

Ampliación de Informática Gráfica

Tema 4. Modelos de Iluminación Global

Índice del Tema.

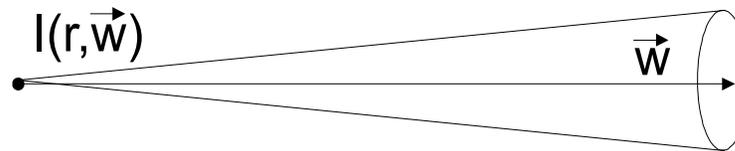
- **Modelo de Radiación de Luz. Ecuación de Iluminación**
- **Aproximación Óptica al modelado global de la luz: Trazado de Rayos.**
 - **Trazado de Rayos Inverso.**
 - **Árbol de Rayos.**
 - **Cálculo de Intersecciones**
 - **Optimización del coste.**
- **Aproximación Termodinámica al modelado global de la luz: Radiosidad.**
 - **Ecuación General de Radiosidad.**
 - **Cálculo de Factores de Forma. Hemicubo.**
 - **Métodos de cálculo de radiosidad.**
 - **Asignación de color por vértice.**

Modelos de Radiación de Luz. Ec. Rendering

- **En este tema vamos a ocuparnos de realizar una revisión más detallada de las causas de la luz existente en un punto del espacio**
 - **Luz directa de fuentes.**
 - **Luz transmitida a través de objetos.**
 - **Luz reflejada por otros objetos.**

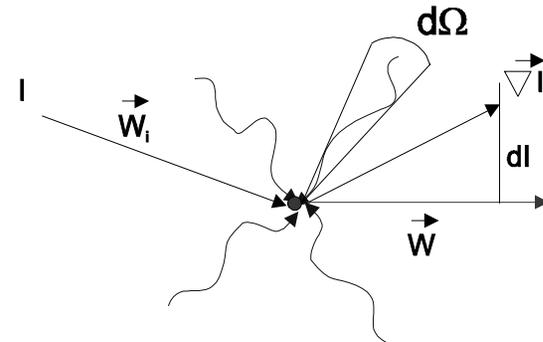
Modelos de Radiación de Luz. Ec. Rendering

- La luz que se transmite a partir de un punto la podemos identificar a partir de una ecuación de radiación



- La transmisión de luz a través del medio la podemos caracterizar por:

$$I = \text{radiancia} = \frac{E}{t \cdot S \cdot \Omega} = f(\nu)$$



$$dI = \vec{\omega} \cdot \vec{\nabla} I(\vec{r}, \vec{\omega}) = -K_t \cdot I(\vec{r}, \vec{\omega}) + K_a I_e(\vec{r}, \vec{\omega}) + (dI)_s$$

$$(dI)_s = \frac{1}{4 \cdot \pi} \cdot \int_{4\pi} f_s(\vec{r}, \vec{\omega}, \vec{\omega}_i) \cdot K_s(\vec{r}, \vec{\omega}_i) \cdot I(\vec{r}, \vec{\omega}_i) d\vec{\omega}_i$$

Modelos de Radiación de Luz. Ec. Rendering

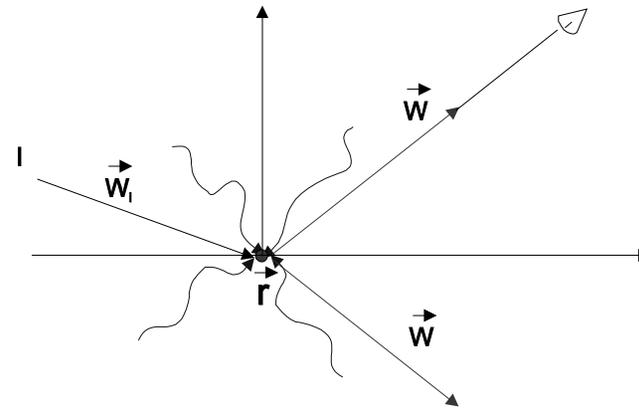
- **Interacción de la luz con la Superficies.**

- **Efectos de Reflexión**

- Difusa
- Perfecta

- **Efectos de Refracción**

- Difusa
- Perfecta



$$I(\vec{w}) = \frac{1}{4 \cdot \pi} \cdot \int_{4\pi} f_s(\vec{r}, \vec{w}, \vec{w}_i) \cdot \rho(\vec{r}, \vec{w}_i) \cdot I(\vec{r}, \vec{w}_i) d\vec{w}_i$$

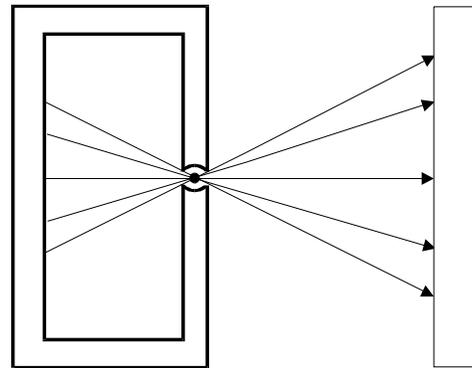
Modelos de Radiación de Luz. Ec. Rendering

- Integración general de todos los aspectos. Ecuación de rendering de kajiva
 - Se representa la luz que llega a un punto x del espacio procedente de otro punto x' .
 - Para resolver la ecuación se debería integrar para todos los puntos o elementos de interés de la escena

$$I(x \leftarrow x') = f_{att}(x, x') \left[I_e(x' \rightarrow x) + \int_s \rho(x', x'' \rightarrow x) I(x' \leftarrow x'') dx'' \right]$$

Trazado de Rayos

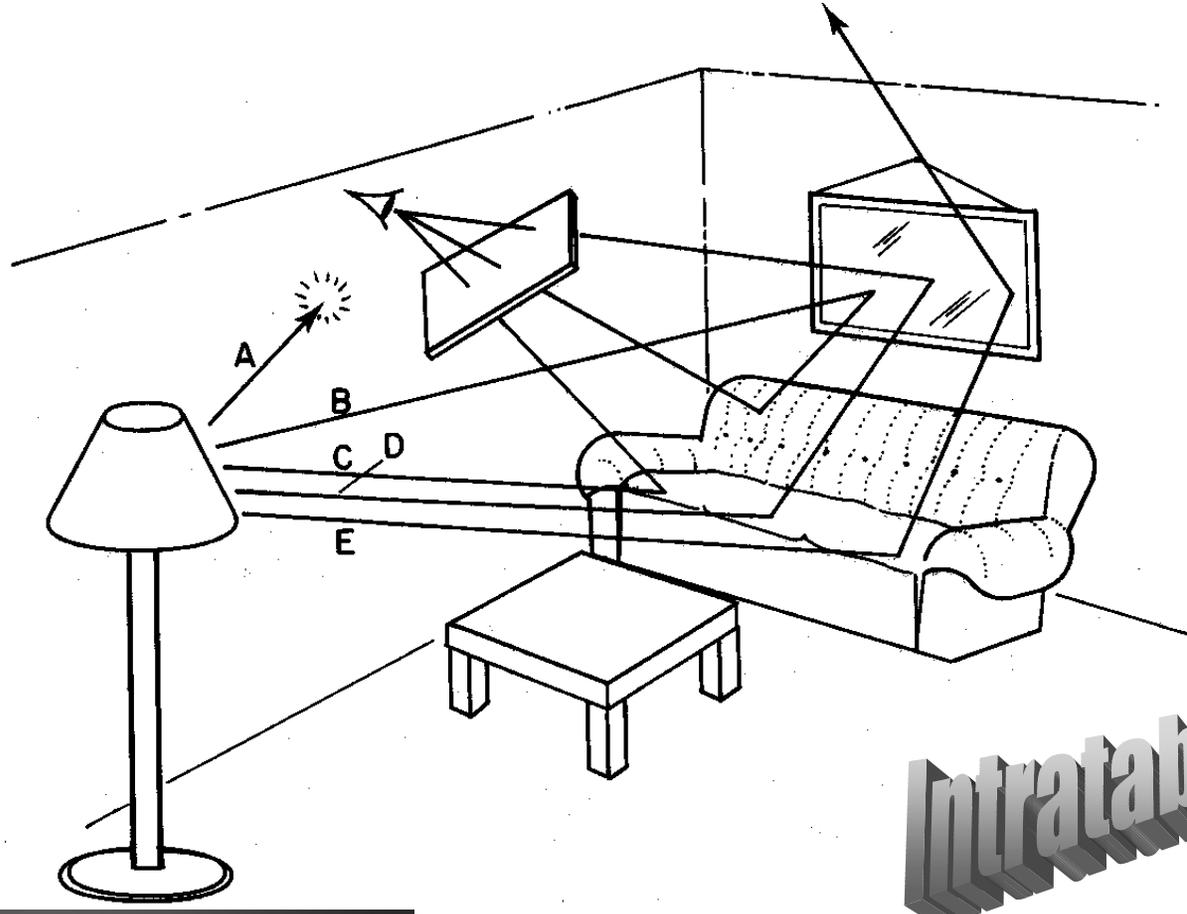
- Se trata de una resolución discreta de la ecuación de rendering de Kajiya.
- Se va a seguir el rastro de los rayos de luz, estos se interpretaran como partículas. Aproximación óptica.



Camara oscura

Trazado de Rayos

- **Aproximación Primera: Trazado de Rayos directo.**
 - Nos vamos a las fuentes de luz vemos los rayos que salen
 - Los seguimos por su viaje en la escena, recogiendo sus cambios de intensidad.
 - Nos quedamos con los que llegan a la cámara.

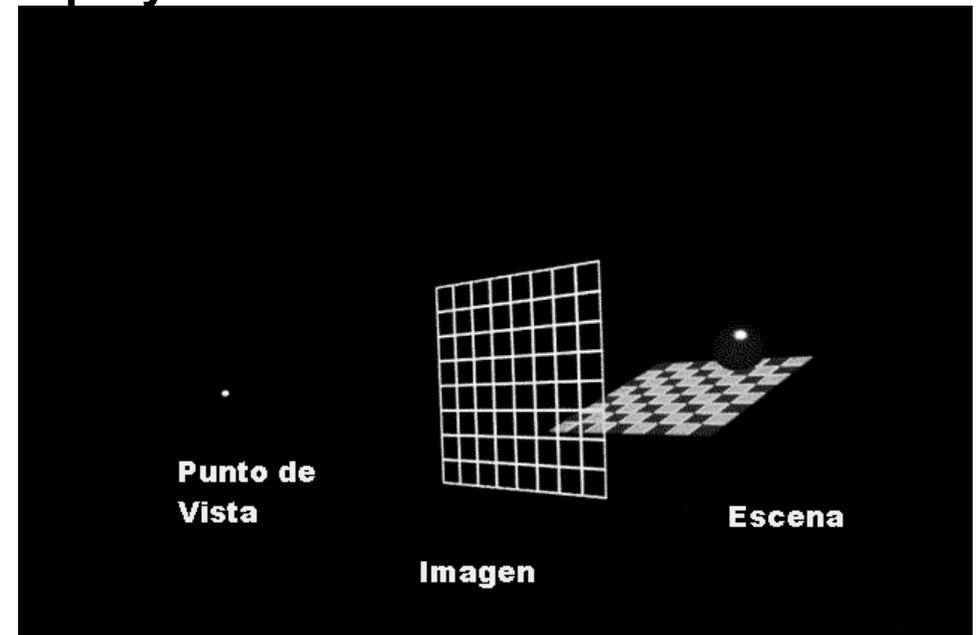


Intratable

Trazado de Rayos

- **Trazado de rayos inverso**

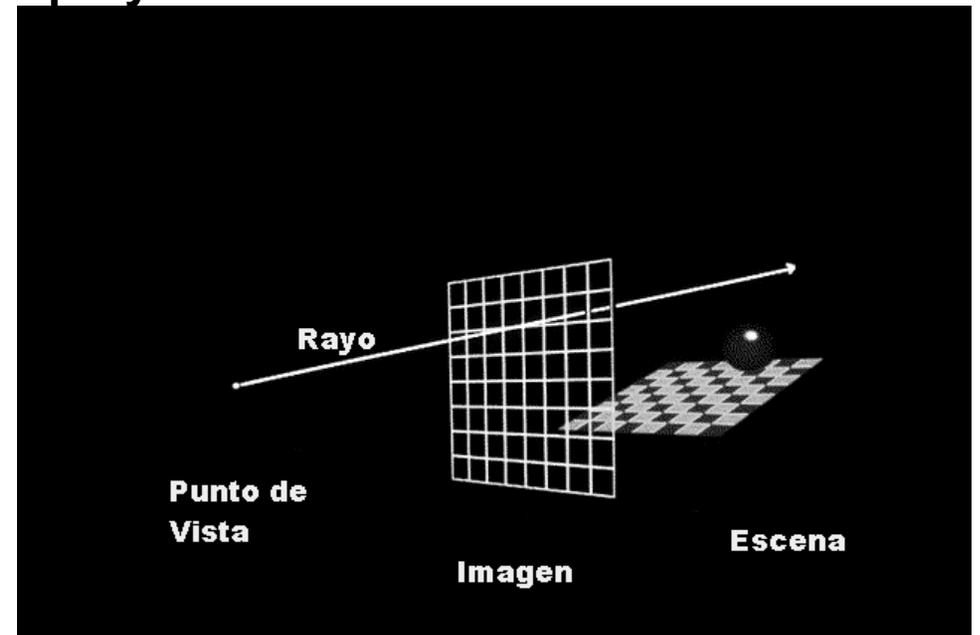
- Vamos a trazar el camino de los rayos de forma inversa.
- Los rayos salen de la cámara.
- Buscamos a través de los rebotes desde donde vienen.
- Múltiples choques.
- Además nos hace la tarea de proyectar



Trazado de Rayos

- **Trazado de rayos inverso**

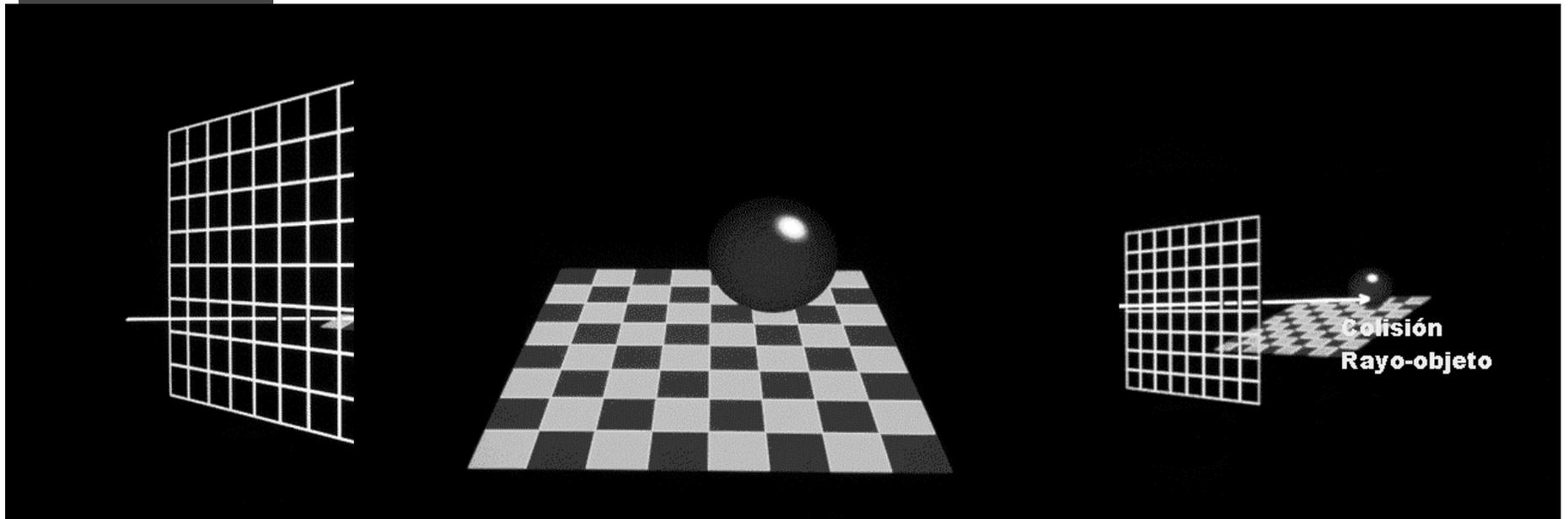
- Vamos a trazar el camino de los rayos de forma inversa.
- Los rayos salen de la cámara.
- Buscamos a través de los rebotes desde donde vienen.
- Múltiples choques.
- Además nos hace la tarea de proyectar



Trazado de Rayos

- **Trazado de rayos inverso**

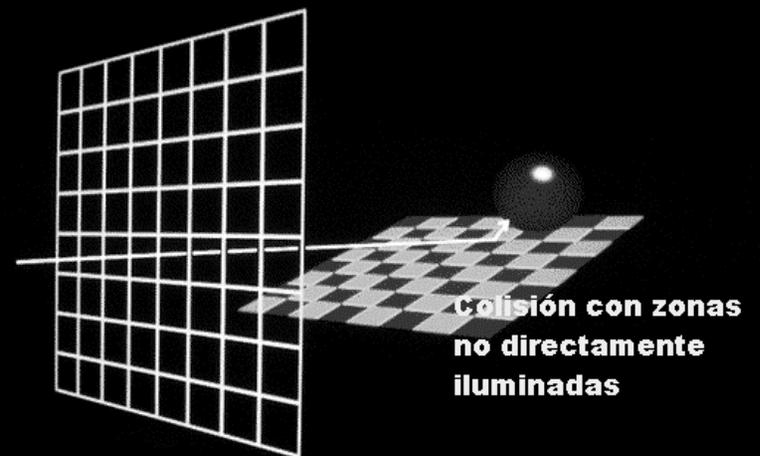
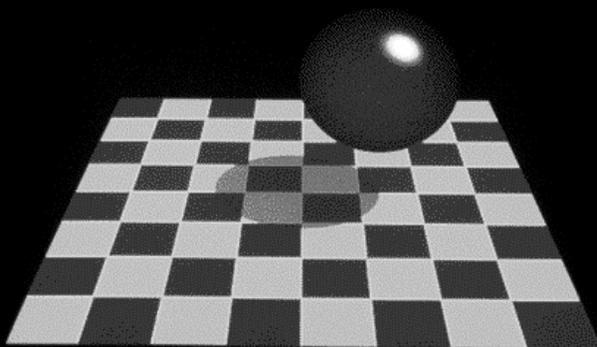
- Vamos a trazar el camino de los rayos de forma inversa.
- Los rayos salen de la cámara.
- Buscamos a través de los rebotes desde donde vienen.
- Múltiples choques.
- Además nos hace la tarea de proyectar



Trazado de Rayos

- **Trazado de rayos inverso**

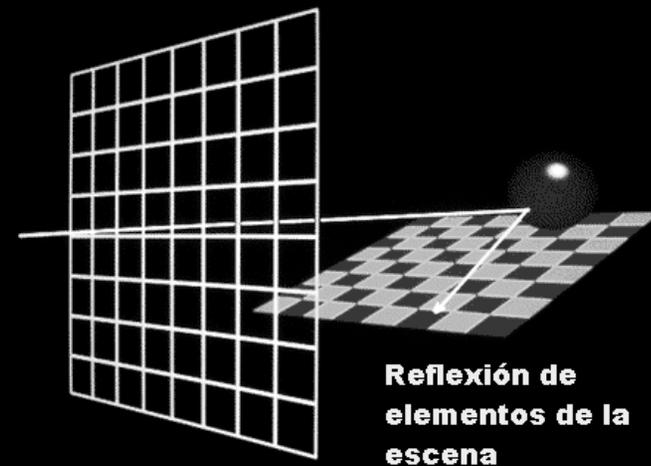
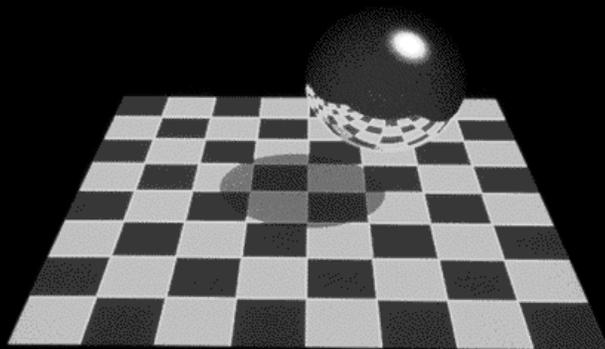
- Vamos a trazar el camino de los rayos de forma inversa.
- Los rayos salen de la cámara.
- Buscamos a través de los rebotes desde donde vienen.
- Múltiples choques.
- Además nos hace la tarea de proyectar



Trazado de Rayos

- **Trazado de rayos inverso**

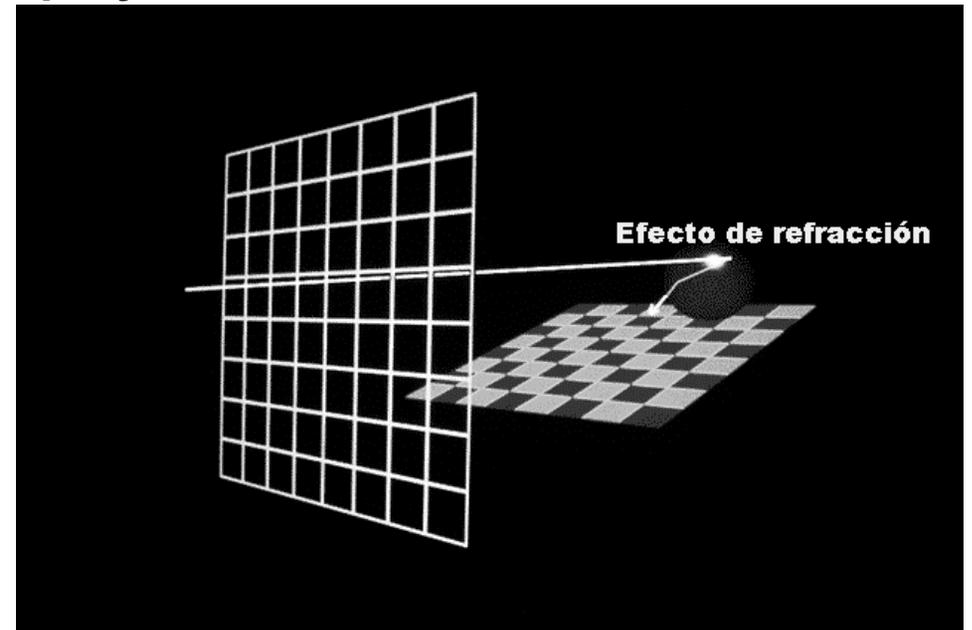
- Vamos a trazar el camino de los rayos de forma inversa.
- Los rayos salen de la cámara.
- Buscamos a través de los rebotes desde donde vienen.
- Múltiples choques.
- Además nos hace la tarea de proyectar



Trazado de Rayos

- **Trazado de rayos inverso**

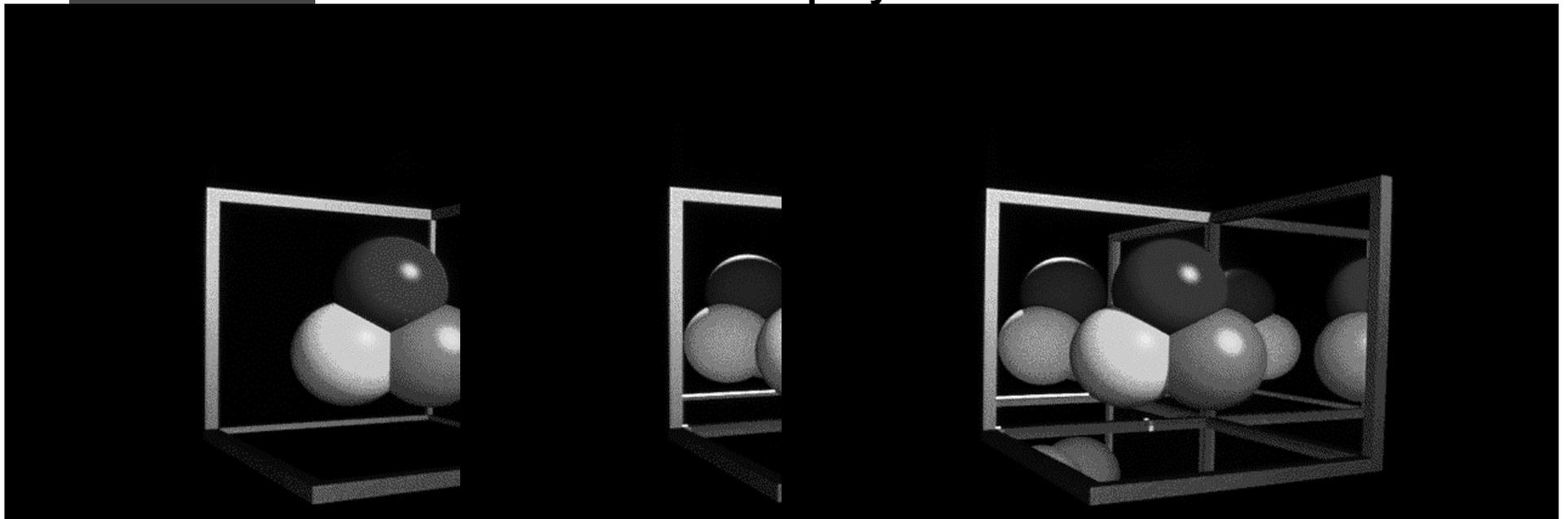
- Vamos a trazar el camino de los rayos de forma inversa.
- Los rayos salen de la cámara.
- Buscamos a través de los rebotes desde donde vienen.
- Múltiples choques.
- Además nos hace la tarea de proyectar



Trazado de Rayos

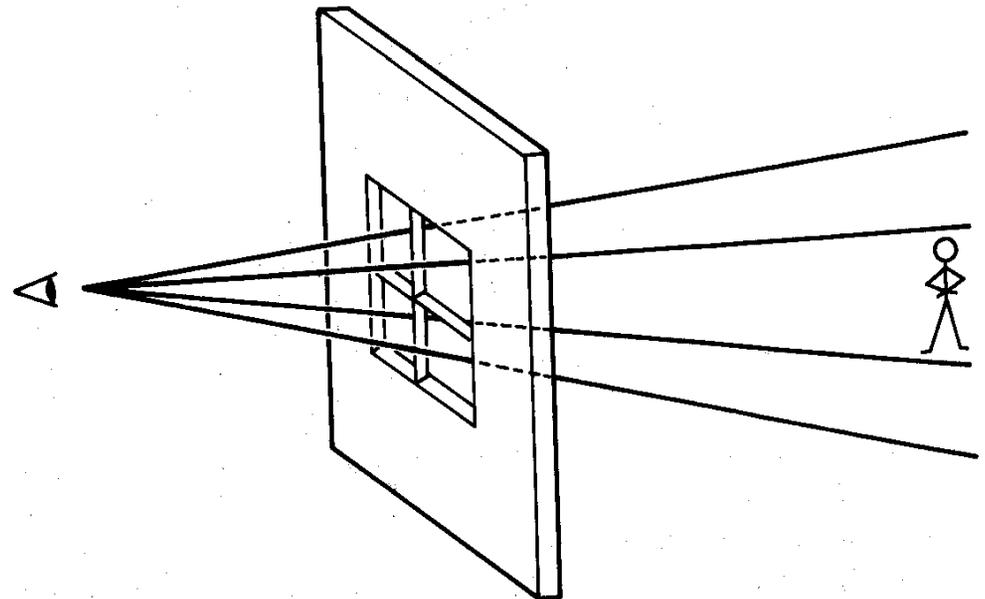
- **Trazado de rayos inverso**

- Vamos a trazar el camino de los rayos de forma inversa.
- Los rayos salen de la cámara.
- Buscamos a través de los rebotes desde donde vienen.
- Múltiples choques.
- Además nos hace la tarea de proyectar



Trazado de Rayos

- **Trazado de rayos inverso. Elementos principales**
 - Definir la dirección de los rayos iniciales. Rayos Primarios
 - Cuantas colisiones detectamos. Rayos secundarios.
 - Solución si no hay intersección: Rayos de sombra
- **El modelo tiene problemas de:**
 - Problemas de cálculo de intersecciones.
 - Problemas de aliasing.
 - Etc.



Trazado de Rayos

● Algoritmo de Trazado de Rayos Inverso

evaluar (rayo)

Determinar Intersección más cercana

```
/* interacción superficie */
```

```
->si no hemos alcanzado la profundidad máxima del árbol  
    crear rayos hijos  $I_r$  ,  $I_t$  y evaluarlos
```

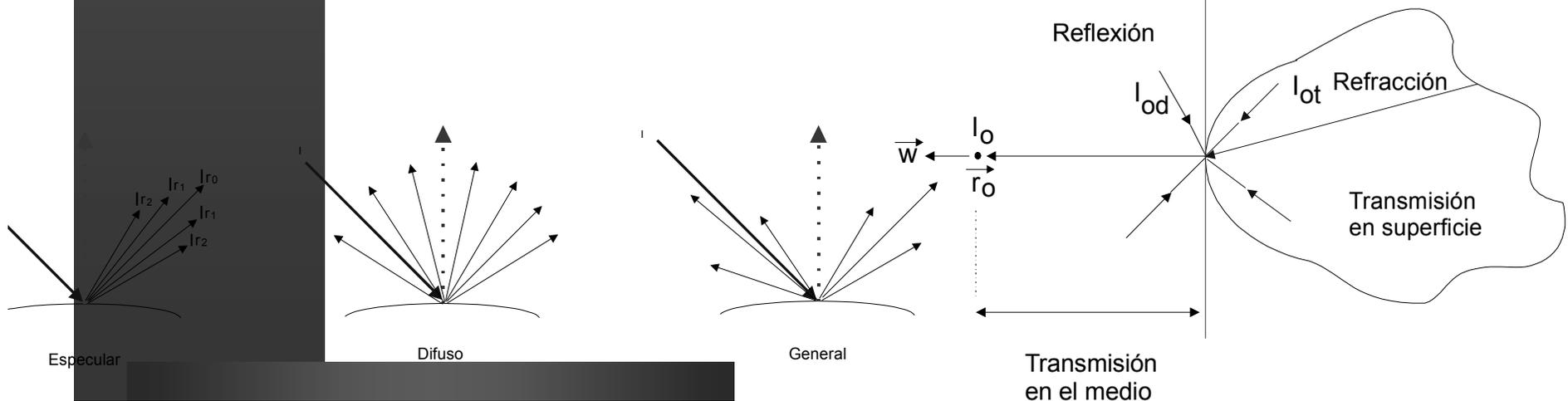
```
->calcular intensidad propia superficie
```

```
    lanzar rayo de sombra a cada fuente de luz
```

```
    si (rayo sombra no intersecciona objeto) aplicar modelo  
    iluminación  $\Rightarrow$  para calcular  $I_{propia}$ 
```

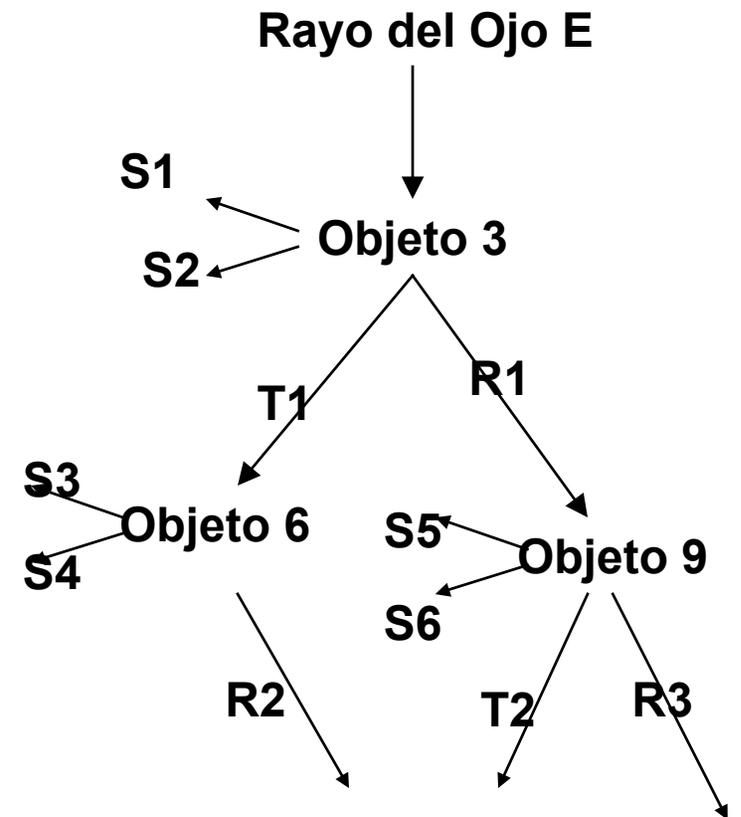
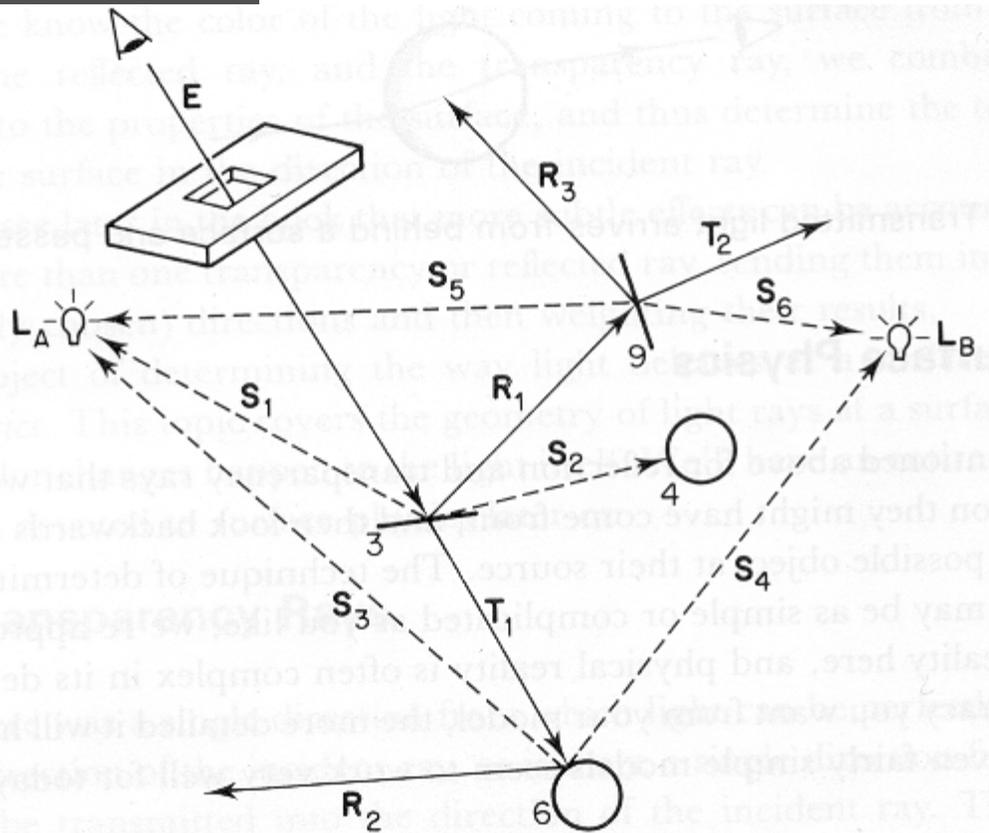
```
    modelo sombreado: componente ambiente + componente difusa  
    + componente especular
```

```
->devolver como color del rayo la intensidad propia  
    combinada con  $I_r$  ,  $I_t$ 
```



Trazado de Rayos

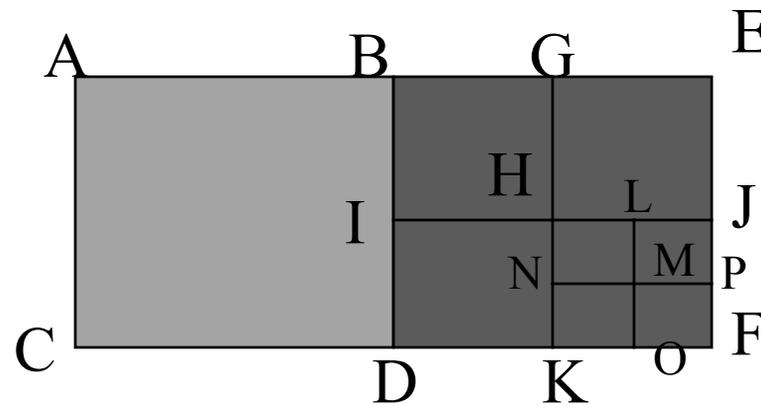
- **Árbol de Rayos** generado por el algoritmo TRI



raytracinjava\raytrace.html

Trazado de Rayos

- **Selección de los Rayos y poda del Árbol**
 - Un único rayo central
 - Sobre muestreo
 - Uniforme
 - Aleatorio
 - Adaptativo



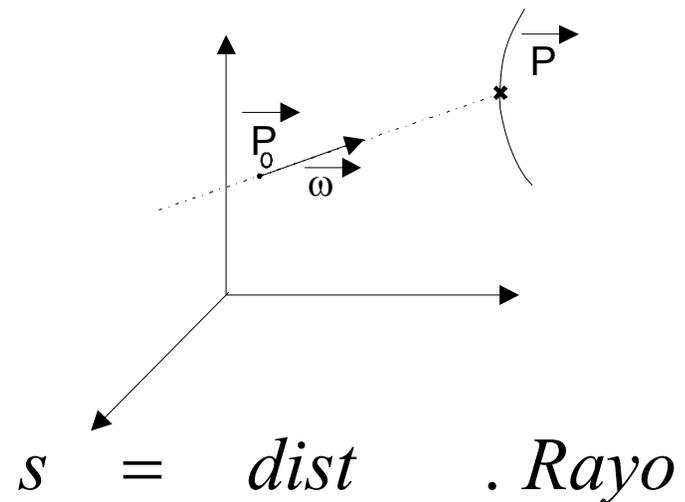
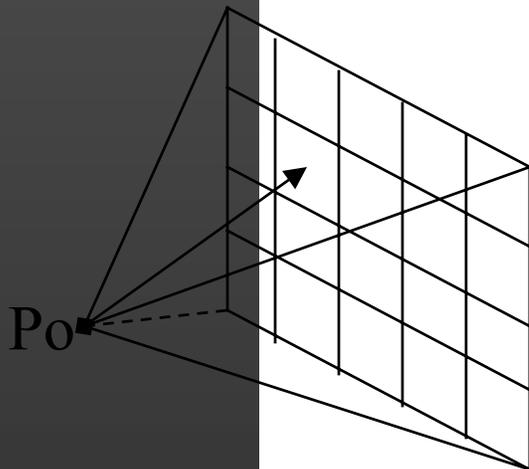
$$ColorP_1 = \frac{1}{4}(A + B + C + D)$$

$$ColorP_2 = \frac{1}{4} \left[\frac{1}{4}(B + I + G + H) + \frac{1}{4}(G + H + E + J) + \frac{1}{4}(D + I + H + K) + \frac{1}{4} \left(\frac{1}{4}(H + L + M + N) \dots \right) \right]$$

Trazado de Rayos

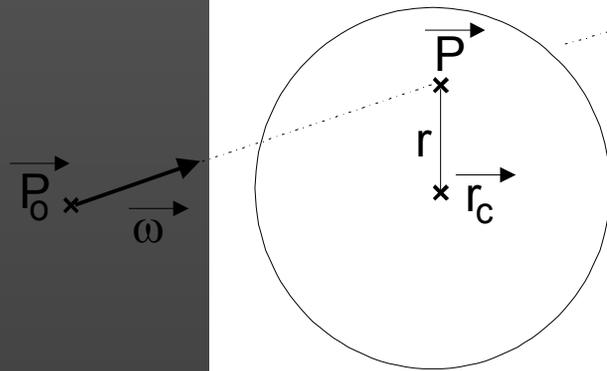
- **Problema Principal calculo de Intersecciones. Aprox. 95% del tiempo.**
- **Veamos como realizar los cálculos de intersección**

$$\vec{P} = \vec{P}_0 + s \cdot \vec{\omega} \quad |\omega| = 1$$



Trazado de Rayos

• Intersección Rayo-Esfera



$$\left. \begin{array}{l} \text{Recta} \rightarrow \vec{P} = \vec{P}_o + s\vec{\omega} \\ \text{Esfera} \rightarrow |\vec{P} - \vec{P}_c|^2 - r^2 = 0 \end{array} \right\} \text{Sustituyendo}$$

$$\left. \begin{array}{l} |\vec{P}_o + s\vec{\omega} - \vec{P}_c|^2 - r^2 = 0 \\ \Delta\vec{P} = \vec{P} - \vec{P}_o \end{array} \right\} |s\vec{\omega} - \Delta\vec{P}|^2 - r^2 = 0$$

$$\left. \begin{array}{l} s^2|\vec{\omega}|^2 - 2 \cdot s \cdot \vec{\omega} \cdot \Delta\vec{P} + |\Delta\vec{P}|^2 - r^2 = 0 \\ \text{como } |\vec{\omega}|^2 = 1 \end{array} \right\} s^2 - 2 \cdot s \cdot \vec{\omega} \cdot \Delta\vec{P} + |\Delta\vec{P}|^2 - r^2 = 0$$

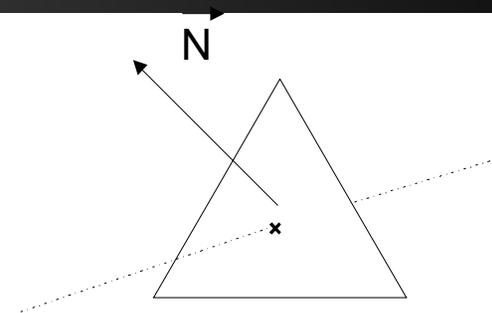
$$A = 1 \quad B = -2 \cdot \vec{\omega} \cdot \Delta\vec{P} \quad C = |\Delta\vec{P}|^2 - r^2 \quad s = \frac{-B \pm \sqrt{\text{discriminante}}}{2 \cdot A}$$

$$\text{discriminante} \left\{ \begin{array}{ll} < 0 & \Rightarrow \text{no existe solución} \\ = 0 & \Rightarrow \text{intersectan en un punto} \\ > 0 & \Rightarrow \text{intersectan en dos puntos} \end{array} \right.$$

Trazado de Rayos

- **Intersección Rayo-Polígono**

$$\left. \begin{array}{l}
 \text{Ec. del plano : } N_x \cdot x + N_y \cdot y + N_z \cdot z + D = 0 \Rightarrow \vec{N} \cdot \vec{P} = -D \\
 \text{Ec. del rayo : } \vec{P} = \vec{P}_0 + s\vec{\omega} \Rightarrow \vec{N} \cdot (\vec{P}_0 + s\vec{\omega}) = -D
 \end{array} \right\} s = \frac{(D + \vec{N} \cdot \vec{P}_0)}{\vec{N} \cdot \vec{\omega}}$$

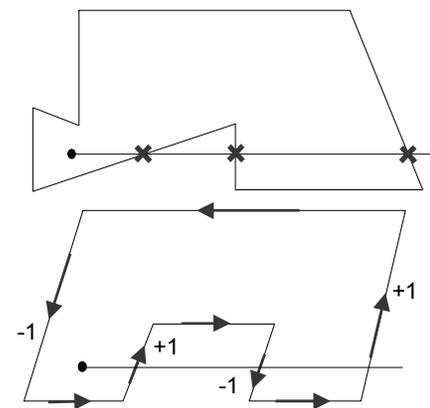
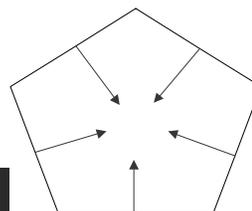


- **Determinar inclusión en polígono**

- Paridad de Intersecciones

- Intersecciones Orientadas

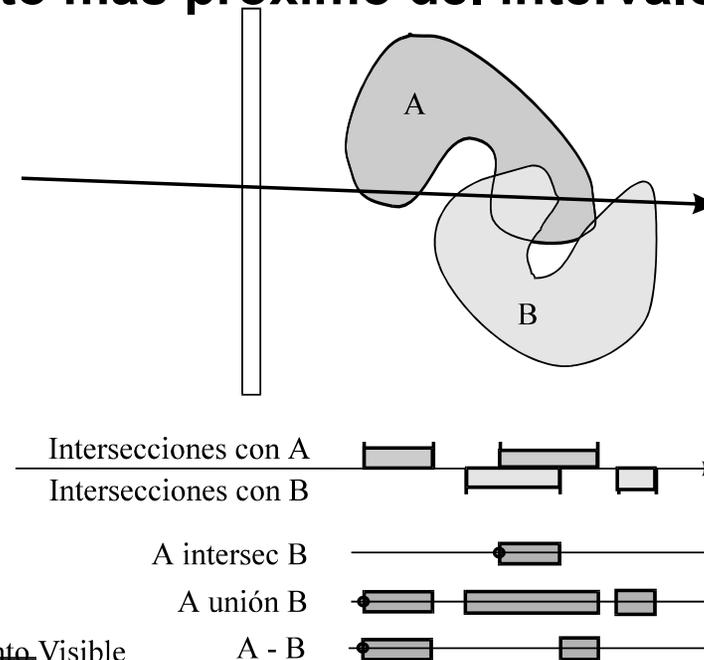
- Polígonos convexos



Trazado de Rayos

- **Intersección con Modelos Sólidos**

- Se calcula la intersección con las primitivas.
- Se combinan los intervalos siguiendo las operaciones booleanas del árbol CSG.
- Se determina el punto más próximo del intervalo resultado.



● Punto Visible

Trazado de Rayos

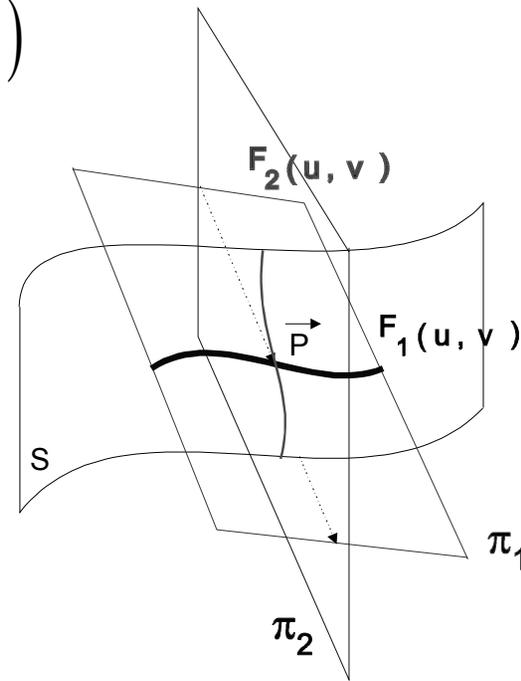
- **Intersección con Superf. Paramétricas**

- **No trivial utilización de métodos numéricos: Newton, etc..**

$$s: (x, y, z) = f(u, v)$$

$$\text{rayo: } (x, y, z) = (p_x, p_y, p_z) + s \cdot (\omega_x, \omega_y, \omega_z)$$

$$\left. \begin{array}{l} \vec{F}_1(u, v) = \pi_1 \cap s \\ \vec{F}_2(u, v) = \pi_2 \cap s \end{array} \right\} \vec{P} = \vec{F}_1 \cap \vec{F}_2$$



Trazado de Rayos

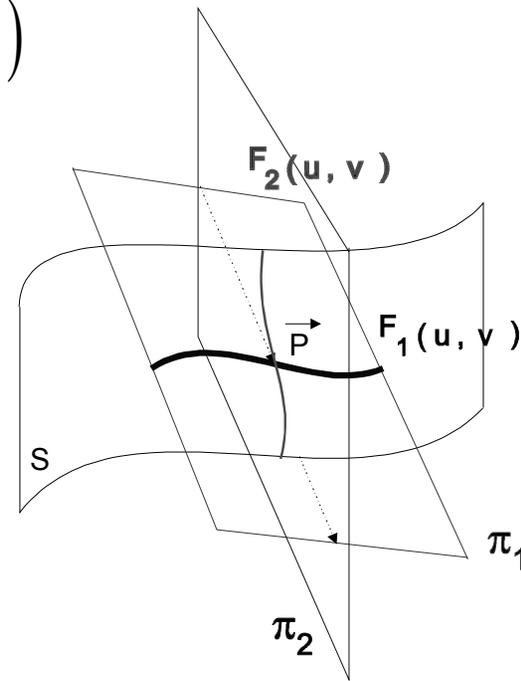
- **Intersección con Superf. Paramétricas**

- **No trivial utilización de métodos numéricos: Newton, etc..**

$$s: (x, y, z) = f(u, v)$$

$$\text{rayo: } (x, y, z) = (p_x, p_y, p_z) + s \cdot (\omega_x, \omega_y, \omega_z)$$

$$\left. \begin{array}{l} \vec{F}_1(u, v) = \pi_1 \cap s \\ \vec{F}_2(u, v) = \pi_2 \cap s \end{array} \right\} \vec{P} = \vec{F}_1 \cap \vec{F}_2$$



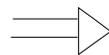
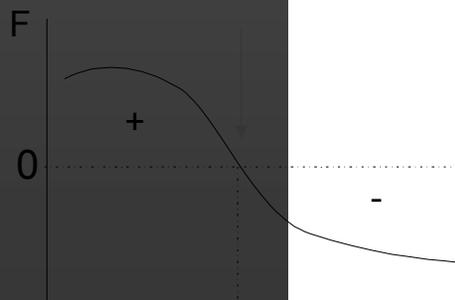
Trazado de Rayos

- **Intersección con Superf. Implícitas**

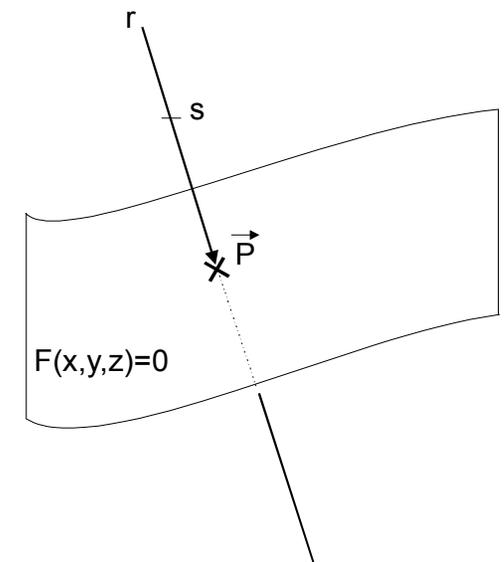
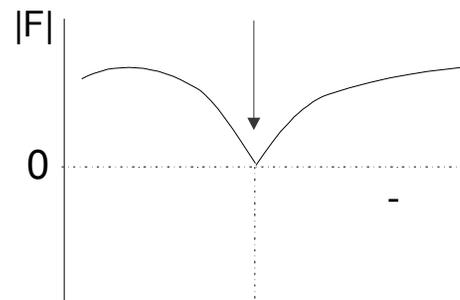
- **Métodos numéricos: Método del Gradiente problemas con mínimos locales.**

$$\left. \begin{array}{l} F(x, y, z) = 0 \\ x, y, z = \vec{P}_0 + s\vec{\omega} \end{array} \right\} F(\underbrace{\vec{P}_0 + s\vec{\omega}}_{\vec{f}(s)}) = 0 \quad \rightarrow \quad F(\vec{f}(s)) = 0$$

Solución



Solución



Trazado de Rayos

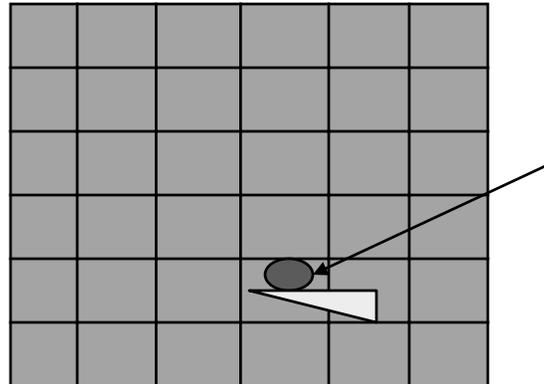
- **Optimización del Cálculo de Intersecciones**
 - **Problemas:**
 - A más rayos, más intersecciones.
 - Intersecciones Inútiles
- **Disminuir nº de Rayos:**
 - **Control Adaptativo.**
 - **Antialiasing Estadístico.**
 - **Poda del árbol de rayo por umbral de contribución.**
 - **Rayos generalizados, hacemos el trabajo para varios a la vez:**
 - Haces, conos, lápices.
- **Minimizar Intersecciones**
 - **Algoritmos de Visibilidad:**
 - Buffers de Items.
 - Buffers de Luz/sombra.
 - **Algoritmos de Organización Espacial.**

Trazado de Rayos

- **Optimización del Cálculo de Intersecciones**
 - **Problemas:**
 - A más rayos, más intersecciones.
 - Intersecciones Inútiles
- **Disminuir nº de Rayos:**
 - **Control Adaptativo.**
 - **Antialiasing Estadístico.**
 - **Poda del árbol de rayo por umbral de contribución.**
 - **Rayos generalizados, hacemos el trabajo para varios a la vez:**
 - Haces, conos, lapices.
- **Minimizar Intersecciones**
 - **Algoritmos de Visibilidad:**
 - Buffers de Items.
 - Buffers de Luz/sombra.
 - **Algoritmos de Organización Espacial.**

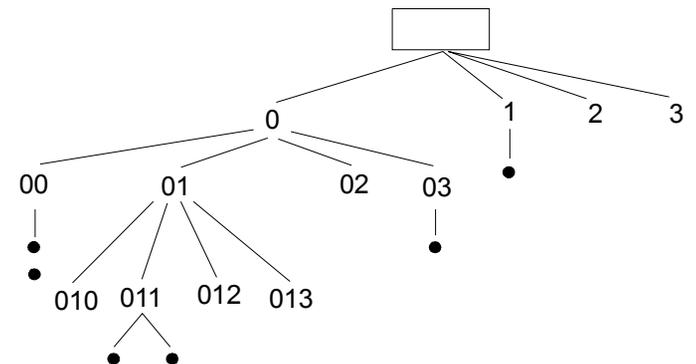
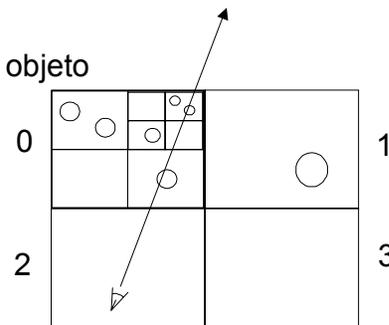
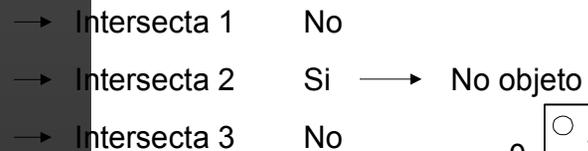
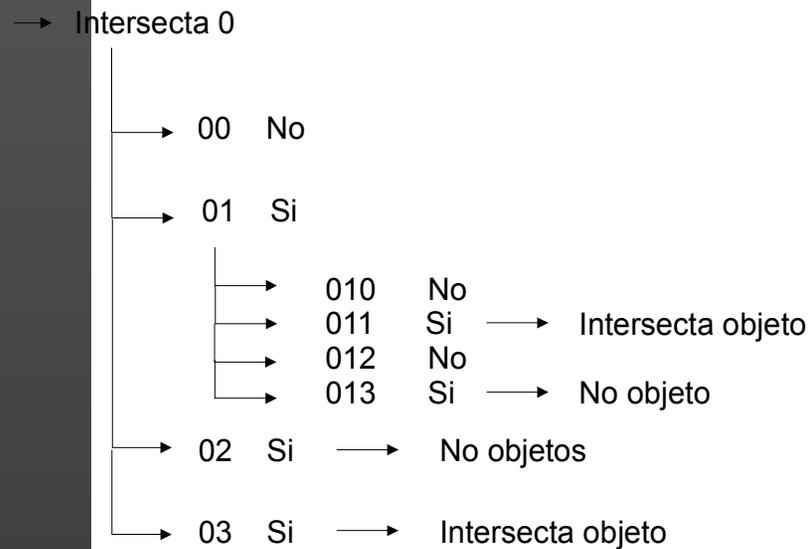
Trazado de Rayos

- **Organización Espacial: Uniforme basada en Voxels.**
 - Recorremos el Rayo utilizando el 3DDA
 - Los objetos están asociados a los Voxels, luego sabemos el orden en el que procesar la intersección.
 - Algunos problemillas de precisión.



Trazado de Rayos

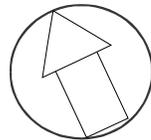
- **Organización Espacial: Jerarquías Octree.**



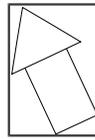
Trazado de Rayos

- **Organización Espacial: Volúmenes envolventes.**

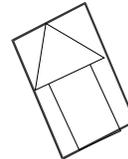
- **Colisionamos primero con elementos más sencillos que envuelven al objeto.**



Esférica



Caja orientada
con ejes



Caja no orientada



Envolvente poligonal
convexa

- **Si lo organizamos de forma jerárquica se puede mejorar el rendimiento.**

```
comprobar (nodo, rayo)
```

```
{
```

```
  si ( existe intersección (nodo, rayo) )
```

```
  {
```

```
    si nodo es terminal
```

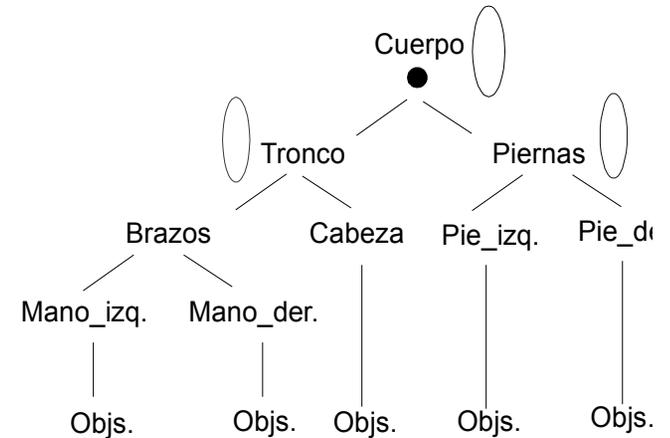
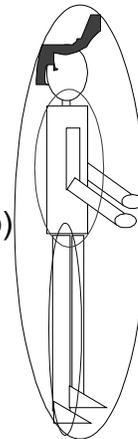
```
      comprobar_objetos(nodo, rayo)
```

```
    sino para cada hijo[i] de nodo
```

```
      comprobar (hijo[i], rayo)
```

```
  }
```

```
}
```



Radiosidad

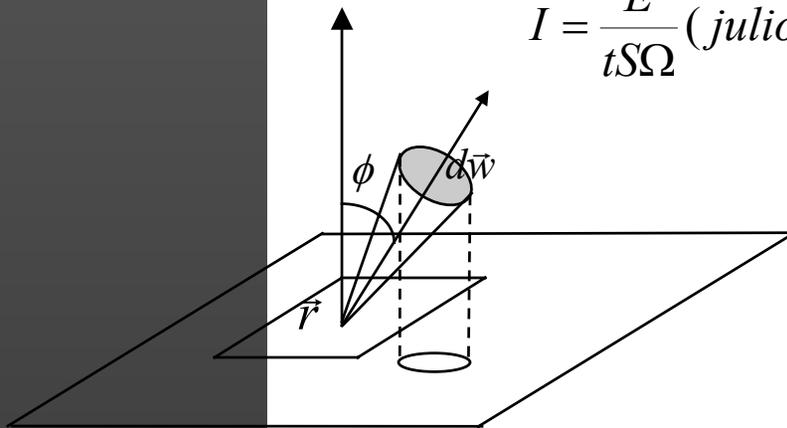
- **Tratamos de resolver la ecuación de rendering utilizando un aproximación distinta. Estudio termodinámico de la radiación de termal que se produce en la superficie de los objetos. La luz también es radiación termal. Aproximación termodinámica.**
- **Esta aproximación se basa por tanto en una magnitud escalar: energía. Por tanto será independiente del punto de vista.**
- **El concepto básico será determina la cantidad de energía que sale de una superficie, esto se deberá tanto a la energía propia que emita como a la que refleje procedente de otros puntos. Si consideramos esta energía por unidad de superficie tenemos lo que se llama Radiosidad**
- **Consideraciones que vamos a hacer para establecer un modelo computacional que nos permita determinar la Radisidad en la escena:**
 - **Todos los objetos son reflectores difusos perfectos.**
 - **La escena esta descompuesta en superficies o *patches*, y el problema será en contrar la radiosidad (B_i) asociado a cada uno de ellos.**
 - **La radiosidad corresponde con el equilibrio energético de la escena**
- **Consecuencias:**
 - **No existen reflexiones especulares.**
 - **Las fuentes se consideran extensas.**
 - **No se realiza un proyección de la escena, solo un balance energético.**

Radiosidad

- Conceptos Radiométricos y Radiosidad

$$B = \frac{E}{tS} \text{ (julios / m}^2\text{s) radiosidad}$$

$$I = \frac{E}{tS\Omega} \text{ (julios / m}^2\text{sSr) irradiancia}$$



$$I(\vec{r}, \vec{w}) = \frac{dB}{d\vec{w} \cos \phi}$$

$$B = \int_{hemi} dB$$

$$B = I \int_{hemi} \cos \phi d\vec{w}$$

$$d\vec{w} = \frac{dA}{r^2} \longrightarrow dA = \sin \phi d\phi d\theta$$

$$B = I \int_0^{2\pi} \int_0^{\pi/2} \frac{1}{r^2} \cos \phi \sin \phi d\phi d\theta = \pi I$$



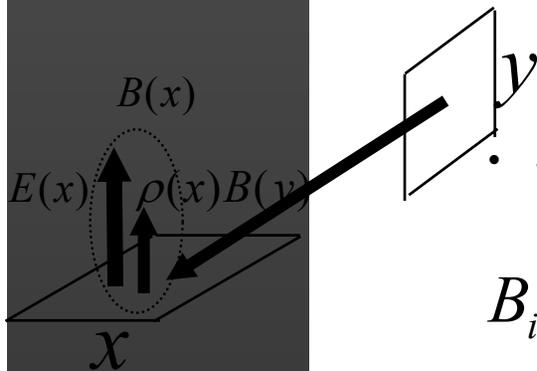
Radiosidad

- Nueva Versión de la Ecuación de Rendering: Ecuación continua de la radiosidad

$$B(x) = E(x) + \rho(x)H(x)$$

$$H(x) = \int_{y \in S} B(y) \frac{\cos \phi \cos \phi'}{\pi r^2} V(x, y) dy$$

$$B(x) = E(x) + \int_S \rho(x) F(x, y) B(y) dS$$



- Solución más adecuada para su computación, una versión discreta:

$$B_i = E_i + \sum_{j=0}^n \rho_i F_{ij} B_j$$

F_{ij} \longrightarrow Factor de Forma de i con respecto a j

Radiosidad

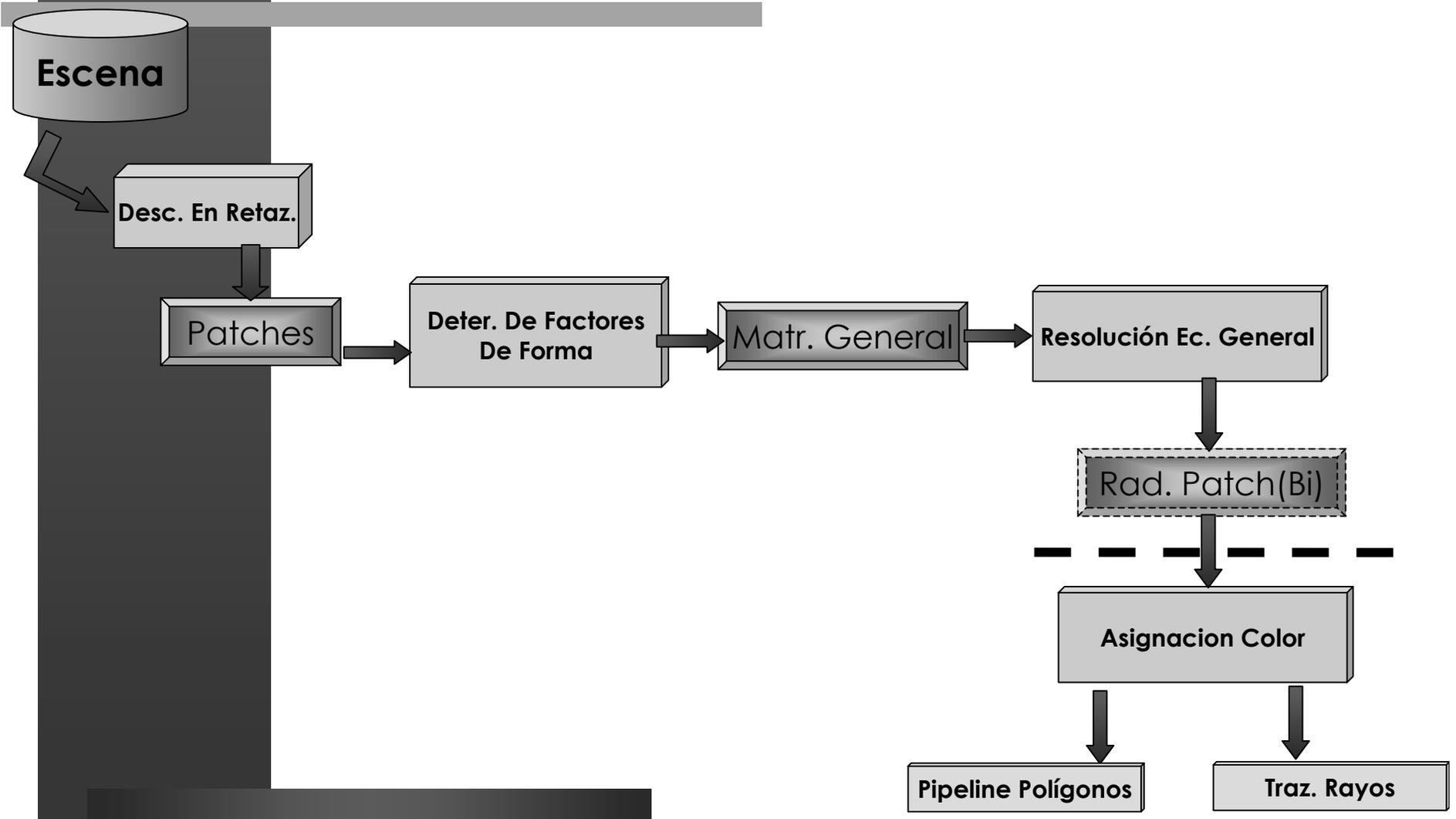
- Si trabajamos un poco más la expresión de la Ec. General de radiosidad podemos obtener la expresión matricial de la misma:

$$B_i = E_i + \rho_i F_{ii} B_i + \sum_{i \neq j}^n \rho_i F_{ij} B_j \longrightarrow E_i = B_i (1 - \rho_i F_{ii}) - \sum_{i \neq j}^n \rho_i F_{ij} B_j$$

$$\begin{bmatrix} E_0 \\ \vdots \\ \vdots \\ E_n \end{bmatrix} = \begin{bmatrix} B_0 \\ \vdots \\ \vdots \\ B_n \end{bmatrix} \begin{bmatrix} (1 - \rho_0 F_{00}) & -\rho_0 F_{01} & \cdots & -\rho_0 F_{0n} \\ -\rho_1 F_{10} & (1 - \rho_1 F_{11}) & \cdots & -\rho_1 F_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ -\rho_n F_{n0} & \cdots & \cdots & (1 - \rho_n F_{nn}) \end{bmatrix}$$

Radiosidad

- Revisemos antes de continuar el proceso general de estimación de la radiosidad en gráficos:

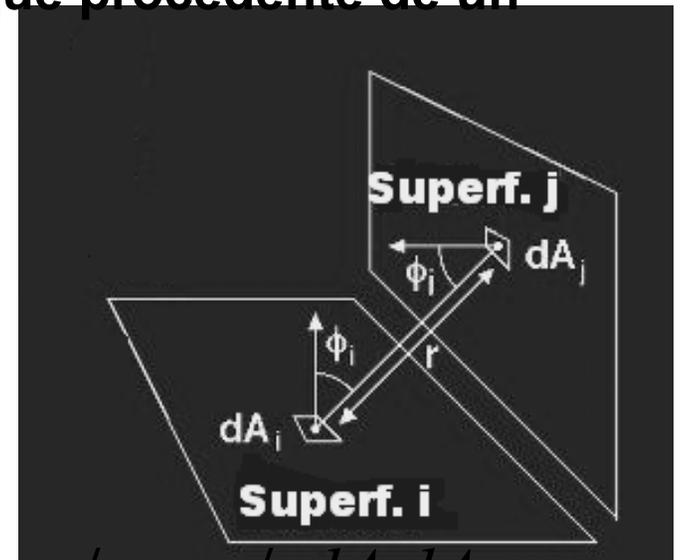


Radiosidad

- Factores de Forma y su cálculo.

- Son un valor que viene a representar la cantidad de superficie útil que un trozo enfrenta a otro. Podemos estimarlo en función de la energía que procedente de un trozo llega a otro.

$$dB_i dA_i = I_i \cos \phi_i dw dA_i$$



$$dw = \frac{dA}{r^2} = \frac{\cos \phi_j dA_j}{r^2} \longrightarrow dB_i dA_i = \frac{I_i \cos \phi_i \cos \phi_j dA_i dA_j}{r^2}$$

$$F_{ij} = \frac{\text{Energía que llega } (A_j)}{\text{Energía que sale de } (A_i)}$$

Radiosidad

- Factores de Forma y su cálculo.

$$F_{dA_i dA_j} = \frac{I_i \cos \phi_i \cos \phi_j dA_i dA_j}{r^2} \cdot \frac{1}{B_i dA_i}$$

$$B_i = \pi I_i$$

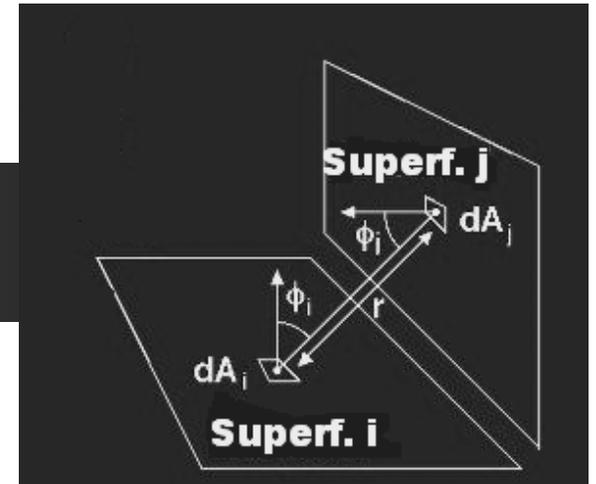
$$F_{dA_i dA_j} = \frac{\cos \phi_i \cos \phi_j dA_j}{\pi r^2} \longrightarrow F_{dA_i, A_j} = \int_{\text{sup}_j} \frac{\cos \phi_i \cos \phi_j dA_j}{\pi r^2}$$

$$F_{ij} = \frac{1}{A_i} \int_{\text{sup}_i} \int_{\text{sup}_j} \frac{\cos \phi_i \cos \phi_j}{\pi r^2} dA_j dA_i,$$

$$F_{ij} = \frac{1}{A_i} \int_{\text{sup}_i} \int_{\text{sup}_j} V_{dA_j dA_i} \frac{\cos \phi_i \cos \phi_j}{\pi r^2} dA_j dA_i$$

$$\sum_{k=1}^n F_{ij} = 1$$

$$A_i F_{ij} = A_j F_{ji}, F_{ii} = 0 (\text{planos, convexos})$$



Radiosidad

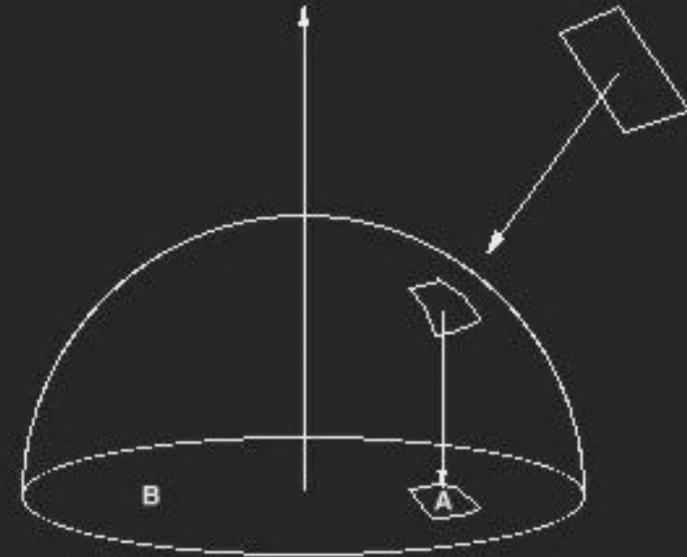
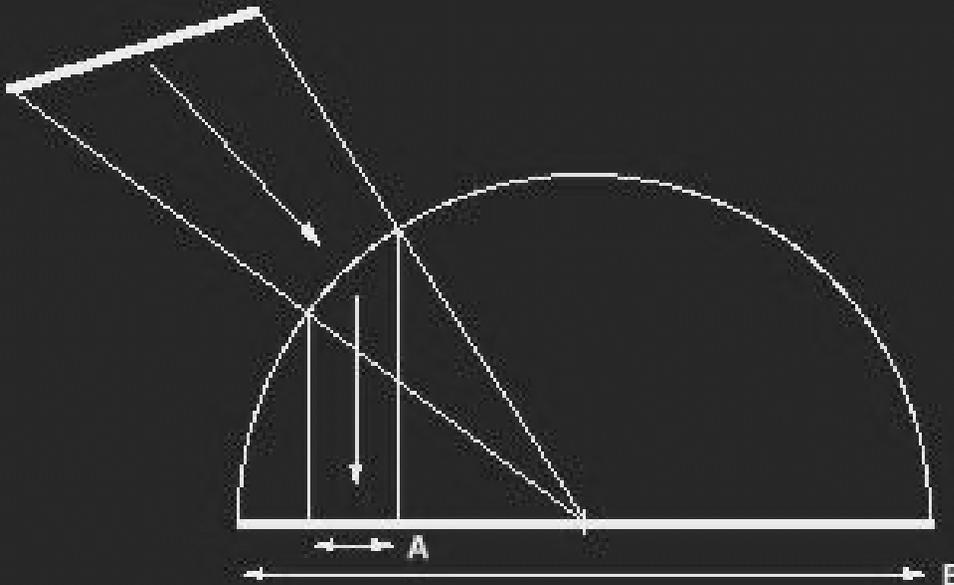
- **Métodos de Estimación de los Factores de Forma.**
 - **Métodos Analíticos**
 - **Casos especiales**
 - **Dif. Área frente a polígono.**
 - **Polígono a Polígono.**
 - **Numéricos:**
 - **Áreas frente a áreas:**
 - **Integrales de Contorno. Teorema de Stokes.**
 - **Descomposición Jerárquica.**
 - **Montecarlo**
 - **Dif. Área Frente a Áreas:**
 - **Muestreo de Áreas.**
 - **Muestreo de Semiesferas:**
 - **Montercarlo.**
 - **Hemicubo.**

Radiosidad

- Muestreo de Semiesferas para el cálculo de Factor de Forma.

– Analogía de Nusselt.

Es una aproximación geométrica. El Factor de forma proporcional al área proyectada partido por el áreas de la base.



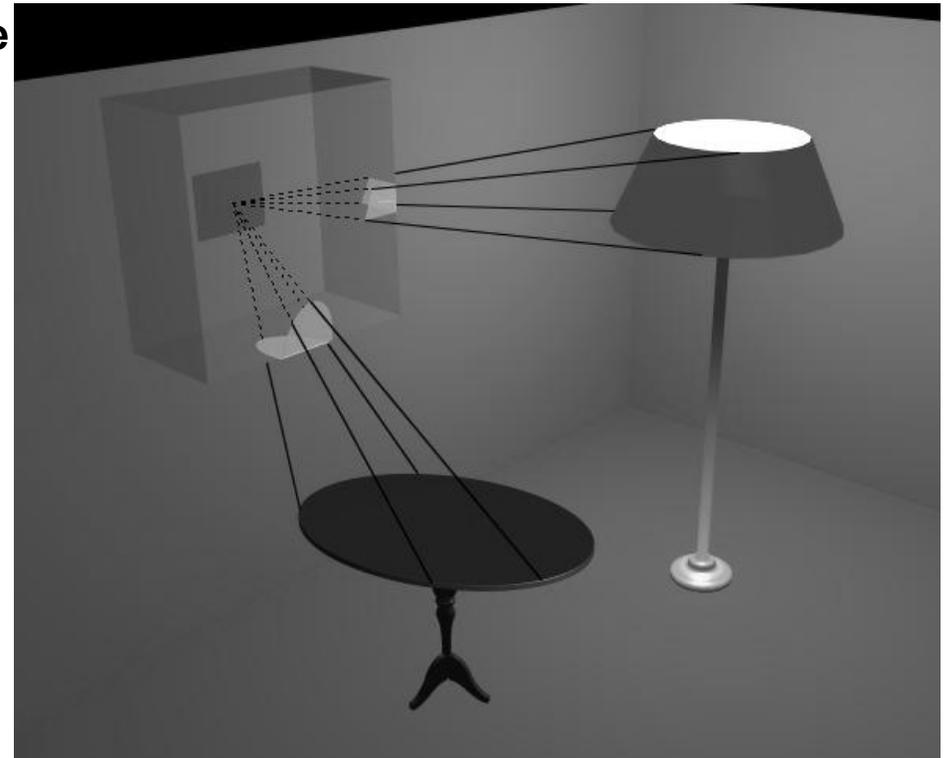
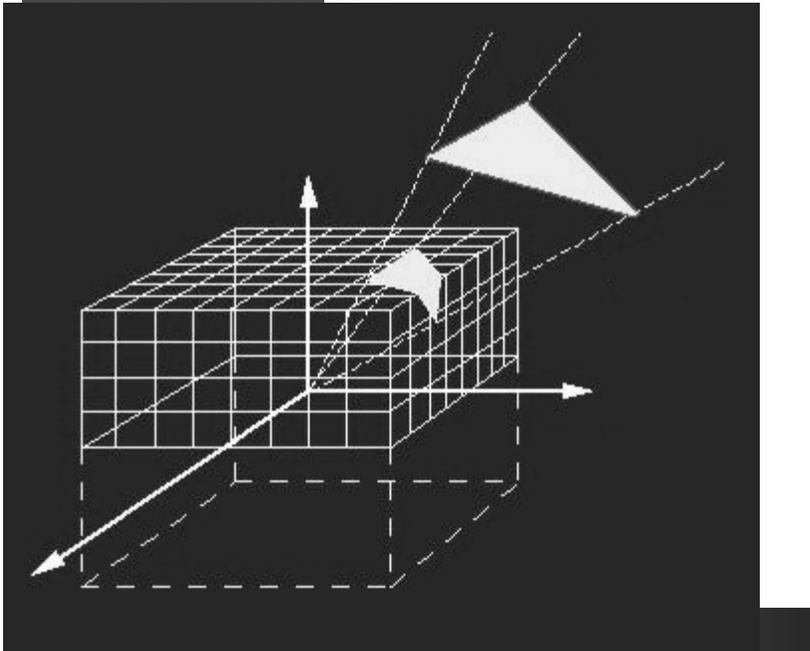
- Esto tiene sentido ya que si nos fijamos cada proyección es un coseno y el área de la base es πr^2

$$\frac{\cos \phi_i \cos \phi_j}{\pi r^2}$$

- Por tanto elementos que tienen la misma área proyectada tienen el mismo factor de forma

Radiosidad

- **Muestreo de Semiesferas para el cálculo de Factor de Forma.**
 - **Método del Hemicubo:** versión modificada de la analogía de Nusselt: Si proyectamos en una superficie intermedia el factor de forma sigue siendo el mismo.
 - Por tanto si discretizamos una serie de trozos y somos capaces de conocer el valor del factor de forma asociado a cada trozo podremos saber el factor de forma sumando las contribución de cada trozo proyectado.
 - Para realizar el algoritmo del hemicubo considerará la cara superior y cuatro medias caras laterales.
 - Lo bueno de esto es que esto se parece a hacer fotos!!!!



Radiosidad

- Determinación de las ΔF asociadas a cada elemento discreto.

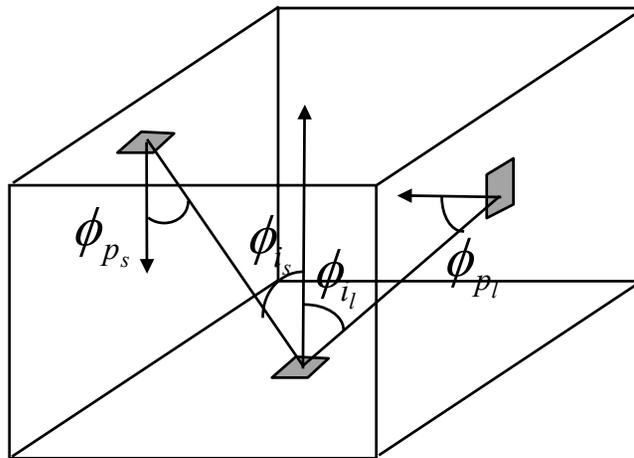
Cara de Arriba

$$\Delta F_p = \frac{\cos \phi_{i_s} \cos \phi_{p_s}}{\pi r^2} \Delta A$$

$$r = \sqrt{x_{p_s}^2 + y_{p_s}^2 + 1}$$

$$\cos \phi_{i_s} = \cos \phi_{p_s} = \frac{1}{r}$$

$$\Delta F_p = \frac{1}{\pi(x_{p_s}^2 + y_{p_s}^2 + 1)^2} \Delta A$$



Cara Lateral

$$\Delta F_p = \frac{\cos \phi_{i_l} \cos \phi_{p_l}}{\pi r^2} \Delta A$$

$$r = \sqrt{x_{p_l}^2 + y_{p_l}^2 + 1}$$

$$\cos \phi_{i_l} = \frac{z_{p_l}}{r}, \cos \phi_{p_l} = \frac{1}{r}$$

$$\Delta F_p = \frac{z_{p_l}}{\pi(y_{p_l}^2 + z_{p_l}^2 + 1)^2} \Delta A$$

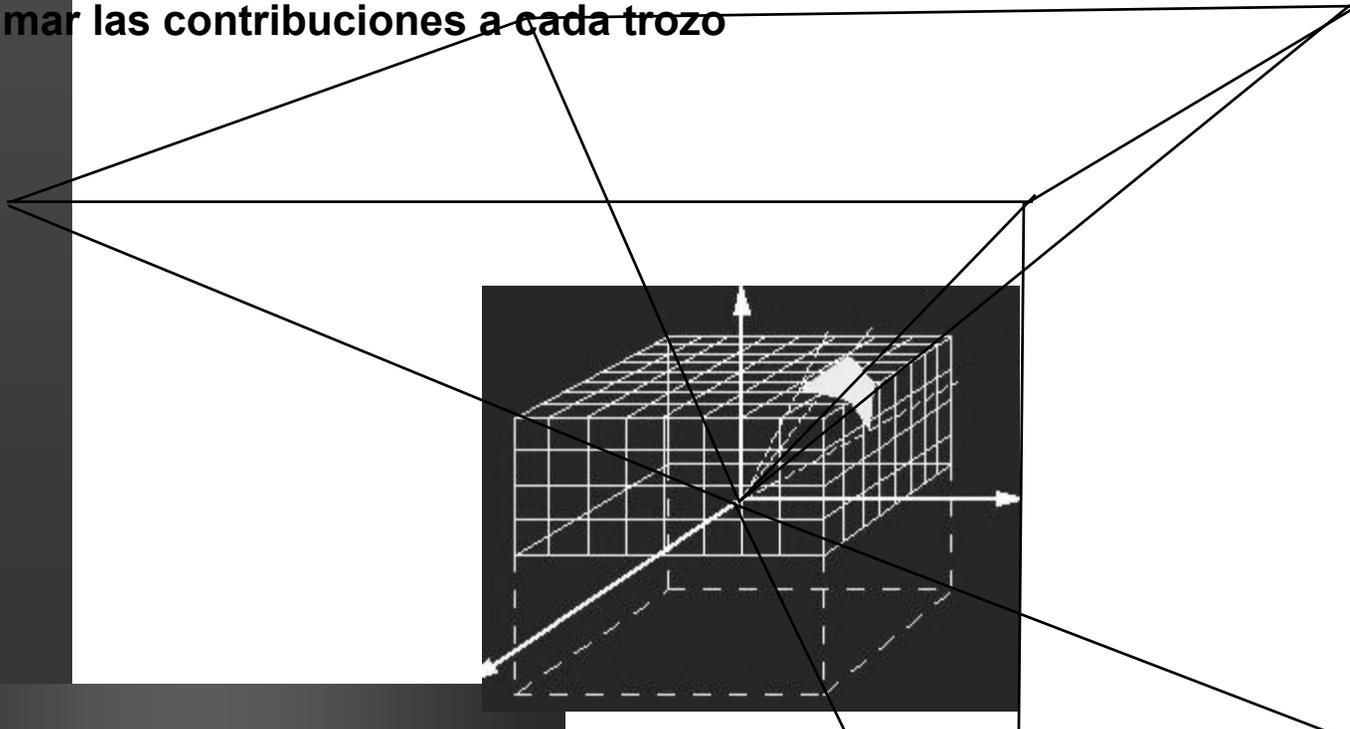
Radiosidad

- **Algoritmo del Hemicubo**

- Iniciar los ΔA para los “pixels” del hemicubo

- Para todos los i

- iniciar los F_{ij} a cero.
- Iniciar todos los item buffer a cero.
- Colocar el hemicubo en el centro del trozo.
- Calcular la imagen sobre cada cara (90°) y sumar las contribuciones a cada item.
- Sumar las contribuciones a cada trozo



Radiosidad

- **Resolución de la Ecuación Discreta de la Radiosidad: algoritmos progresivo.**
 - **Problema de la repr. Matricial NxN y resolución de ecuaciones $O(n^3)$**
 - **Buscaremos la solución empleando métodos iterativos que vayan buscando la solución progresivamente.**
 - **Idea básica, partimos de una suposición para las B miramos cuanto nos equivocamos y en la siguiente iteración corregimos. (Jacobi, Gauss-Steindell, Southell). Esto implica por ejemplo que en la primera iteración la única energía presente es la de las fuentes. En la segunda tendremos fuentes más lo que no hayamos recibido de otras fuentes, después de esto hay que reajustar todo ya que nosotros hemos contribuido menos de lo debido a la escena. Esto genera en cada paso un vector de residuos, que iremos minimizando de una interacción a otra. El proceso finalizará cuando los residuos sean suficientemente pequeños.**
 - **Arreglamos el problema computacional pero seguimos necesitando un montón de almacenamiento (todos los factores de forma).**

Radiosidad

- Resolución de la Ecuación Discreta de la Radiosidad: algoritmos progresivo.

- El algoritmos progresivo lo vamos a refinar un poco más considerando que solamente emite un trozo cada vez, el que más energía le queda por emitir en cada paso y veremos como contribuye a todos los demás, de esta forma cada vez sólo necesitamos los factores de forma de este con respecto a los demás...

Inicio

$B_i = E_i$ ← la energía que tiene cada uno es la que emite por sí mismo
 $\Delta_i = E_i$ ← y es la misma que le queda por enviar

Repetir

elegir $\Delta_1 = \text{maximo}\{\Delta_i\}$ ← para que converja más rápido

enviar la energía de este trozo a todos los demás trozos mediante :

$$B_j \leftarrow B_j + \Delta_1 \cdot \rho_i \cdot F_{lj}$$

$$\Delta_j \leftarrow \Delta_j + \Delta_1 \cdot \rho_i \cdot F_{lj}$$

hacer : **error** ← Δ_1

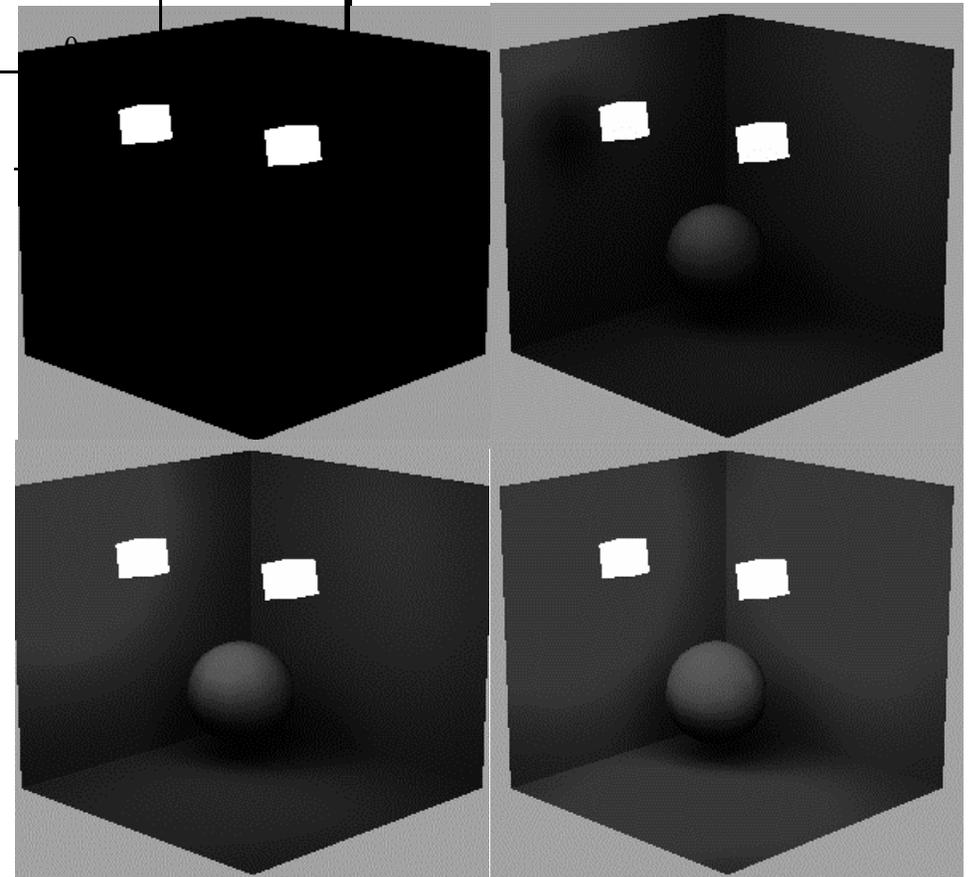
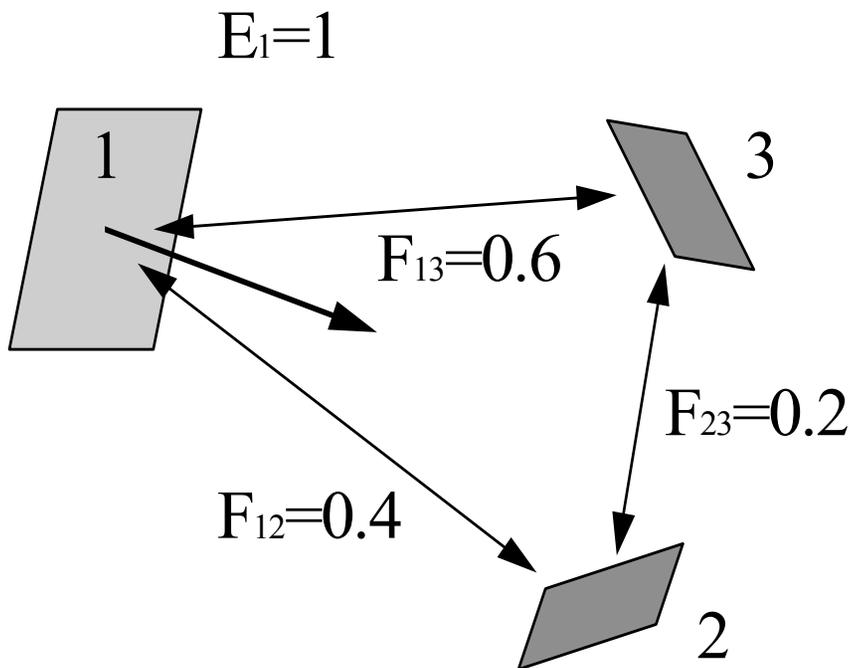
poner a cero Δ_1 : $\Delta_1 \leftarrow 0$

hasta (**error** < umbral)

Radiosidad

- Resolución de la Ecuación Discreta de la Radiosidad: algoritmos progresivo. Ejemplo

Quién envía?	B ₁	B ₂	B ₃	Δ ₁	Δ ₂	Δ ₃
1	1	.2	.3	.0	.2	.3
3	1.09	.23	.3	.09	.23	.0
2	1.136	.23	.323	.136	.0	.023
1	1.136	.2572	.3638	.0	.0272	.0638
3	1.15514	.26358	.3638	.01914	.03358	.0
2	1.16186	.26358	.36716	.025856	.0	.00336
...						
∞	1.16745	.27123	.37736	.0		

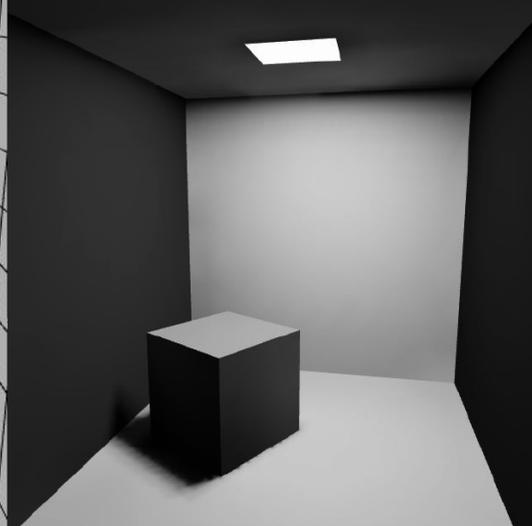
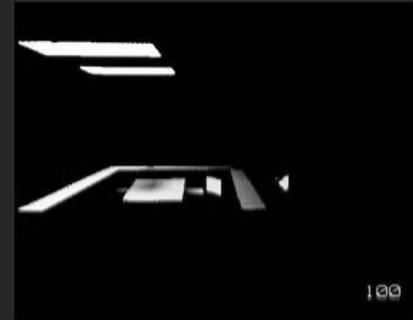
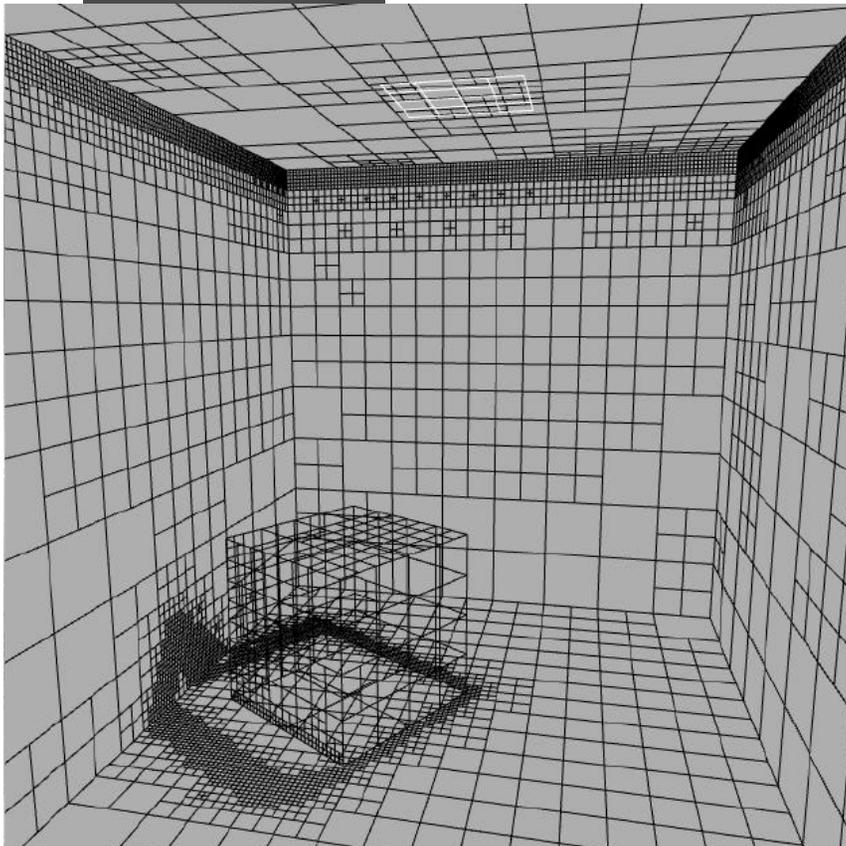


Radiosidad

- **Mejoras adicionales al algoritmo de radiosidad**

- Mejora de tiempos de convergencia considerando energía ambiente inicial.

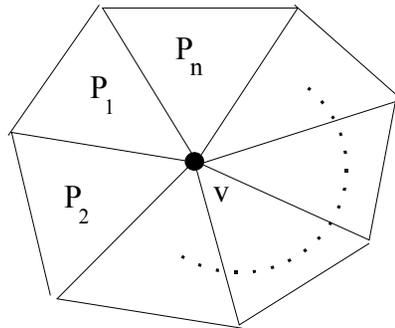
- Mejora del aspecto final:
subestructuración de trozos cuando sea necesario por excesivos gradientes de energía entre trozos.



Radiosidad

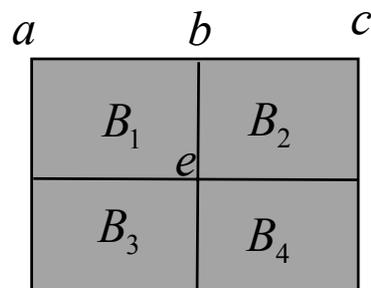
- **Asignación de Color por vértice:**

- Se trata de uno de los posibles usos de los valores obtenidos con la radiosidad, consiste en realizar un proceso de adaptación de los valores a colores por vértice.
- **Solución1:Trivial, el color de cada vértice el del trozo al que pertenece. No hay buenos resultados.**
- **Solución más elaborada: estimar para cada vértice la contribución de los trozos a los que pertenece:**



$$I_v = f(I_{p1}, I_{p2}, \dots, I_{pn})$$

$$I_v = \sum_i^n \int \frac{I_i}{d^2} dS \longrightarrow I_v = \frac{\sum_i^n I_i}{n}$$



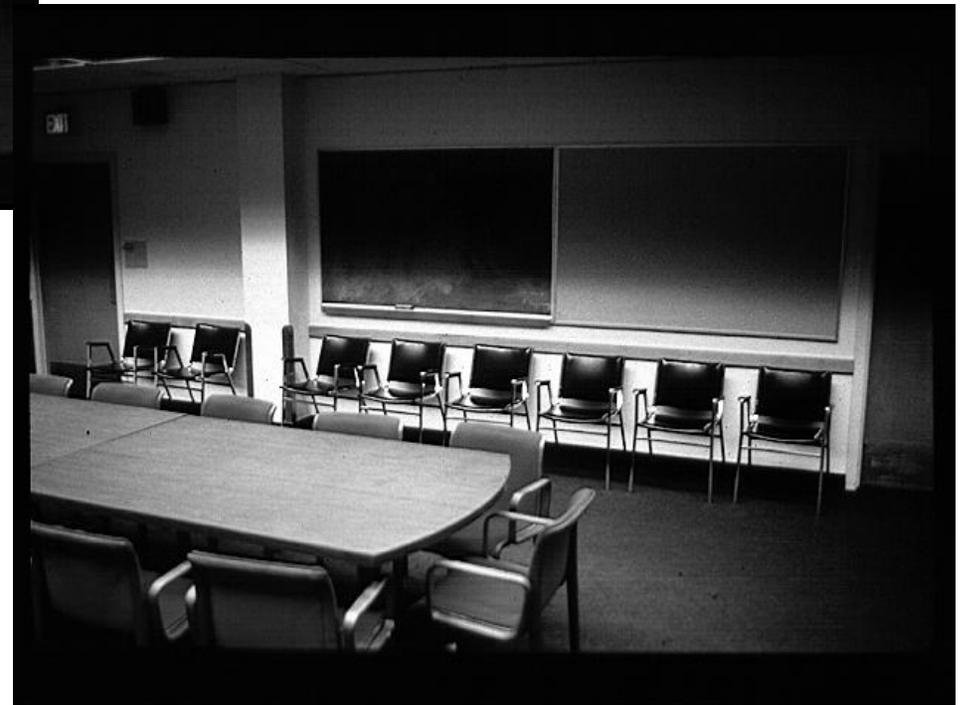
$$B_e = \frac{B_1 + B_2 + B_3 + B_4}{4}$$

$$\frac{B_b + B_e}{2} = \frac{B_1 + B_2}{2} \longrightarrow B_b = B_1 + B_2 - B_e$$

$$B_1 = \frac{B_a + B_e}{2} \longrightarrow B_a = 2B_1 - B_e$$

Radiosidad

- Ejemplos de radiosidad:



Radiosidad

- Ejemplos de radiosidad:

