
Fondamenti di Grafica Tridimensionale

Francesco Banterle

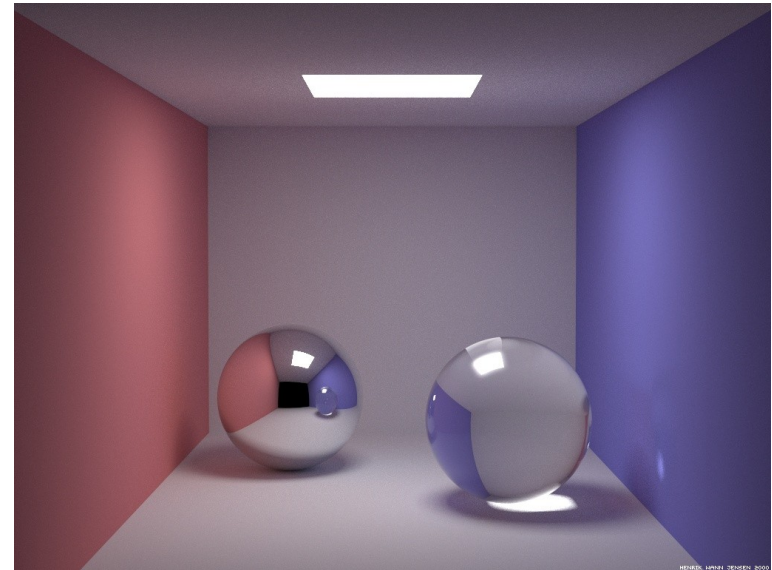
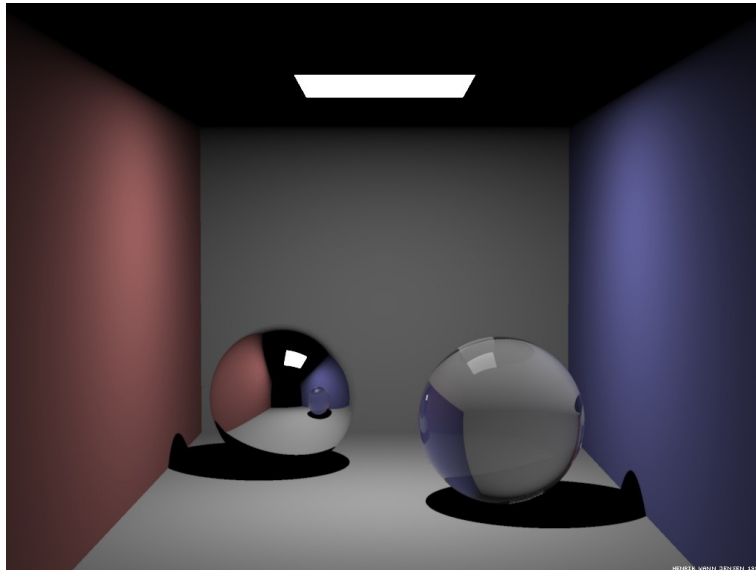
Slides basate su quelle di:

Paolo Cignoni

p.cignoni@isti.cnr.it

<http://vcg.isti.cnr.it/~cignoni>

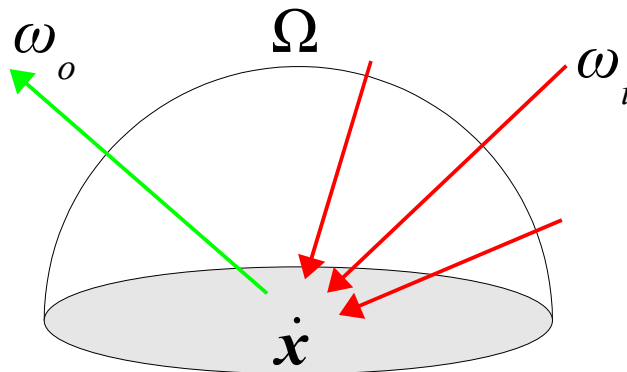
Global Illumination: Perché?



The Rendering Equation

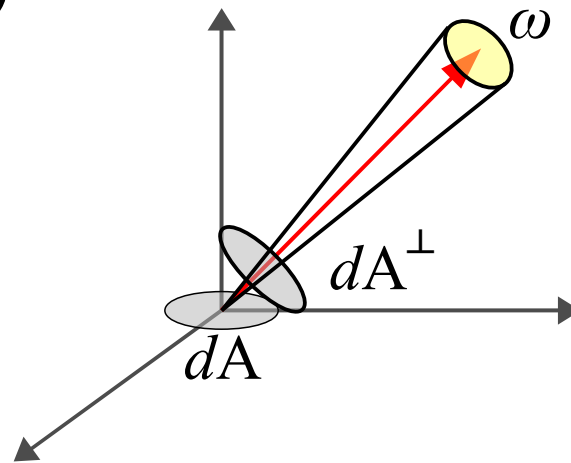
“To provide an unified context for viewing rendering algorithms as more or less accurate approximations to the solution for a single equation” [Kajiya 1986]

$$L_o(\mathbf{x}, \vec{n}, \vec{\omega}_o) = L_e(\mathbf{x}, \vec{\omega}_o) + \int_{\Omega(\vec{n})} f_r(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) (\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i$$



Radianza

- La *radianza* L è una delle 8 unità fisiche radiometriche del S.I.
- Misura l'intensità di un raggio luminoso definita come la potenza per angolo solido unitario per area proiettata.
- Si misura in: $\text{W}/(\text{sr m}^2)$



Irradianza

- **Irradianza**, E , è l'integrale della radianza su tutte le direzioni (pesato con il cos della direzione)

$$E(\mathbf{x}, \vec{n}) = \int_{\Omega(\vec{n})} L_i(\mathbf{x}, \vec{\omega}) \cos\theta \, d\vec{\omega}$$

- θ è l'angolo tra la normale di \mathbf{x} e ω
- $L_i(\omega)$ è la radianza in ingresso
- Nota: $d\vec{\omega} = \sin\theta \, d\theta \, d\phi$
- L'irradianza si misura in W/m^2

Radiosity

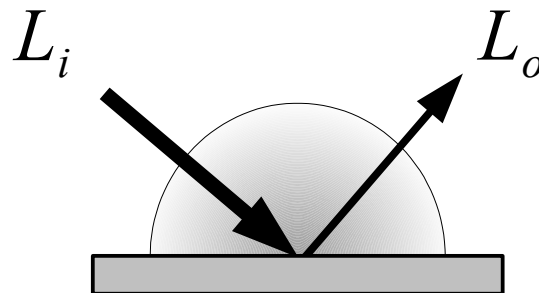
- *Radiosity*, B , è l'energia per unità di area che lascia una superficie per unità di tempo
- Unità di misura: W/m^2

Radianza Riflessa

- La radianza riflessa e' definita:

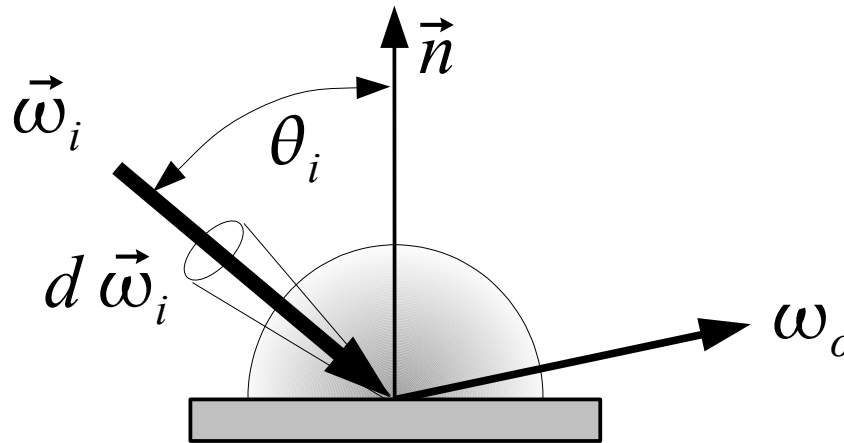
$$L_o(\mathbf{x}, \vec{\omega}_o) = \int \rho L_i(\mathbf{x}, \vec{\omega}_i) d\vec{\omega}_i$$

dove ρ e' il coefficiente di riflessione, L_o e rappresenta la quantità di energia (radianza) che viene riflessa verso una direzione.



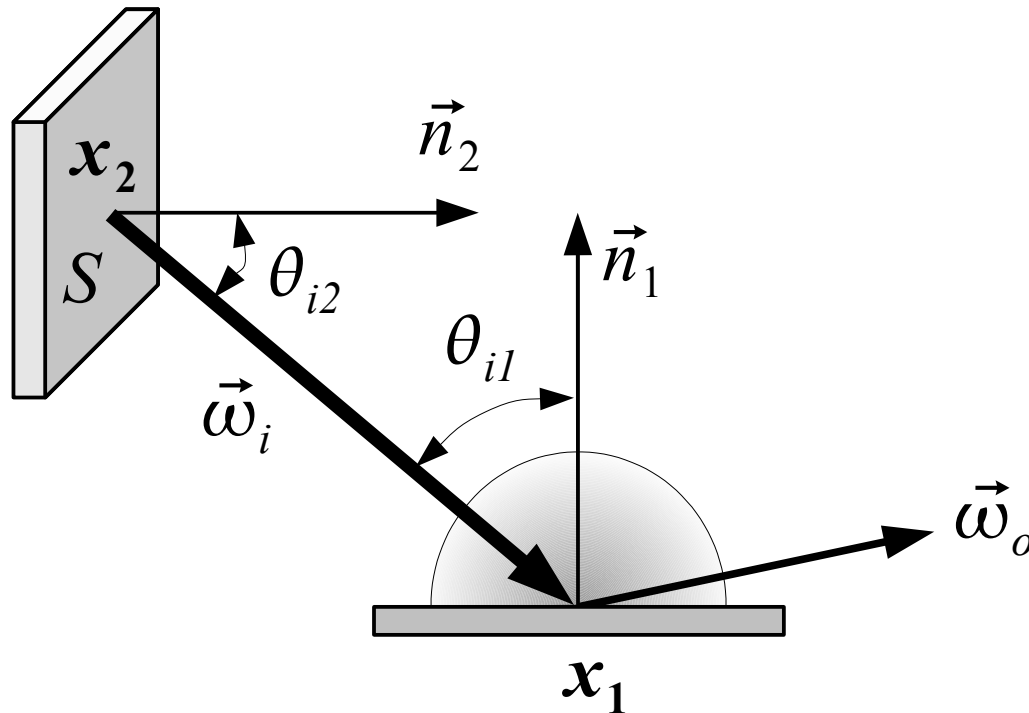
Con variazione direzionale

$$L_o(\mathbf{x}, \vec{\omega}_o) = L_e(\mathbf{x}, \vec{\omega}_o) + \int_{\Omega} f_r(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) \cos\theta \, d\vec{\omega}_i$$



Integrazioni di Superficie

$$L_o(\mathbf{x}, \omega_o) = L_e(x, \omega_o) + \int_S f_r(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o) L_i(\mathbf{x}, \vec{\omega}_i) g(\mathbf{x}, \mathbf{x}') \frac{\cos\theta_{i1} \cos\theta_{12}}{\|\mathbf{x} - \mathbf{x}'\|^2} dA$$



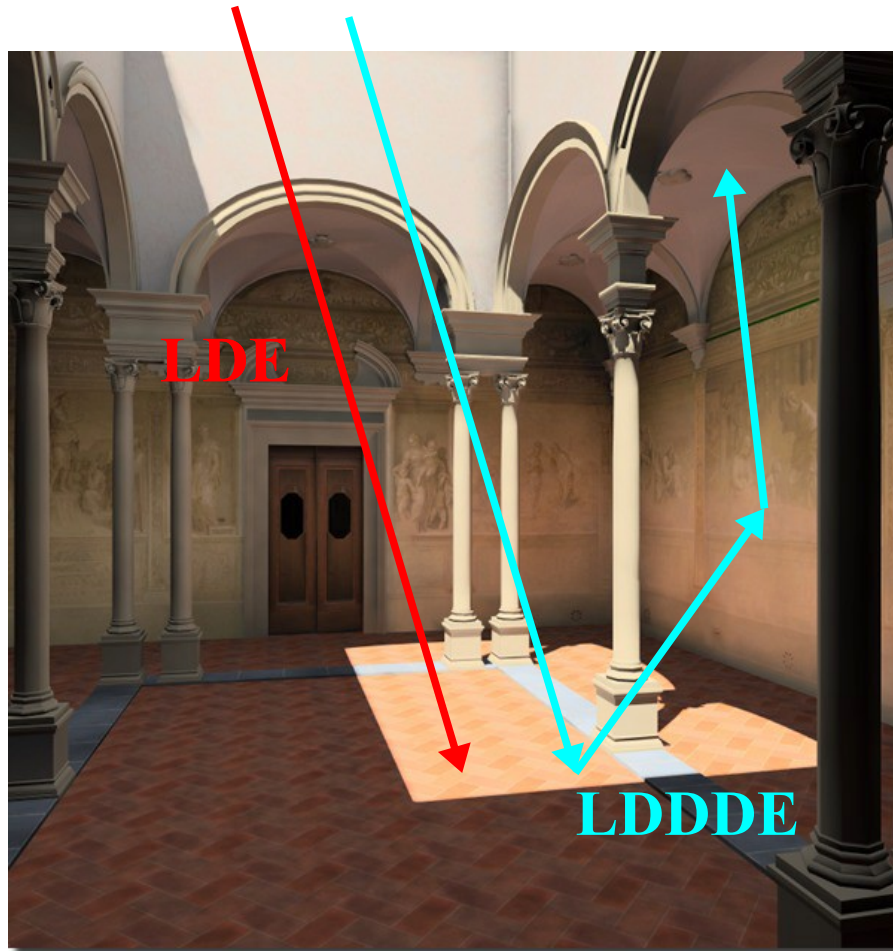
Notazione Light Paths

- E' una grammatica regolare introdotta da Heckbert, utile per classificare gli algoritmi di *Global Illumination*
- Si descrive il percorso della luce dalla sorgente luminosa all'occhio:

$$L(S|D)*E$$

- L: sorgente luminosa
- D: riflessione diffusa
- S: riflessione speculare
- E: occhio

Esempi di Light Paths



Esempi di Light Paths: Caustics

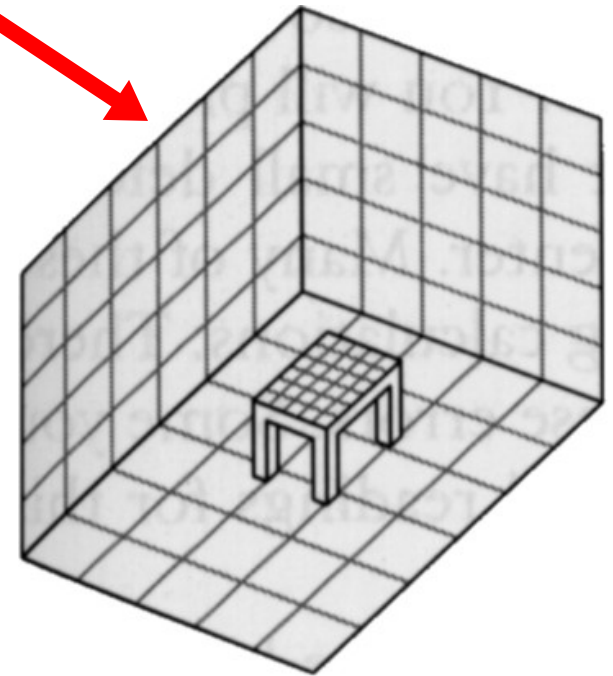
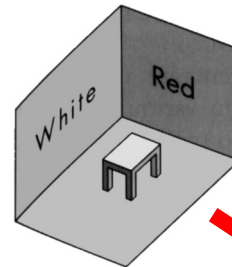


Algoritmi di Global Illumination

- Radiosity
- Whitted Ray-Tracing
- Distributed Ray-Tracing
- Path-Tracing + Irradiance Caching
- Photon Mapping
- Instant Radiosity

Radiosity

- Nel Radiosity, la scena viene suddivisa in *patch*, ovvero in poligoni piatti e di dimensioni limitate, ciascuno dei quali è considerato perfettamente diffusivo
- Questo implica che:



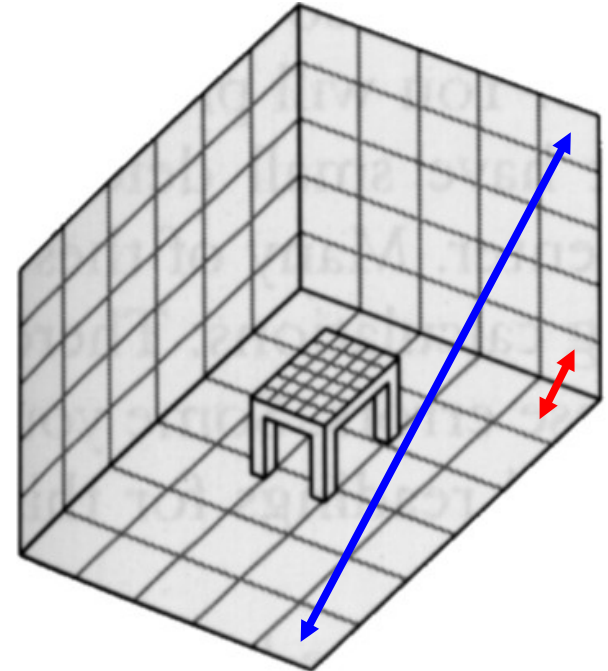
$$f_r(\mathbf{x}, \vec{\omega}_i \rightarrow \vec{\omega}_o) = \frac{\rho}{\pi}$$



$$L_o(\mathbf{x}, \vec{n}, \vec{\omega}_o) = L_e(\mathbf{x}, \vec{\omega}_o) + \frac{\rho}{\pi} \int_{\Omega(\vec{n})} L_i(\mathbf{x}, \vec{\omega}_i) (\vec{\omega}_i \cdot \vec{n}) d\vec{\omega}_i$$

Radiosity: Primo Passo

- Il primo passo consiste nel determinare, per ogni coppia di patch, i **fattori di forma** (*form factor* o *FF*).
- I form factor definiscono quanta energia esce da un patch tenendo conto di:
 - Occlusioni
 - Orientamento
 - Distanza
- Il calcolo dei FF e' $O(n^2)$, ma la maggior parte dei FF sono praticamente nulli:
 - Patch lontane non si influenzano



Radiosity: Form Factor

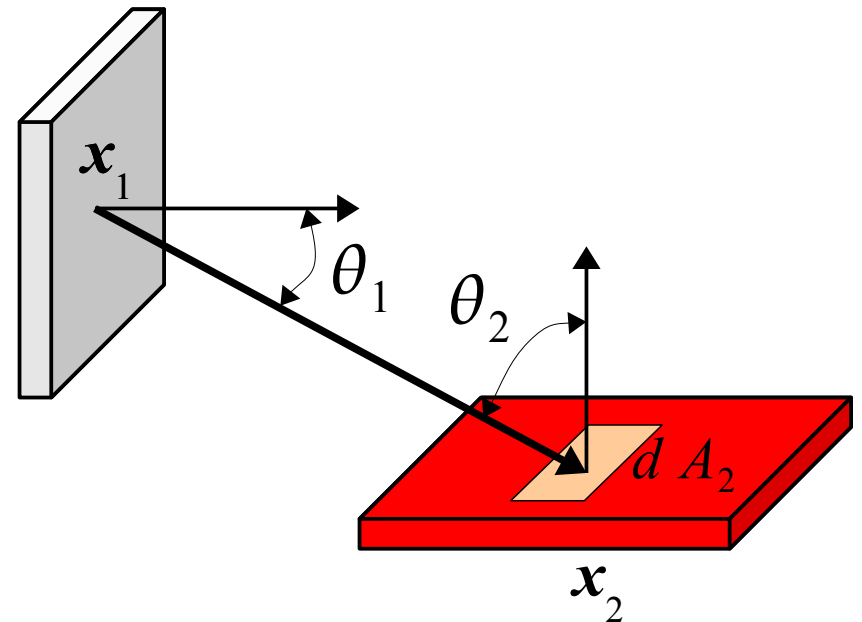
- Form Factor:

$$F(\mathbf{x}_1, \mathbf{x}_2) = V(\mathbf{x}_1, \mathbf{x}_2) \frac{\cos \theta_1 \cos \theta_2}{\pi \|\mathbf{x}_1 - \mathbf{x}_2\|^2} dA_2$$

- Vale che:

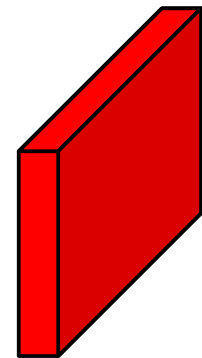
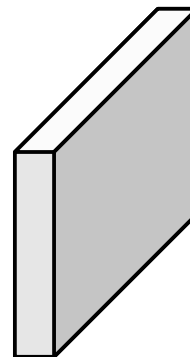
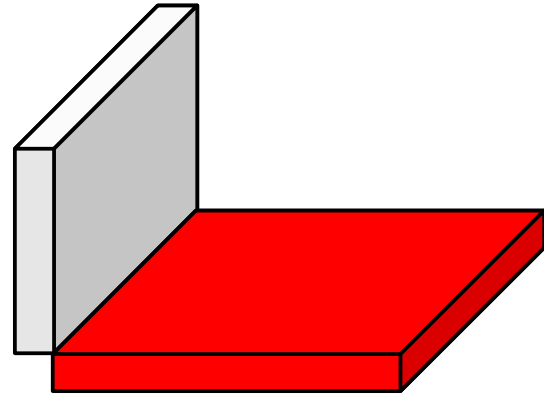
$$F(\mathbf{x}_1, \mathbf{x}_2) A_1 = F(\mathbf{x}_2, \mathbf{x}_1) A_2$$

$$F(\mathbf{x}_1, \mathbf{x}_2) = F(\mathbf{x}_1, \mathbf{x}_2) \frac{A_2}{A_1}$$



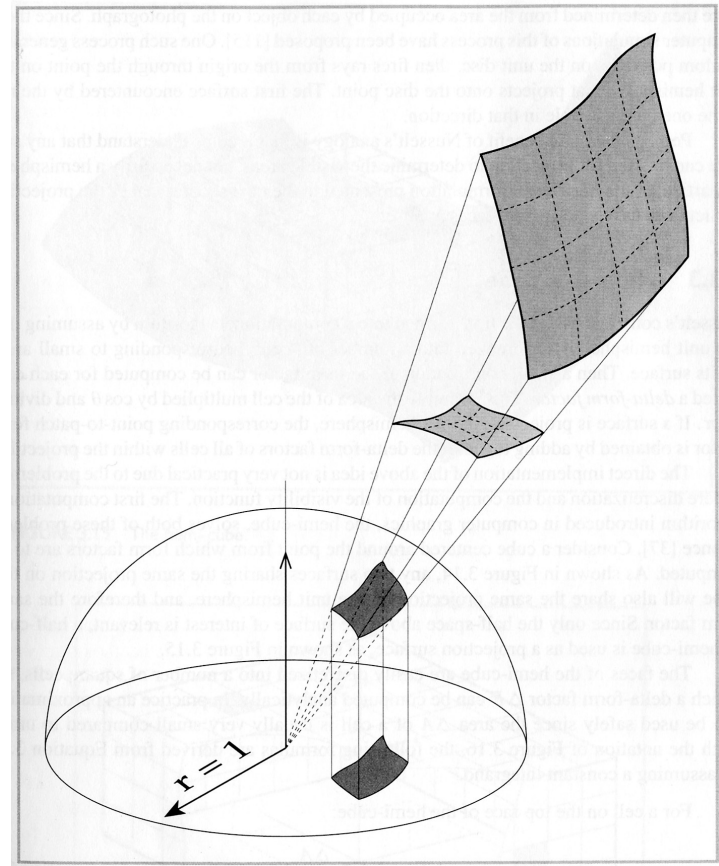
Radiosity: Calcolo del Form Factor

- In casi molto semplici esiste una forma chiusa
 - Caso rettangolo-rettangolo, che si vedono perfettamente
 - In generale sono risultati solo teorici, utili per come benchmark degli algoritmi generali



Radiosity: Calcolo del Form Factor

- Ogni patch con la stessa proiezione sulla semisfera ha lo stesso FF
- Metodo Montecarlo:
 - Campionare (uniformemente in area proiettata sul piano) la semisfera
 - Lanciare un raggio
 - FF per ogni patch è la somma pesata del numero di raggi che colpisce quella patch



Radiosity: Secondo Passo

- Nel secondo passo, la distribuzione della luce all'interno di una scena corrisponde ad un sistema di equazioni lineari
- La somma della radianza che arriva su una patch deve essere uguale alla radianza che esce più la radianza assorbita dalla patch stessa:

$$B(\mathbf{x}') d\mathbf{x}' = \varepsilon(\mathbf{x}') d\mathbf{x}' + \rho(\mathbf{x}') \int_S B(\mathbf{x}) F(\mathbf{x}, \mathbf{x}') d\mathbf{x}$$

Radiosity: Equazione

$$B(\mathbf{x}') d\mathbf{x}' = \varepsilon(\mathbf{x}') d\mathbf{x}' + \rho(\mathbf{x}') \int_S B(\mathbf{x}) F(\mathbf{x}, \mathbf{x}') d\mathbf{x}$$

- La soluzione effettuando la discretizzazione dell'integrale (patch discrete invece che di porzioni infinitesime della superficie):

$$B(p_i) A(p_i) = \varepsilon(p_i) A(p_i) + \rho(p_i) \sum_{j=1}^n B(p_j) A(p_j) F(p_j, p_i)$$

Radiosity: Discretizzazione

- Sfruttando il principio di reciprocità dei FF, e dividendo per

$$A(p_i)$$

si ottiene che:

$$B(p_i) = \varepsilon(p_i) + \rho(p_i) \sum_{j=1}^n B(p_j) F(p_i, p_j)$$

Radiosity: Sistema Lineare

$$B(p_i) = \varepsilon(p_i) + \rho(p_i) \sum_{j=1}^n B(p_j) F(\mathbf{x}_i, \mathbf{x}_j)$$

$$B_i = E_i + R_i \sum_{j=1}^n B_j F_{ij}$$

$$\begin{bmatrix} 1 - R_1 F_{11} & -R_1 F_{12} & \cdots & -R_1 F_{1n} \\ -R_2 F_{21} & 1 - R_2 F_{22} & \cdots & -R_2 F_{2n} \\ \vdots & & \ddots & \vdots \\ -R_n F_{n1} & \cdots & & 1 - R_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

Radiosity: Approccio Iterativo

- Approccio iterativo:
 - Evitare di calcolarsi e memorizzarsi tutti i FF subito
 - Avere una soluzione approssimata subito che migliora ad ogni iterazione
- Per Ogni Iterazione
 - Selezionare la patch i (massima differenza di unshot)
 - Calcolare F_{ij} , per Ogni patch j :
 - Aggiornamento del radiosity della patch j
 - Aggiornamento dell'emissione della patch j
 - Assegnare l'emissione della patch i a zero

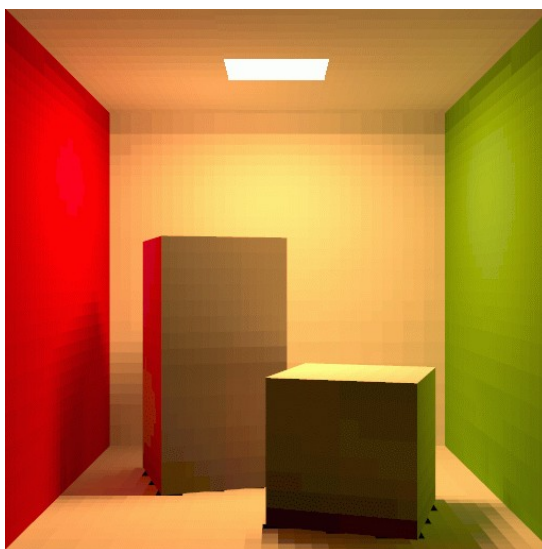
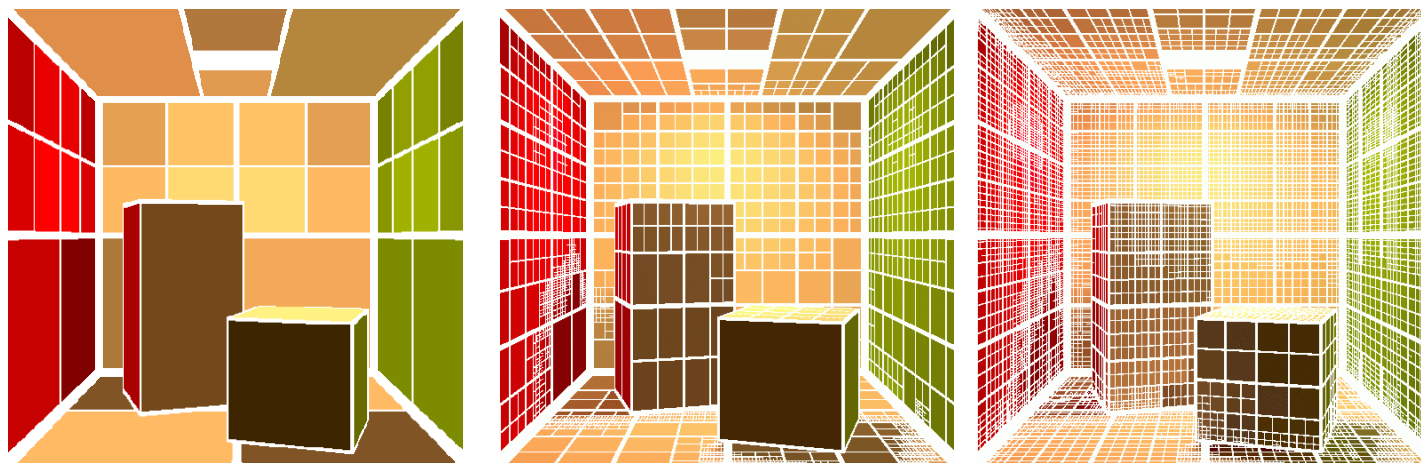
Radiosity: Approccio iterativo



PROGRESSIVE SOLUTION

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

Radiosity: Raffinamento della Mesh



Radiosity: Conclusioni

- Path Notation: LD^*E
- Invece di seguire raggi considera l'interazione tra le superfici della scena
- La soluzione e' generalmente *view-independent*
- Problemi:
 - Discretizzazione della scena in patch (dipendenti o meno dalla vista)
 - Non modella riflessioni speculari e trasmissione
 - Calcolare FF per tutte le coppie di patch
 - Risolvere il sistema

Whitted Ray-Tracing

- E' un algoritmo ricorsivo che parte dalla telecamera (E)
- Per ogni pixel del piano immagine viene lanciato un raggio, R , che viene intersecato con la scena
- Per ogni superficie diffusa, D , colpita da R viene calcolata l'illuminazione diretta, lanciando un raggio S (*shadow ray*) verso la luce
- Se la superficie e' riflettente o rifrattiva viene lanciato un raggio riflesso o rifratto

Whitted Ray-Tracing: Intersezioni

- Raggio definito come:

$$\mathbf{R} = \begin{bmatrix} d_x \\ d_y \\ d_z \\ 0 \end{bmatrix} t + \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix}$$

\vec{d} la direzione del raggio

\mathbf{c} punto di partenza del raggio

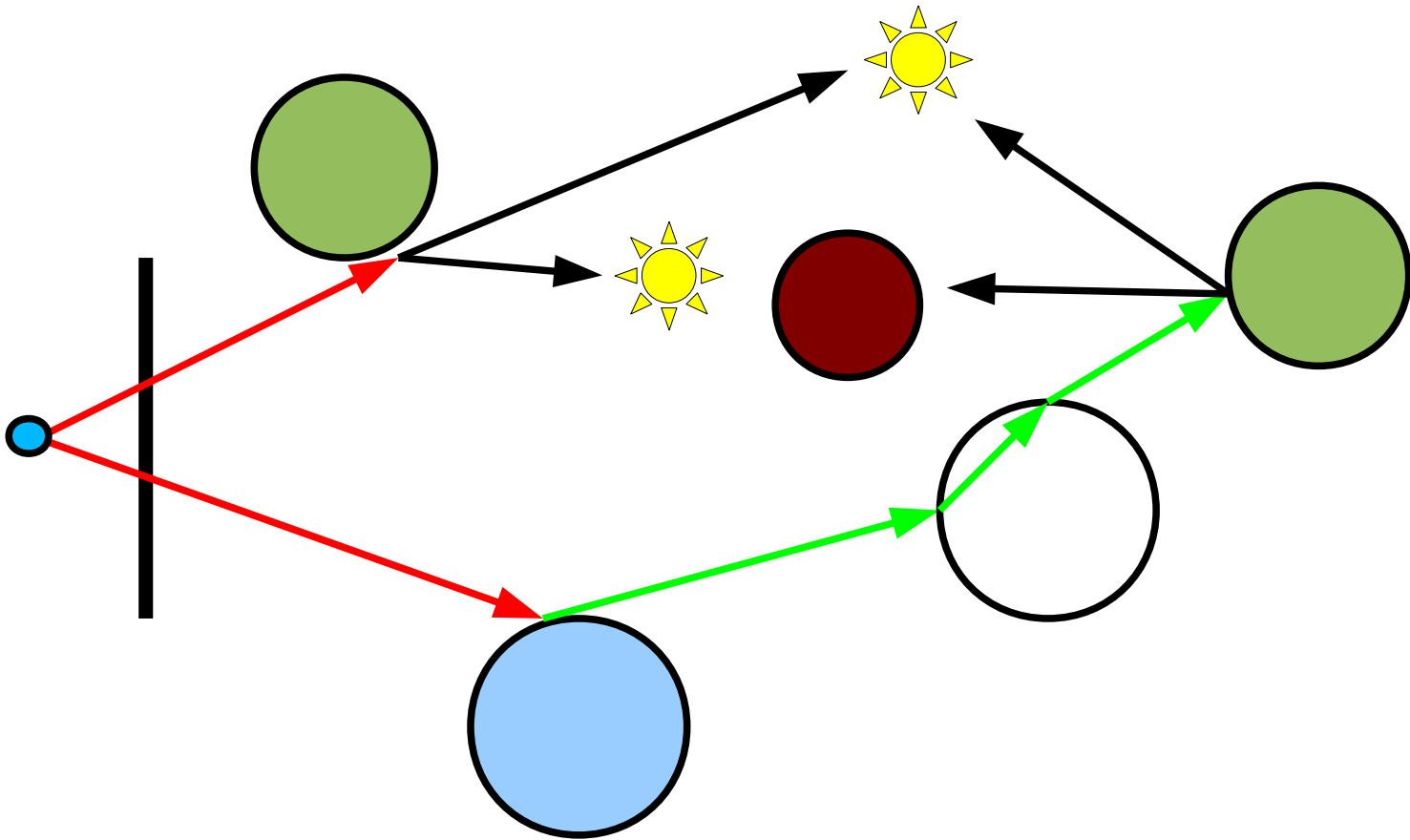
$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = r^2 \quad \text{Equazione della sfera in } (x_0, y_0, z_0)^T \text{ e raggio } r$$

Si sostituisce x , y e z nell'equazione della sfera con \mathbf{R} , e si risolve l'equazione di secondo grado in t :

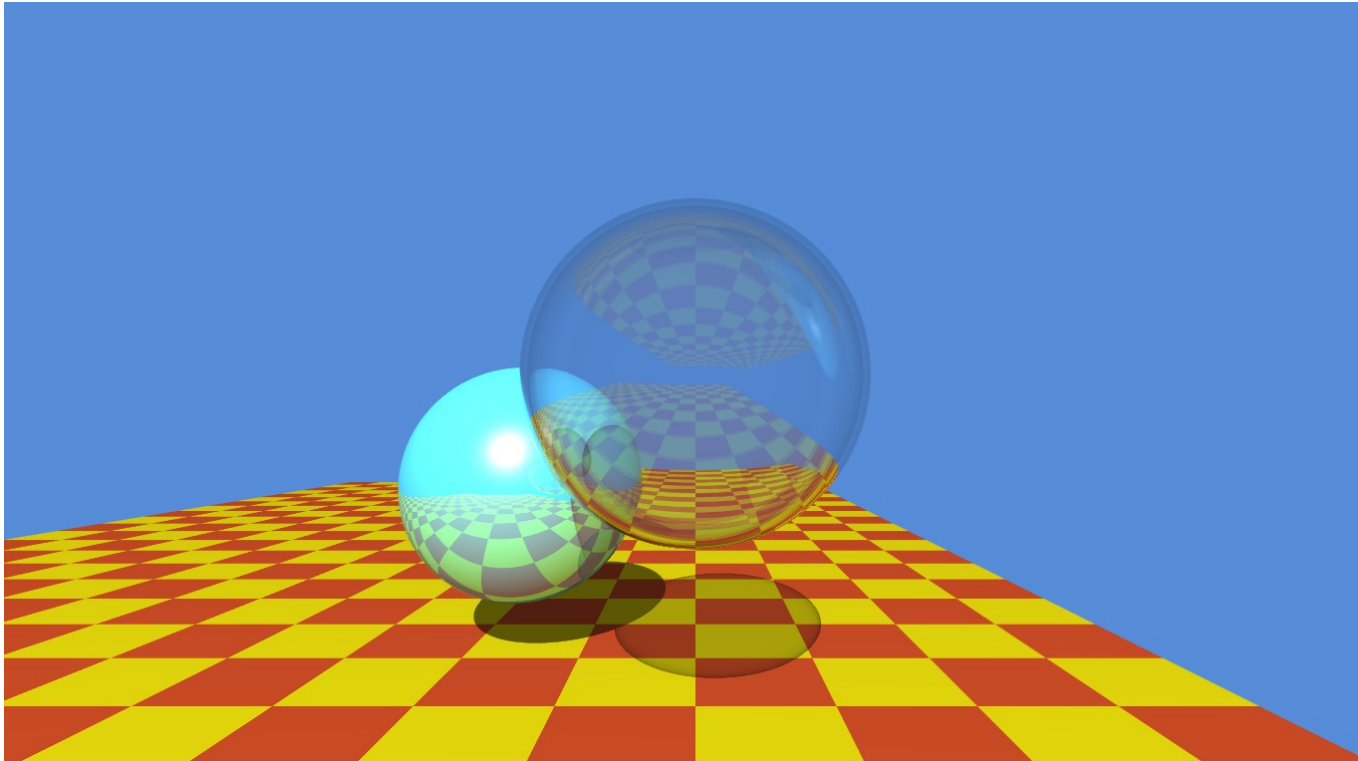
- Se esistono soluzioni vi è intersezione
- Si prende la soluzione minore con $t > 0$

Whitted Ray-Tracing

- Path Notation: LD?S*E



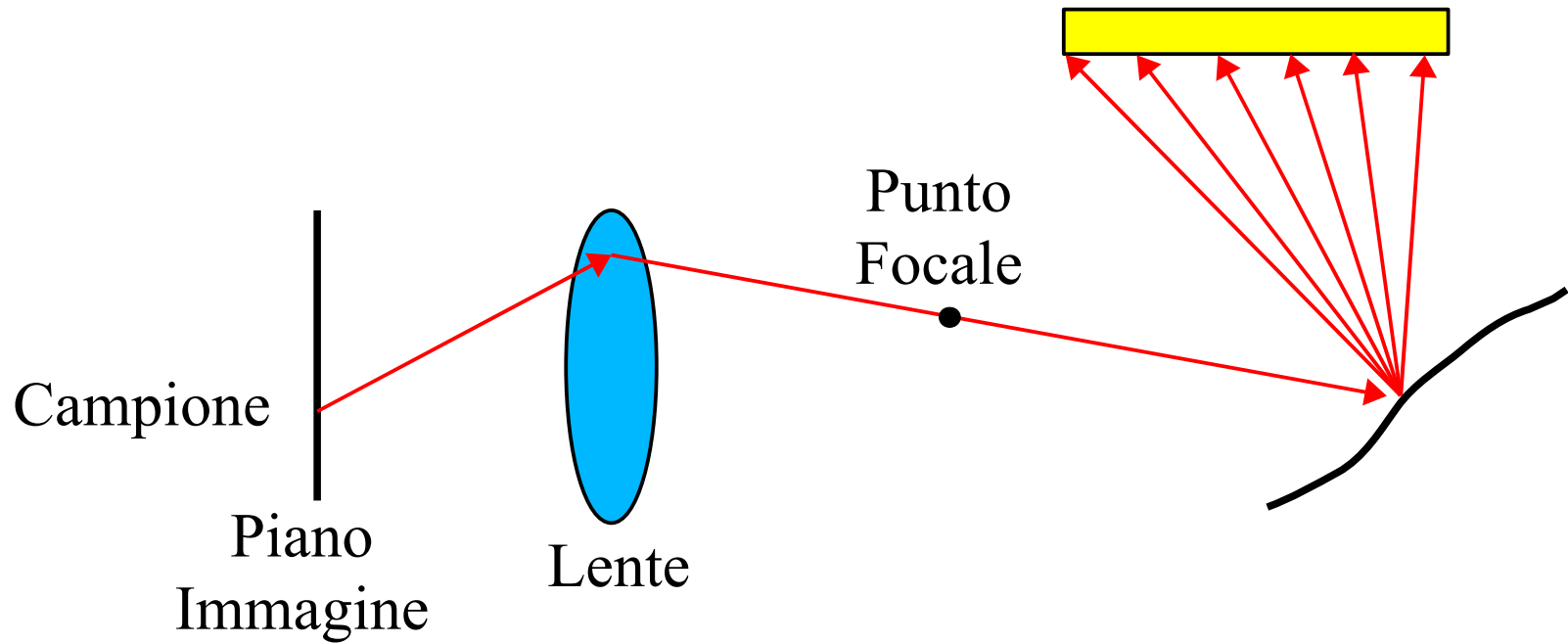
Ray-Tracing: Esempio



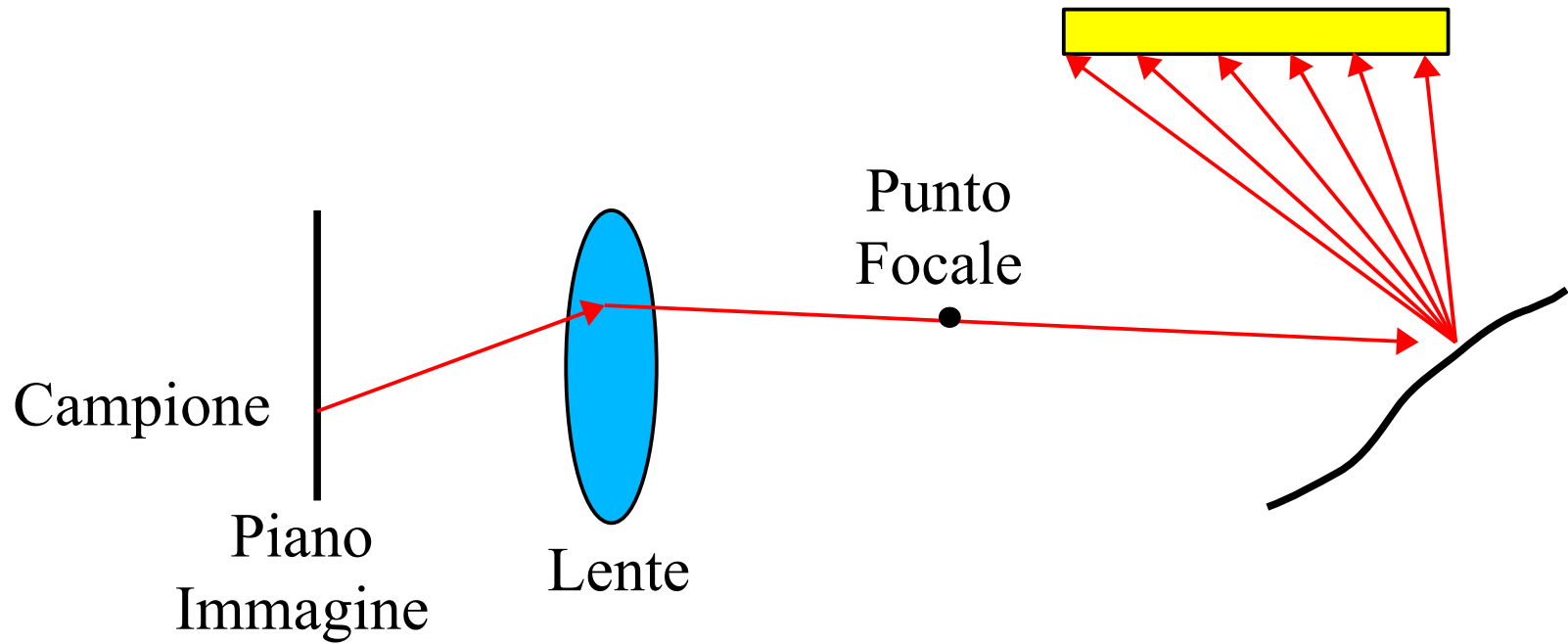
Distributed Ray-Tracing

- E' un'estensione del Whitted Ray-Tracing, per simulare i seguenti effetti:
 - Anti Aliasing
 - Area Lights
 - Soft Reflections
 - Motion Blur
 - Depth of Field
- L'idea di base e' di lanciare piu' raggi utilizzando metodi di Monte-Carlo:
 - Buon generatore di numeri casuali (Mersenne Twister)
 - Distribuzione di punti (Jittering, Hammersly, Halton, Poisson)

Distributed Ray-Tracing: Diagramma



Distributed Ray-Tracing: Diagramma



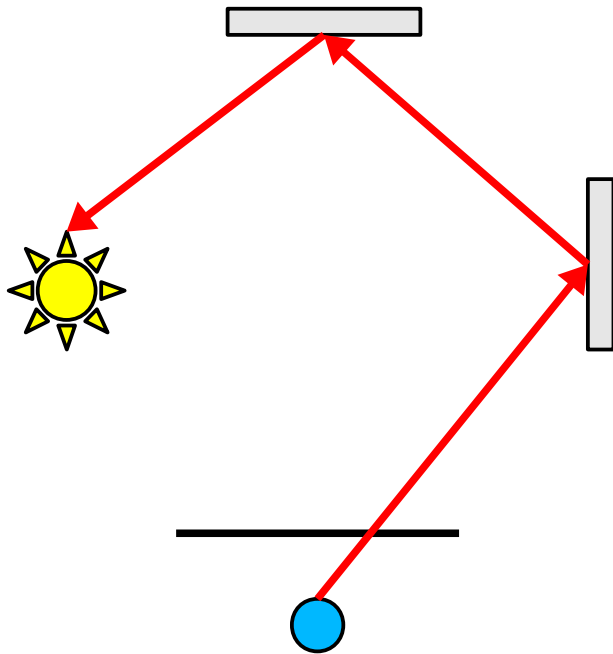
Distributed Ray-Tracing: Esempio



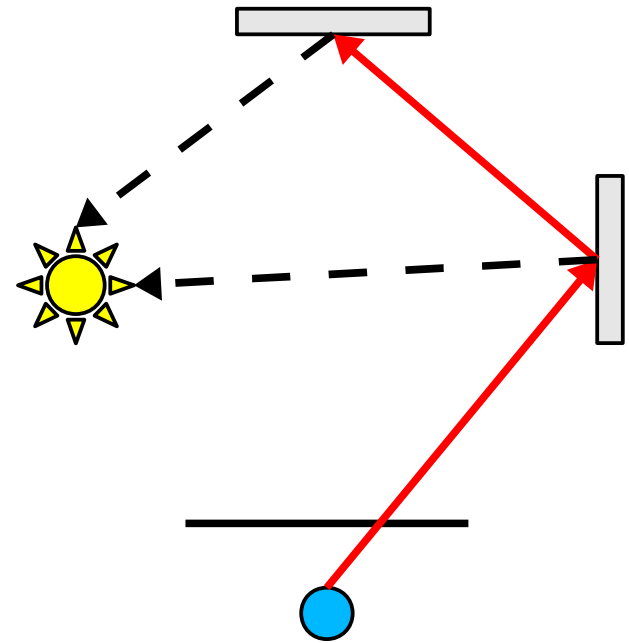
Path-Tracing

- E' la naturale estensione del Ray-Tracing e del Distributed Ray-Tracing
- Per ogni pixel vengono generati N path come nel Distributed Ray-Tracing
- **Differenza:** quando viene colpita una superficie diffusa si crea un raggio di riflessione diffusiva (random sull'emisfero)
- Path Notation: $L(S|D)*E$

Path-Tracing: Implicito ed Esplicito

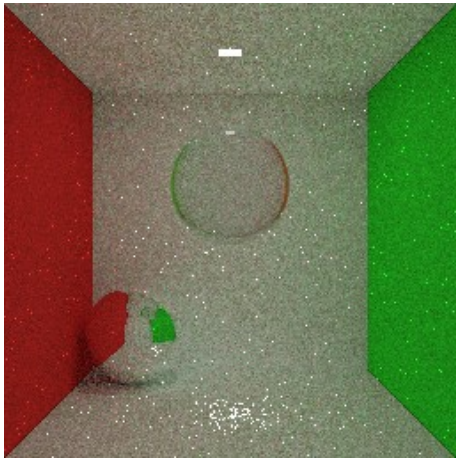


Path-Tracing Implicito

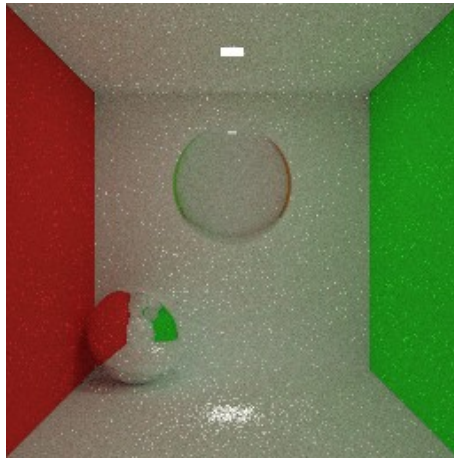


Path-Tracing Esplicito

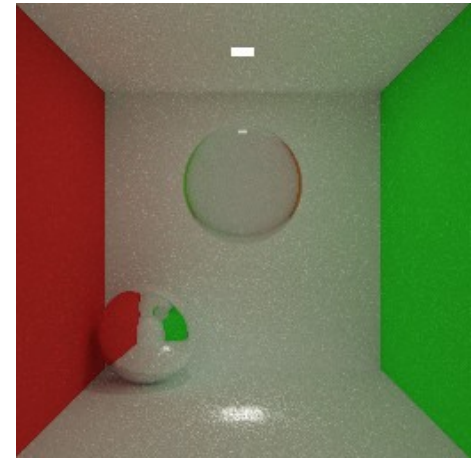
Path-Tracing: Esempio



4 path per pixel



16 path per pixel



64 path per pixel

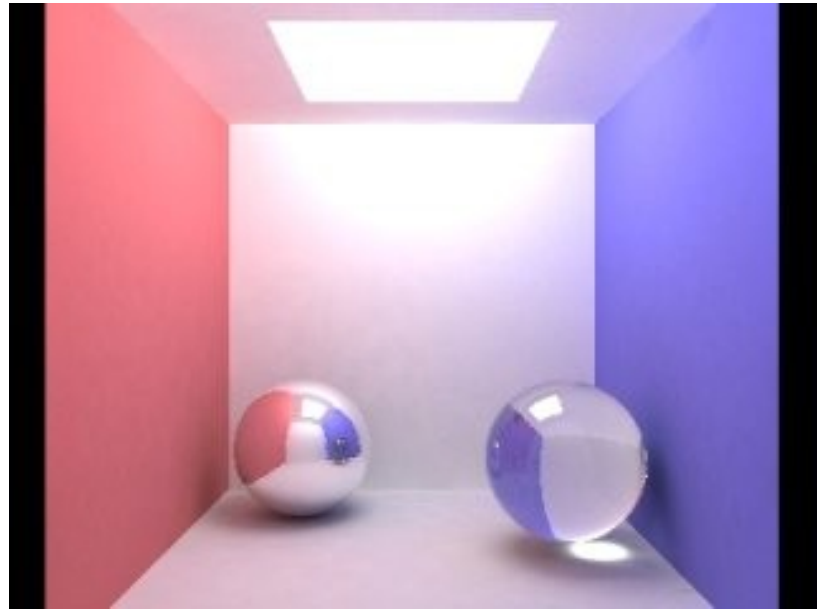
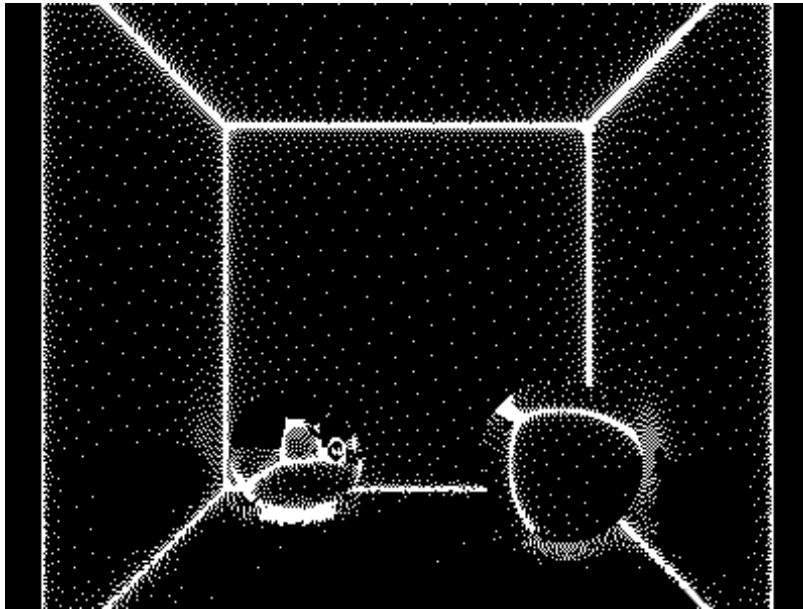
Path-Tracing Conclusioni

- Simulazione di tutti i path di illuminazione in modo un-biased
- Le immagini richiedono un'elevato numero di path per ridurre il rumore, circa 1000 path per pixel. La varianza del rumore è proporzionalmente a $\frac{1}{\sqrt{N}}$
- Non appena vi è un oggetto riflettente o rifrattivo il rumore aumenta in modo elevato, più di 10000 path per pixel sono necessari per ridurre il rumore!

Irradiance Cache

- Si calcola l'irradianza dei path solo in alcuni punti della scena in spazio oggetto assumendo che:
 - Le superfici sono diffuse o con BRDF speculare a bassa frequenza
 - La geometria non è ad alta frequenza
- Per il resto della scena l'irradianza viene calcolata usando interpolazione, di ordine zero o primo
- Punti di cache:
 - Dove la geometria cambia (spigoli ed lati)
 - Dove l'illuminazione cambia (piano a punti distanti)

Irradiance Cache: Esempio

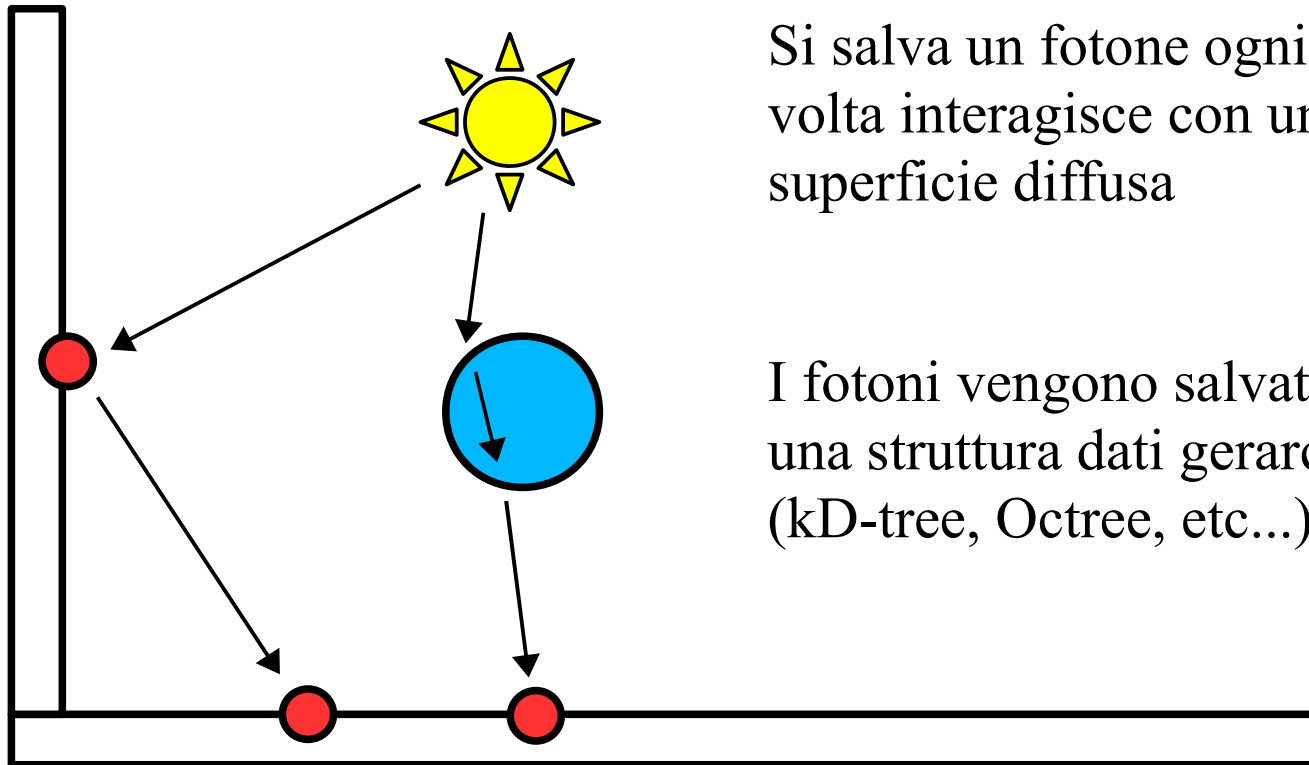


Photon Mapping

- E' un algoritmo a due passate:
 - **Prima passata o Photon Tracing:** si lanciano delle particelle, dette fotoni, che partono dalla luce e rimbalzano nella scena finché non vengono assorbite da un materiale:
$$e < \rho$$

e casuale in $[0,1]$ e ρ e' la riflettanza media del materiale
 - **Seconda Passata o Density Evaluation:** viene eseguito il Ray-Tracing (Whitted o Distributed) ma quando si colpisce una superficie diffusa si calcola la stima di densità dei fotoni anziché l'illuminazione diretta
- Path Notation: $L(S|D)*E$

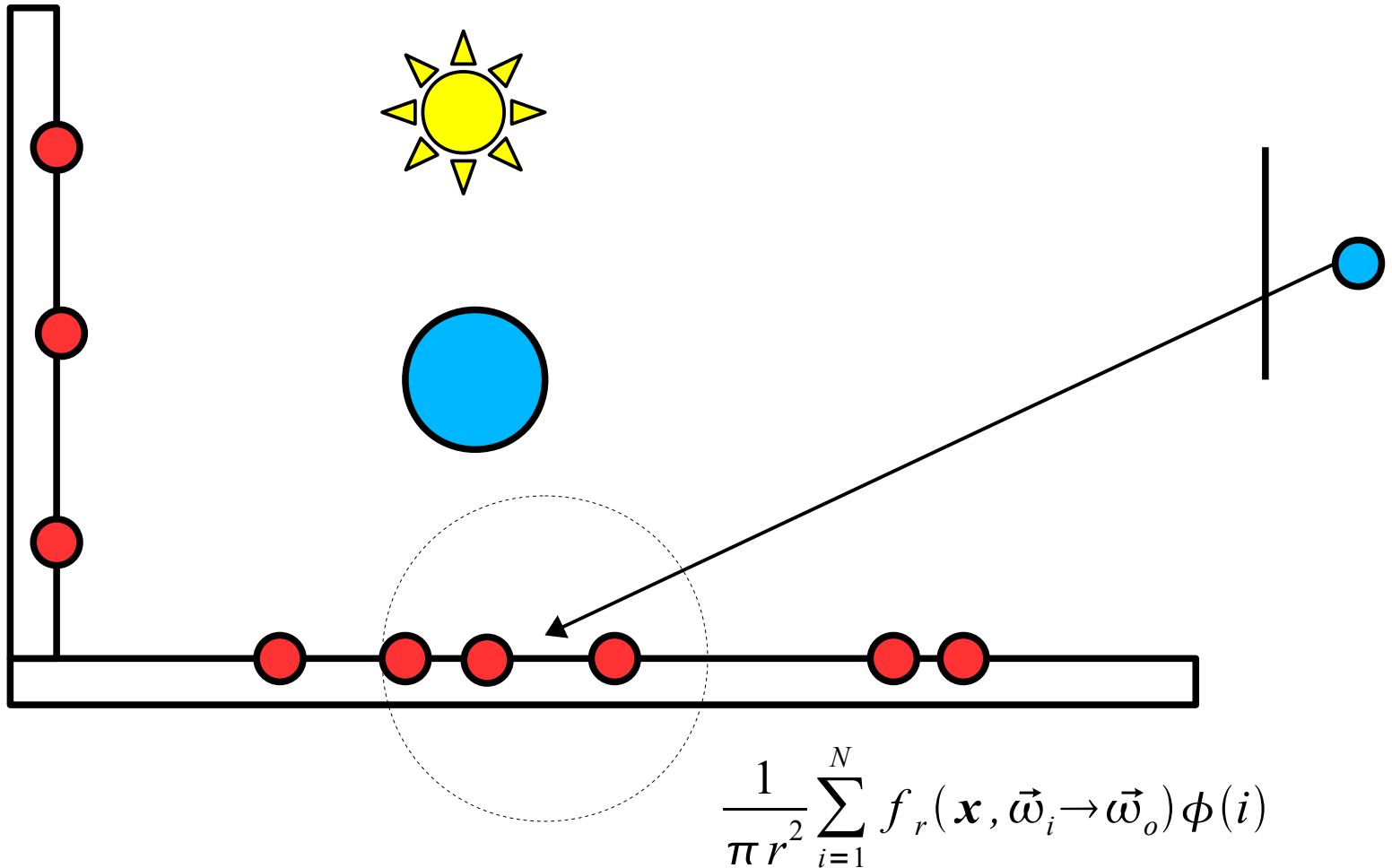
Photon Mapping: Photon Tracing



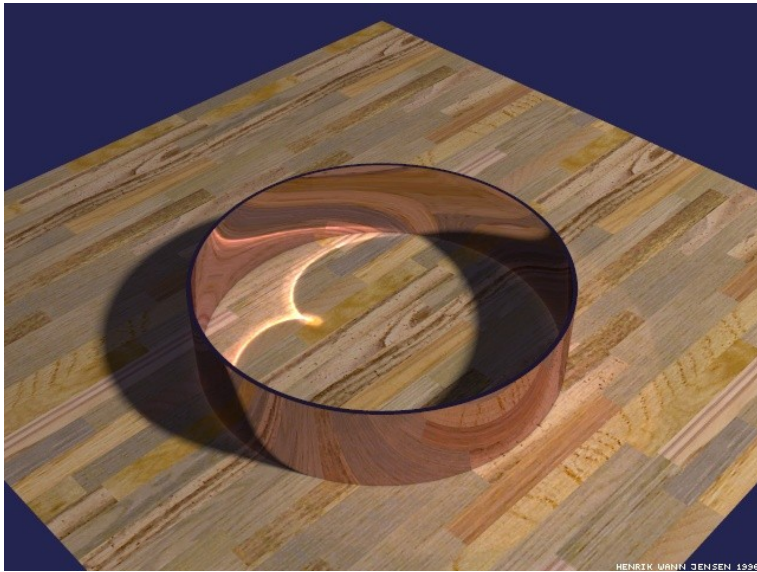
Si salva un fotone ogni qual volta interagisce con una superficie diffusa

I fotoni vengono salvati in una struttura dati gerarchica (kD-tree, Octree, etc...)

Photon Mapping Density Estimation



Photon Mapping: Esempi



Photon Mapping: Conclusioni

- Il metodo riesce a simulare tutti i path di illuminazione, molto efficace per le caustiche (LS⁺DE)
- Le cache possono essere utilizzate per velocizzare
- Non vi è rumore, ma è sostituito dal bias
- I parametri della stima di densità dipendono dalla scena: raggio di gathering e numero di fotoni
- Elevato consumo di memoria (Gbyte per avere almeno 50M di fotoni):
 - Soluzione: Progressive Photon Mapping

Instant Radiosity

- E' un algoritmo a due passate:
 - **Prima Passata o Virtual Light tracing:** simile al photon mapping, si lanciano delle luci virtuali dalla sorgente luminosa. Vengono salvate quando vengono assorbite dal materiale
 - **Seconda Passata:** è uguale al Ray-Tracing (Whitted o Distributed), nel quale si valutano anche le luci virtuali oltre a quelle reali (inclusi gli *shadow ray!*)
- Path Notation: $L(S|D)*E$

Instant Radiosity: Esempio



Instant Radiosity: Conclusioni

- L'algoritmo è veloce e può essere:
 - Implementato su GPU
 - Usato con irradiance/radiance cache
 - Produce risultati convincenti
- L'algoritmo tuttavia non riproduce bene:
 - BRDF ad alta frequenza
 - Caustiche
- Inoltre l'algoritmo può produrre aliasing se si generano pochi VPL:
 - Almeno 256 VPL per gli interni
- In questo caso un numero elevato di VPL è necessario