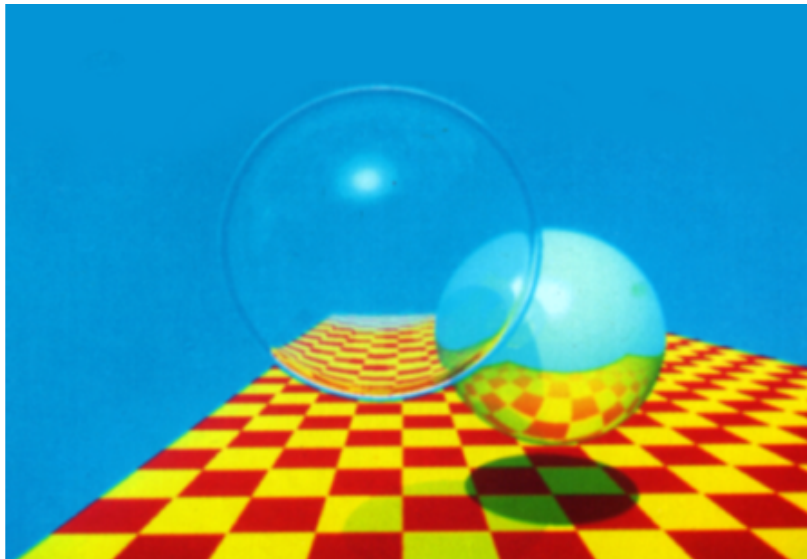


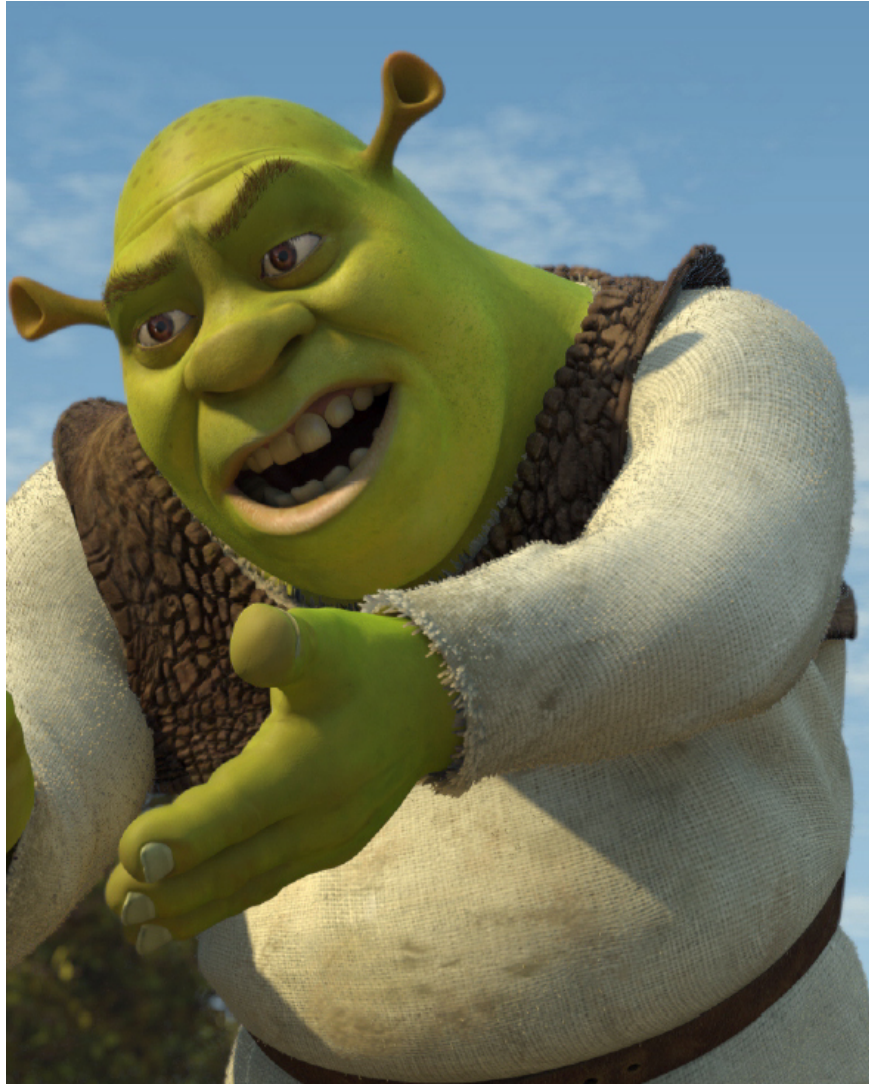
Trazado de rayos

TAIG

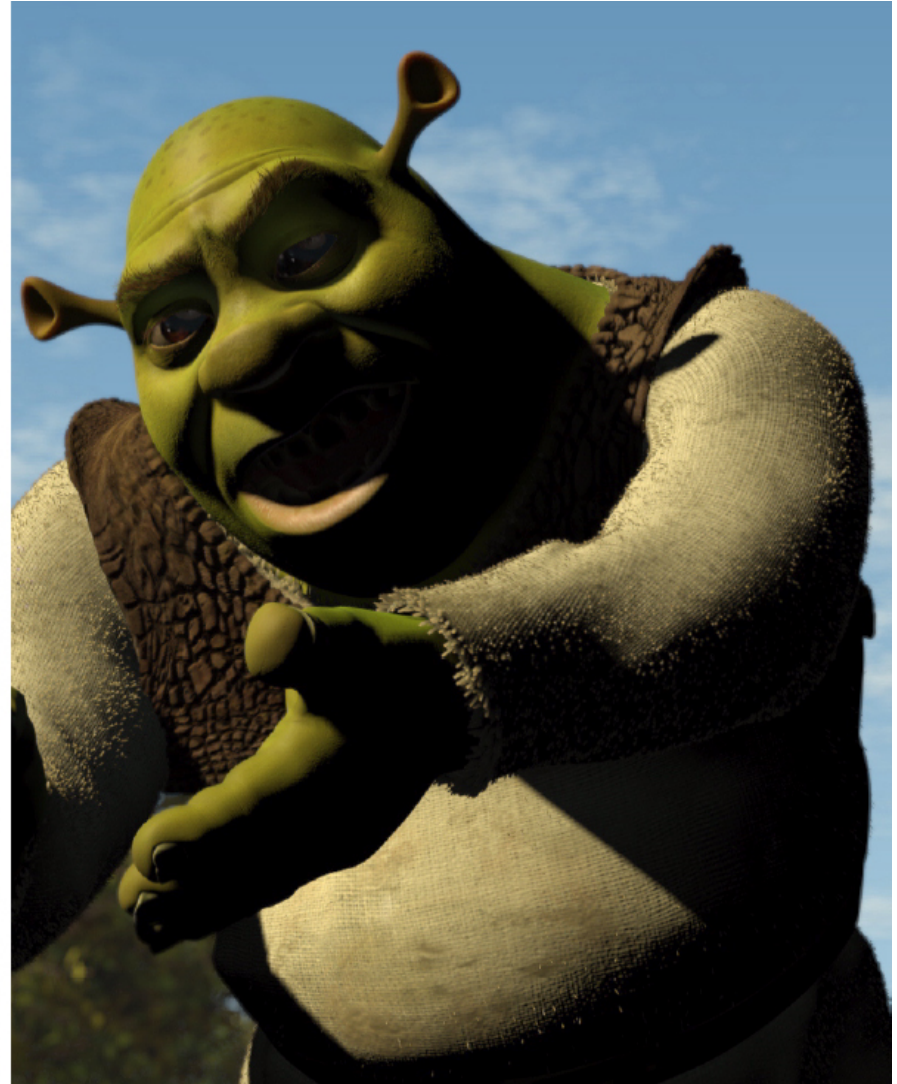


[Woop 2005]

Iluminación Global

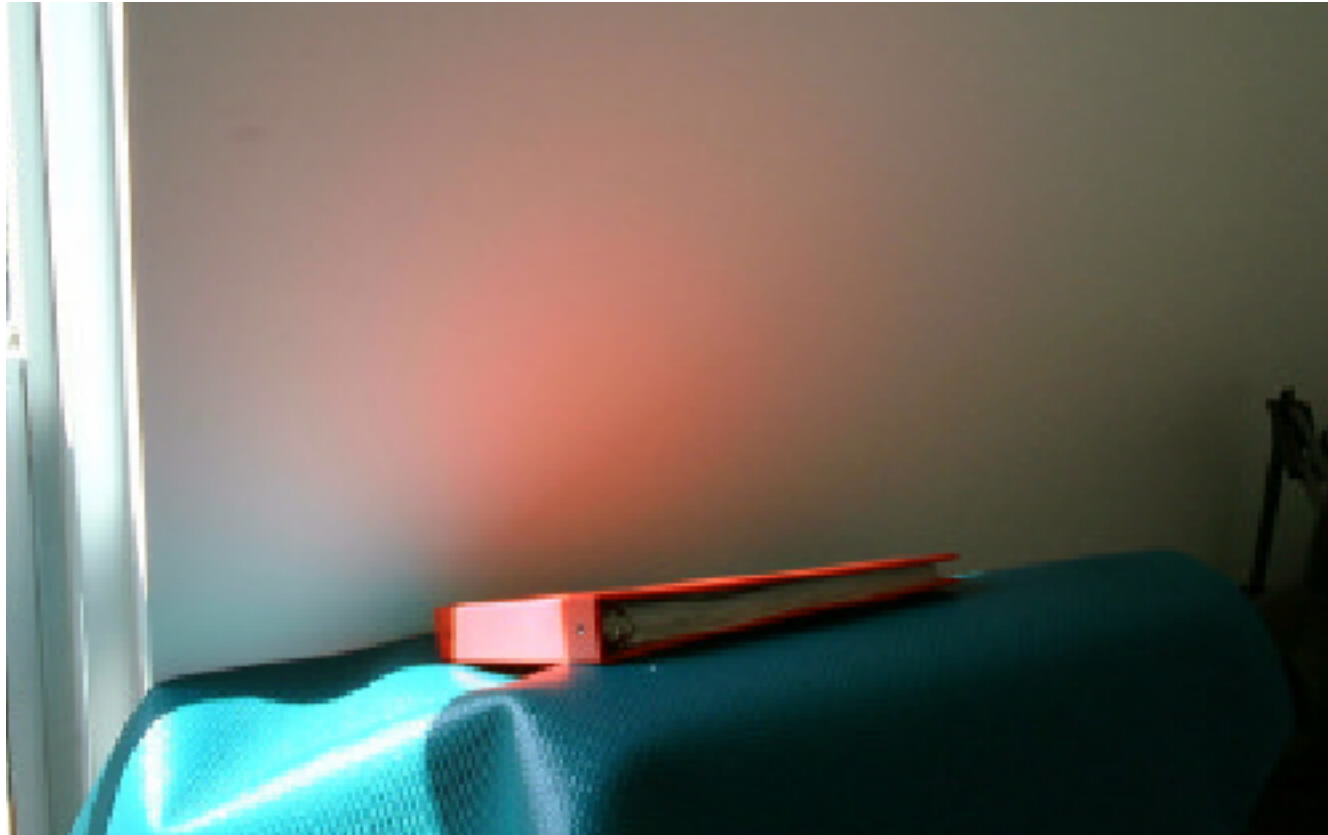


Iluminación global



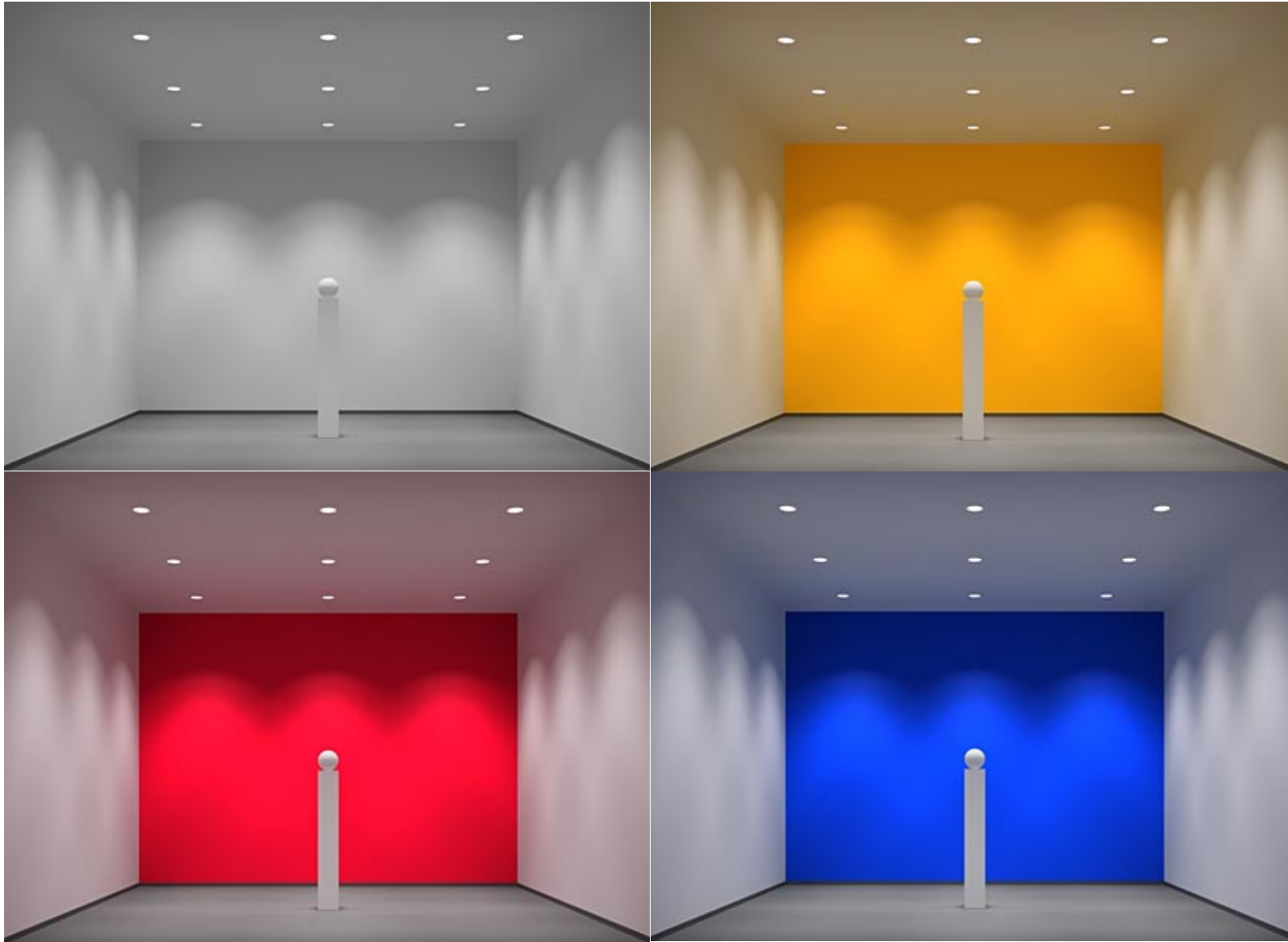
Iluminación directa

Iluminación Global



Color bleeding

Iluminación Global



Color bleeding

Iluminación Global



HENRIK WANN JENSEN 1995

Henrik W. Jensen

Cáusticas

Iluminación Global



Medios participativos

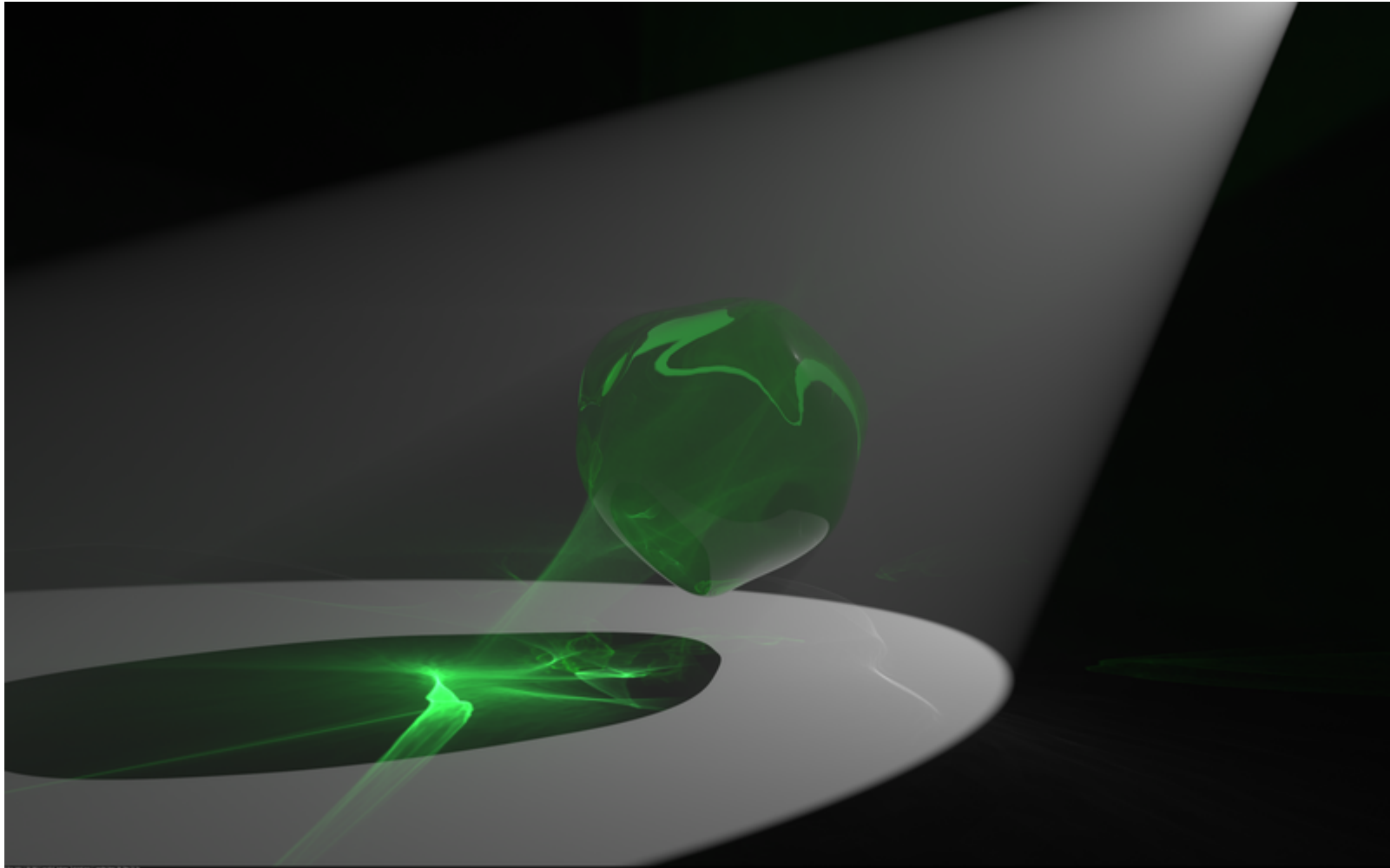
Iluminación Global



Diego Gutiérrez

Medios participativos

Iluminación Global



Francesco Legrenzi @ copyright 2009

Cáusticas en medios participativos

Iluminación Global

- ¿Por qué nos gusta?
 - Es bonita
 - Mucha fenomenología
 - Reflexiones y refracciones: cáusticas
 - Iluminación indirecta: *color bleeding*
 - ...
 - Fotorrealismo
 - Corrección física

Iluminación Global

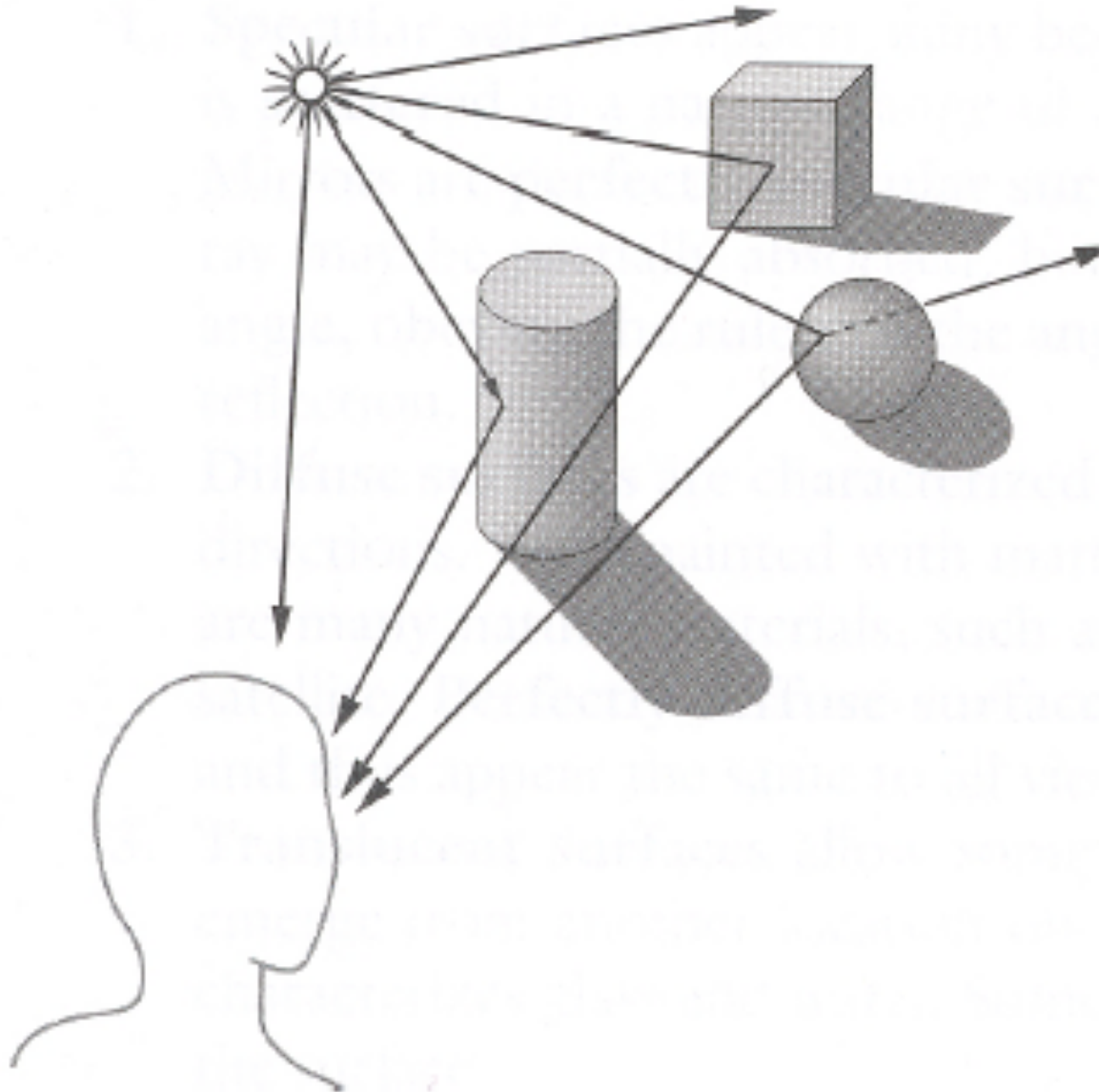
- ¿Por qué NO no gusta?
 - Costosa de simular
 - Costosa de simular
 - Costosa de simular
 - Costosa de simular
- PERO:
 - Entendiendo la iluminación global se pueden hacer “trucos” para aproximarla (videojuegos)

Iluminación Global

- ¿Cómo simularla?
 - Trazado de rayos / técnicas de Montecarlo.
 - ~~• Radiosidad / técnicas basadas en transferencia de calor.~~
 - ~~• Con papel y boli.~~

WARNING:
Dos de estas tres
técnicas están obsoletas

Trazado de rayos



Trazado de rayos

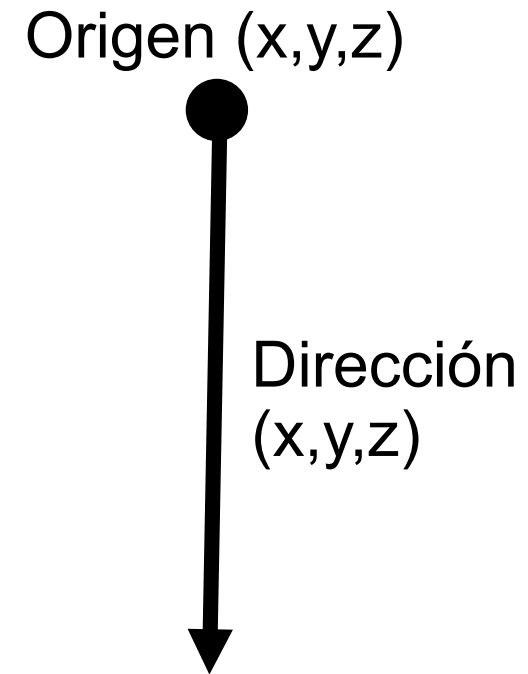
- 3 fases:
 - Generación de rayos
 - Cálculo de intersecciones más cercanas
 - Cálculo de color (con potencial recursividad)

Trazado de rayos: Generación

- ¿Qué es un rayo?

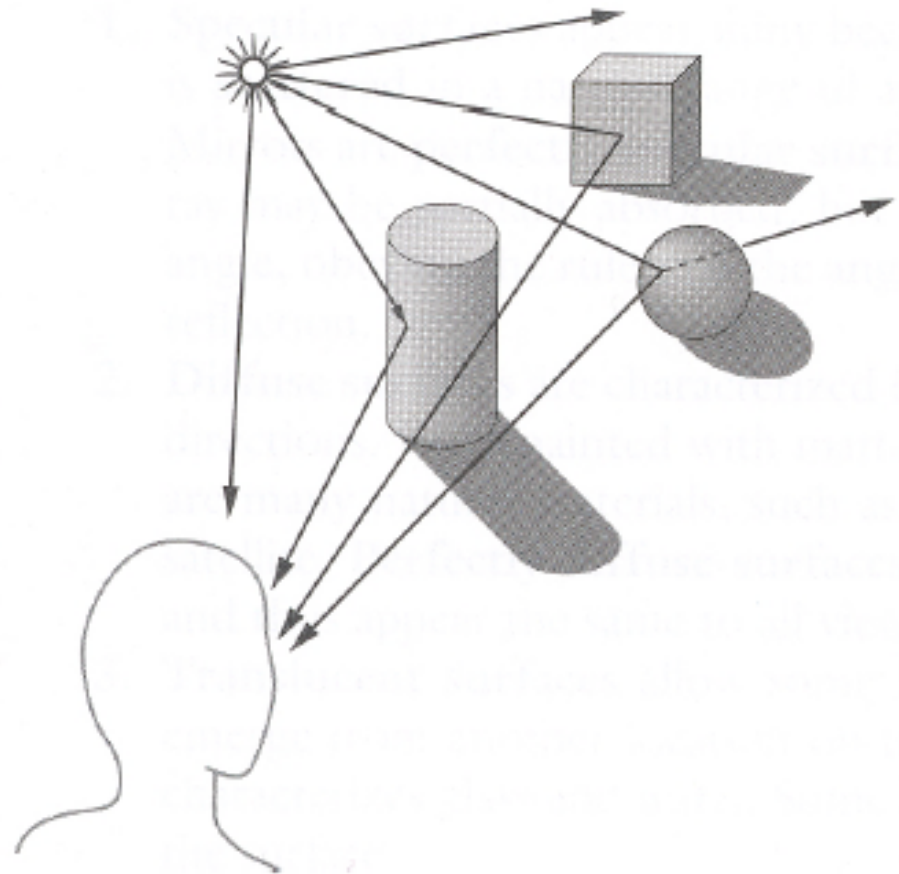


<http://pixdaus.com/single.php?id=37295>



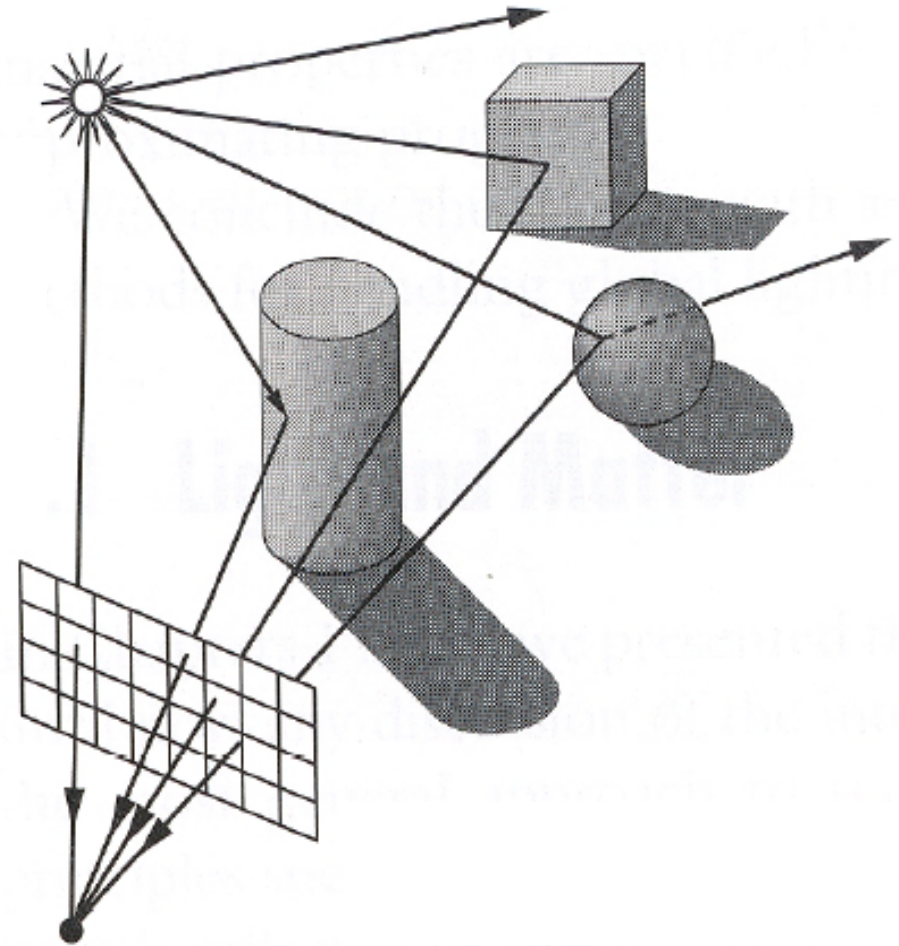
Trazado de rayos: Generación

- Origen
 - En la vida real: de la luz al objeto al ojo
 - En un trazador de rayos: del ojo al objeto a la luz.
 - ¿Por qué?



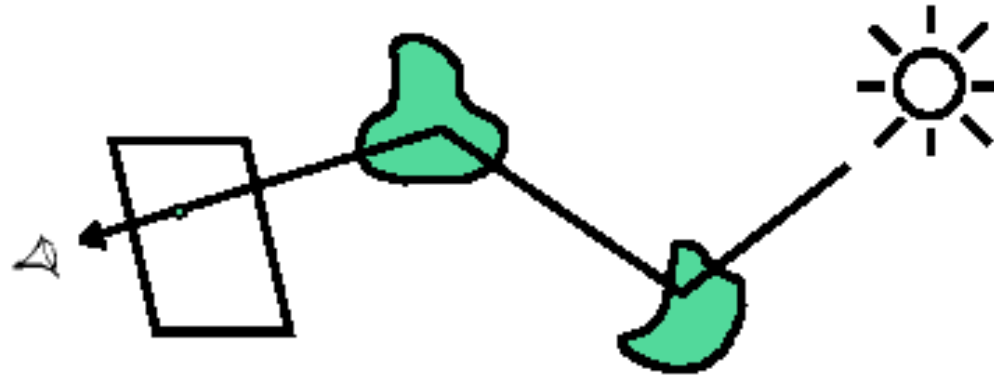
Trazado de rayos: Generación

- Origen
 - Trazar solamente desde la luz es ineficiente.
 - Habitualmente se trazan rayos desde la cámara.
 - Proceso bidireccional

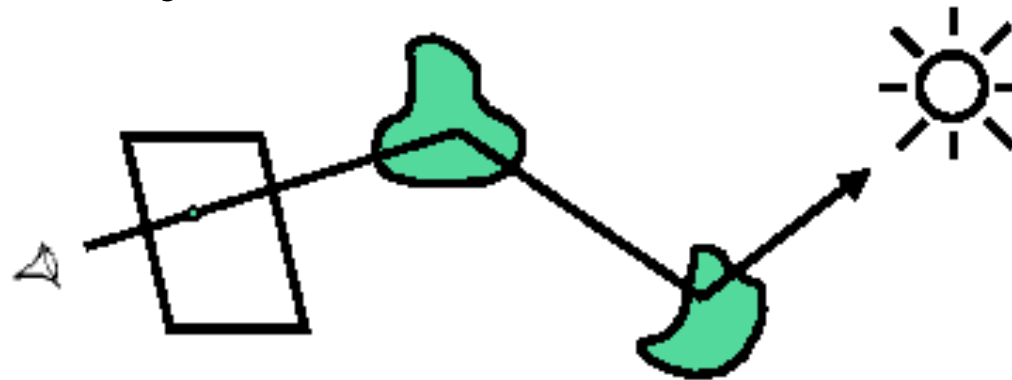


Trazado de rayos: Generación

- Fenómeno real



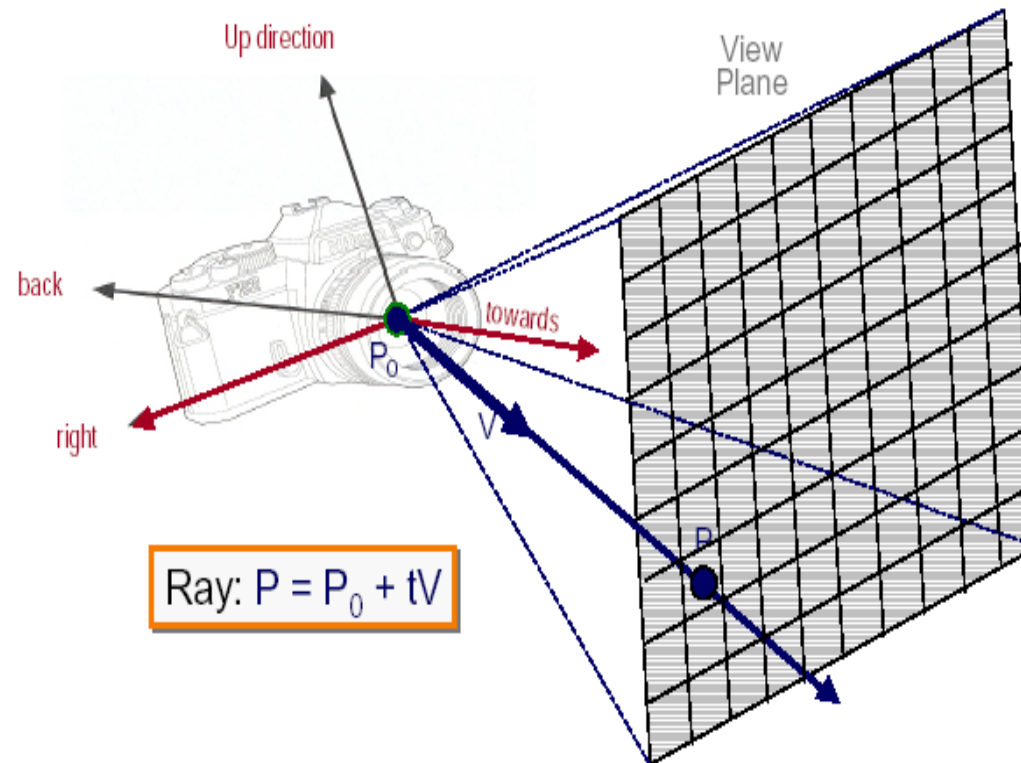
- Trazado de rayos básico



Trazado de rayos: Generación

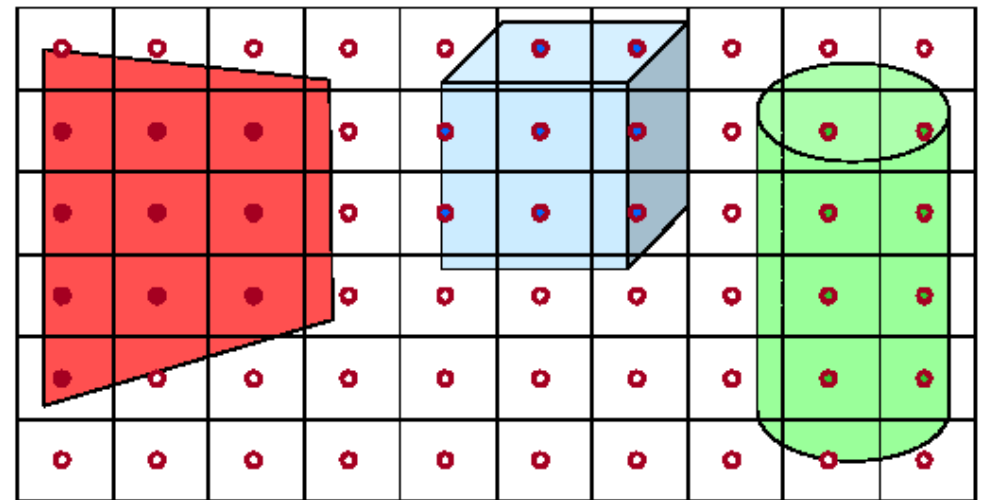
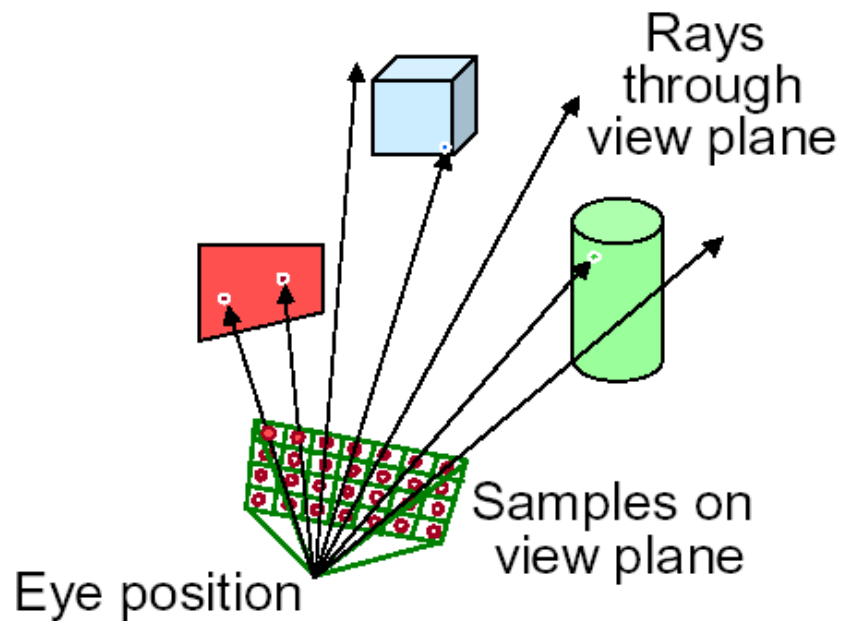
- La cámara

Rayo = $f(x,y)$



Trazado de rayos

- Intersecciones



Trazado de rayos: Intersecciones

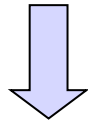
- Sistema de ecuaciones
 - Rayo: $P = O + tD$ (3 ecuaciones, para x, y, z)
 - Superficie implícita: $f(x,y,z)=0$ (1 ecuación más)
- Nos quedamos siempre con la intersección más cercana (con $t>0$)

Trazado de rayos: intersecciones

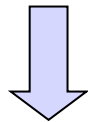
- Rayo-plano:

$$P = O + tD$$

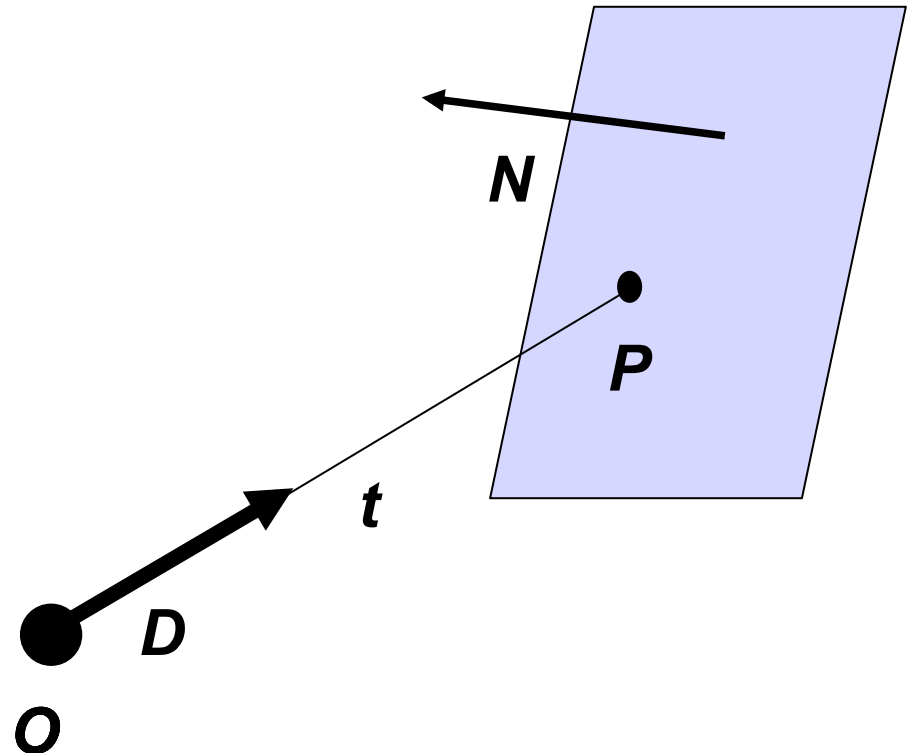
$$P \cdot N + d = 0$$



$$(O + tD) \cdot N + d = 0$$



$$t = -(O \cdot N + d) / (D \cdot N)$$

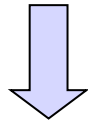


Trazado de rayos: intersecciones

- Rayo-esfera:

$$P = O + tD$$

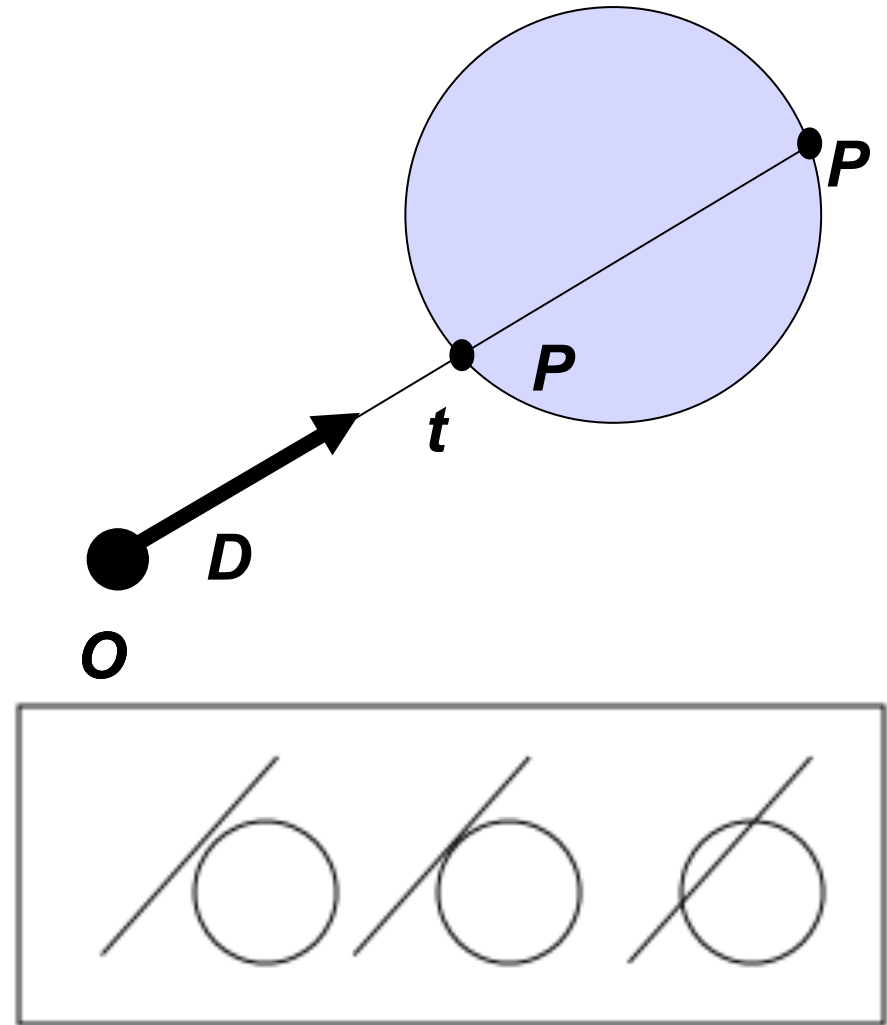
$$P \cdot P - r^2 = 0$$



$$(O + tD) \cdot (O + tD) - r^2 = 0$$

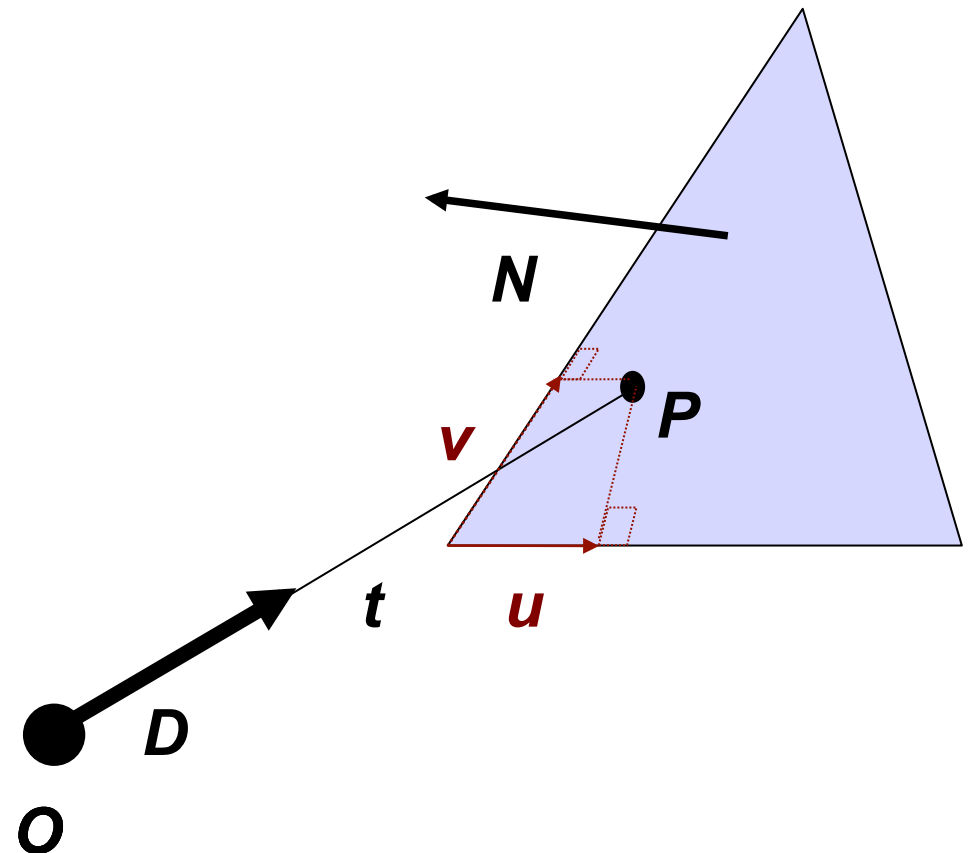
2º grado:

0, 1 ó 2 soluciones



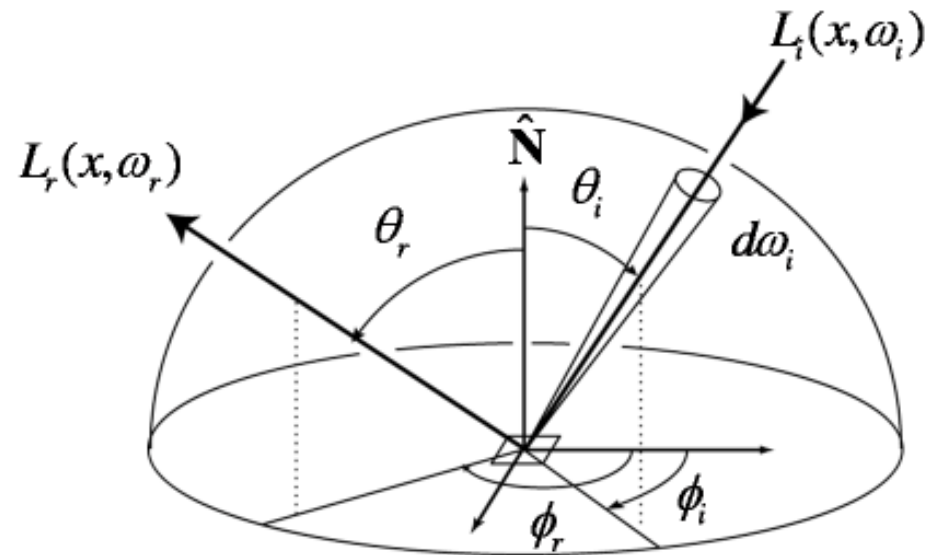
Trazado de rayos: intersecciones

- Rayo-triángulo:
 - Intersección con plano.
 - Comprobación de rangos



Trazado de rayos: color

- Ecuación de reflexión



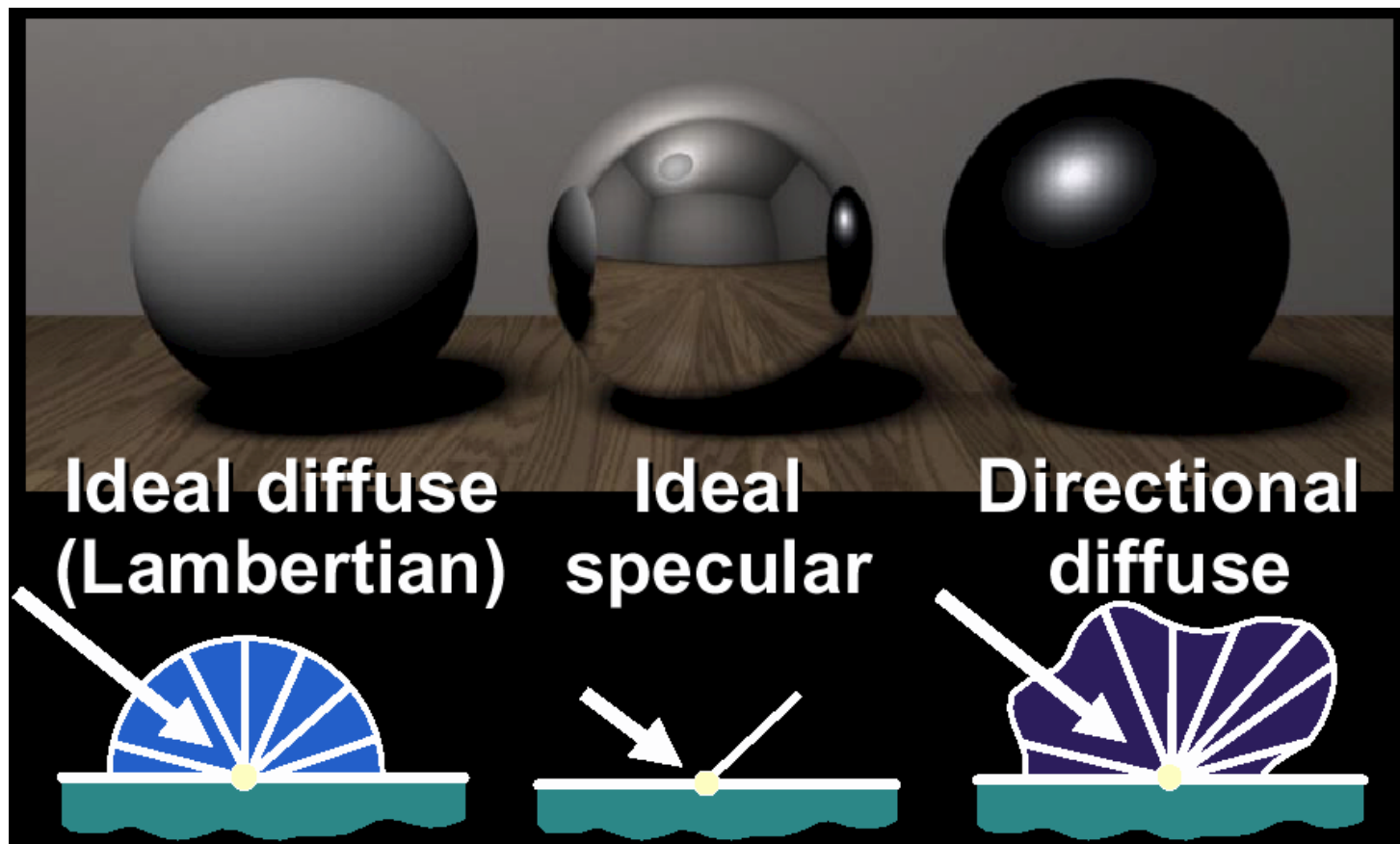
$$L_r(x, \omega_r) = \int_{H^2} \underbrace{f_r(x, \omega_i \rightarrow \omega_r)}_{\text{BRDF}} L_i(x, \omega_i) \cos \theta_i d\omega_i$$

Trazado de rayos: color

- BRDF – Bidirectional Reflectance Distribution Function:
 - Para una dirección de entrada y otra de salida, define la intensidad de luz reflejada.
 - Puede cambiar según longitud de onda → color
- Análogamente BTDF – Bidirectional Transmission Distribution Function
 - Lo mismo, pero en la semiesfera opuesta (refracción)

Trazado de rayos: color

- BRDFs

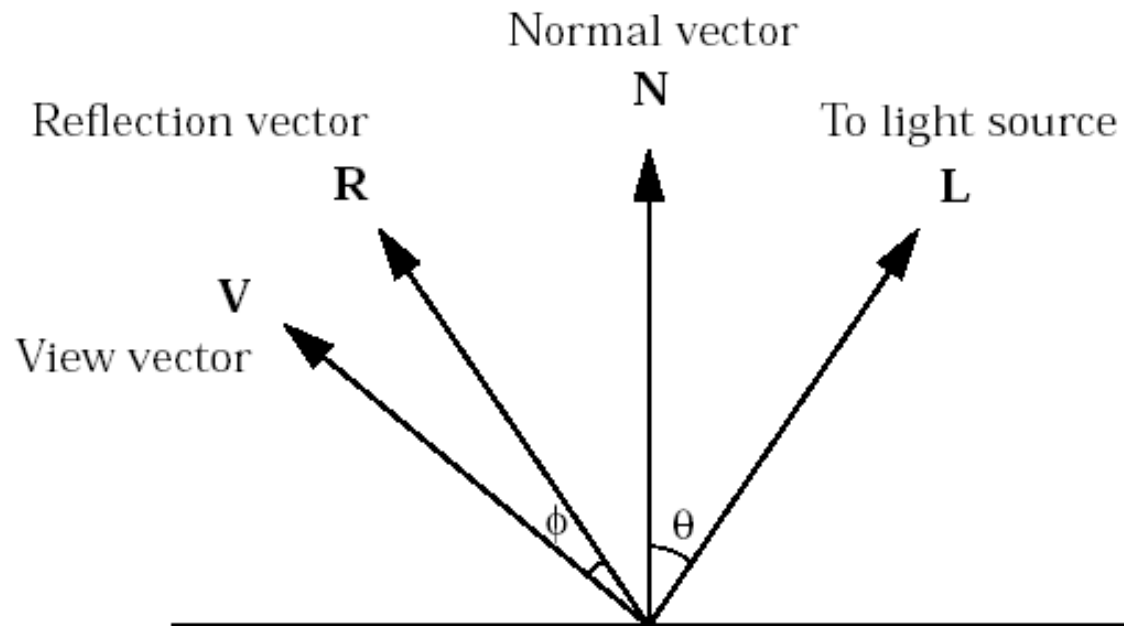


Trazado de rayos: color

- BRDFs: Phong

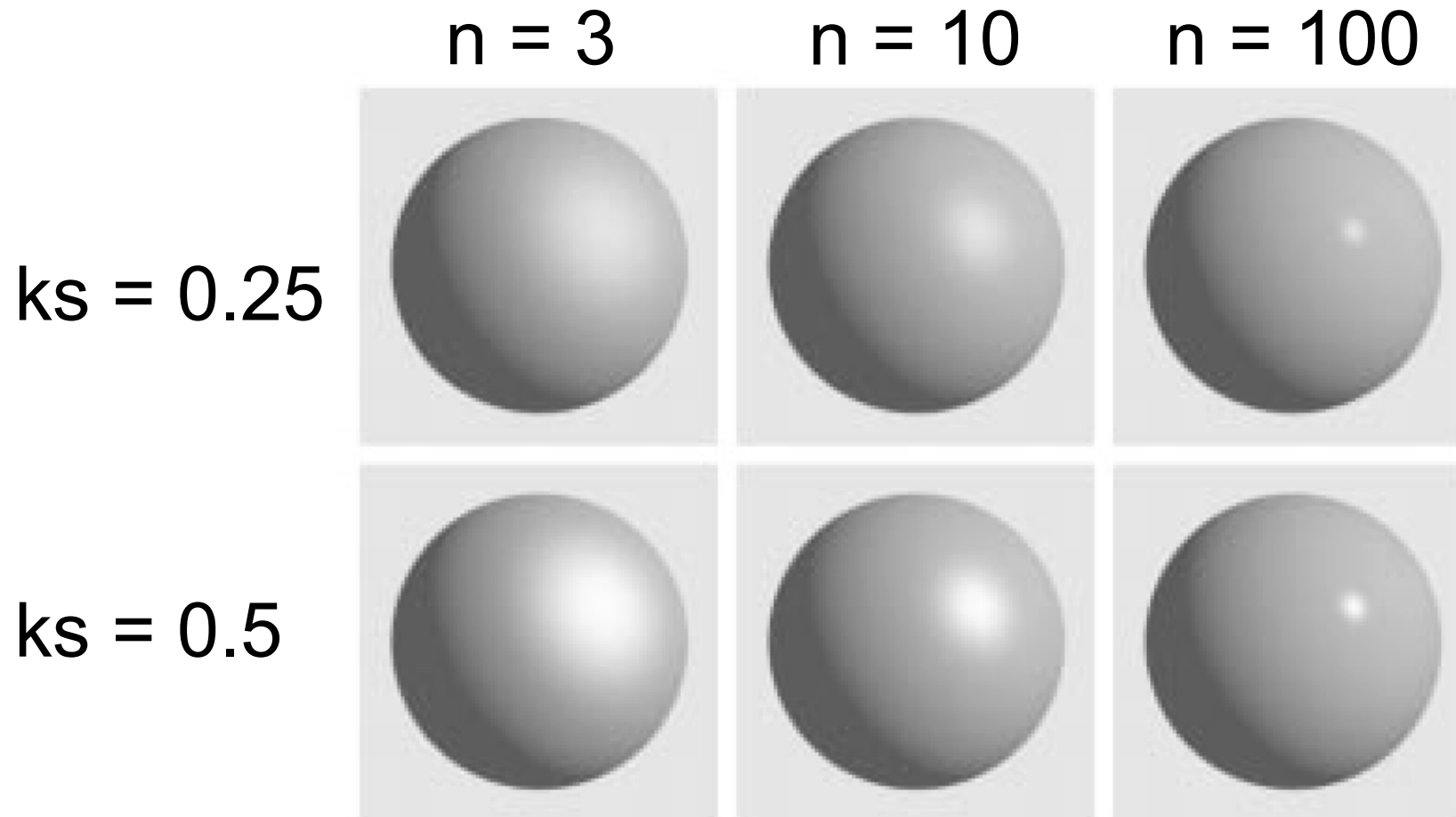
$I = \text{ambient} + \text{diffuse} + \text{specular}$

$$I = k_d I_a + I_p [k_d (\mathbf{N} \cdot \mathbf{L}) + k_s (\mathbf{R} \cdot \mathbf{V})^n]$$



Trazado de rayos: color

- BRDFs: Phong ($k_d = 0.4$, $I_a = 0.2$)



$${}^i\omega p^i \theta \text{soc}({}^i\omega'x)^i \mathcal{T}({}^i\omega \leftarrow {}^i\omega'x)^i f \int = ({}^i\omega'x)^i \mathcal{T}$$

- Figure 10 illustrates the derivation of the radiation field equation for a dipole antenna. The diagram shows a spherical coordinate system with a unit vector $\hat{N}$ and a differential area element $d\omega_i$ on the surface of a sphere. The angle between the direction of radiation $L_i(x, \omega_i)$ and the normal vector $\hat{N}$ is θ_i . The angle between the direction of radiation $L_i(x, \omega_i)$ and the direction of the incident wave $L_i(x, \omega_r)$ is ϕ_i . The angle between the direction of radiation $L_i(x, \omega_i)$ and the direction of the incident wave $L_i(x, \omega_r)$ is ϕ_i . The angle between the direction of radiation $L_i(x, \omega_i)$ and the direction of the incident wave $L_i(x, \omega_r)$ is ϕ_i .
- The radiation field equation is given by:
- $$L_r(x, \omega_r) = \int_{H^2} f_r(x, \omega_i) \rightarrow$$
- where $f_r(x, \omega_i)$ is the radiation field function, and H^2 is the set of all possible directions of radiation.
- The radiation field function $f_r(x, \omega_i)$ is defined as:
- $$f_r(x, \omega_i) = \int_{\mathbb{R}^3} f_r(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i d\omega_i$$
- where θ_i is the angle between the direction of radiation $L_i(x, \omega_i)$ and the normal vector $\hat{N}$.
- The radiation field function $f_r(x, \omega_i)$ is also defined as:
- $$f_r(x, \omega_i) = \int_{\mathbb{R}^3} f_r(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i d\omega_i$$
- where θ_i is the angle between the direction of radiation $L_i(x, \omega_i)$ and the normal vector $\hat{N}$.
- Figure 10 illustrates the derivation of the radiation field equation for a dipole antenna. The diagram shows a spherical coordinate system with a unit vector $\hat{N}$ and a differential area element $d\omega_i$ on the surface of a sphere. The angle between the direction of radiation $L_i(x, \omega_i)$ and the normal vector $\hat{N}$ is θ_i . The angle between the direction of radiation $L_i(x, \omega_i)$ and the direction of the incident wave $L_i(x, \omega_r)$ is ϕ_i . The angle between the direction of radiation $L_i(x, \omega_i)$ and the direction of the incident wave $L_i(x, \omega_r)$ is ϕ_i . The angle between the direction of radiation $L_i(x, \omega_i)$ and the direction of the incident wave $L_i(x, \omega_r)$ is ϕ_i .

$$w' \cos \theta w'$$

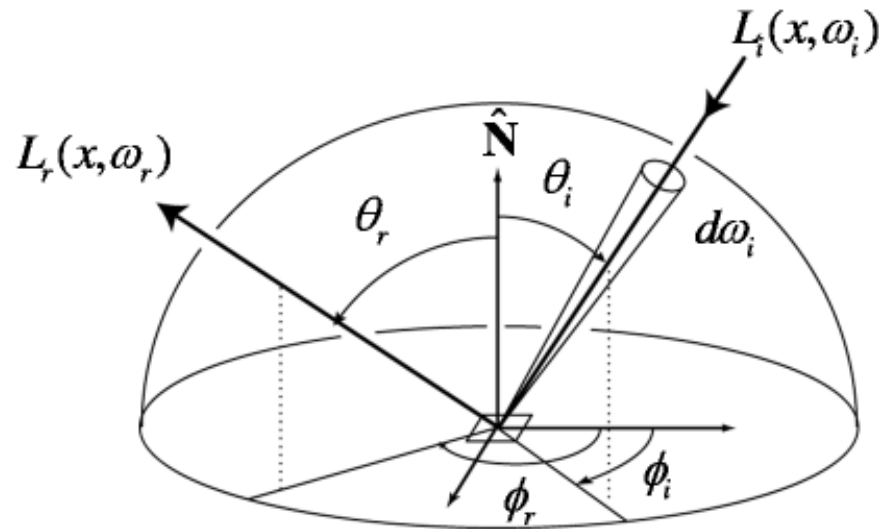
$$L_r(x, \omega_r) = \int_{H^2} f_r(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i d\omega_i$$

$$L_r(x, \omega_r) = \int_{H^2} f_r(x, \omega_i) \rightarrow \omega_r) L_i(x,$$

$$= \int_{H^2} f_r(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i d\omega_i$$

Trazado de rayos: color

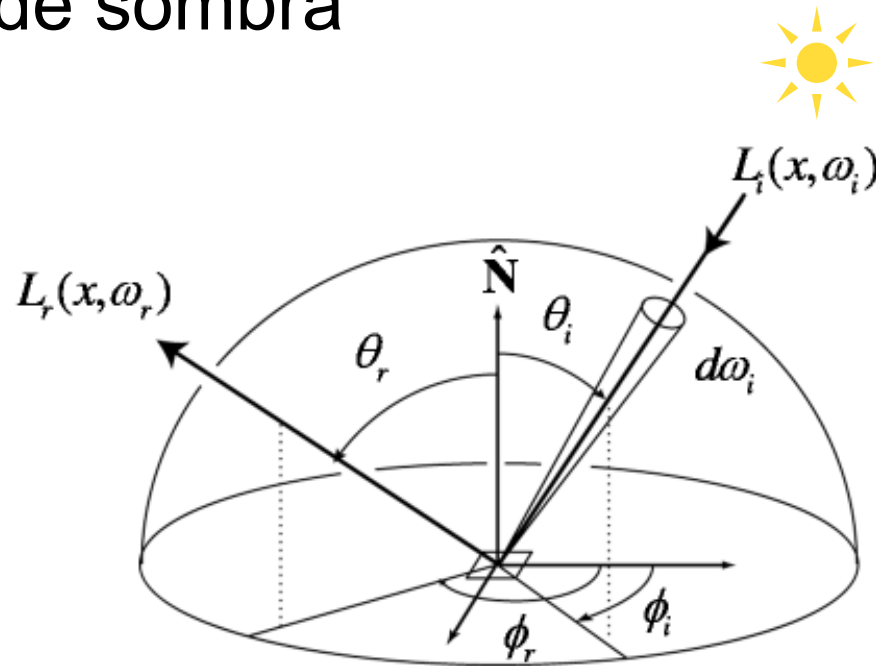
- Costoso de simular
- Excepción: funciones *delta*



$$L_r(x, \omega_r) = \int_{H^2} f_r(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i d\omega_i$$

Trazado de rayos: color

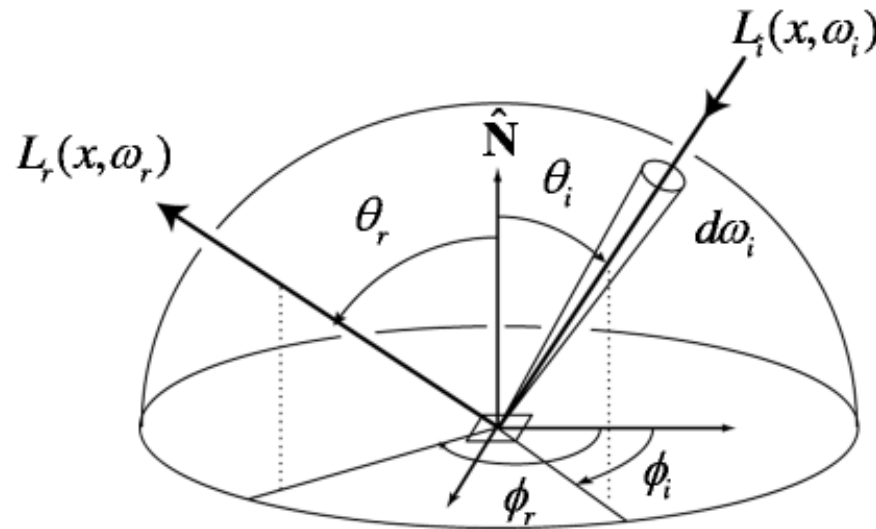
- Fuentes de luz delta: puntuales/direccionales:
 - Rayos de sombra



$$L_r(x, \omega_r) = \int_{H^2} f_r(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i d\omega_i$$

Trazado de rayos: color

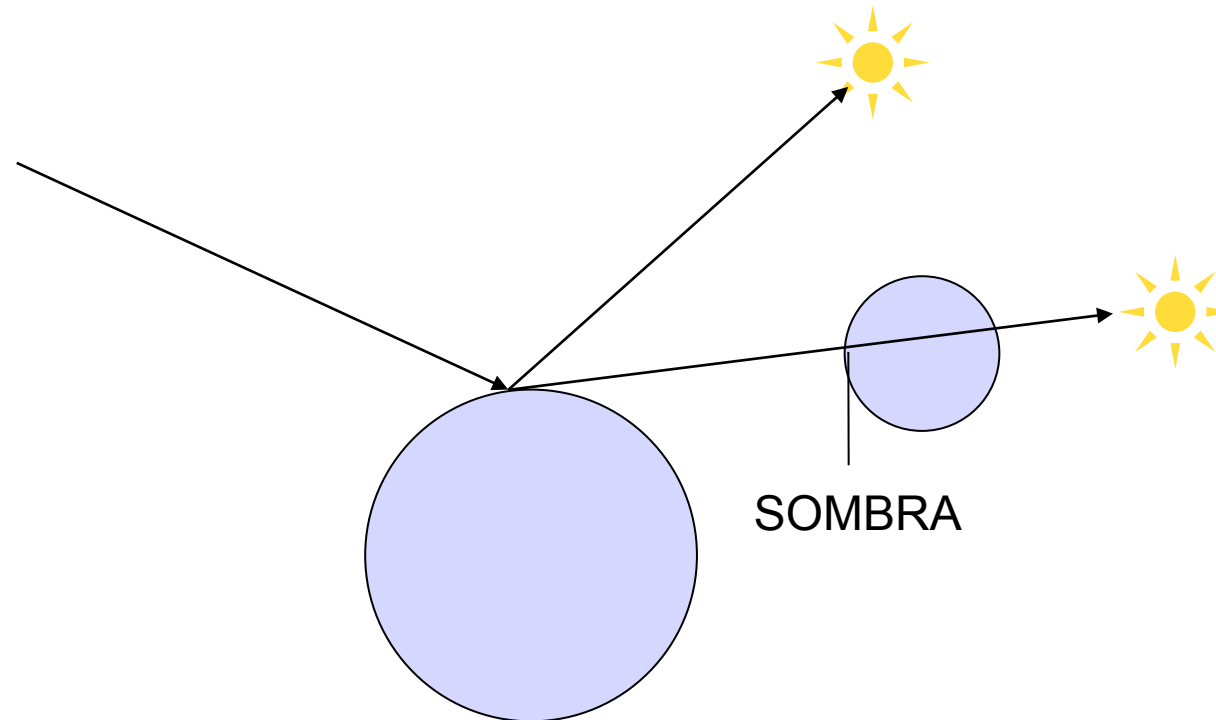
- BRDF delta / BTDF delta: caminos especulares perfectos
 - Rayos secundarios



$$L_r(x, \omega_r) = \int_{H^2} \boxed{f_r(x, \omega_i \rightarrow \omega_r)} L_i(x, \omega_i) \cos \theta_i d\omega_i$$

Trazado de rayos: color

- Rayos de sombra
 - Trazamos rayos, buscamos cualquier intersección
 - Luces delta: puntuales o direccionales

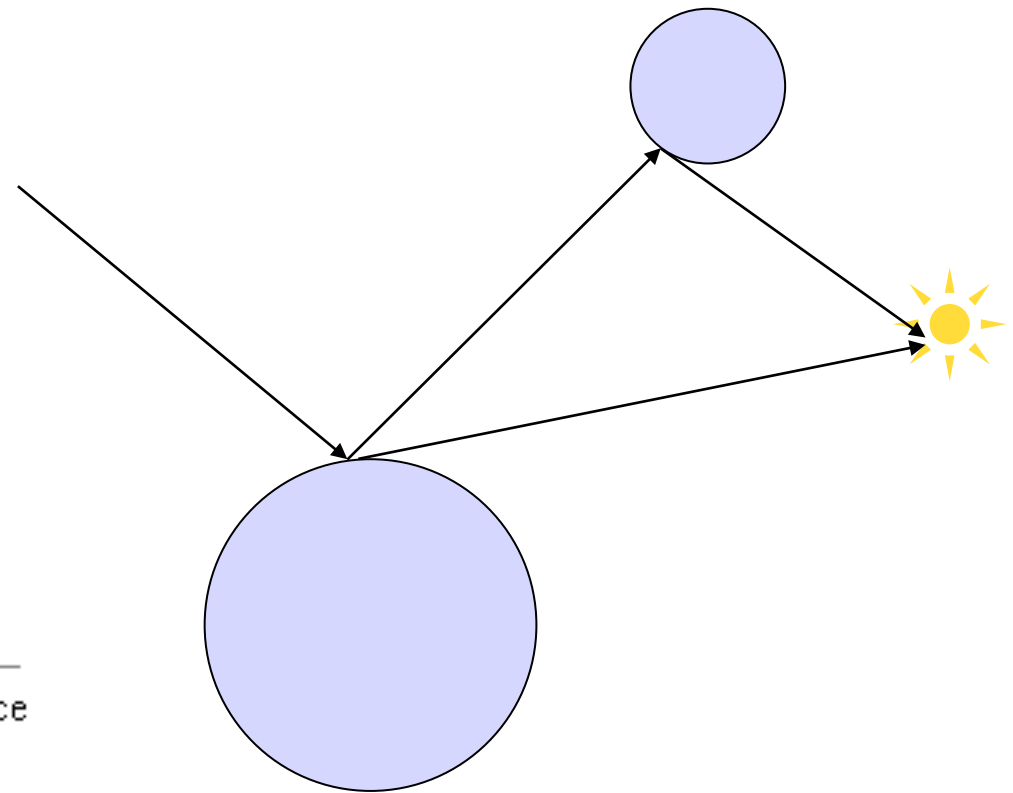
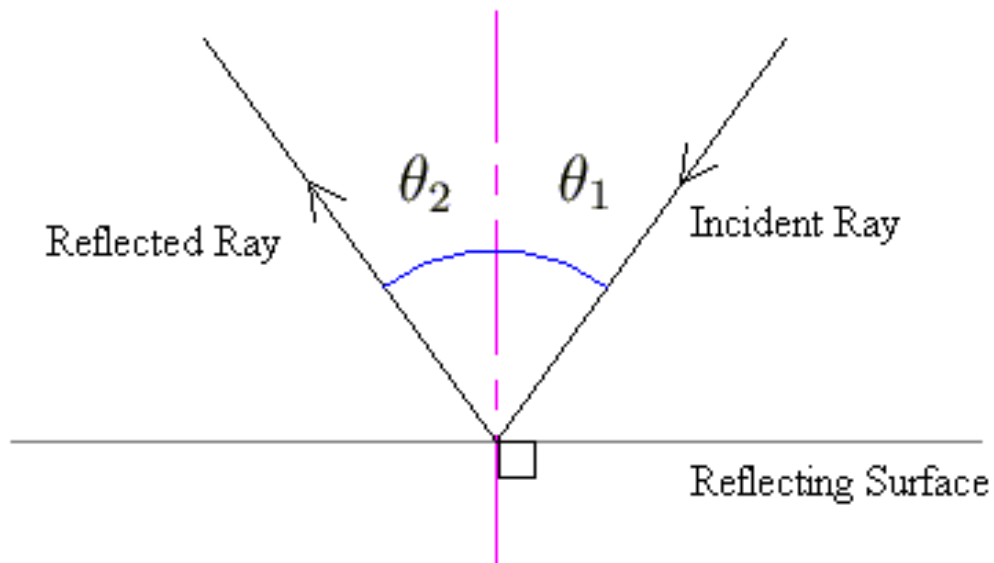


Trazado de rayos: color

- Delta BRDF – Reflexión especular perfecta

Law of Reflection:

$$\theta_1 = \theta_2$$



Trazado de rayos: color

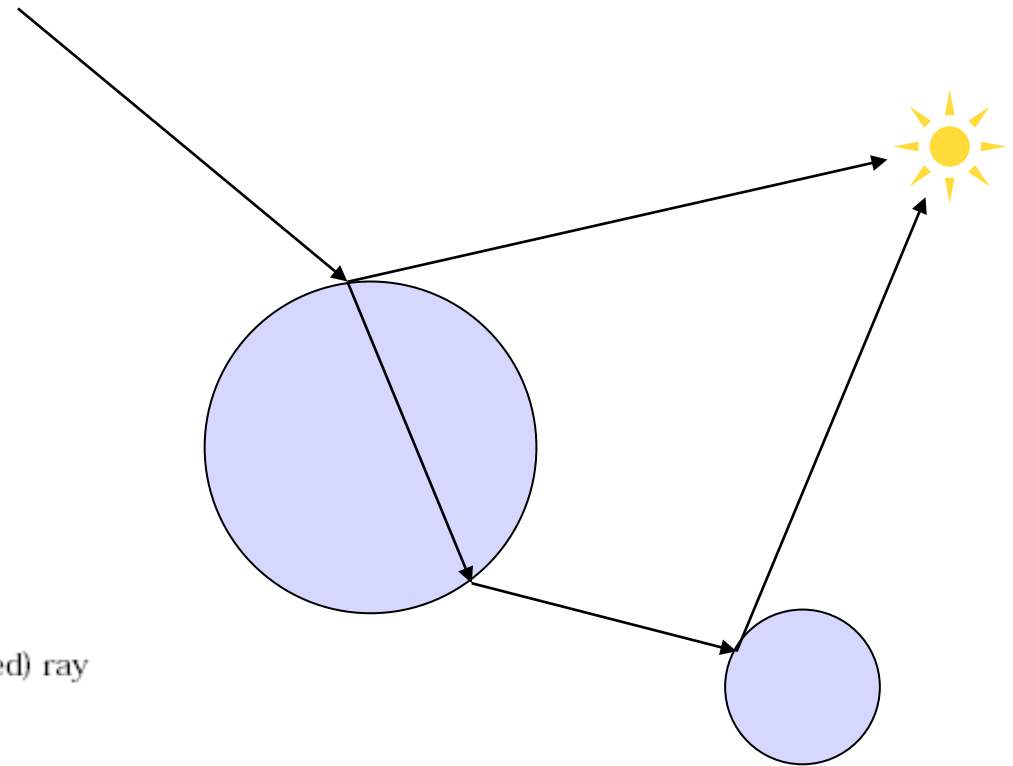
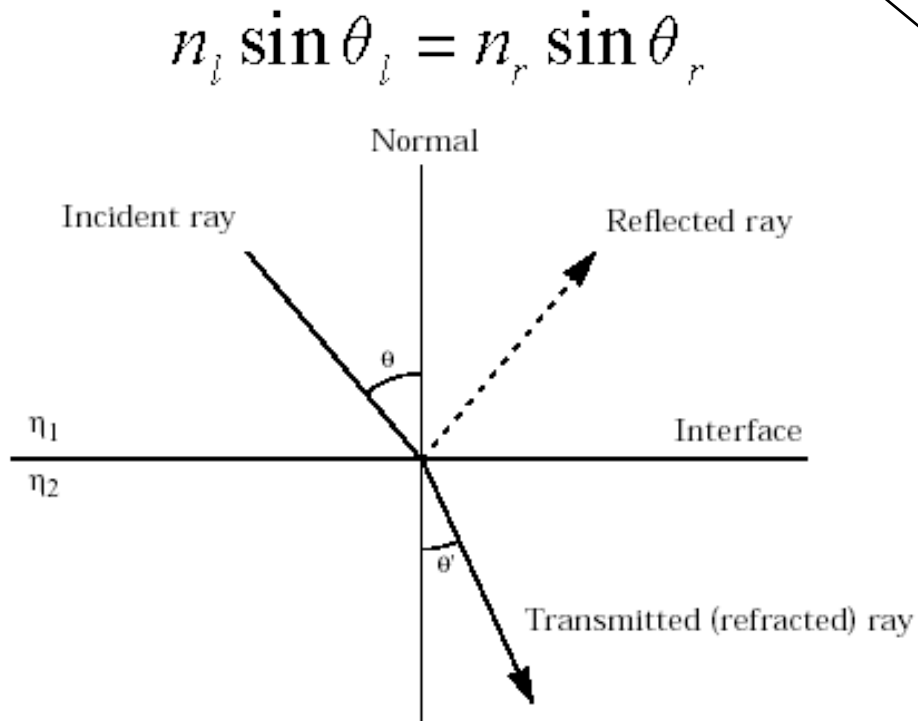
- Delta BRDF – Reflexión especular perfecta



Partial installation view
of Mirror Universe as
seen in a convex mirror,
Caren Golden Fine Art,
NYC

Trazado de rayos: color

- Delta BTDF – Transmisión perfecta, ley de Snell



Trazado de rayos: color

- Delta BTDF – Transmisión perfecta, ley de Snell

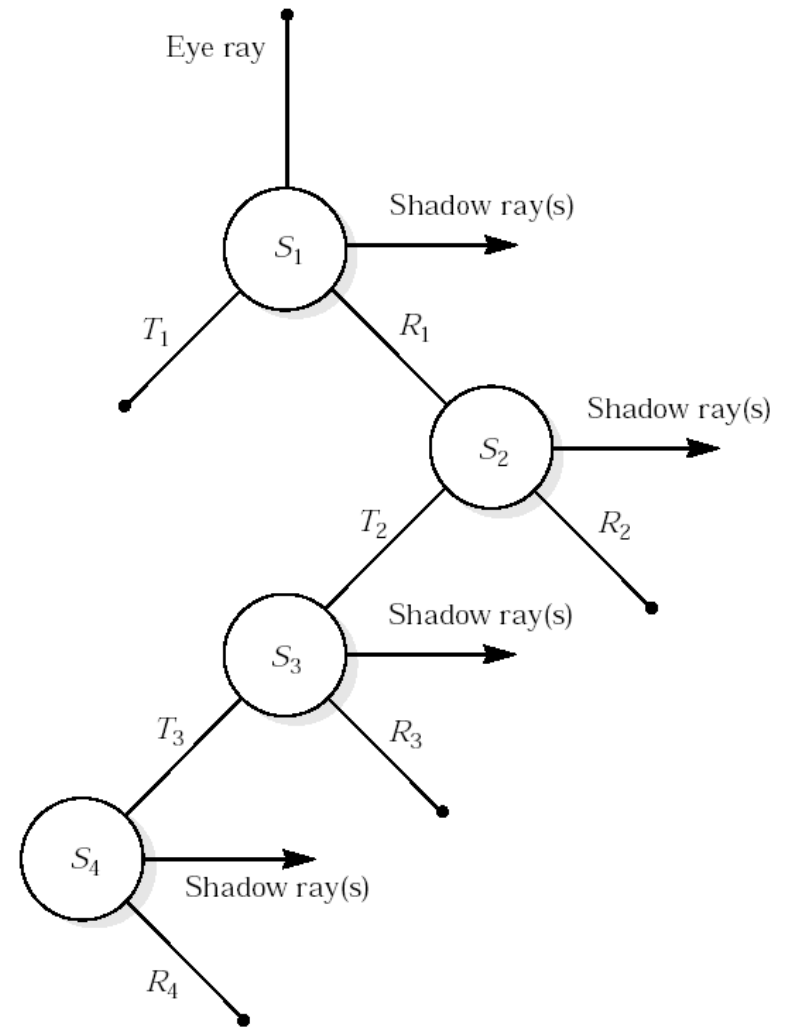


Trazado de rayos: color

- Delta BRDF / BTDF:
 - Se calcula el *rayo secundario* aplicando la ecuación correspondiente (ley de reflexión, ley de Snell).
 - Se vuelve a trazar sobre la escena.
 - Es recursivo, puesto que el nuevo rayo intersecta y busca de nuevo información de color.

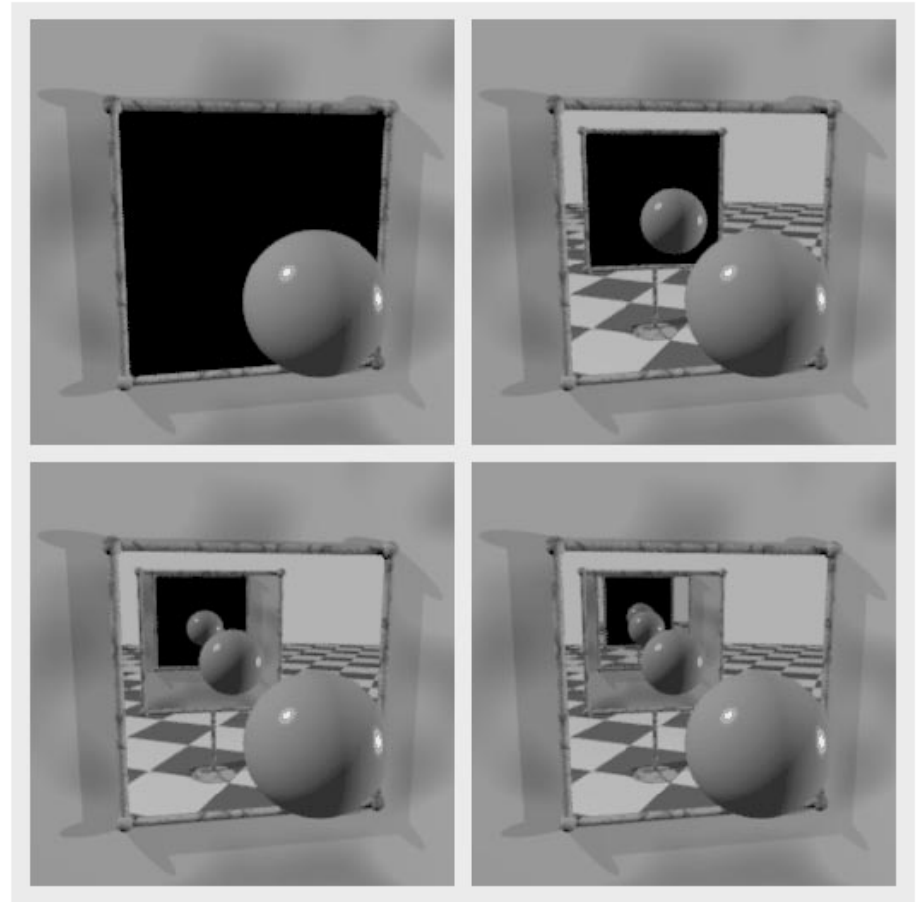
Trazado de rayos: color

- Delta BRDF / BTDF:
Arbol de rayos



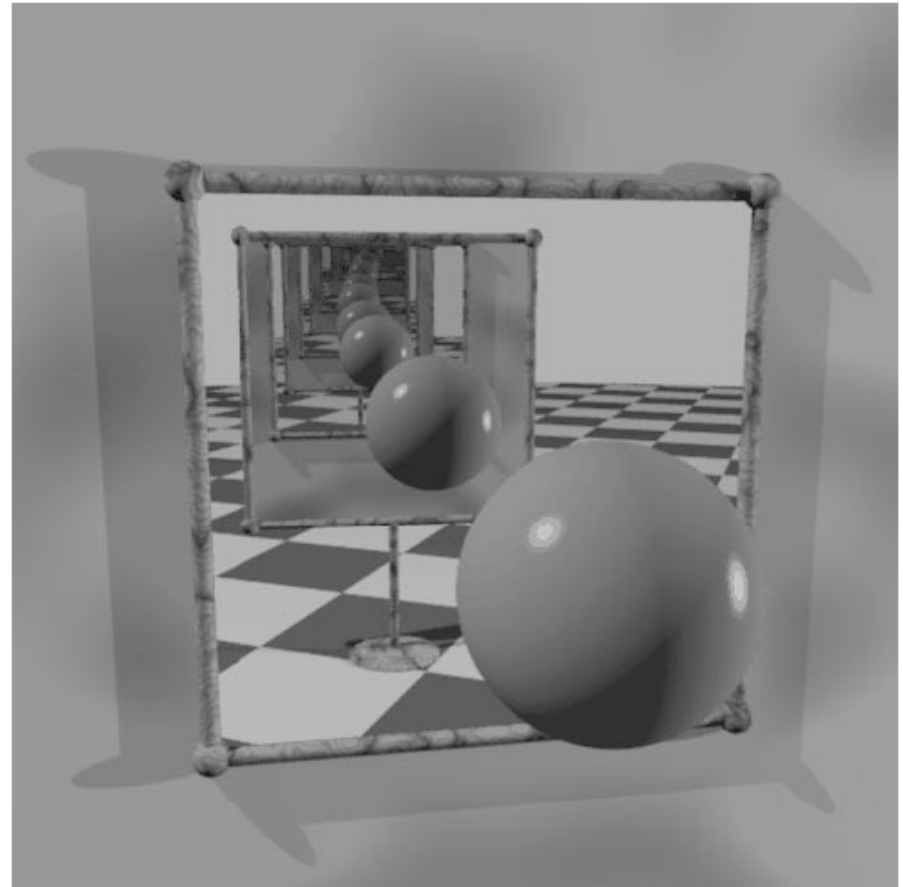
Trazado de rayos: color

- Ojo al número de rebotes secundarios calculados.
- ¿Por qué hay que limitarlos? ¿Cuántos ponemos?



Trazado de rayos: color

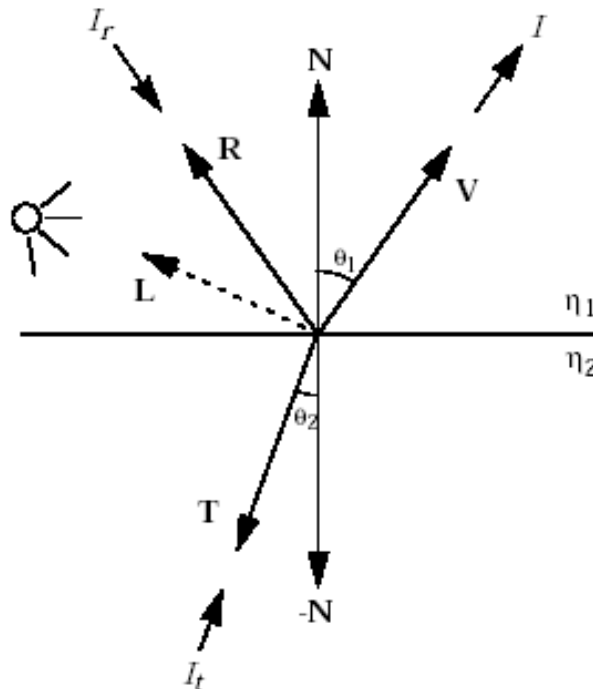
- Infinitos rebotes costarían infinito tiempo.
- Dos opciones:
 - Poner un máximo número.
 - Poner un umbral de luz.



Trazado de rayos: color

- Sumando todo (hasta ahora)

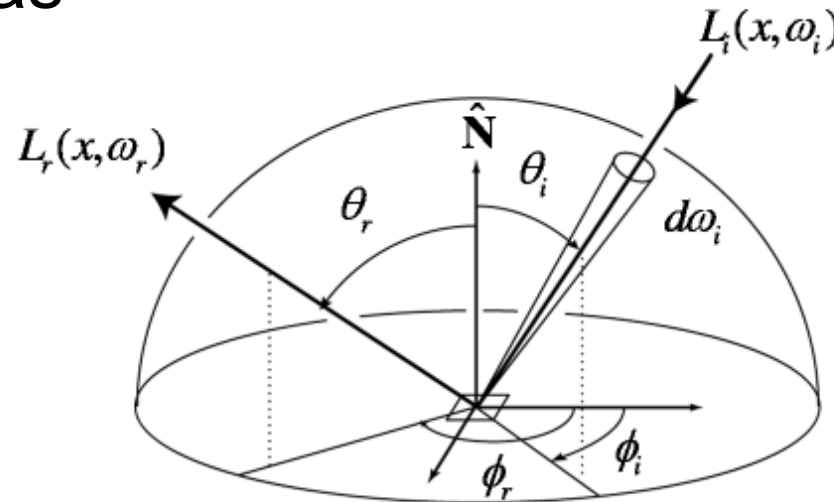
$$I = k_d I_a + k_d \sum_{i=1}^m I_{p_i} [(\mathbf{N} \cdot \mathbf{L}_i) + k_s (\mathbf{R} \cdot \mathbf{V})^n] + k_r I_r + k_t I_t$$



¿Qué falta aquí?

Trazado de rayos: color

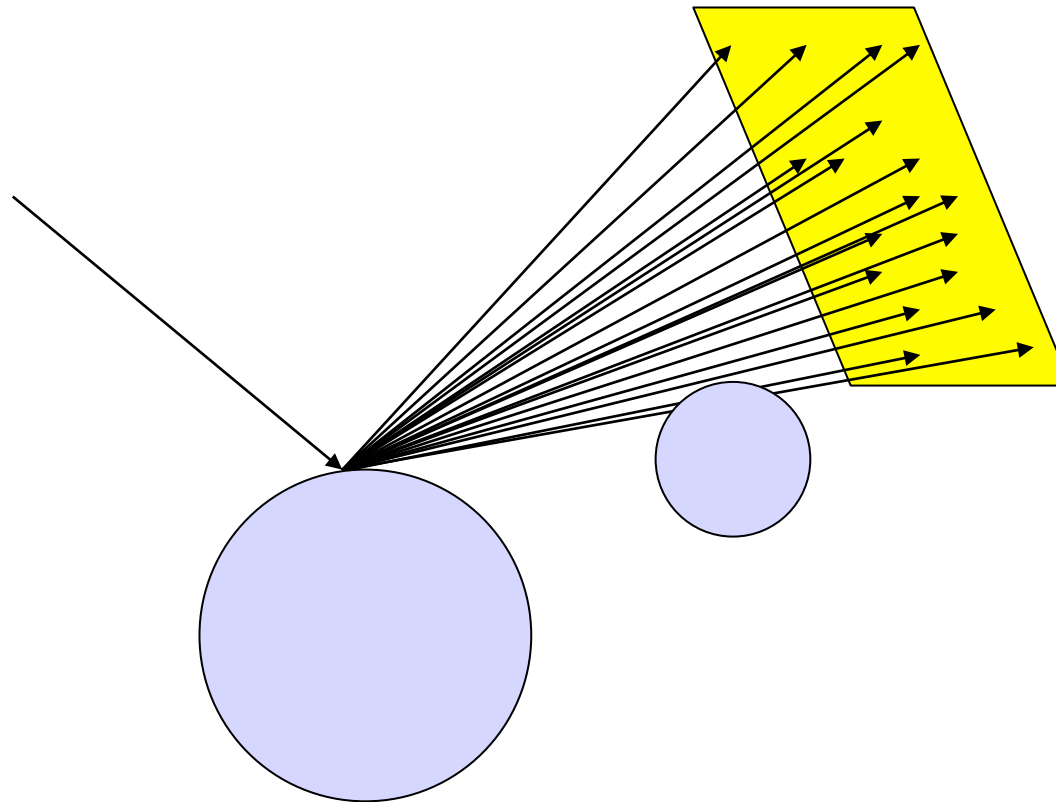
- Todo lo que NO es delta:
 - Luces de área / mapas de entorno / volumétricas
 - Interreflexión difusa (o casi difusa)
 - Cáusticas
 - ...



$$L_r(x, \omega_r) = \int_{H^2} f_r(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos \theta_i d\omega_i$$

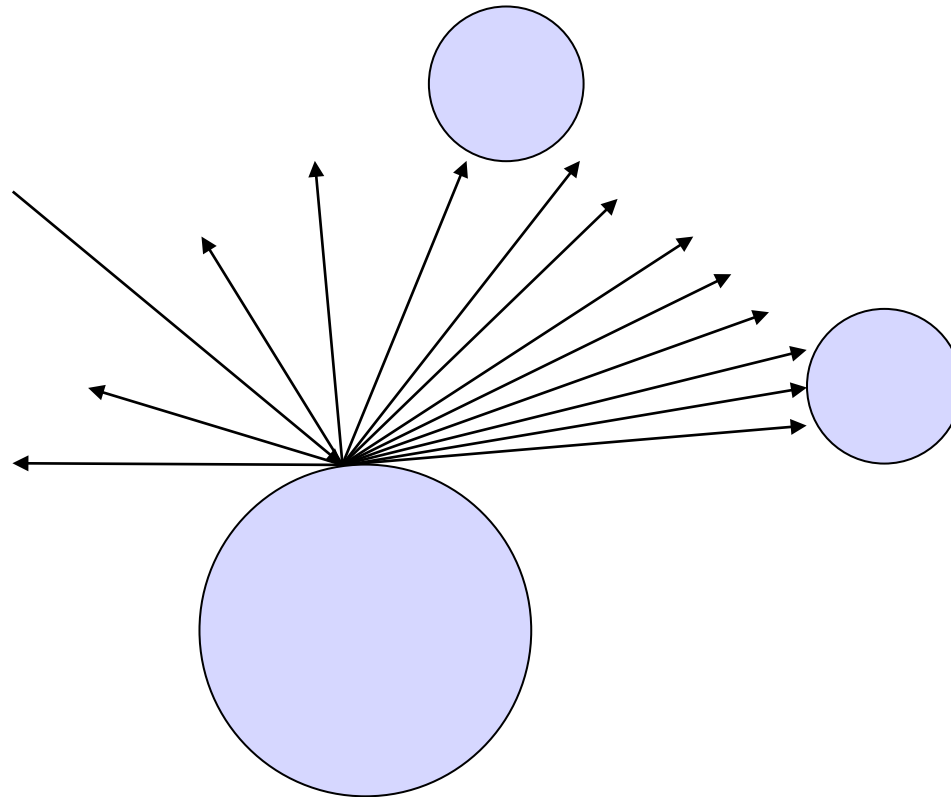
Trazado de rayos: color

- Luces de área (luz directa):
 - Muestrear mediante muchos rayos de sombra.
 - Equivalente a muchas luces puntuales “delta”



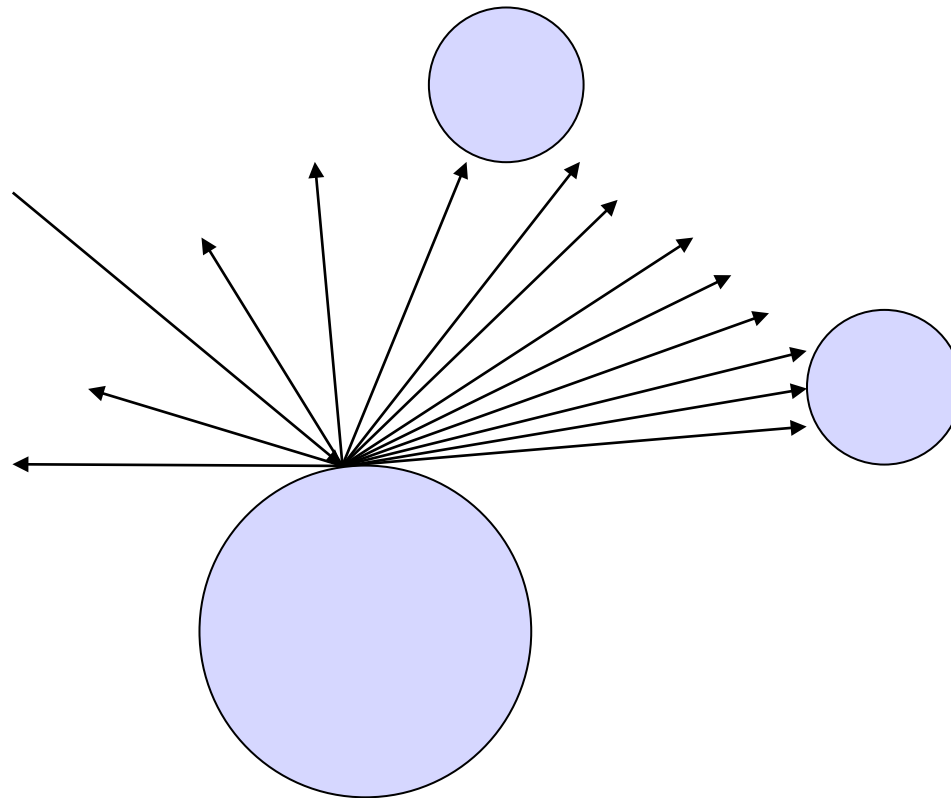
Trazado de rayos: color

- Interreflexión difusa
 - Muestrear toda la semiesfera mediante rayos secundarios



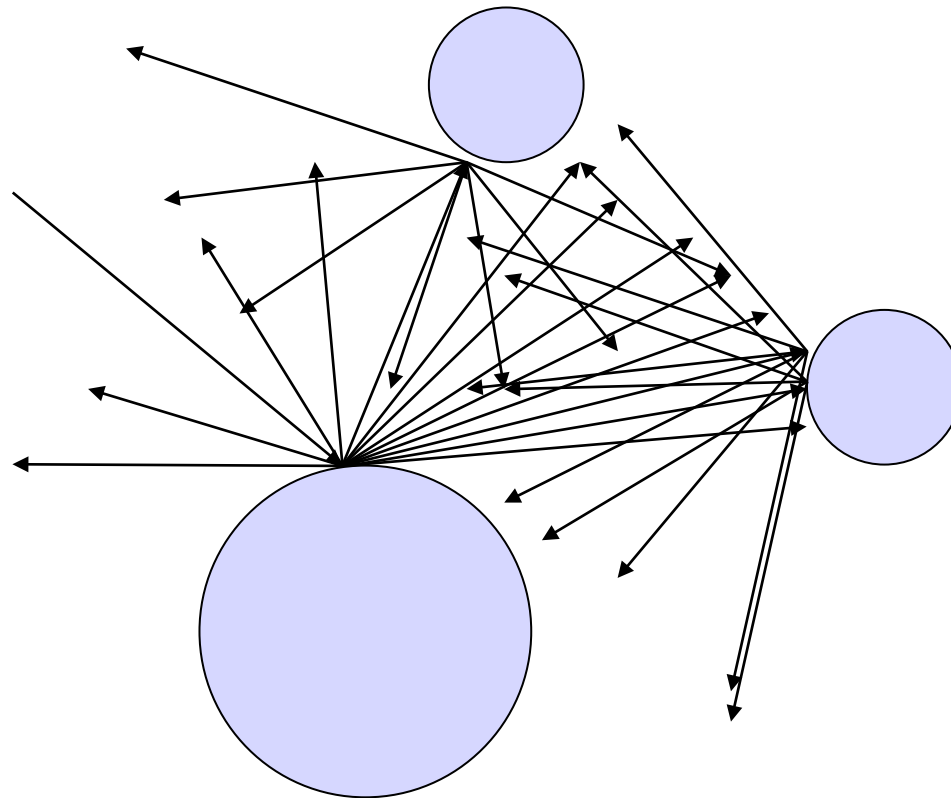
Trazado de rayos: color

- Interreflexión difusa
 - Crecimiento exponencial con respecto a rebotes



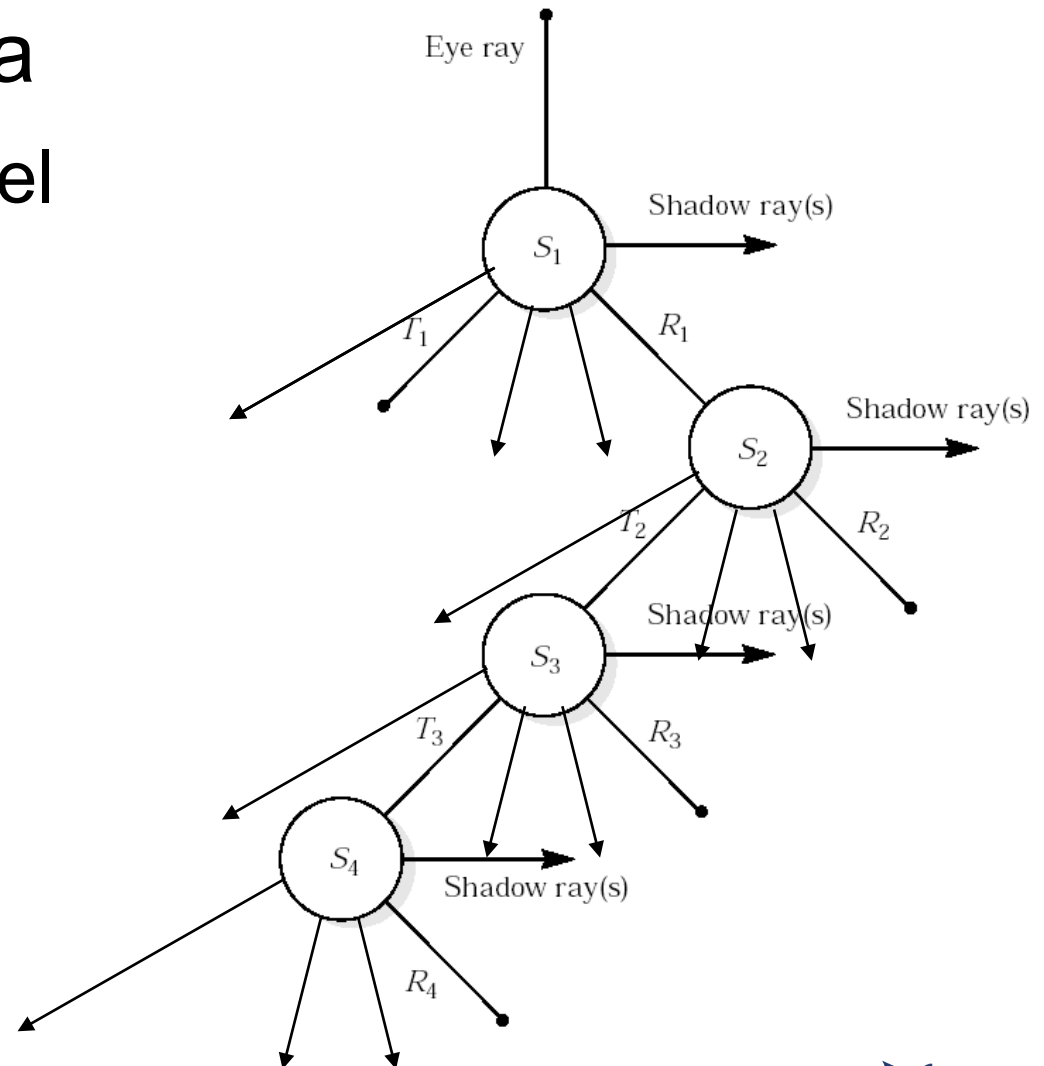
Trazado de rayos: color

- Interreflexión difusa
 - Crecimiento exponencial con respecto a rebotes



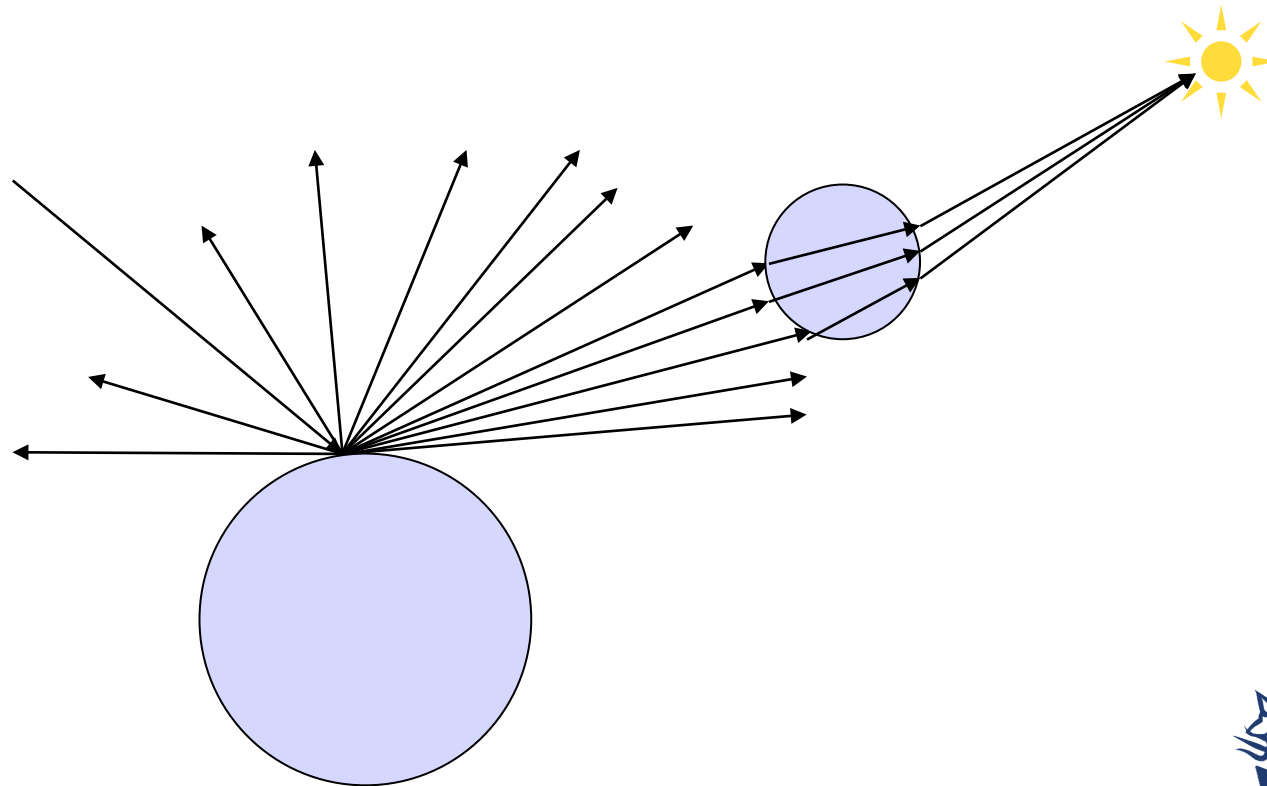
Trazado de rayos: color

- Interreflexión difusa
 - Muchas ramas en el arbol de rayos



Trazado de rayos: color

- Cáusticas
 - Caminos difíciles de encontrar, requieren muchas muestras difusas



Trazado de rayos: color

- Cáusticas

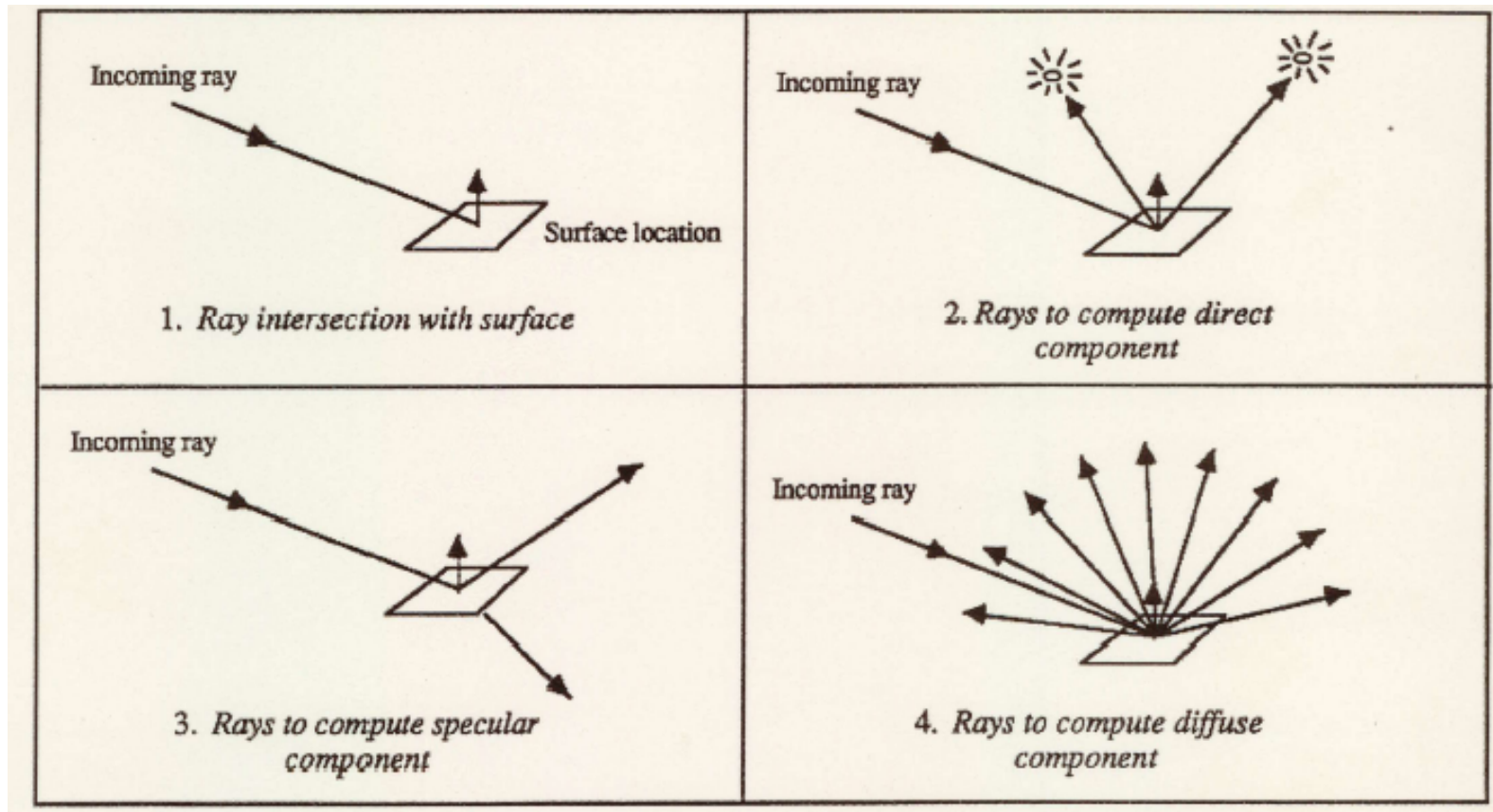


HENRIK WANN JENSEN 1995

Henrik W. Jensen

Trazado de rayos

- Resumen



Trazado de rayos

- Algoritmo:
 1. Se lanza un rayo primario por pixel (en el mejor caso).
 2. Cada rayo primario intersecta todos los objetos de la escena.
 3. Cada intersección genera un rayo de sombra por cada fuente de luz.
 4. Cada intersección genera varios rayos secundarios (luz especular / difusa)
 5. Cada uno de estos rayos secundarios vuelve al punto 2, hasta llegar a un número máximo de rebotes

Trazado de rayos

- Tiempo de cálculo:

$$\text{tinterseccion} * \text{npixels} * \\ (\text{nobjetos} * \text{nsecundarios} * \text{nluces})^{\text{nrebotes}}$$

Oh my god!!!

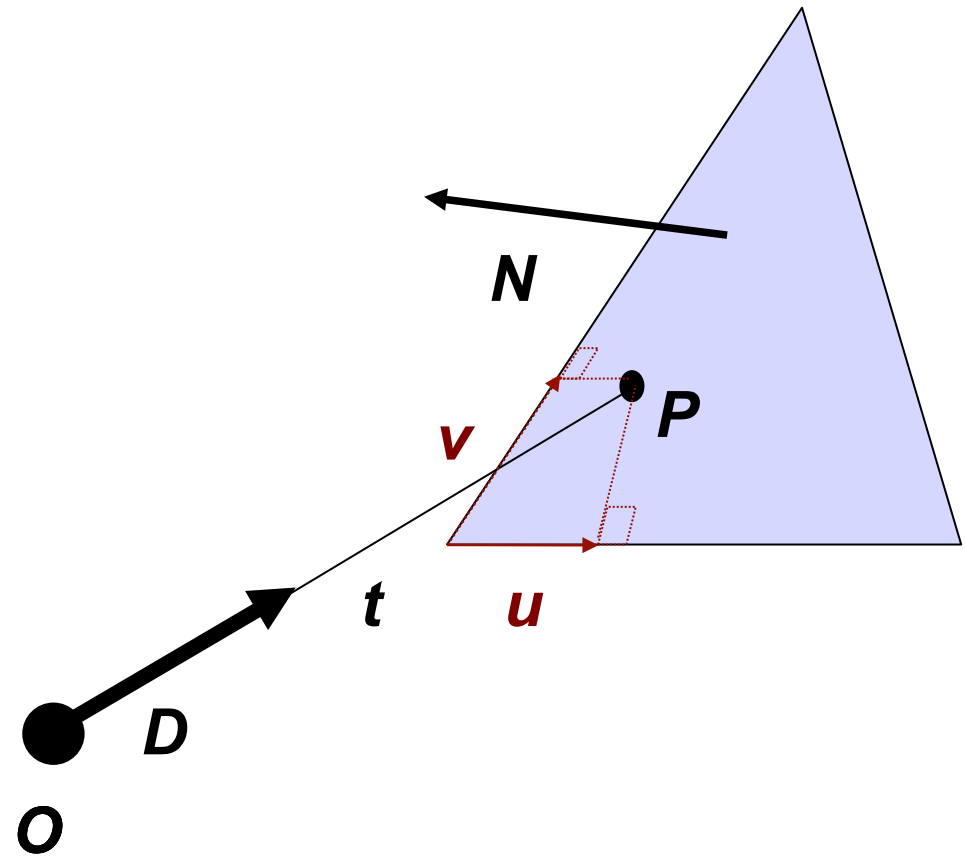
Trazado de rayos

- ¿Cómo reducimos el tiempo de cálculo?
 - Respuesta trivial: reduciendo cada uno de los factores.
 - La realidad: reducir alguno de esos factores no es tarea fácil

$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (n_{\text{objetos}} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos

- Reduciendo tiempo de intersección:
 - Ser cuidadoso con la programación.
 - Reducir accesos a memoria.
 - Programar en ensamblador (SSE)
 - Descargarse código ya eficiente de internet.



$$\text{tinterseccion} * \text{npixels} * \\ (\text{nobjetos} * \text{nsecundarios} * \text{nluces})^{\text{nrebotes}}$$

Trazado de rayos: Anti-aliasing

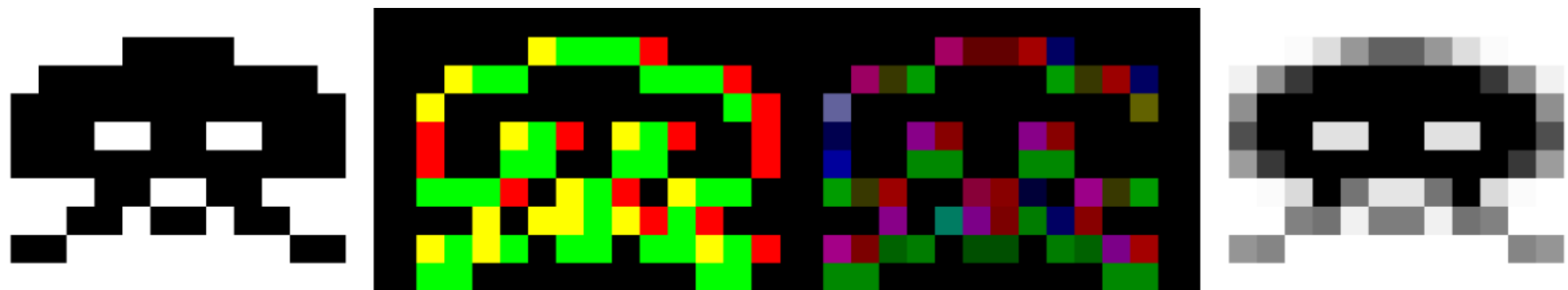
- Reduciendo número de píxeles:
 - Poca resolución: aliasing.
 - Alta resolución: mejor calidad pero más tiempo de cálculo.



$$\text{tinterseccion} * \text{npixels} * (\text{nobjetos} * \text{nsecundarios} * \text{nluces})^{\text{nrebotes}}$$

Trazado de rayos: Anti-aliasing

- Más inteligentemente
 - Jittering – random sampling por pixel.
 - Anti-aliasing adaptativo (por zonas de imagen).
 - Anti-aliasing como post-proceso de imagen (MLAA).



$$\text{tinterseccion} * \text{npixels} * \\ (\text{nobjetos} * \text{nsecundarios} * \text{nluces})^{\text{nrebotes}}$$





Trazado de rayos

- Reduciendo número de objetos:
 - Borrar objetos de la escena.

$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{objetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos

- Reduciendo número de objetos:
 - Borrar objetos de la escena.
 - Bueno, si no se ven...

$$t_{\text{interseccion}} * n_{\text{pixels}} * (\text{nobjetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos

- Reduciendo número de objetos:

- Borrar objetos de la escena.

- Bueno, si no se ven...

- Ideas mejores:

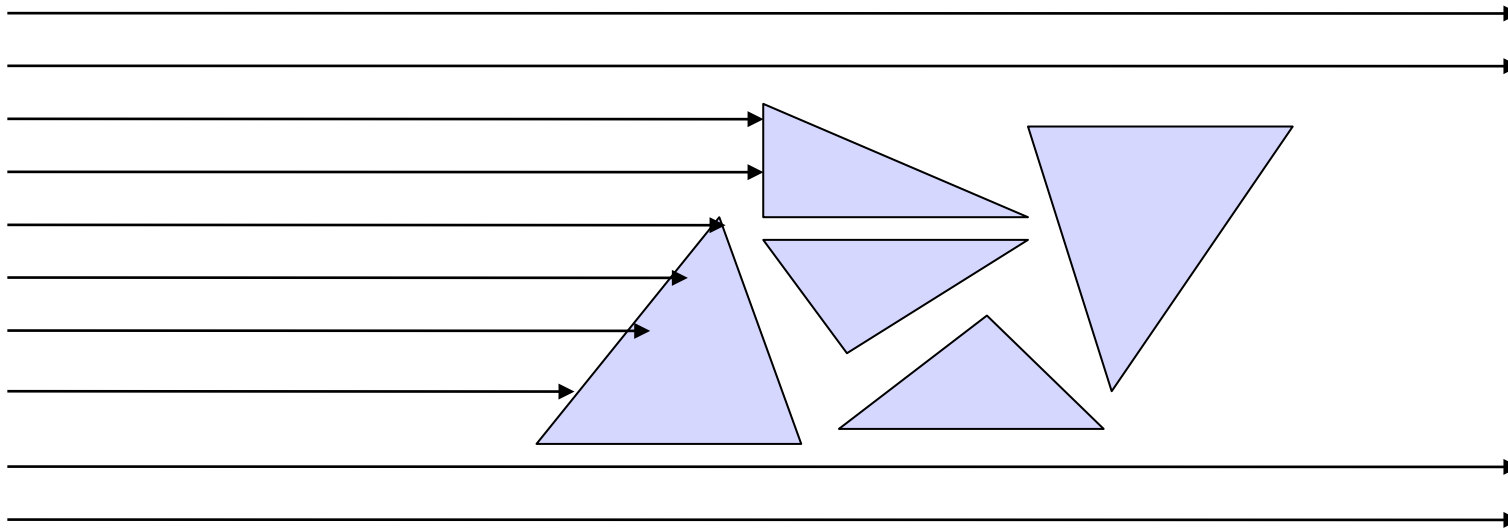
- BVH (Bounding Volume Hierarchies)

- Octree, kd-tree, ...

$$t_{\text{interseccion}} * n_{\text{pixels}} * (n_{\text{objetos}} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: BVH

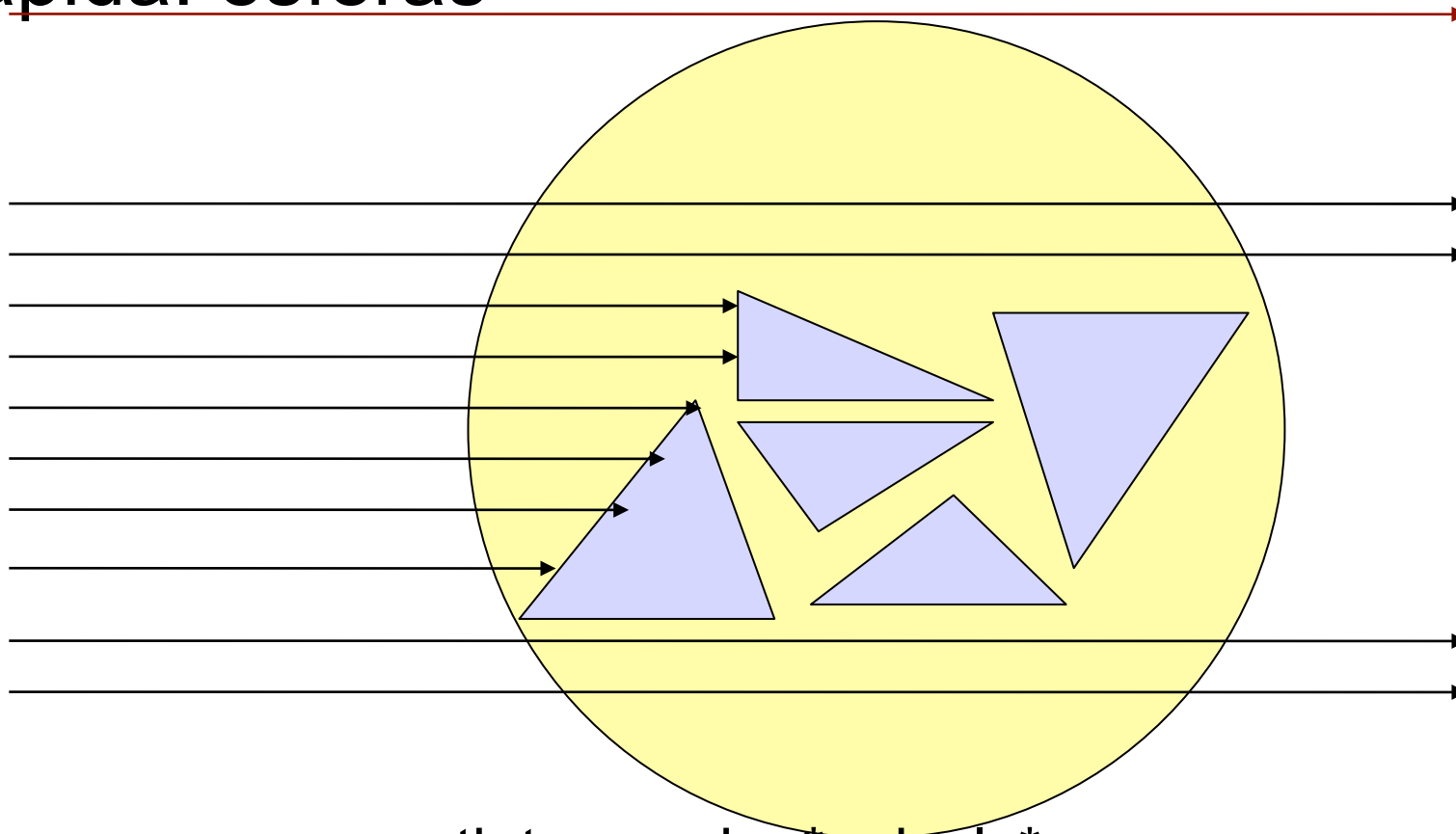
- Idea: un rayo no intersecta con ningún polígono de cierta geometría si no intersecta con un “bounding volume” de esta



$$t_{\text{interseccion}} * n_{\text{pixels}} * (\text{nobjetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: BVH

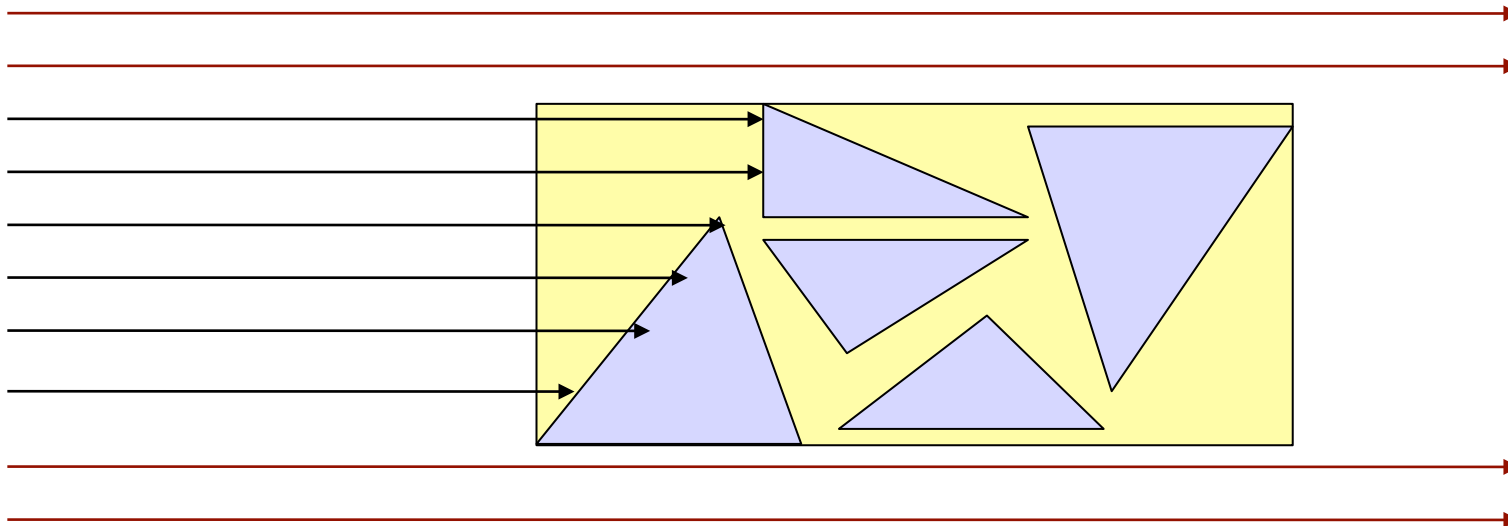
- Utilizar “bounding volumes” de intersección rápida: esferas



$t_{interseccion} * npixels *$
 $(n_{objetos} * n_{secundarios} * n_{luces})^{n_{rebotes}}$

Trazado de rayos: BVH

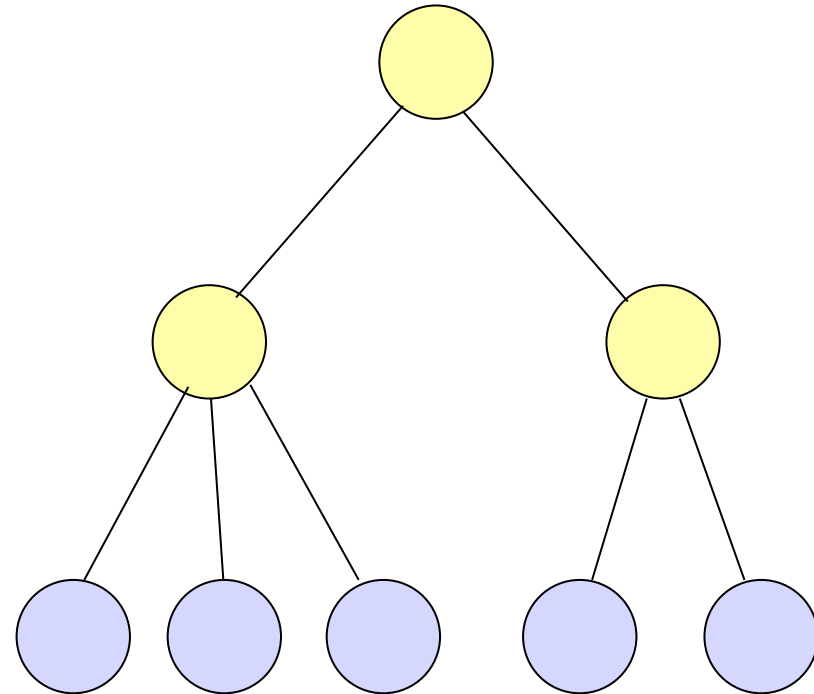
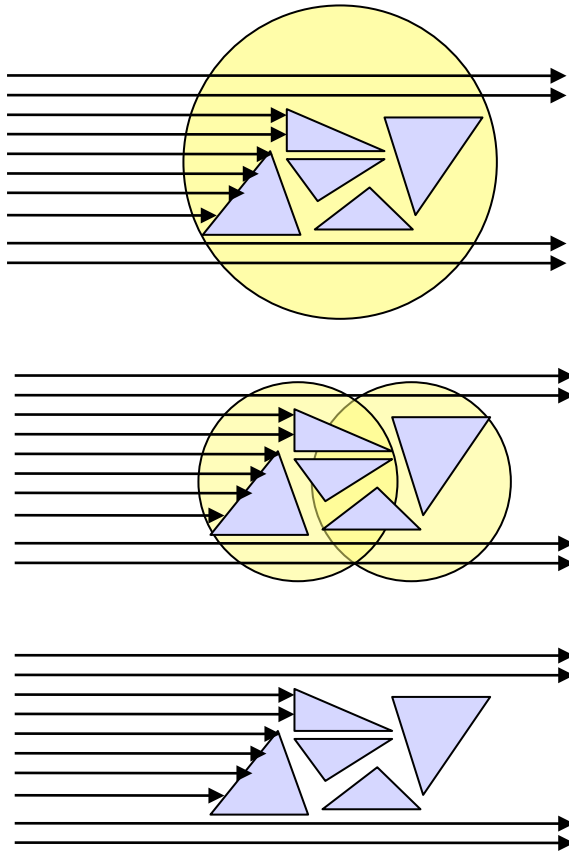
- Utilizar “bounding volumes” de intersección rápida: cajas alineadas con los ejes



$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{nobjetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: BVH

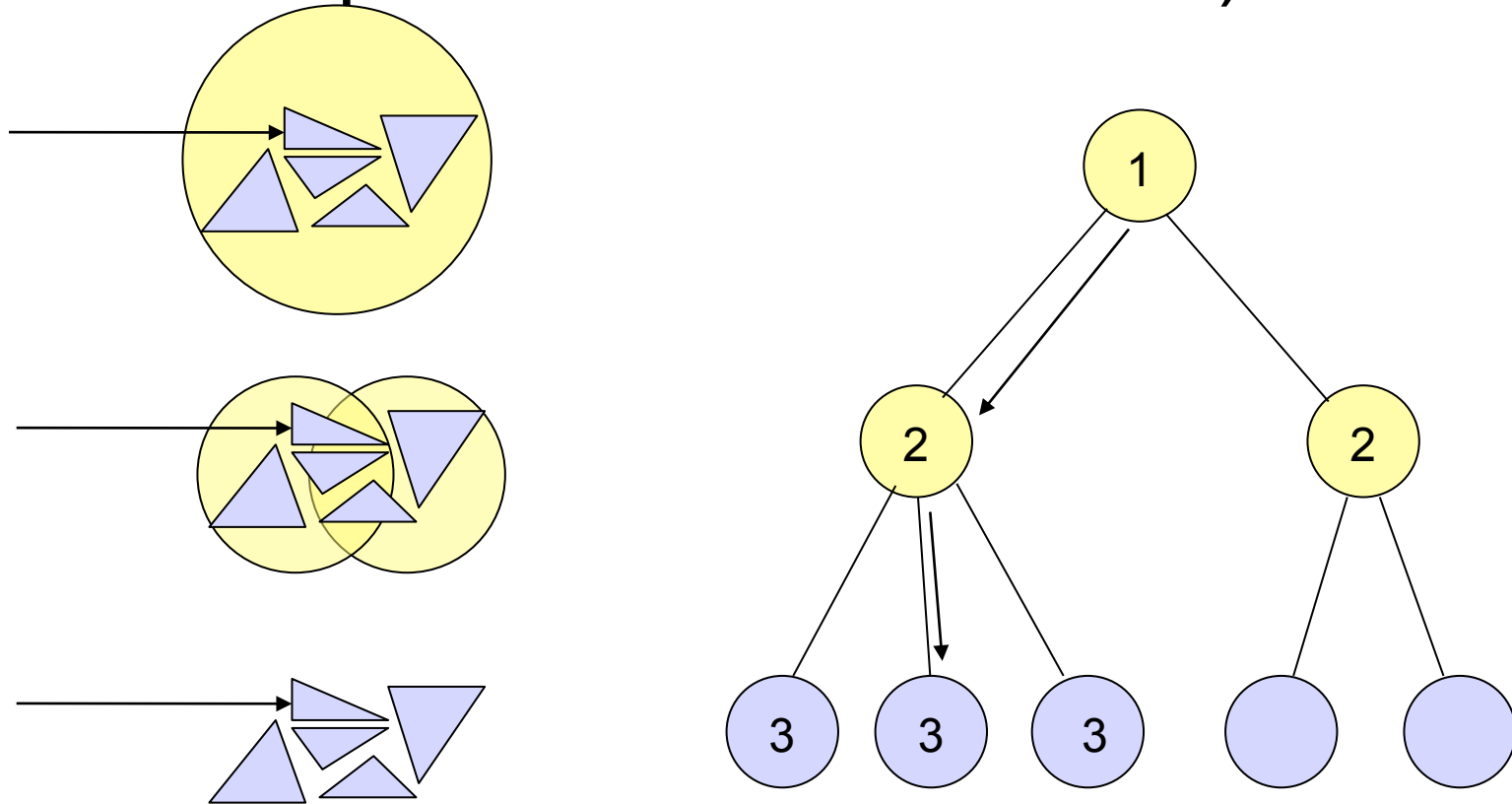
- Estructura jerárquica: árbol



$$t_{\text{interseccion}} * n_{\text{pixels}} * (\text{objetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: BVH

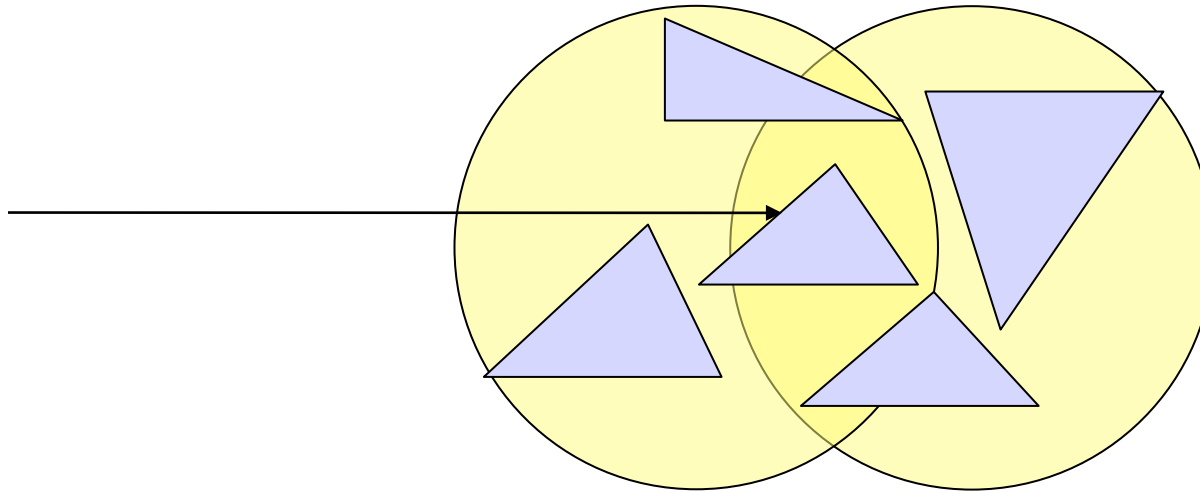
- Estructura jerárquica: búsqueda (interceptar y luego buscar primero el más cercano)



$t_{interseccion} * n_{pixels} * (n_{objetos} * n_{secundarios} * n_{luces})^{n_{rebotes}}$

Trazado de rayos: BVH

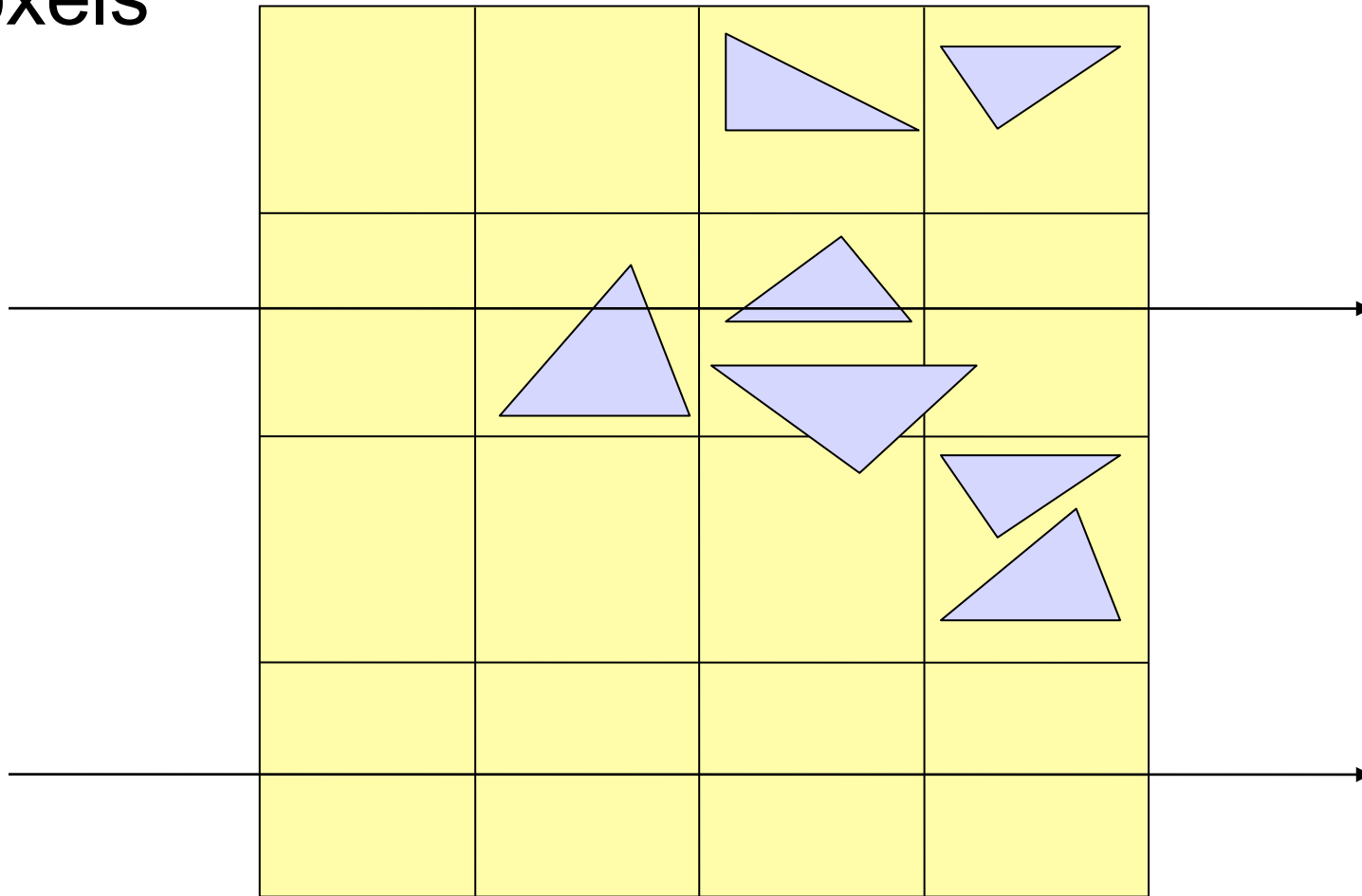
- ¿Problemas?
 - Areas de intersección.
 - A veces toca comprobar el otro nodo, incluso habiendo encontrado una intersección.



$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{nobjetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: Partición espacial

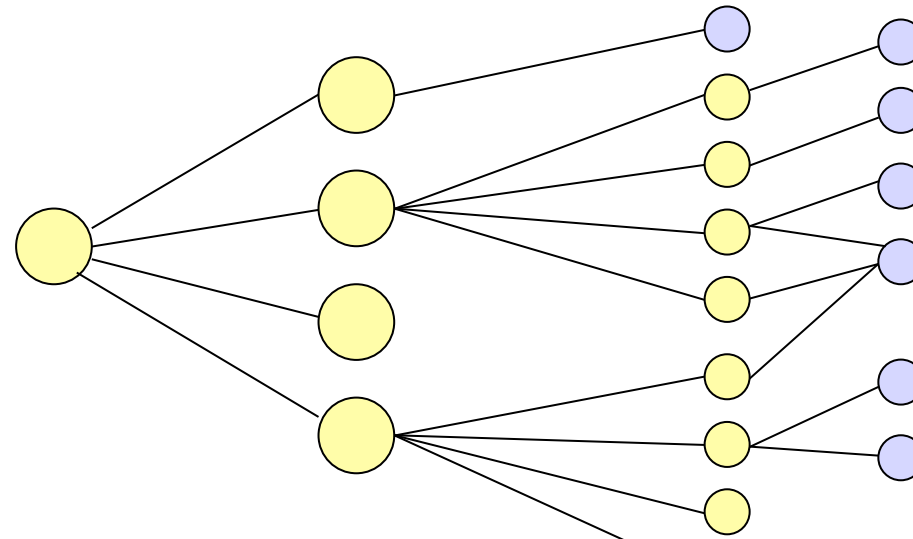
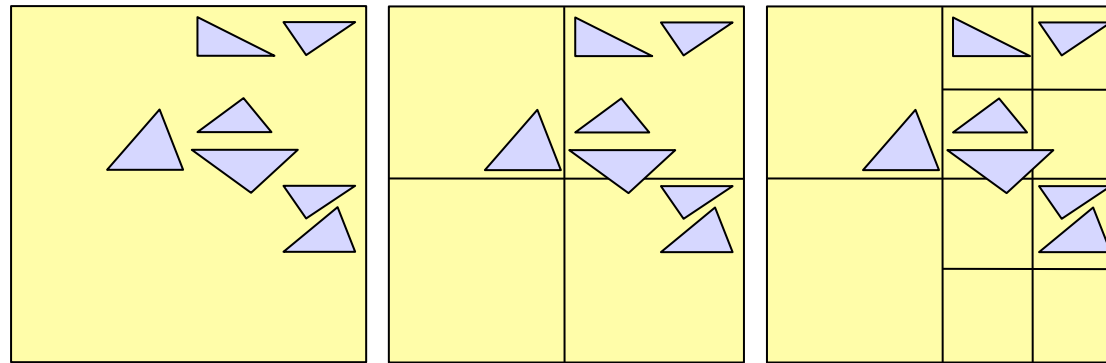
- Voxels



$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{objetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: Partición espacial

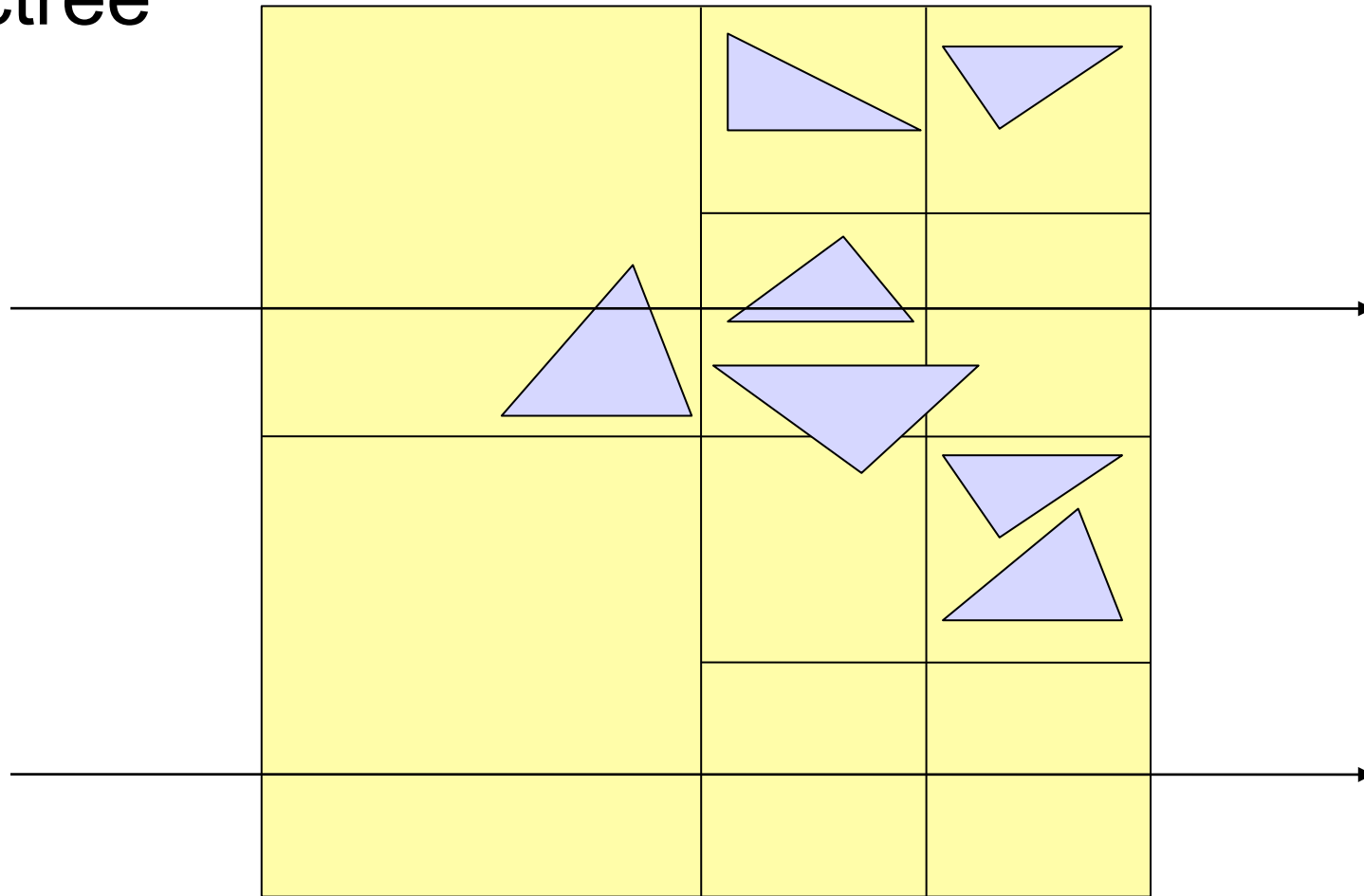
- Octree (voxels jerárquicos)



$$t_{\text{interseccion}} * n_{\text{pixeles}} * (\text{nobjetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: Partición espacial

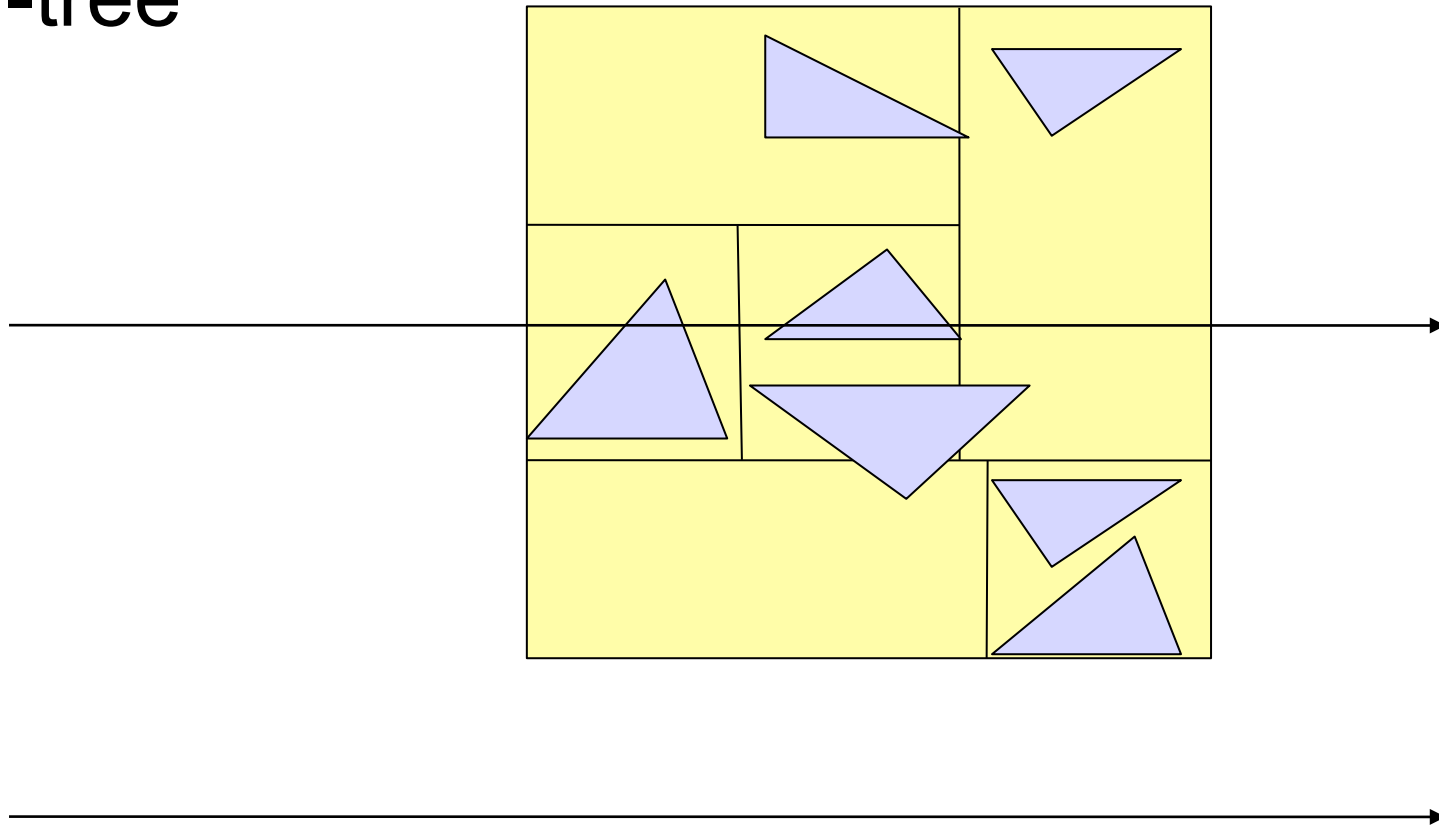
- Octree



$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{objetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: Partición espacial

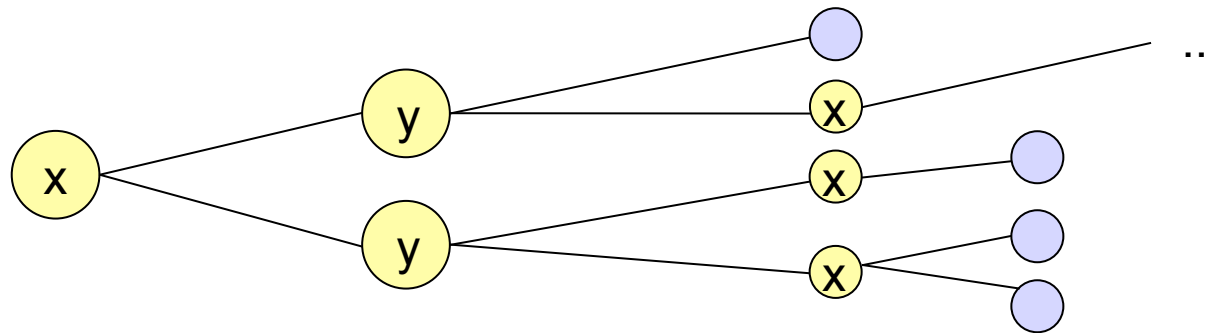
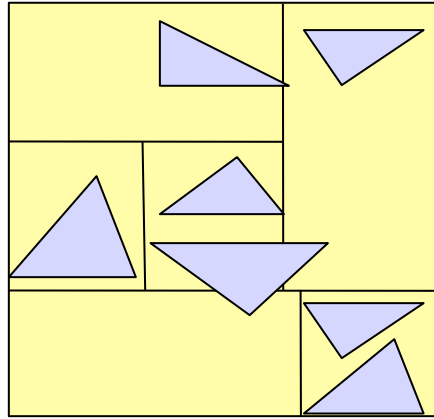
- Kd-tree



$$t_{\text{interseccion}} * n_{\text{pixels}} * (\text{objetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos: Partición espacial

- Kd-tree

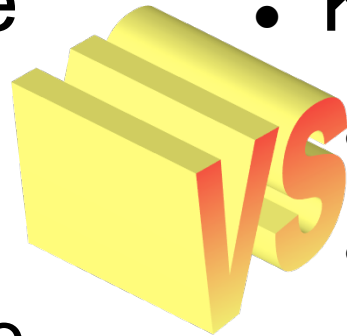
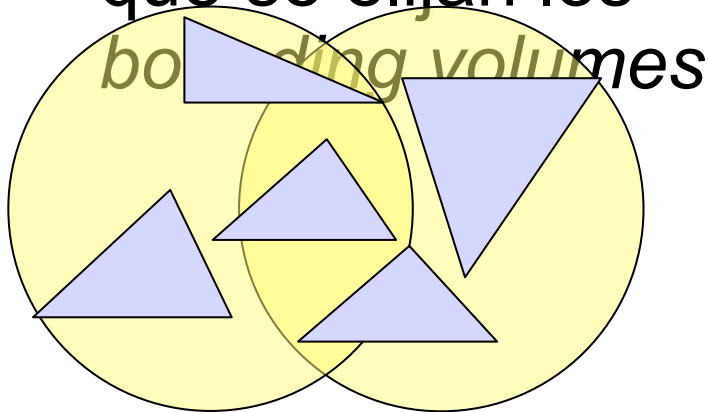


$$t_{\text{interseccion}} * n_{\text{pixels}} * (\text{nobjetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos

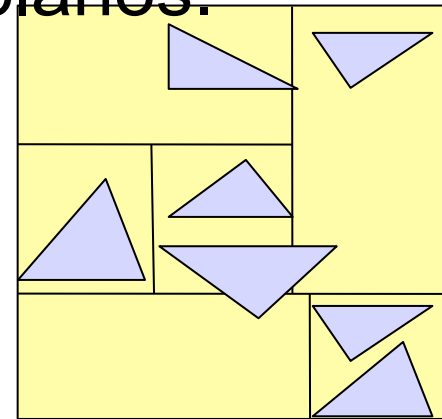
- Bounding Volume Hierarchy

- Area compartida.
- Acceso más lento.
- Depende de lo bien que se elijan los



- Kd-tree

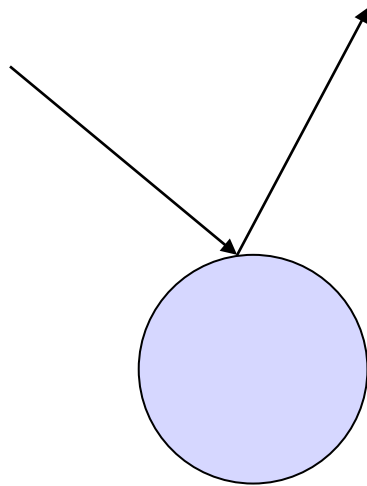
- Objetos duplicados.
- Programación difícil.
- Depende de lo bien que se elijan los planos.



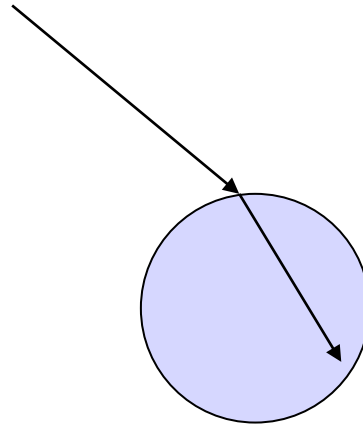
$$t_{\text{interseccion}} * n_{\text{pixels}} * (\text{objetos} * n_{\text{secundarios}} * n_{\text{luces}})^{n_{\text{rebotes}}}$$

Trazado de rayos

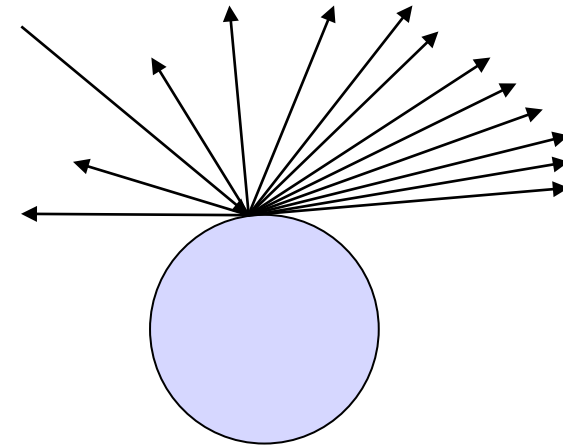
- Reduciendo el número de rayos secundarios



REFLEXIÓN
PERFECTA



REFRACCIÓN
PERFECTA

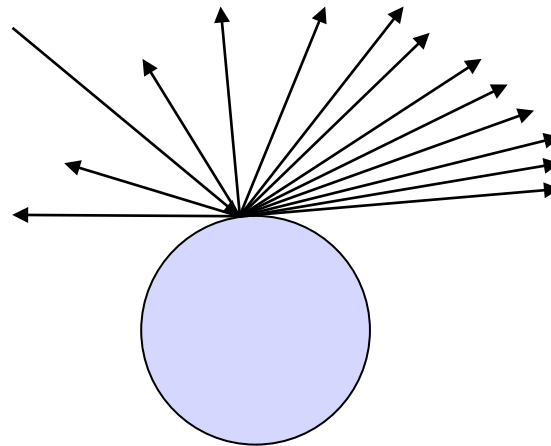


BRDF DIFUSA
GENÉRICA
(INTERREFLEXIÓN)

$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{nobjetos} * \mathbf{n_{\text{secundarios}}} * n_{\text{luces}})^{\mathbf{n_{\text{rebotes}}}}$$

Trazado de rayos

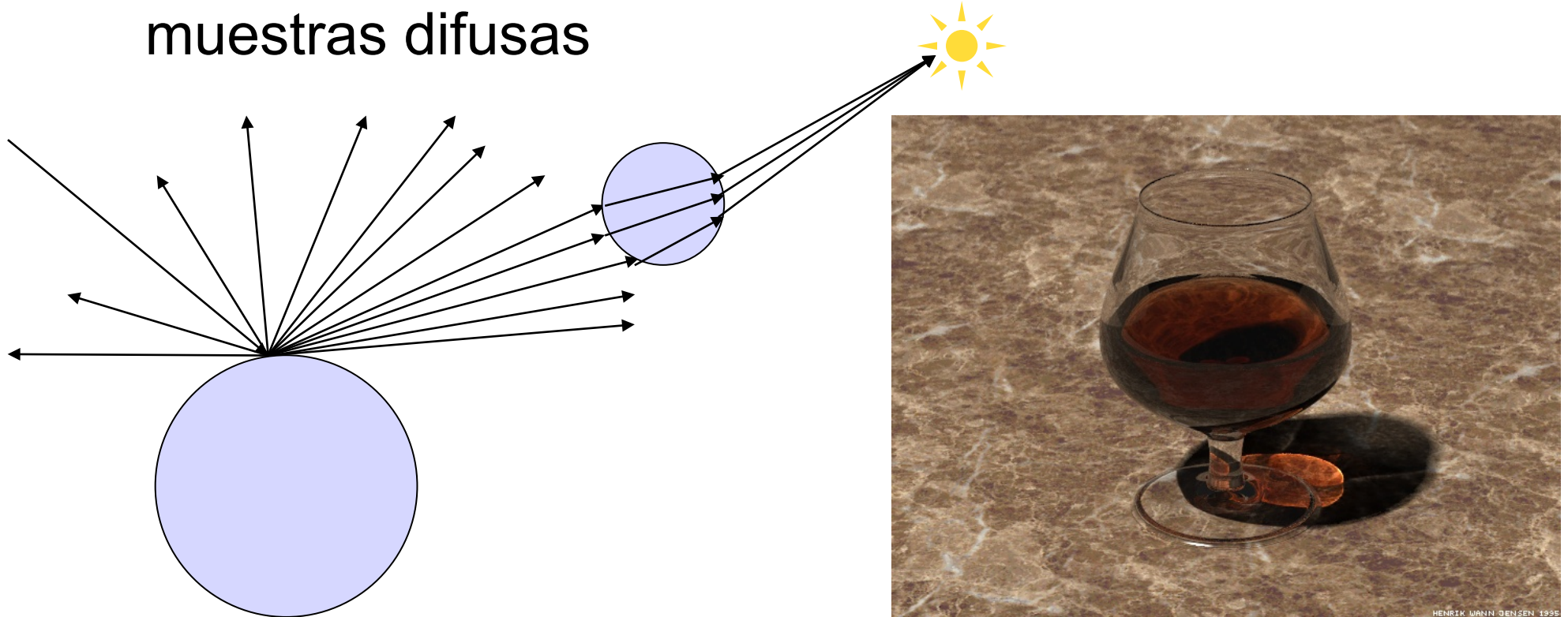
- Idea: elegir las mejores muestras, o las más importantes.
- Técnicas de Montecarlo (más adelante en el temario)



$$\text{tinterseccion} * \text{npixels} * \\ (\text{nobjetos} * \text{nsecundarios} * \text{nluces})^{\text{nrebotes}}$$

Trazado de rayos: color

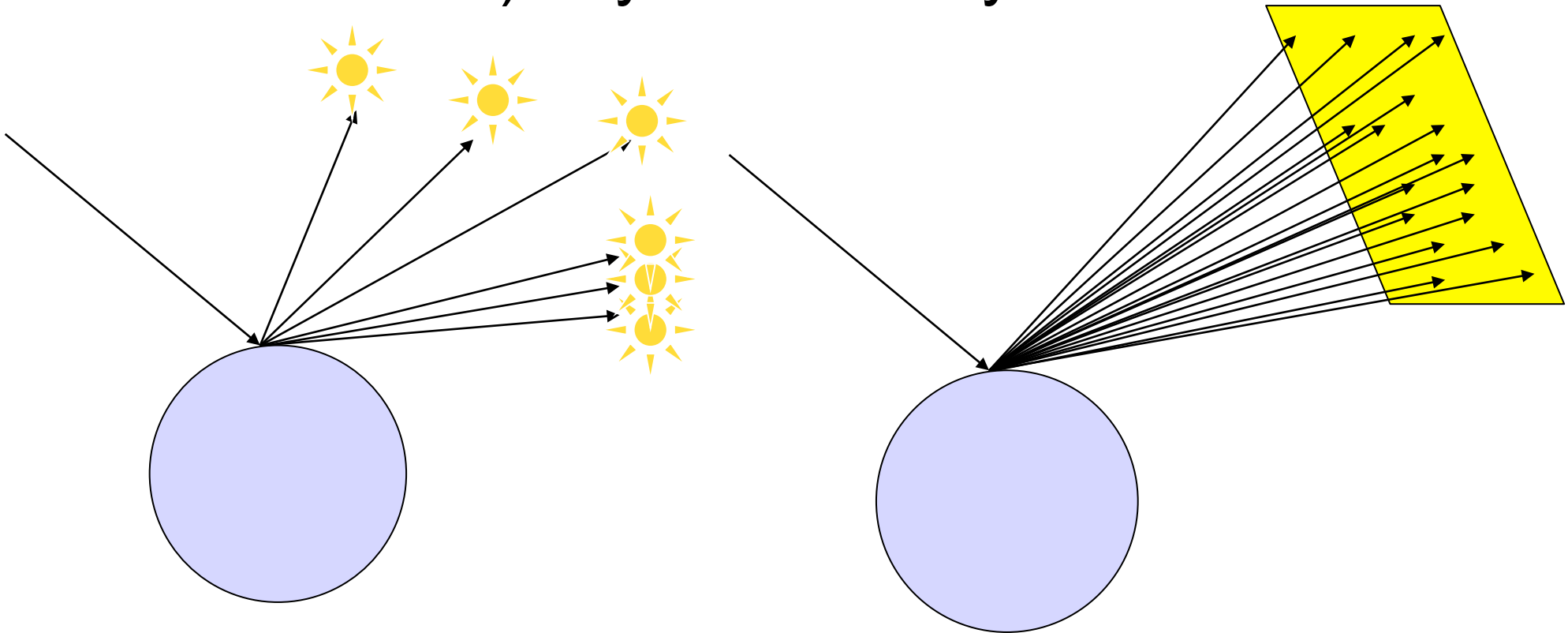
- Aún eso: problemas con cáusticas
 - Caminos difíciles de encontrar, requieren muchas muestras difusas



$$\text{tinterseccion} * \text{npixels} * \\ (\text{nobjetos} * \text{nsecundarios} * \text{nluces})^{\text{nrebotes}}$$

Trazado de rayos

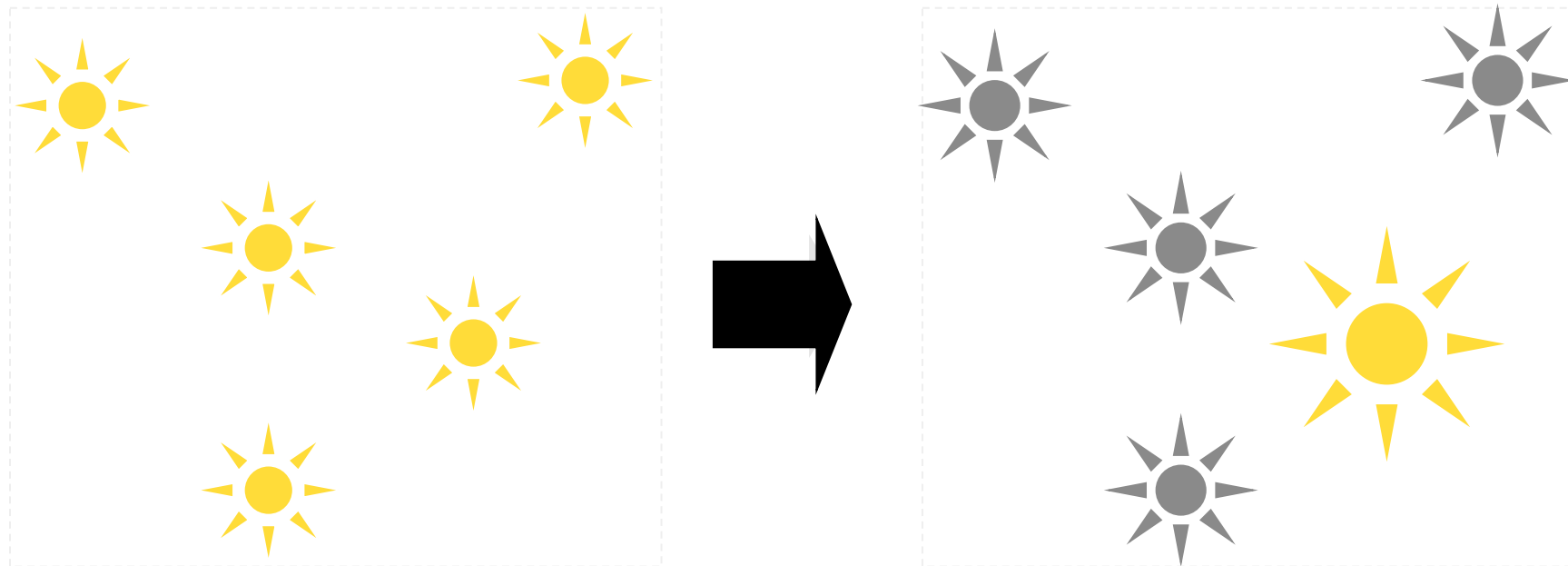
- Si hay muchas luces (o luces de área muestreadas) hay muchos rayos de sombra.



$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{nobjetos} * n_{\text{secundarios}} * \mathbf{n_{\text{luces}}})^{\mathbf{n_{\text{rebotes}}}}$$

Trazado de rayos: Lightcuts

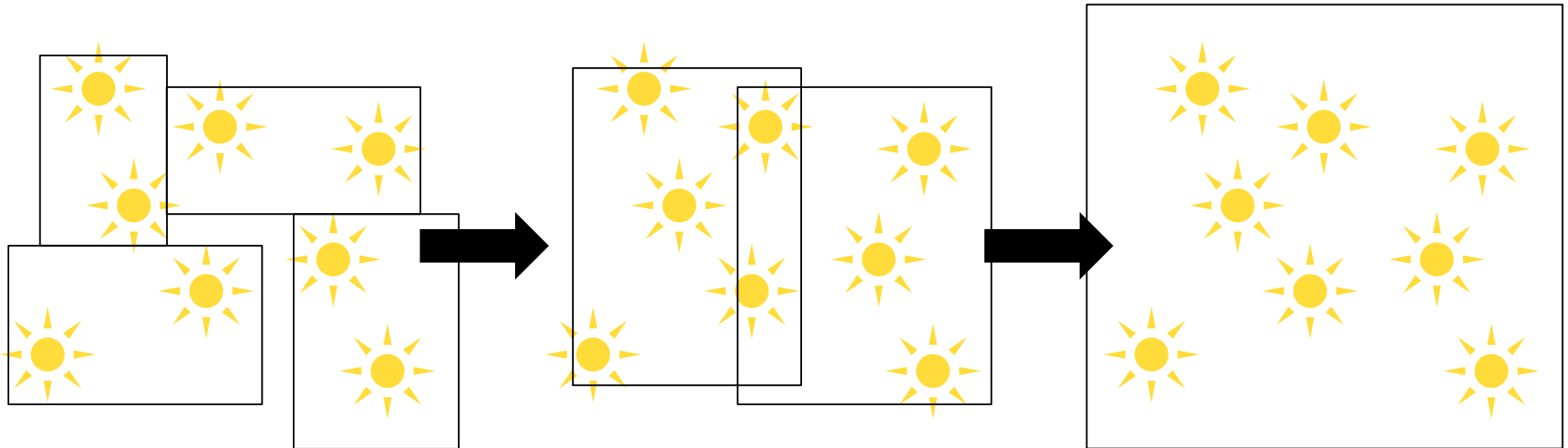
- Idea: Agrupar luces cercanas que aportan radiancias similares (*clusters*)



$$\text{tinterseccion} * \text{npixels} * \\ (\text{nobjetos} * \text{nsecundarios} * \mathbf{n\text{luces}})^{\text{nrebotes}}$$

Trazado de rayos: Lightcuts

- Idea mejor: Agrupación jerárquica.
- Según el punto, elegir una u otra subdivisión.



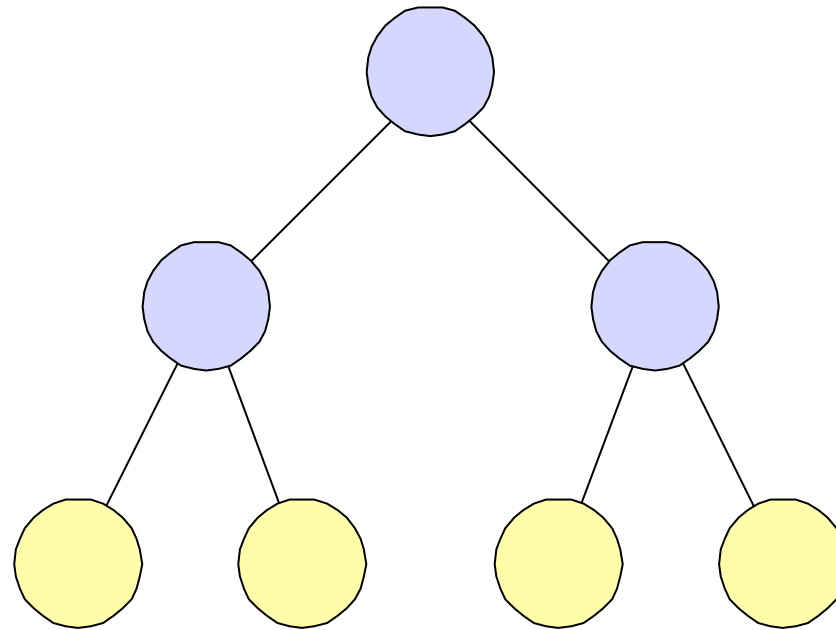
$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{nobjetos} * n_{\text{secundarios}} * \mathbf{n_{\text{luces}}})^{\mathbf{n_{\text{rebotes}}}}$$

Trazado de rayos: Lightcuts

- Estructura de árbol

Clusters

Luces
individuales



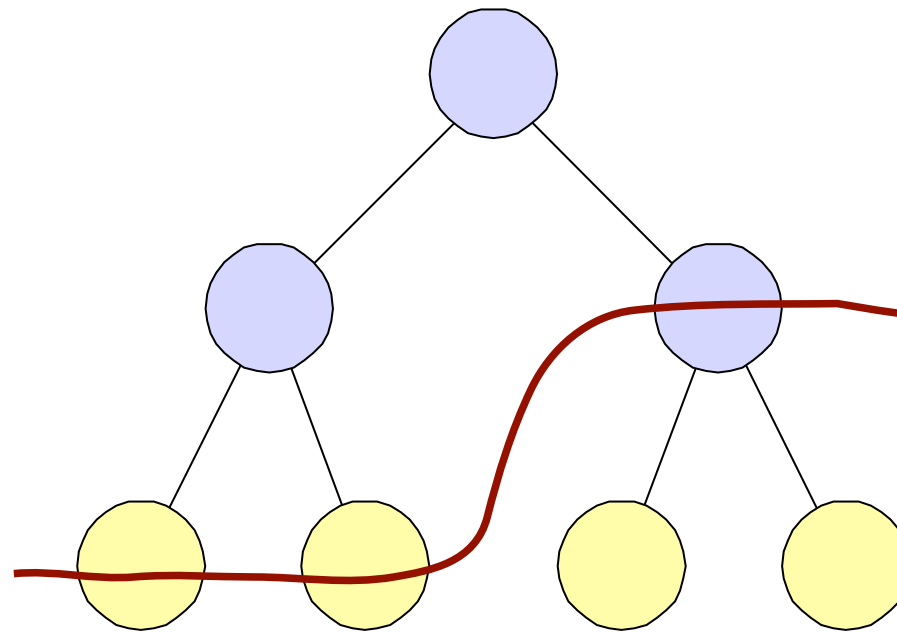
$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{nobjetos} * \text{nsecundarios} * \mathbf{n_{\text{luces}}})^{\text{nrebotes}}$$

Trazado de rayos: Lightcuts

- Para cada punto iluminado buscamos el corte óptimo en el árbol: *lightcut*

Clusters

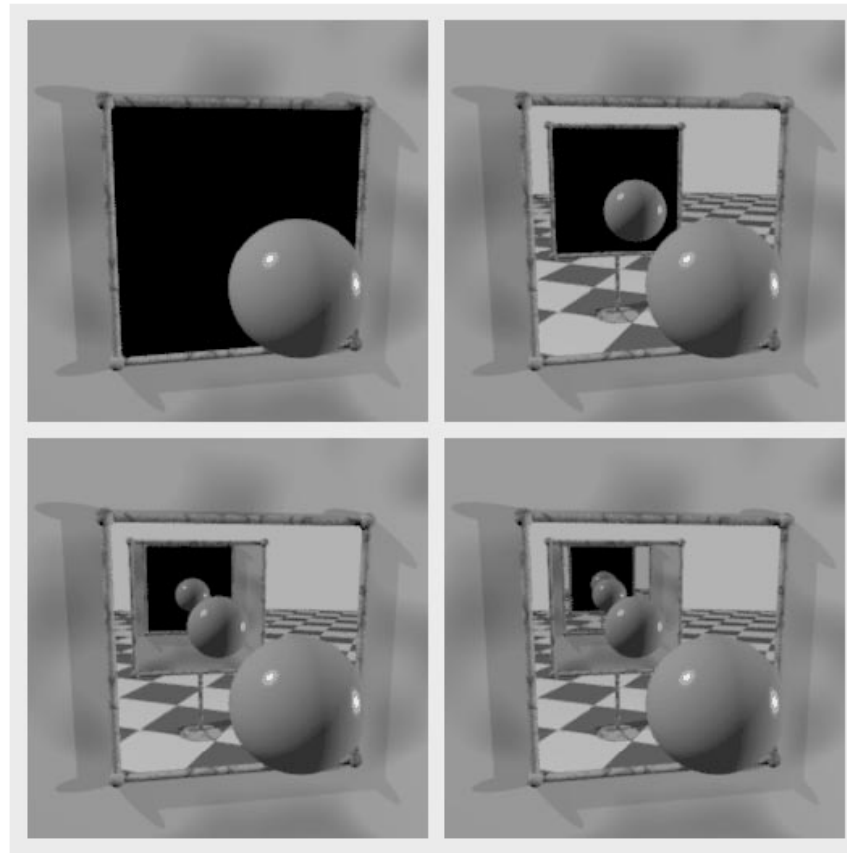
Luces
individuales



$$t_{\text{interseccion}} * n_{\text{pixels}} * \\ (\text{nobjetos} * n_{\text{secundarios}} * \mathbf{n_{\text{luces}}})^{\text{nrebotes}}$$

Trazado de rayos

- Elegir el número de rebotes óptimo no es tarea fácil...



$$\text{tinterseccion} * \text{npixels} * \\ (\text{nobjetos} * \text{nsecundarios} * \text{nluces})^{\text{nrebotes}}$$

Trazado de rayos

- Próximamente...
 - Reducimos varias magnitudes del tiempo a la vez (nuevos algoritmos)
 - Aprendemos a pintar cáusticas.
 - Aprendemos Montecarlo.
 - Aprendemos a renderizar medios participativos.

Trazado de rayos

