



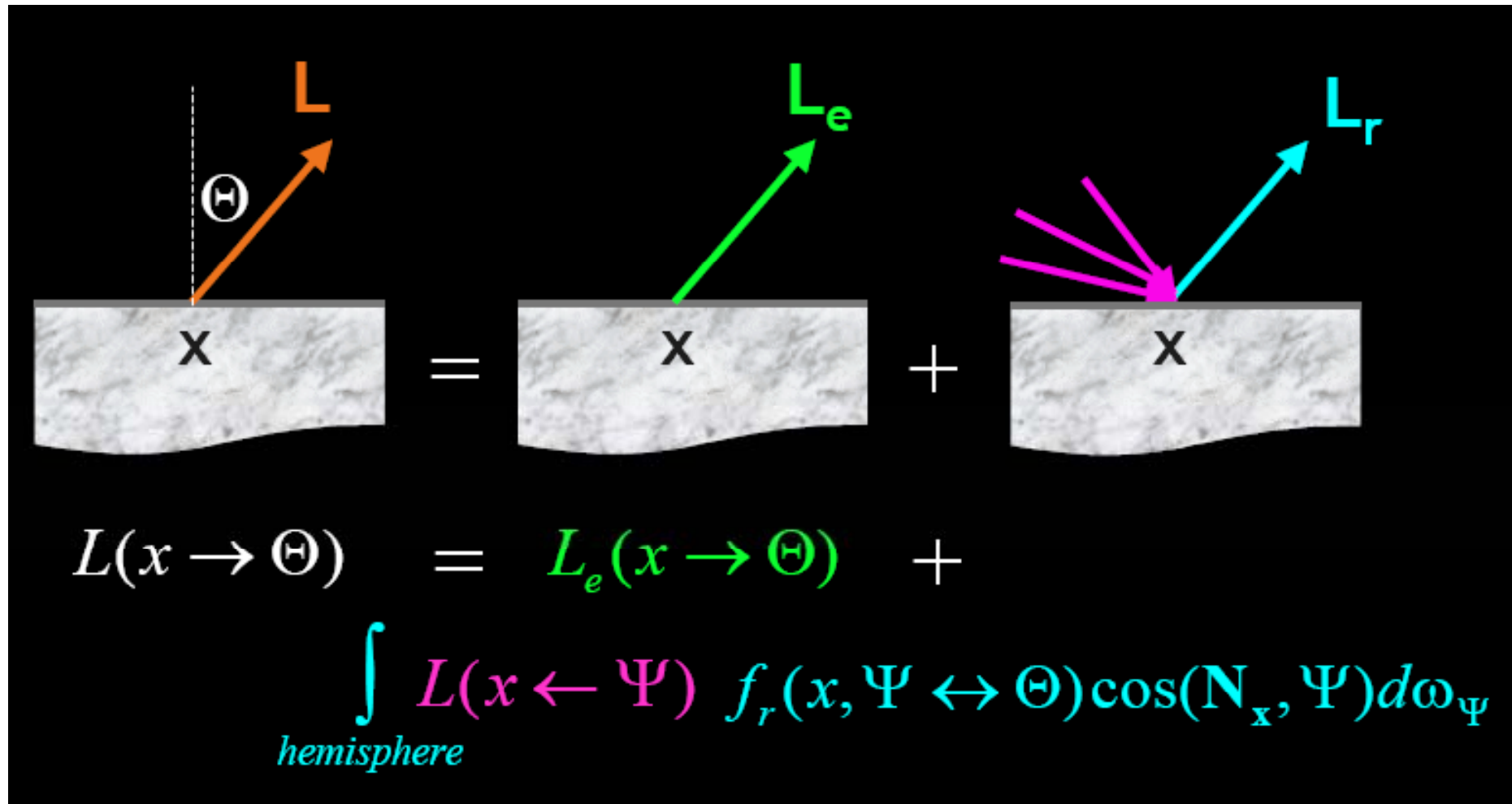
Traza de Rayos

Técnicas Avanzadas de Gráficos en 3D

Miguel Ángel Otaduy
15 Marzo 2010



Ecuación de Rendering



Ecuación de rendering completa

Traza de Rayos Básica

- Cálculo de la luz reflejada
 - a) Aproximar integral de la ecuación de rendering
 - b) Utilizar sólo rayos a las luces puntuales y rayo de reflexión especular
- Trazado de rayos recursivo: realizar lo mismo por cada rayo especular



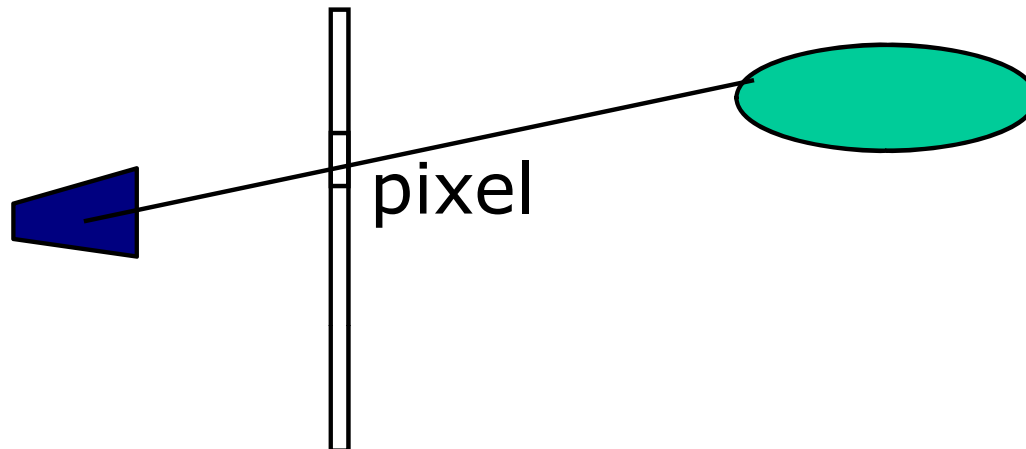
Traza de Rayos Básica

- Algoritmo de trazado recursivo de Whitted:
 - Se genera un árbol de rayos, que termina con una profundidad marcada por el usuario o cuando la energía es muy pequeña
 - Se evalúa el color en las hojas, y luego se asciende por el árbol



Traza de Rayos Básica

- Envío de rayos:
 - Desde la cámara, pasando por un pixel, hasta la intersección en la escena.
 - El plano de imagen puede estar en cualquier posición, y no hacen falta planos cercano y lejano como en OpenGL.

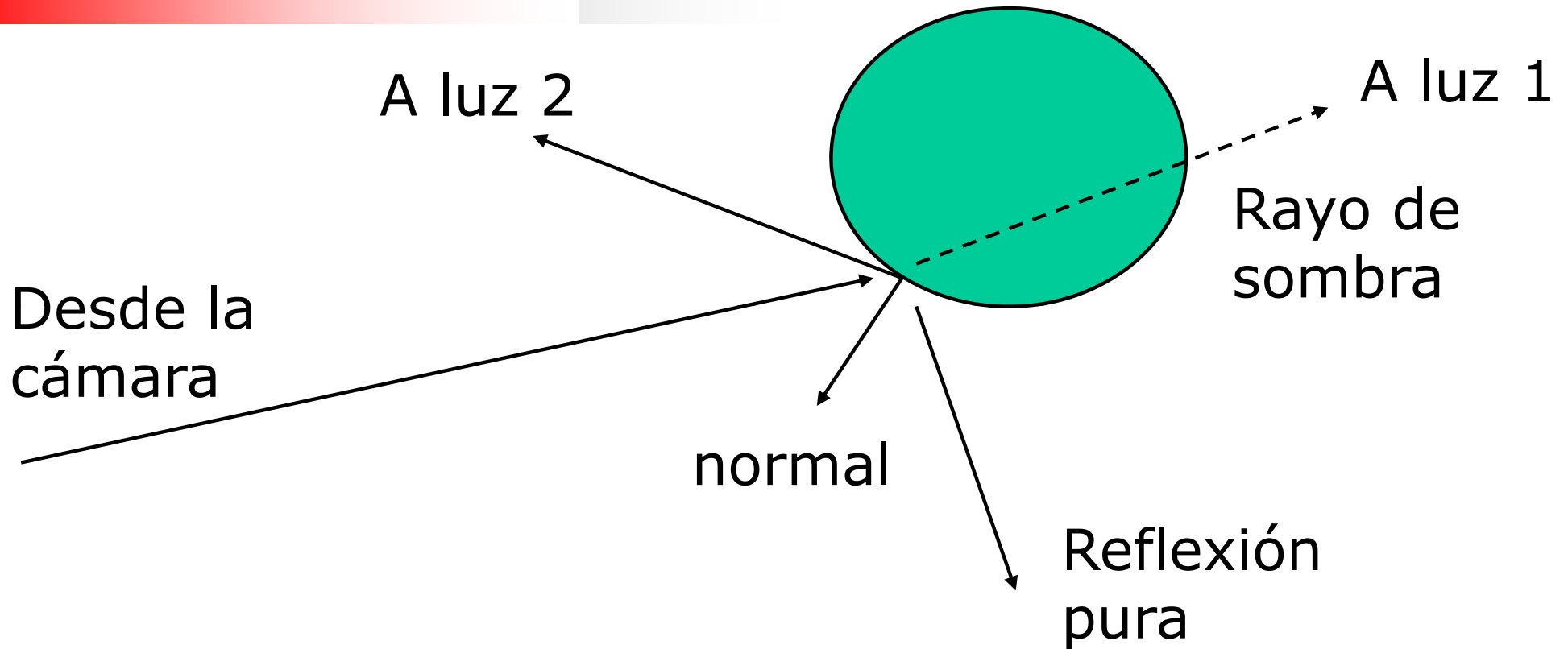


Traza de Rayos Básica

- **Árbol de rayos:**
 - Desde un punto de intersección en la escena, se envían rayos a las luces y el rayo de reflexión pura (si el material es reflectante).
 - Por cada rayo incidente, se evalúa un modelo de iluminación (p.ej. Phong), tomando la luz del rayo incidente y las propiedades del material, y se suman todas las contribuciones.



Traza de Rayos Básica



$$\text{Luz rayo} = \omega_{\text{reflexión}} * \text{Phong}(\text{reflexión}) + \text{Phong}(\text{rayo luz}_1) + \text{Phong}(\text{rayo luz}_2)$$



Traza de Rayos Básica

- Comparación de la traza de rayos básica con la Ec. de rendering:
 - Se aproxima la integral tomando sólo la contribución de unos pocos rayos (los de luz y la reflexión pura), pues se supone que esos rayos contribuyen la mayor parte de la intensidad.
 - La aproximación suele ser buena, pero se omiten los efectos de luz indirecta, que dan realismo.



Intersección de Rayos

- Descripción de la escena:
 - Mediante primitivas algebraicas: esferas, conos, planos... o sus combinaciones (CSG)
 - Mediante mallas de triángulos
- Intersecciones habituales: rayo-plano, rayo-esfera, rayo-triángulo



Rayo-Plano

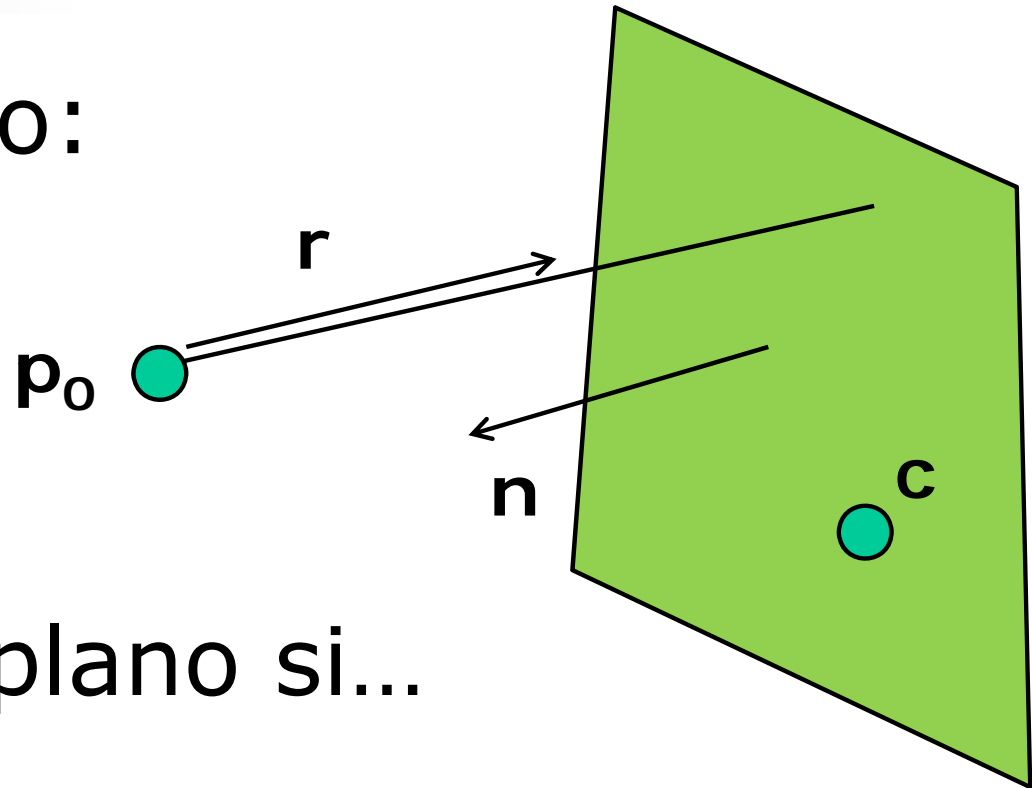
- Parametrizar el rayo
- Ecuación del plano
- ¿Se cumple la ecuación del plano para algún valor del rayo?
- Nota: sólo se admiten valores positivos del parámetro



Rayo-Plano

Ecuación del rayo:

$$\mathbf{p} = \mathbf{p}_0 + t \mathbf{r}$$



El rayo cruza el plano si...

$$\mathbf{n}^T (\mathbf{p} - \mathbf{c}) = 0$$

Rayo-Esfera

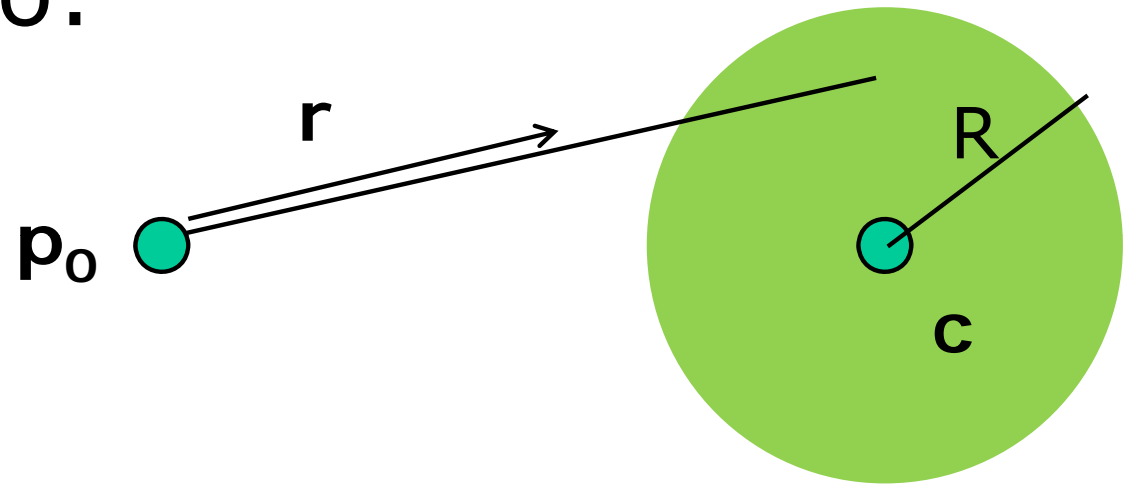
- Parametrizar el rayo
- Ecuación de la esfera
- ¿Se cumple la ecuación de la esfera para algún valor del rayo?
- Nota: mirar los valores del discriminante; 0, 1, 2 soluciones



Rayo-Esfera

Ecuación del rayo:

$$\mathbf{p} = \mathbf{p}_0 + t \mathbf{r}$$



El rayo cruza el plano si...

$$(\mathbf{p} - \mathbf{c})^T(\mathbf{p} - \mathbf{c}) - R^2 = 0$$

Rayo-Triángulo

- Como en el caso del plano, pero se ha de mirar si la intersección entra dentro del triángulo, empleando coordenadas baricéntricas



Rayo-Triángulo

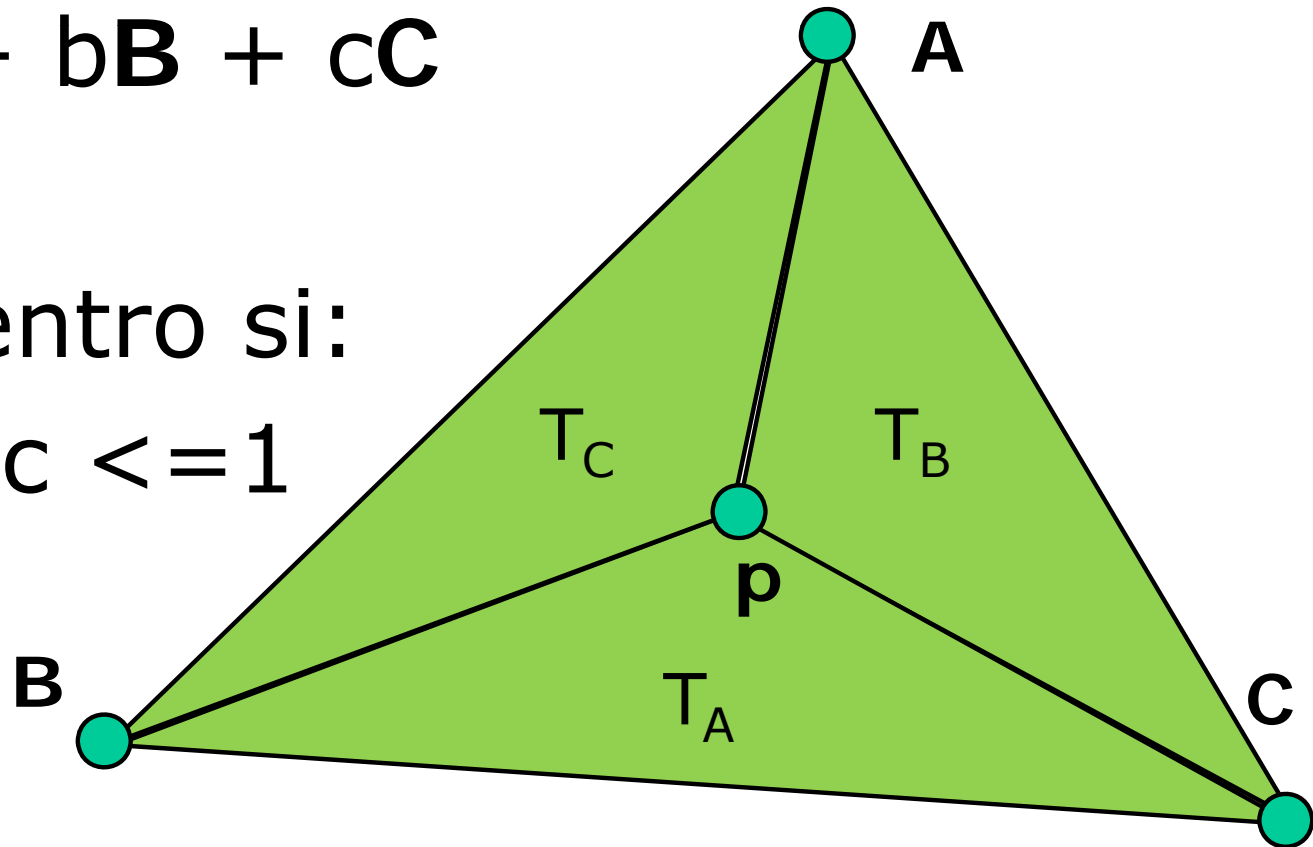
Coordenadas baricéntricas:

$$a = \text{area}(T_A) / \text{area}(T_A + T_B + T_C)$$

$$\mathbf{p} = a\mathbf{A} + b\mathbf{B} + c\mathbf{C}$$

\mathbf{p} está dentro si:

$$0 \leq a, b, c \leq 1$$



Complejidad de Ray Casting

- Ejercicio: Cuantificar la complejidad de ray casting
 1. Factores que influyen
 2. Función de dependencia



Complejidad de Ray Casting

1. Factores que influyen

- Número de rayos (pixels)
- Número de objetos
- Número de triángulos de los objetos
- Número de luces

2. Función de dependencia

- Coste por pixel
- Coste por triángulo
- ...



Complejidad de Ray Tracing

1. Gran número de intersecciones rayo – primitiva → se requieren métodos de aceleración (particiones espaciales, volúmenes envolventes)
2. Decisión de qué rayos emitir, para eliminar problemas de ruido → problema de muestreo (Montecarlo)

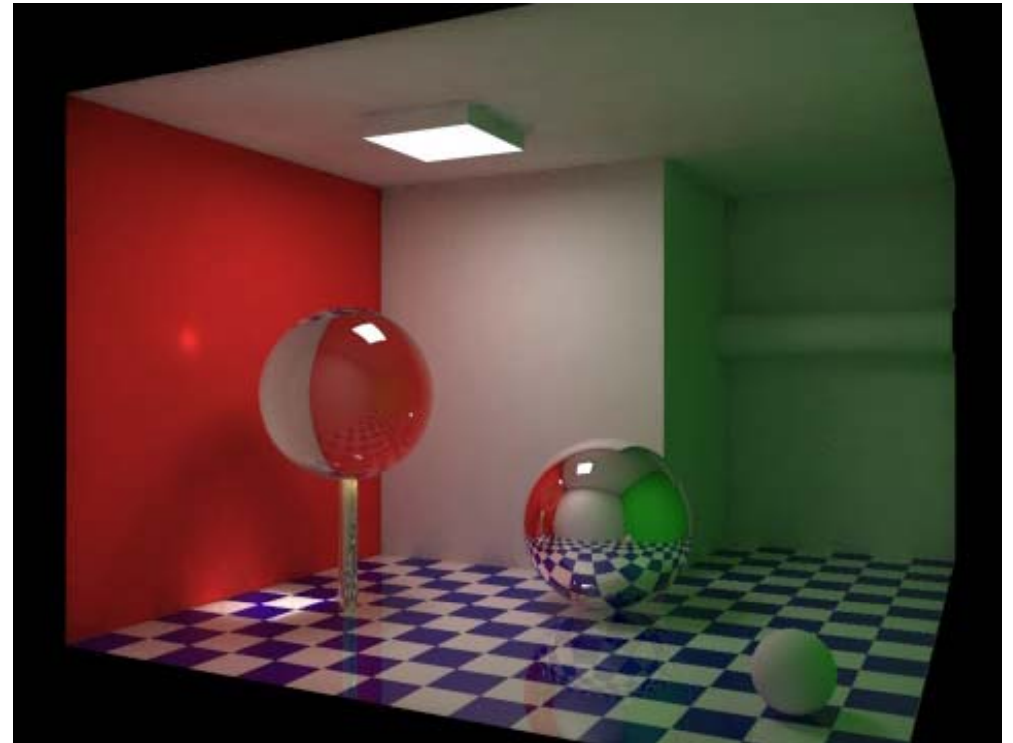
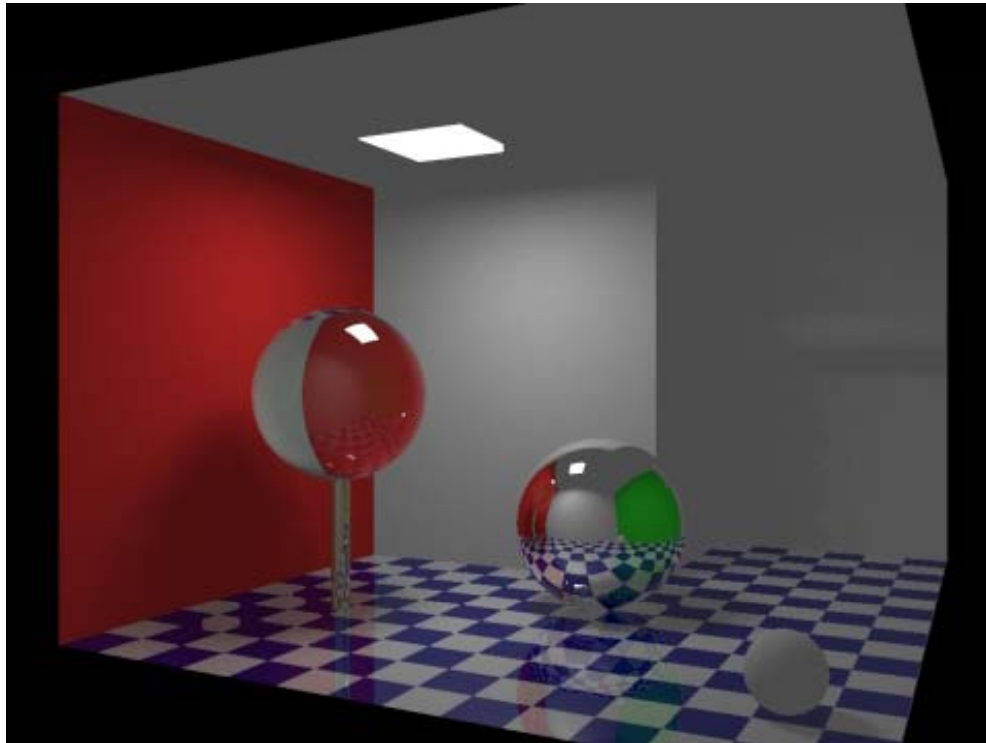


Traza de Rayos

- ¿Qué pasa con luces de área o en casos más complejos?
 - Traza de rayos múltiples
 - Traza de caminos, basada en la Ec. de transporte
 - Traza de rayos bidireccional
 - ...



¿Por qué Iluminación Global?



Traza de Caminos

Traza de rayos: define la radianza en la imagen utilizando un árbol de rayos.

Traza de caminos: el objetivo es definir la radianza como una integral de caminos de rayos (no un árbol!)

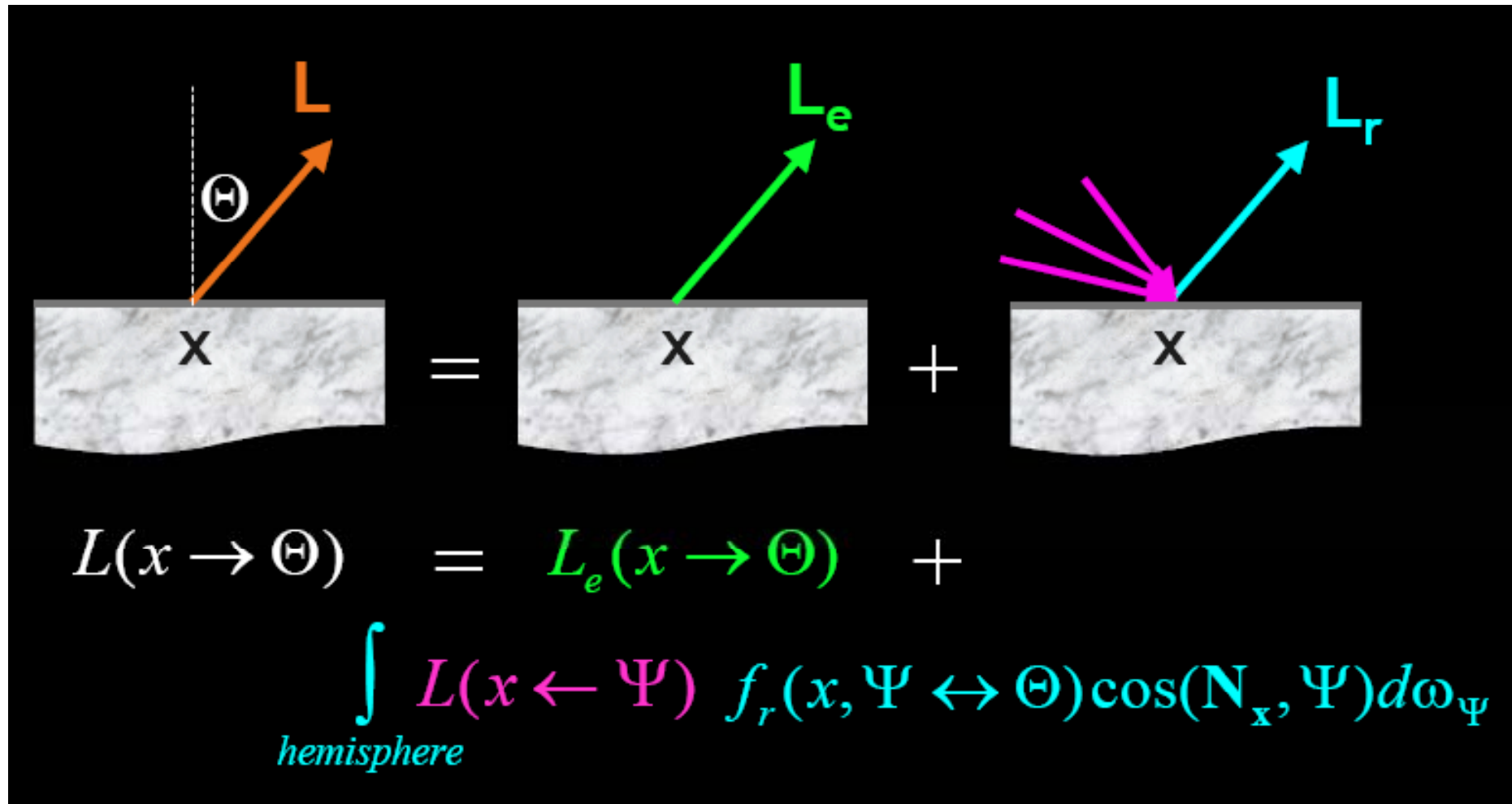


Traza de Caminos

Partiremos de la Ec. de rendering, y pasaremos a escribir una ec. que nos define la radianza en toda la escena. Luego la interpretaremos...



Ecuación de Rendering



Ecuación de rendering completa

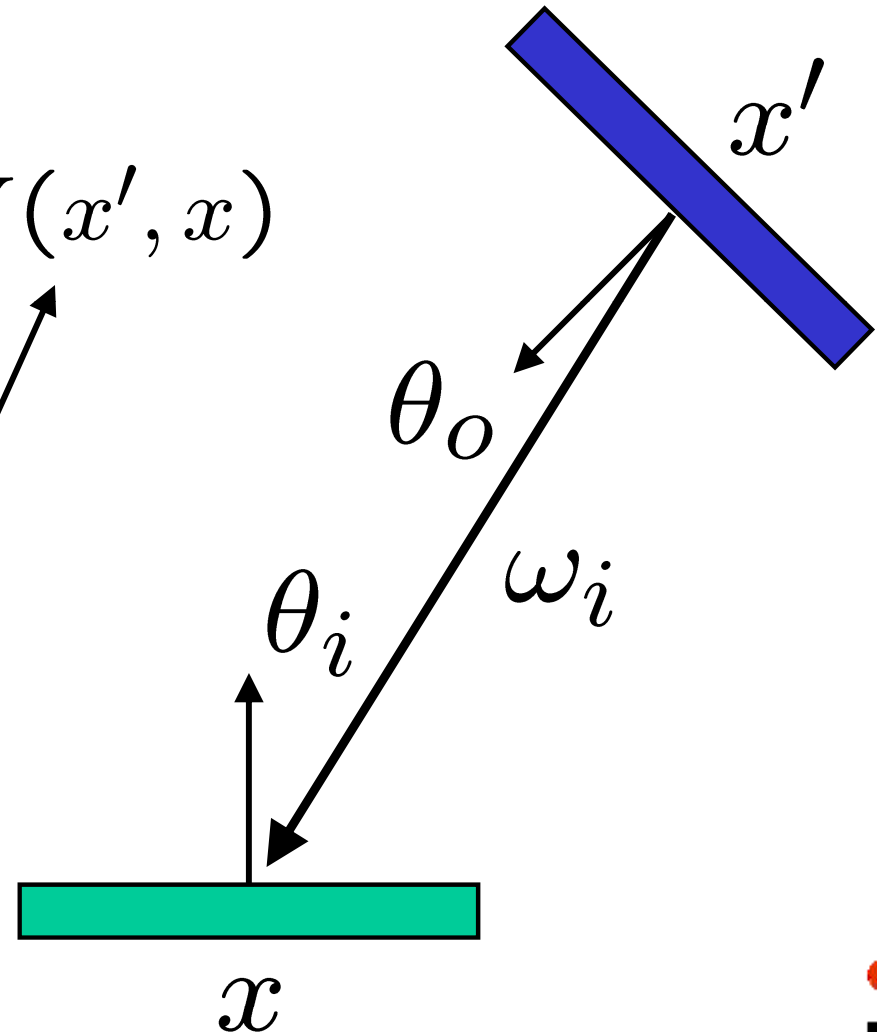
Transporte de Radianza

$$L_i(x, \omega_i) = L_o(x', \omega'_o) V(x', x)$$

Transporte en el vacío:
radianza entrante igual a
radianza saliente

Visibilidad (0 ó 1)

$$\omega_i = -\omega'_o$$

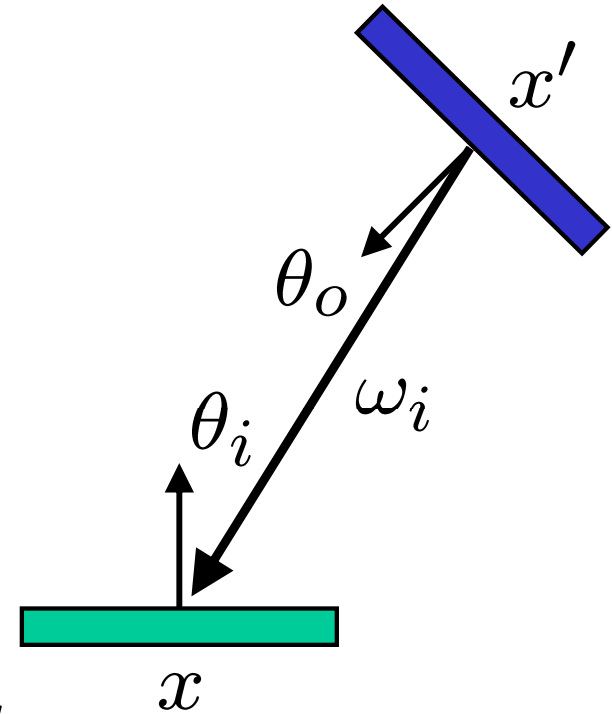


Ec. de Transporte

Radianza sin emisión

$$L_o(x, \omega_o) = \int_{\Omega} \frac{L_i(x, \omega_i) f_r(x, \omega_o, \omega_i) \cos \theta_i d\omega_i}{}$$

$$L_o(x', \omega'_o) V(x', x)$$



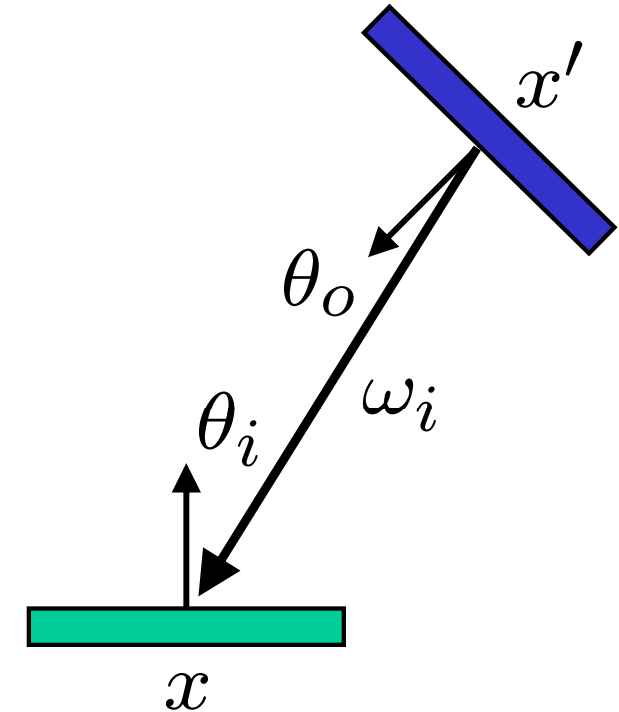
Integral sobre Superficie

Integrando sobre el área de la escena en lugar de sobre ángulo sólido:

$$d\omega = \frac{dA}{r^2} = \frac{\cos \theta_o dA'}{|x' - x|^2}$$

$$L_o(x, \omega_o) =$$

$$\int_S L_o(x', \omega'_o) V(x', x) f_r(x, \omega_o, \omega_i) \frac{\cos \theta_o \cos \theta_i}{|x' - x|^2} dA'$$



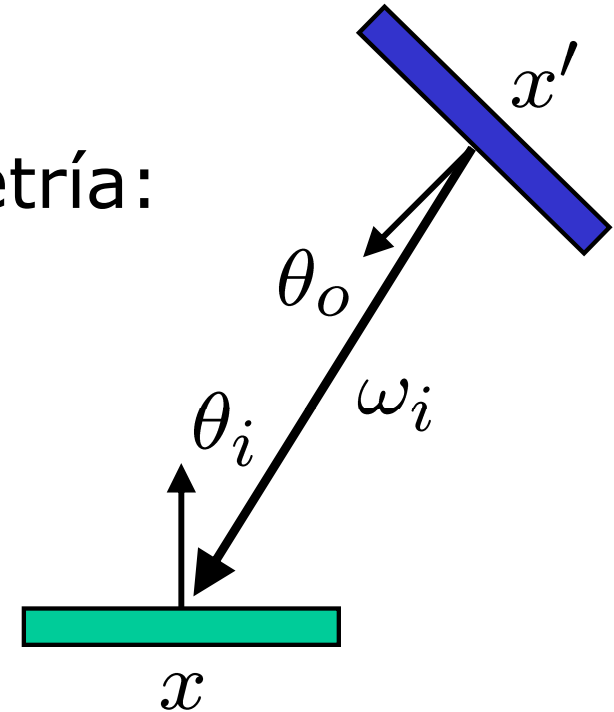
Este es el paso clave. Ahora definimos la radianza reflejada en x en función de la radianza reflejada en todos los puntos de la escena!



Término de Geometría

Término dependiente de la geometría:

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



$$L_o(x, \omega_o) =$$

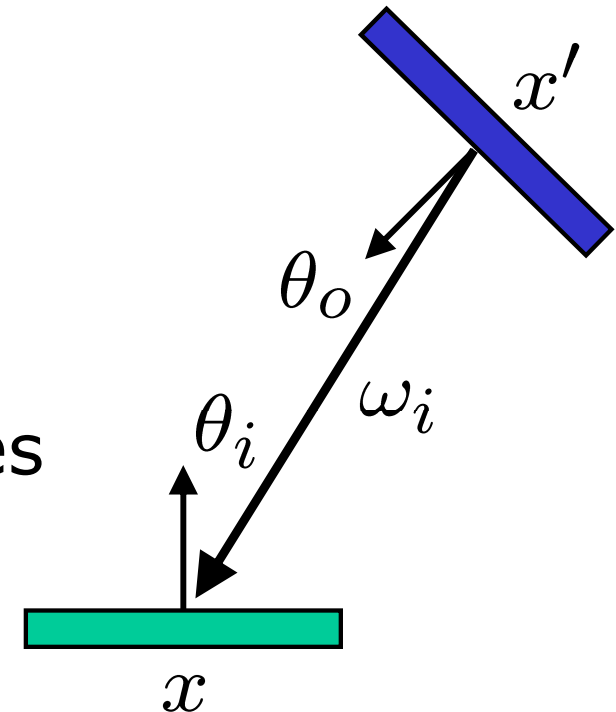
$$\int_S L_o(x', \omega'_o) f_r(x, \omega_o, \omega_i) G(x, x') dA'$$



Caso particular para Radiosidad: Asumiendo superficies difusas

$$f_r(x, \omega_o, \omega_i) = \rho(x)$$

BRDF independiente de las direcciones



$$L_o(x) = \rho(x) \int_S L_o(x') G(x, x') dA'$$

El conjunto de la ec. es independiente de las direcciones



Ec. de Transporte

$$L_o(x, \omega_o) = \int_S L_o(x', \omega'_o) f_r(x, \omega_o, \omega_i) G(x, x') dA'$$

Añadimos la radianza emitida, y lo expresamos todo con un operador de transferencia T. Esto es una notación muy general, donde lo importante es que es una función lineal.

$$L = L_e + T \circ L$$

L representa ahora la radianza en toda la escena



Ec. de Transporte

Se puede expresar de forma iterativa:

$$L_0 = L_e$$

$$L_1 = L_e + T \circ L_e$$

...

$$L_n = \sum_{i=0}^n T^i \circ L_e$$

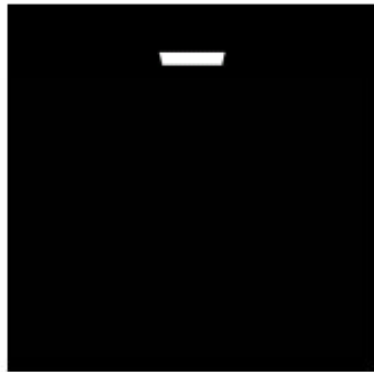
$$L = L_\infty = \sum_{i=0}^{\infty} T^i \circ L_e$$

$$L = (1 - T)^{-1} \circ L_e$$

Esta última es una expresión teórica, no tiene una solución práctica.



Ejemplos de Transferencias



$$L_e$$



$$TL_e$$

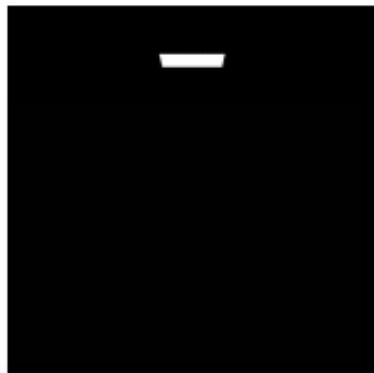


$$T^2 L_e$$



$$T^3 L_e$$

[Cornell PCG]



$$L_e$$



$$L_e + TL_e$$



$$L_e + \dots + T^2 L_e$$



$$L_e + \dots + T^3 L_e$$



Path Tracing

Radianza como suma infinita de transferencias:

$$L = \sum_{i=0}^{\infty} T^i \circ L_e$$

Un término es:

$$L(n) = T^n \circ L_e$$

Indica la radianza en todos los puntos debida a caminos de n rebotes. En un punto x_n en dirección a x_{n+1} , requiere calcular n integrales anidadas sobre toda la superficie de la escena:

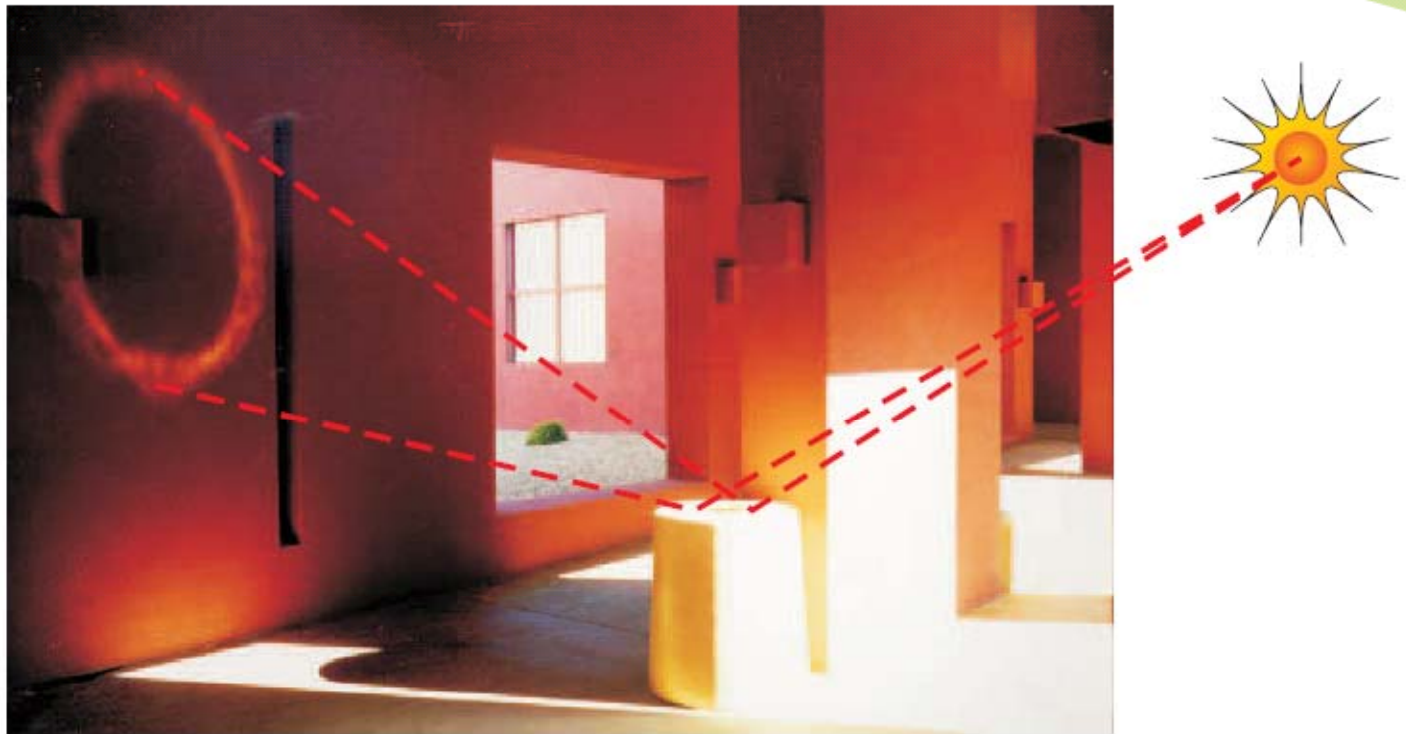
$$L(x_n, x_{n+1}) = \int_S \dots \int_S L_e(x_0 \rightarrow x_1) f_r(x_0 \rightarrow x_1 \rightarrow x_2) G(x_0, x_1) \dots f_r(x_{n-1} \rightarrow x_n \rightarrow x_{n+1}) G(x_{n-1}, x_n) dA(x_0) \dots dA(x_n)$$



Path Tracing

Por cada valor de radianza debida a caminos de n rebotes se ha resolver una integral en $2n$ dimensiones!

A veces es complicado encontrar los caminos que contribuyen....



Path Tracing y Montecarlo

Veremos más sobre el problema de path tracing en el contexto de Montecarlo. Montecarlo es un buen método para aproximar las integrales complejas en altas dimensiones.

Pero hay dos problemas a resolver:

- Muestrear los rayos sin que el muestreo sea sesgado (unbiased rendering).
- Realizar un muestreo para maximizar la precisión con un mínimo número de rayos.



Referencias

- Computer Graphics, Principles and Practice (Foley, van Dam, Feiner, Hughes).
- Physically-based Rendering (Pharr, Humphreys).
- Curso de SIGGRAPH 2001:

<http://www.cs.princeton.edu/courses/archive/fall02/cs526/papers/course29sig01.pdf>

