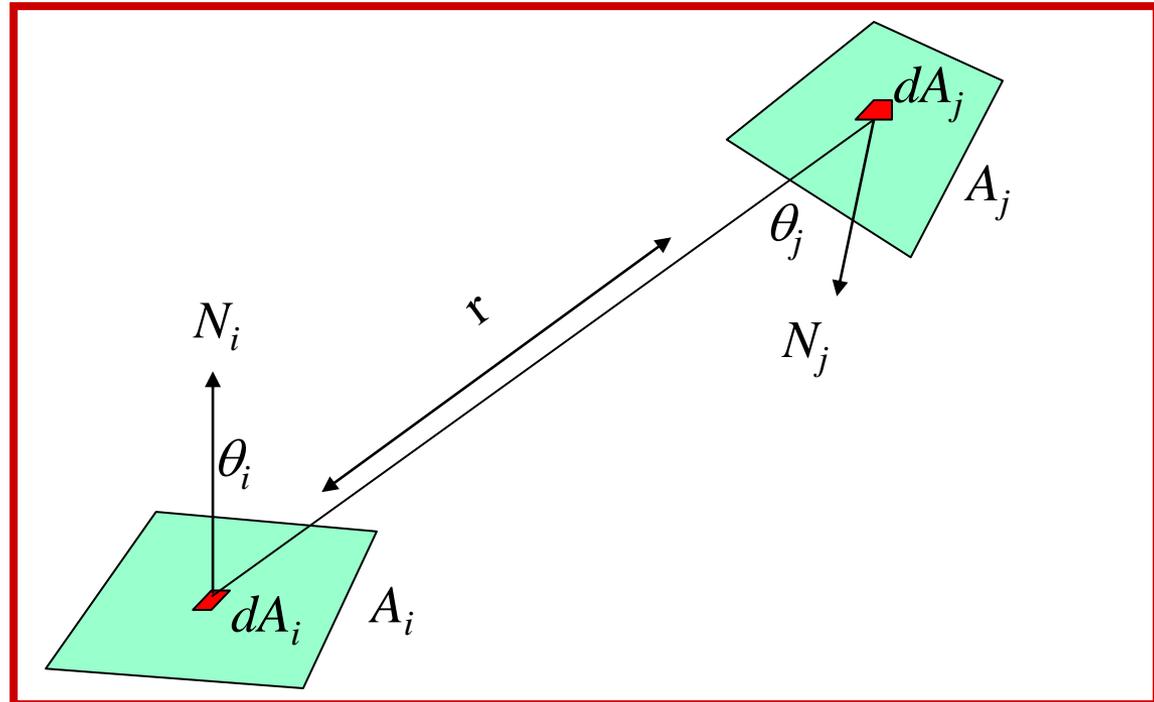
$$(B_a + B_e)/2 = B_1 \quad \Rightarrow \quad B_a = 2B_1 - B_e$$

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.

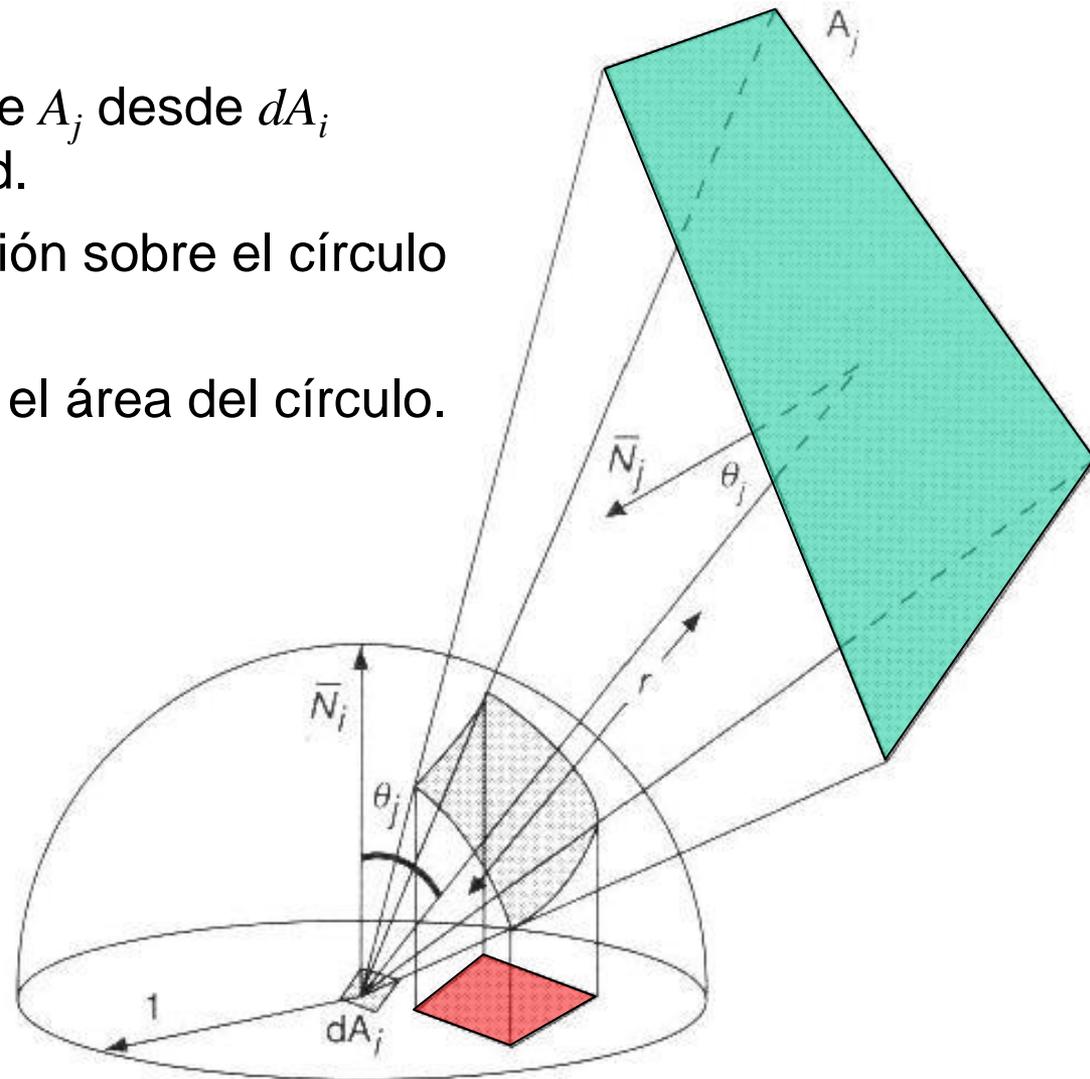
=>

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

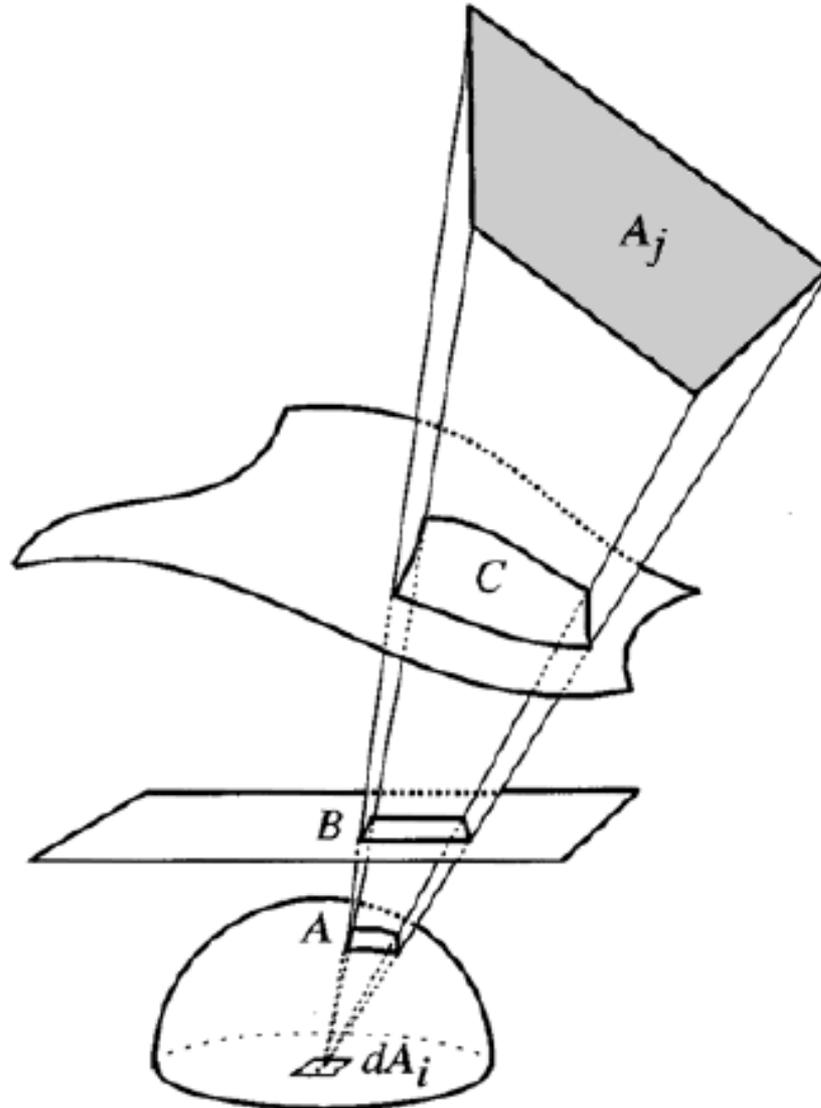
Cálculo de los *F*actores de *F*orma

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.



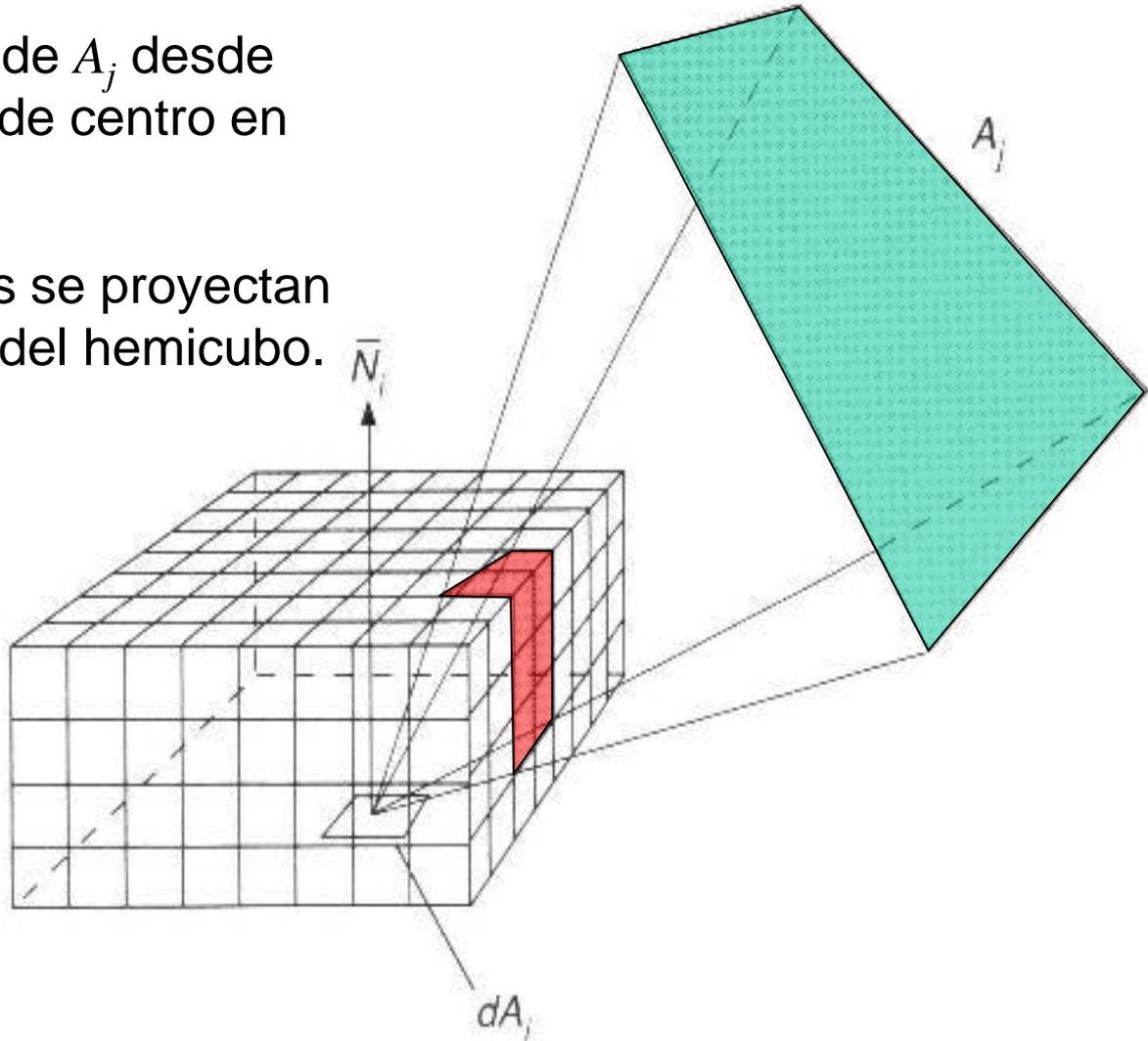
Cálculo de los *F*actores de *F*orma



Cálculo de los *F*actores de *F*orma

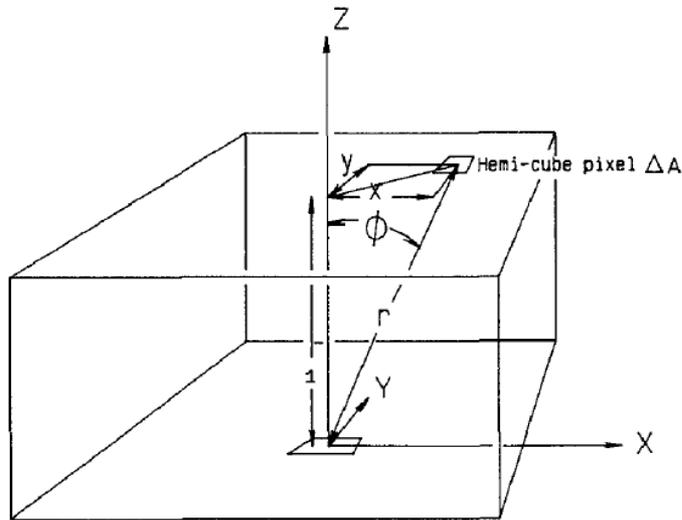
Calcular F_{di-j} equivale a:

- 1) proyectar el área visible de A_j desde dA_i sobre 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 *Factor de Forma* (*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 Factores de Forma

Cálculo del FdeF de cada celda o píxel



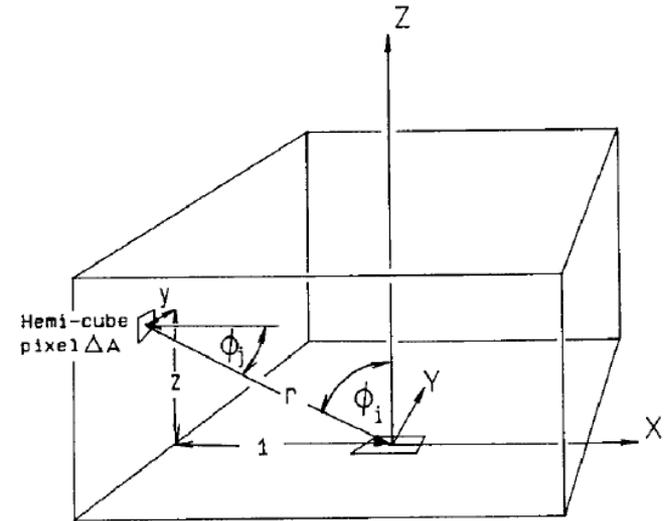
TOP OF HEMI-CUBE

$$r = \sqrt{x^2 + y^2 + 1}$$

$$\cos \phi_i = \cos \phi_j$$

$$\cos \phi = \frac{1}{\sqrt{x^2 + y^2 + 1}}$$

$$\begin{aligned} \Delta \text{Form-factor} &= \frac{\cos \phi_i \cos \phi_j}{\pi r^2} \Delta A \\ &= \frac{1}{\pi (x^2 + y^2 + 1)^2} \Delta A \end{aligned}$$



SIDE OF HEMI-CUBE

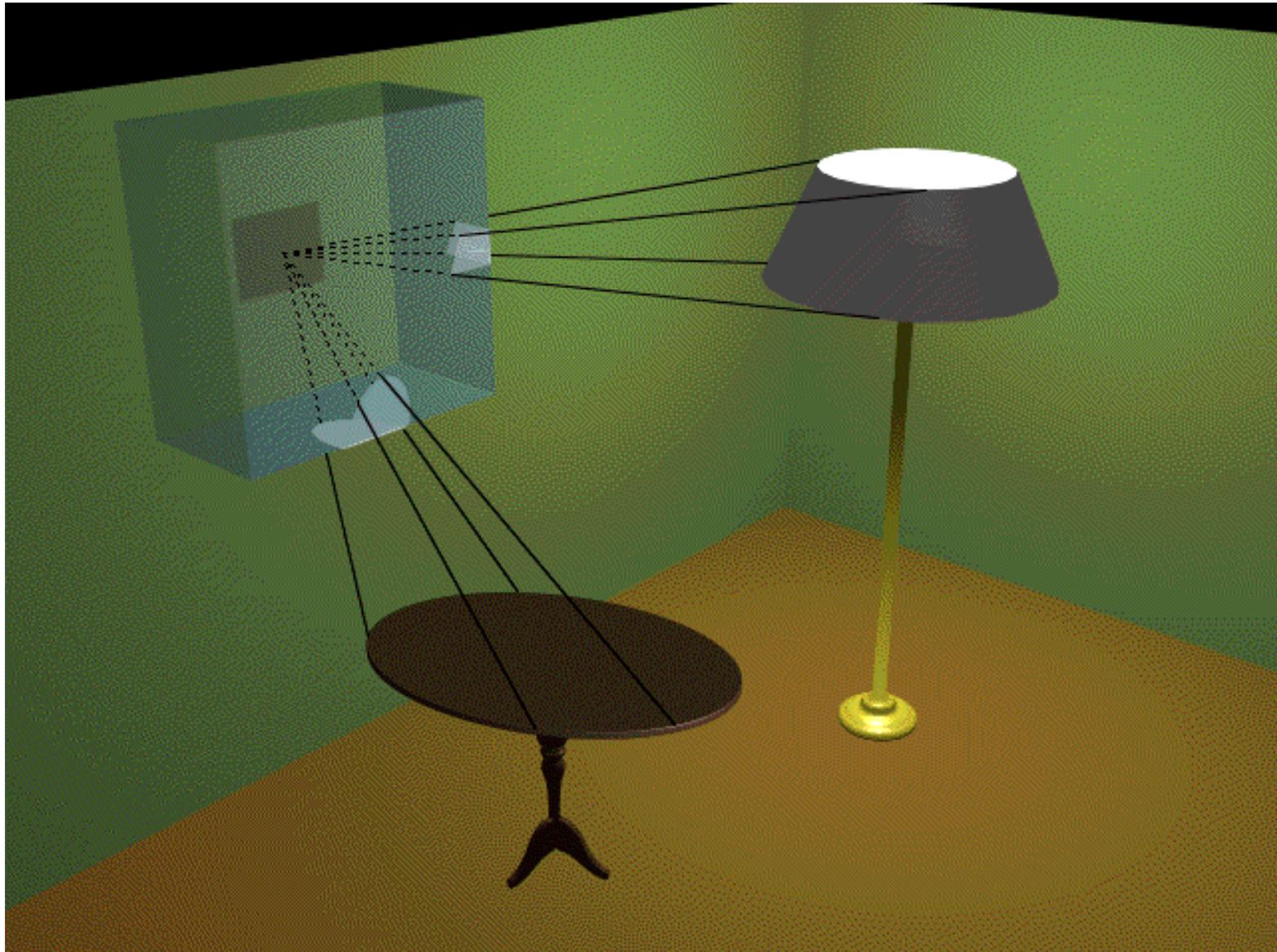
$$r = \sqrt{y^2 + z^2 + 1}$$

$$\cos \phi_i = \frac{z}{\sqrt{y^2 + z^2 + 1}}$$

$$\cos \phi_j = \frac{1}{\sqrt{y^2 + z^2 + 1}}$$

$$\begin{aligned} \Delta \text{Form-factor} &= \frac{\cos \phi_i \cos \phi_j}{\pi r^2} \Delta A \\ &= \frac{z}{\pi (y^2 + z^2 + 1)^2} \Delta A \end{aligned}$$

Cálculo de los *F* Factores de *F*orma

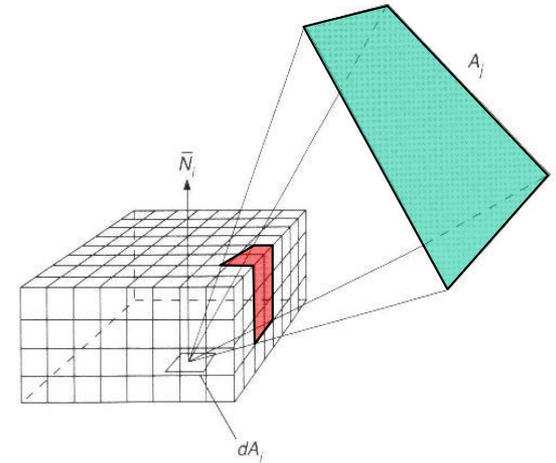


Cálculo de los *F*actores de *F*orma

Este algoritmo se puede realizar aplicando el algoritmo de z-buffer a cada lado del hemicubo, 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.



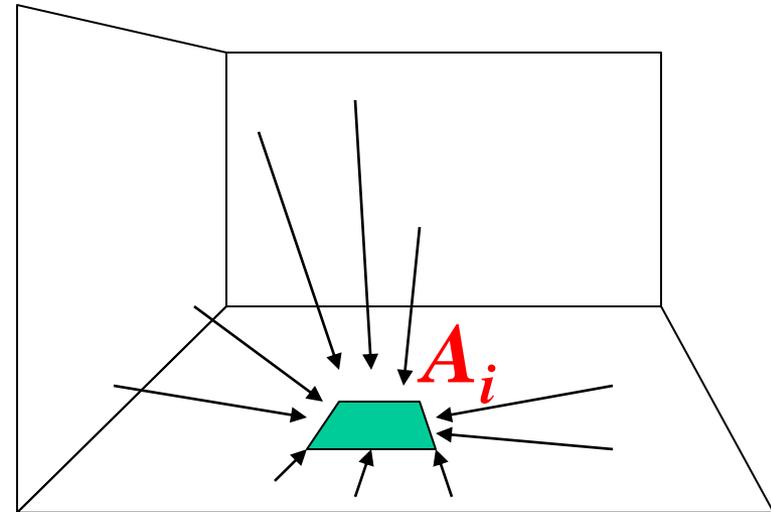
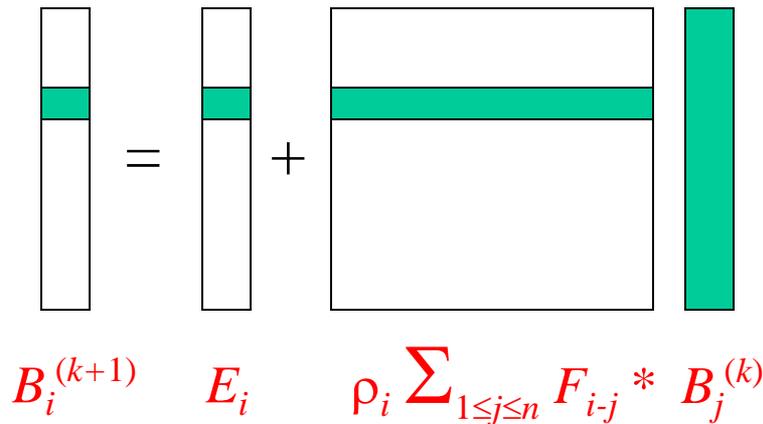
Soluciones parciales = Refinamiento progresivo

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 .

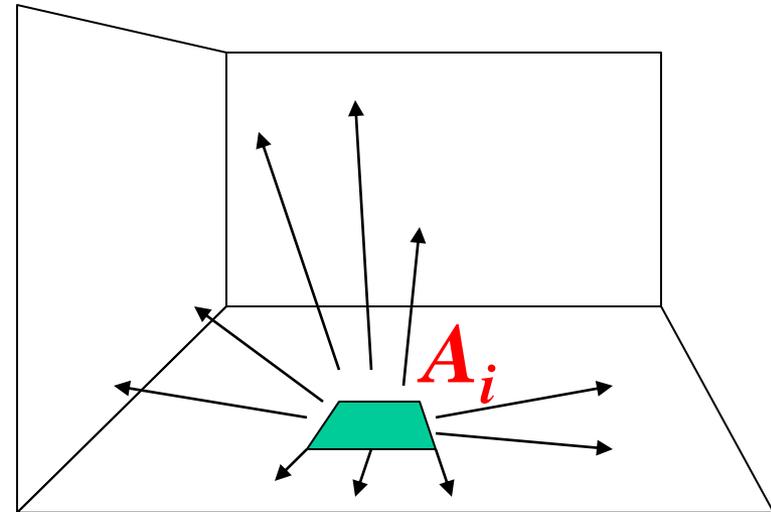
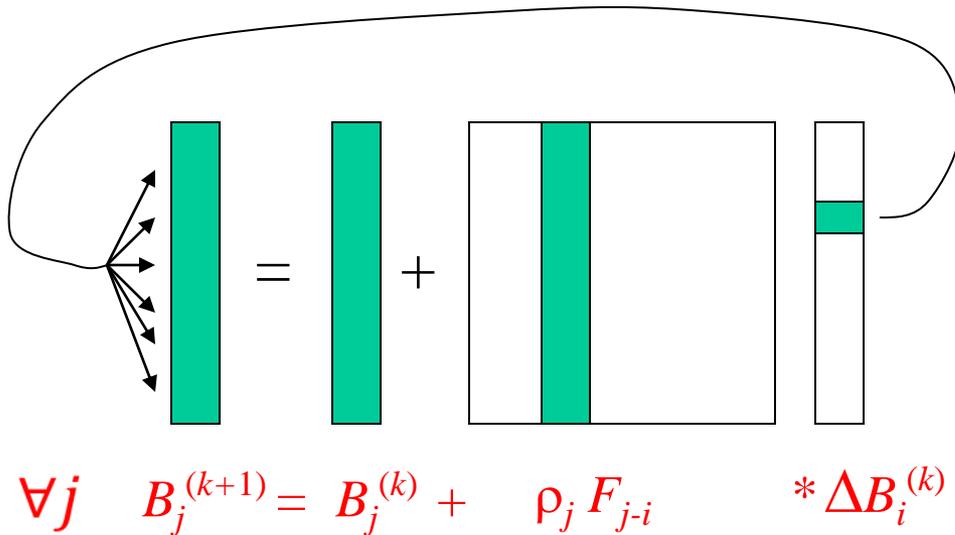
Por cada fila i procesada de ρF , se obtiene un nuevo valor de radiosidad para el parche i .



Soluciones parciales = Refinamiento progresivo

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

Se realiza $\rho_j F_{j-i} \Delta B_i^{(k)}$ para cada celda j de la columna i de la matriz ρF , y el resultado se suma a la radiosidad del parche j .



Soluciones parciales = Refinamiento progresivo

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

While (no hay convergencia)

Hallar i tal que $\Delta B_i * A_i$ sea máximo.

Calcular F_{i-j} para todo j

For ($j = 1:n$)

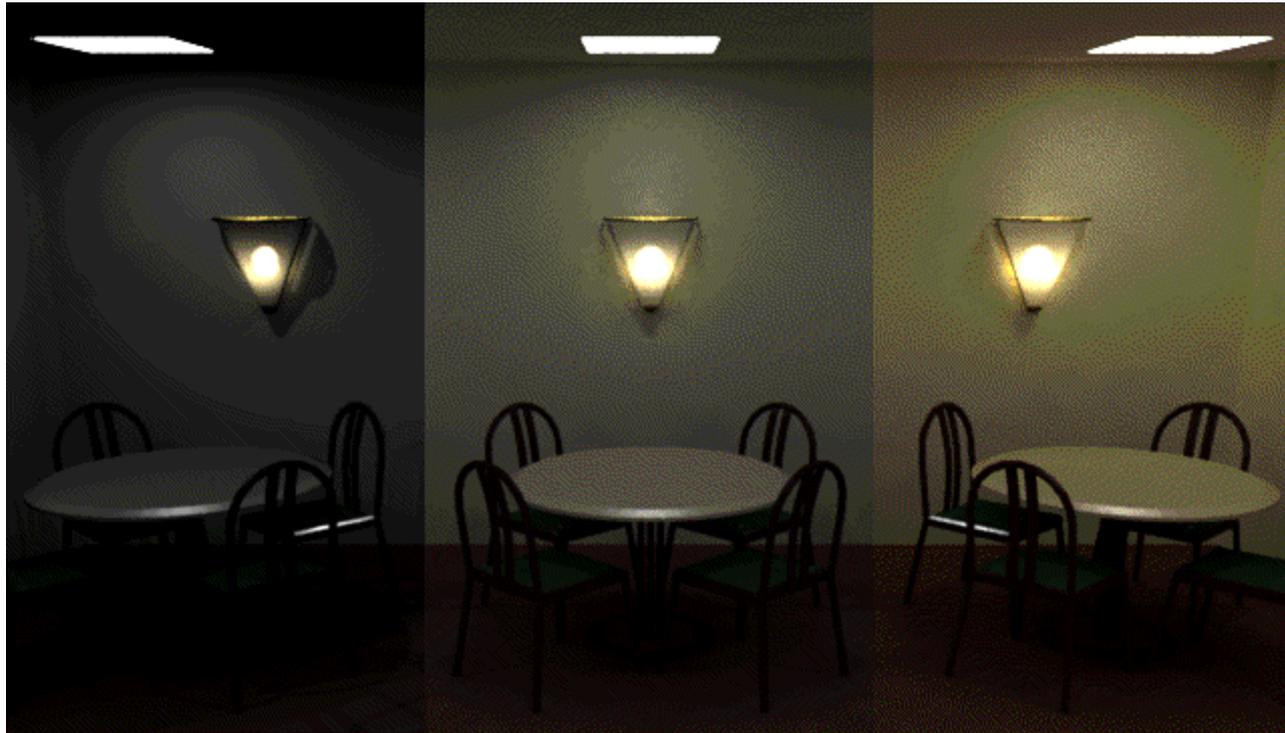
$$\Delta Rad = \rho_j (F_{i-j} A_i / A_j) \Delta B_i ;$$

$$B_j += \Delta Rad ;$$

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

$$\Delta B_i = 0 ;$$

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 irradió.
- 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 normales 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.**

Traza de rayos

Los objetos visibles en cada píxel 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.