

Grafica computazionale

Lezione 5

(slide parzialmente basate su
Computer Graphics - MIT OpenCourseWare
Grafica Computazionale - Massimiliano Corsini – Università di Siena)

1

L'Equazione di Rendering

- La luce visibile in un punto della scena per una particolare direzione è data dalla luce riflessa più la luce emessa in quella direzione

$$\begin{aligned}L_o(x, \vec{\omega}_r) &= L_e(x, \vec{\omega}_r) + L_r(x, \vec{\omega}_r) \\ &= L_e(x, \vec{\omega}_r) + \int_{\Omega} f_r(x, \vec{\omega}_i, \vec{\omega}_r) L_i(x, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i\end{aligned}$$

- Negli schemi locali
 - Approssimiamo la BRDF f_r come somma di termini semplici (diffusione, riflessione e ambiente)
 - Consideriamo solo la radiosità incidente L_i da sorgenti luminose puntiformi

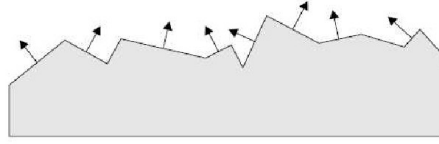
$$\int_{\Omega} f_r(x, \vec{\omega}_i, \vec{\omega}_r) L_i(x, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i = \sum_i f_r(x, \vec{\omega}_i, \vec{\omega}_r) I_i(\vec{n} \cdot \vec{\omega}_i)$$

- Gli oggetti nella scena non emettono luce ($L_e=0$)
- Generalizzazioni
 - Modelli di illuminazione avanzati (Cook-Torrence, Oren-Nayar, Minnaret, BRDF misurate,...)
 - Luci non puntiformi (soft shadows)
 - Integrazione esplicita di L_r (illuminazione globale)
 - Ray Tracing
 - Radiosity
 - Photon Tracing

2

Modelli di Cook-Torrance

- James Blinn (1977) propone un modello di illuminazione basato su un modello della riflessione della luce sviluppato da Torrance e Sparrow.
- Il modello di Torrance-Sparrow assume che la superficie sia composta da un insieme di piccolissime facce (microfacce) che si comportano come piccoli specchi. La distribuzione dell'orientazione delle microfacce determina la componente speculare della luce.



- Cook e Torrance migliorarono il modello di Blinn introducendo due ulteriori principi fisici:
 - Conservazione dell'energia (luce incidente = luce riflessa)
 - Variazioni del colore dello specular highlight

$$R_s = \frac{FDG}{(\vec{N} \cdot \vec{L})(\vec{N} \cdot \vec{V})}$$

- $F \rightarrow$ fattore di Fresnel
- $D \rightarrow$ dipende dalla distribuzione delle microfacce e modella la rugosità del materiale
- $G \rightarrow$ termine che dipende dalla geometria della superficie

3

Il fattore di rugosità (D)

- Il termine D modella la rugosità del materiale utilizzando una distribuzione delle microfacce
- Cook-Torrance utilizzarono la distribuzione di Beckmann-Spizzichino:

$$R_s = \frac{1}{m^2 \cos^2 \alpha} \exp \left(-\frac{\tan^2 \alpha}{m^2} \right)$$

- m è lo slope medio delle microfacce
- α è l'angolo tra la normale N e l'half vector $H = (V+L)/2$
- NOTA: L'half vector è utile ad approssimare l'angolo di riflessione, in particolare:

$$(\vec{R} \cdot \vec{V})^n \approx (\vec{N} \cdot \vec{H})^{4n}$$

Il fattore geometrico (G)

- Il fattore G è un fattore di attenuazione che tiene conto degli effetti di masking e shadowing delle microfacce:

$$G_m = \frac{2(\vec{N} \cdot \vec{H})(\vec{N} \cdot \vec{V})}{(\vec{V} \cdot \vec{H})}$$

Effetto di Masking

Effetto di Shadowing

$$G_s = \frac{(\vec{N} \cdot \vec{H})(\vec{N} \cdot \vec{L})}{(\vec{V} \cdot \vec{H})}$$

$$G = \min(1, G_m, G_s)$$

4

Il fattore di Fresnel

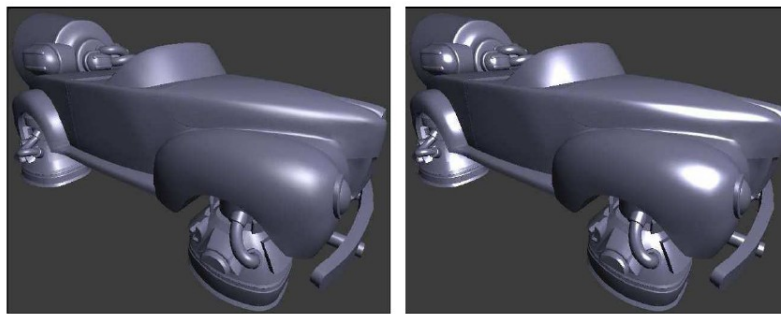
- Il coefficiente di Fresnel tiene conto delle variazioni di colore dello specular highlight in funzione dell'angolo di incidenza e della lunghezza d'onda della luce.
- In generale il colore delle specular highlight dipende dalle proprietà del materiale eccetto quando l'angolo di incidenza della luce è molto basso.
- Formula approssimata: $F = \rho + (1 - \rho)(1 - (\vec{N} \cdot \vec{L}))^5$
-

Conservazione Energia

- Per tenere conto della conservazione dell'energia la luce incidente = luce riflessa
 - Luce riflessa formata da una componente diffusiva (Rd) e da una componente speculare (Rs)

$$K_d R_d + K_s R_s = R$$

- Quindi $K_d + K_s = 1$



Phong

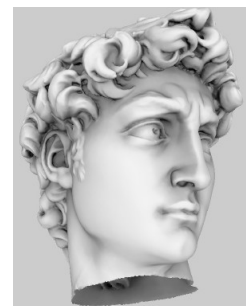
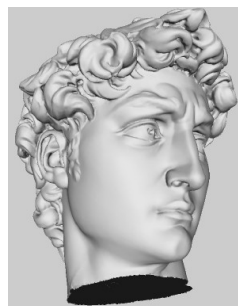
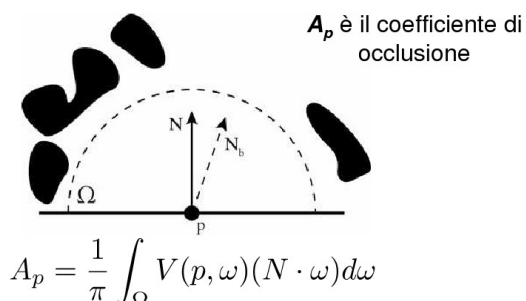
Cook-Torrance

Altri modelli di illuminazione

- Sono stati presentati altri modelli basati su considerazioni fisiche per rappresentare il comportamento di materiali per cui il modello Lambertiano non funziona
- Modello di Oren-Nayar
 - Materiali con effetti di retroriflessione (Porcellana).
 - Questo crea un effetto di “appiattimento” (es. la luna).
- Modello di Minneart
 - Utile per oggetti come il velluto $I_o = K_D I_L ((\vec{N} \cdot \vec{L})(\vec{N} \cdot \vec{V}))^{m-1}$

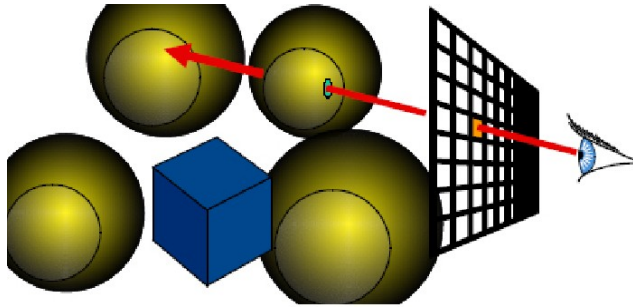
Ambient Occlusion

- Modelli fisici migliorano componenti diffuse e riflessive (l riflessione). Cosa fare per la riflessioni multiple?
 - Phong propone componente ambiente: luce uguale da tutte le direzioni
 - Poco credibile se concavità (forti occlusioni locali)
 - Possiamo calcolare un coefficiente di attenuazione della luce ambientale che prenda in considerazione le occlusioni locali



Ray Casting (Appel 1968)

- Modello view-based di rendering e hidden surface removal:
 - Per ogni pixel, emetti un raggio verso la scena
 - Rileva il primo oggetto colpito dal raggio, accendi il pixel asseconda di questa intersezione

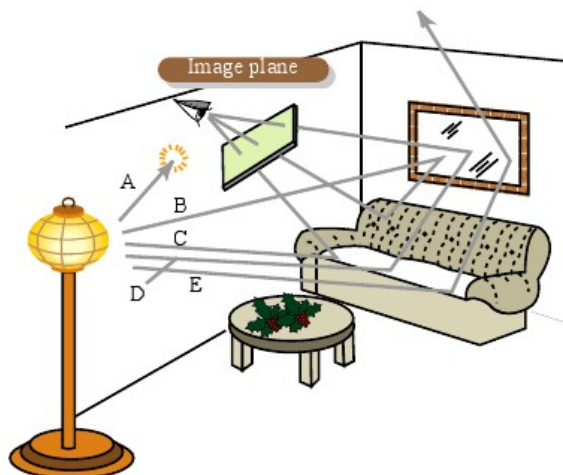


- Si emette solo un raggio e poi si procede con illuminazione locale.
- Possibile aggiungere ombre con un solo raggio secondario verso la luce (primo algoritmo per ombre)
- Rispetto a modello raster:
 - Svantaggi:
 - non sfrutta la coerenza locale dei dati (lento)
 - Vantaggi:
 - altamente parallelizzabile, è possibile fare supersampling solo dove serve
 - Si possono rendere facilmente effetti delle ottiche come distorsioni e fuoco.

7

Recursive Ray Tracing (Whitted 1980)

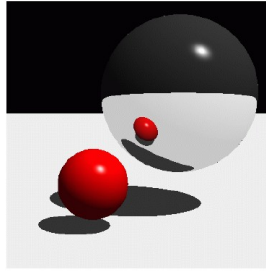
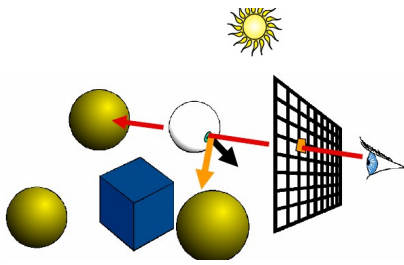
- Le ombre in Ray casting vengono calcolate facendo partire nuovi raggi dal punto alla luce:
 - Se c'è un oggetto tra punto e luce il punto è in ombra (approccio non è più locale!)
- Un approccio simile può essere usato per risolvere riflessioni e rifrazioni:
- Si mandano raggi e si producono nuovi raggi nelle direzioni di riflessione e rifrazione



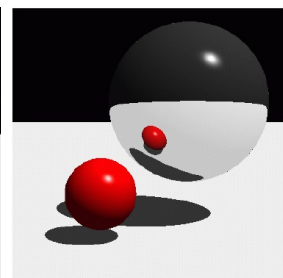
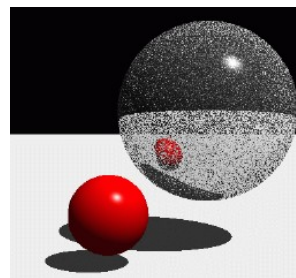
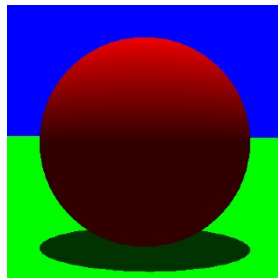
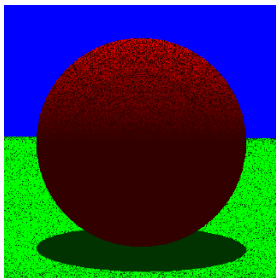
8

Riflessioni

- Emetti raggio in direzione simmetrica rispetto alla normale
- Moltiplica per il coefficiente riflessivo (colore)



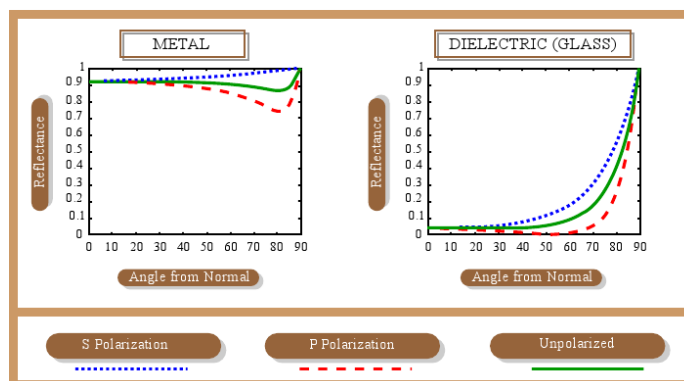
- Non dimenticarsi di aggiungere una epsilon (distanza minima) per evitare auto-ombreggiature/riflessioni



9

Coefficiente di riflessione

- Due modi:
 - Approssimato: costante (molto usato)
 - Più realistico: termine di Fresnel (quantità di riflessione dipende dall'angolo)
 - Approssimazione di Schlick: $R(\theta) = R_0 + (1 - R_0)(1 - \cos \theta)^5$

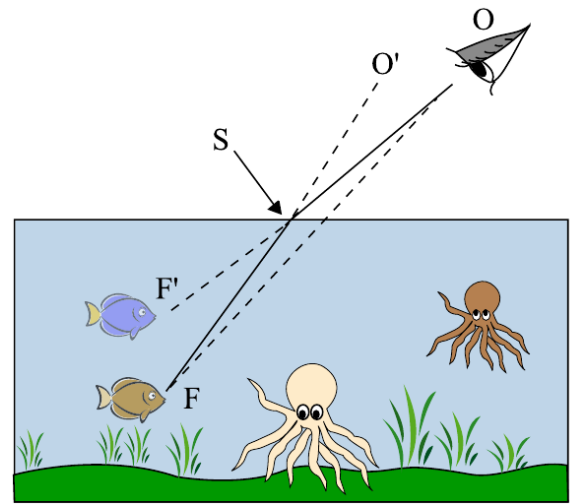
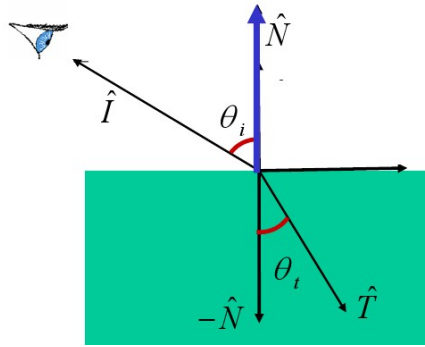


Rifrazione

- Quando la luce attraversa l'interfaccia tra due sostanze trasparenti cambia direzione

- Legge di Snell

$$\frac{\sin \theta_i}{\sin \theta_t} = \frac{\eta_t}{\eta_i}$$



- Da cui $\hat{T} = \left(\eta_r (\hat{N} \cdot \hat{I}) - \sqrt{1 - \eta_r^2 (1 - (\hat{N} \cdot \hat{I})^2)} \right) \hat{N} - \eta_r \hat{I}$
 - Si ha riflessione totale se la radice quadrata è immaginaria

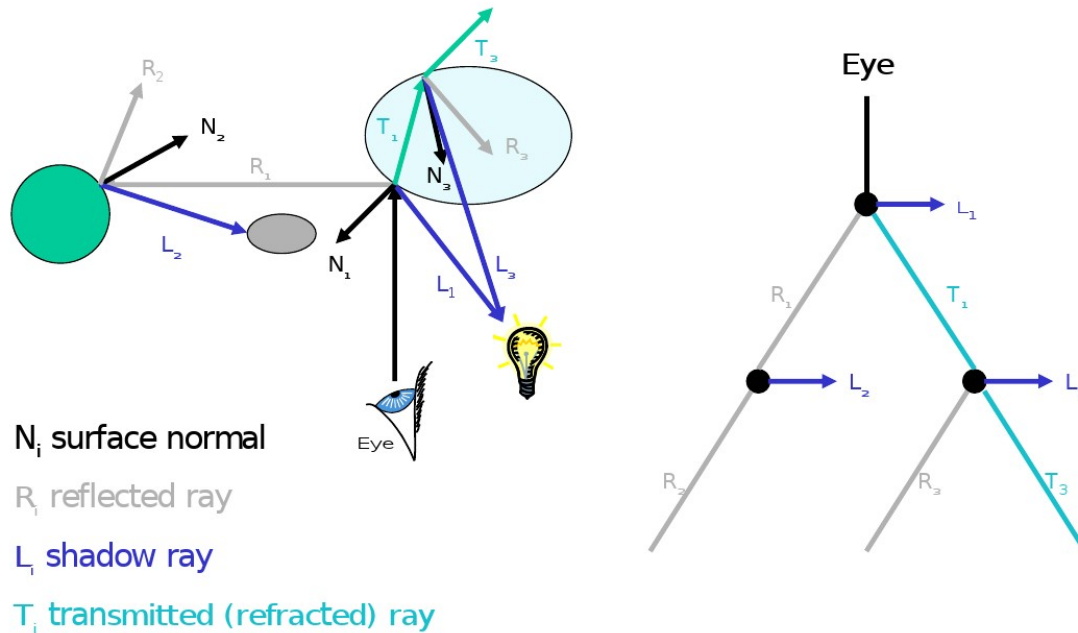
- L'angolo di riflessione è dipendente dalla frequenza (prisma)
 - Di solito ignorato in grafica!

11

Ray tracing ricorsivo

- Per ogni pixel
 - emetti un raggio e trova la prima intersezione con una primitiva (P)
 - Per ogni luce (L)
 - Emetti un raggio di ombra da P verso L e controlla se intercetta qualche primitiva (non serve la prima)
 - Se in luce calcola componente diffusiva
 - Se la superficie in P è riflettente
 - Emetti raggio in direzione del riflesso e **ripeti ricorsivamente**
 - Usa risultato per componente riflessiva
 - Se la superficie in P è trasparente
 - Emetti raggio in direzione rifrazione e **ripeti ricorsivamente**
 - Usa risultato per componente Trasmissiva
 - Restituisci colore pixel come somma componenti diffusiva, riflessiva e trasmissiva (più eventualmente componente ambientale)
- Ci sono due rami che effettuano chiamate ricorsive
- Condizioni di terminazione ricorsione:
 - Superato un certo numero di riflessioni
 - Superata una certa attenuazione

12



13

Ray Tracing Simula il Fenomeno Fisico?

- I fotoni vanno dalla sorgente luminosa agli occhi e non viceversa
 - Principio di reversibilità ottica
- Quello descritto si chiama s backward (eye-based) ray tracing
- Forward (light-based) ray tracing
 - Emettiamo raggi dalla luce => bassa probabilità di colpire gli occhi
 - Ad ogni intersezione mandiamo un raggio verso gli occhi.
- Recursive Ray Tracing è pieno di trucchetti
 - Gestione oggetti opachi
 - Riflessioni solo ideali
- Recursive Ray Tracing e l'equazione di rendering
 - Recursive Ray Tracing risolve l'integrale di radianza incidente come somma di componenti lambertiane per ogni luce (calcolando occlusioni) e due componenti riflessiva e trasmissiva in direzioni ideali

$$\int_{\Omega} f_r(x, \vec{\omega}_i, \vec{\omega}_r) L_i(x, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i = \sum_i I_i(\vec{n} \cdot \vec{\omega}_i) + L_r(x_r, \vec{\omega}_r) + L_t(x_t, \vec{\omega}_t)$$

- BRDF molto limitata ed irrealistica!

14

Monte Carlo Ray Tracing

- Cerchiamo di approssimare l'integrale della radianza incidente per qualsiasi BRDF

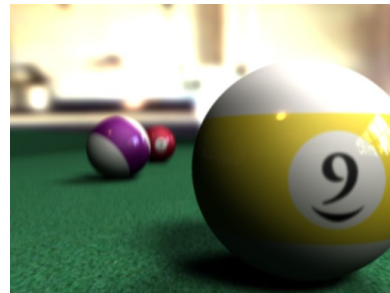
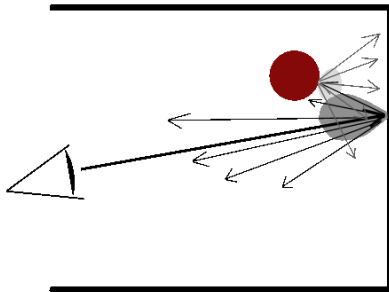
$$\int_{\Omega} f_r(x, \vec{\omega}_i, \vec{\omega}_r) L_i(x, \vec{\omega}_i) (\vec{n} \cdot \vec{\omega}_i) d\vec{\omega}_i$$

- **Integrazione Monte Carlo**

- Vogliamo calcolare la quantità $I = \int f(x) dx$
- Lo rendiamo nell'equivalente calcolo del valore atteso $I = \int \frac{f(x)}{p(x)} dp(x) = E \left[\frac{f(x)}{p(x)} \right]$
- Estrahendo un campione x_1, \dots, x_n con densità $p(x)$, abbiamo

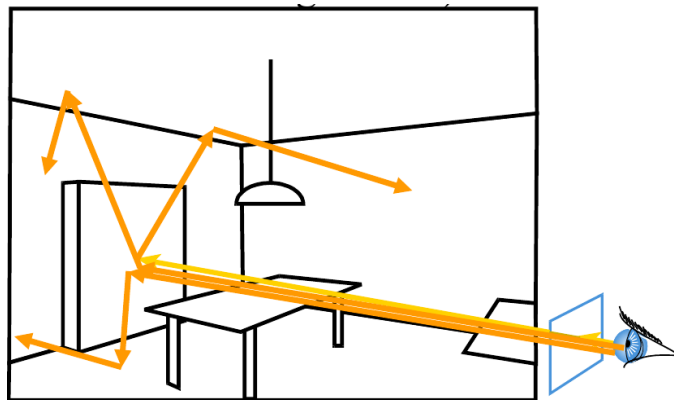
$$I = E \left[\frac{f(x)}{p(x)} \right] \approx \frac{1}{n} \sum_{i=1}^n \frac{f(x)}{p(x)}$$

- Monte Carlo Ray Tracing: Ad ogni intersezione, invece di emettere raggi nelle direzioni di riflessione e trasmissione ideali, emettiamo n raggi in direzioni casuali distribuite come la BRDF



15

- Monte Carlo path tracing
- Abbiamo visto Monte Carlo ray tracing come approssimazione della rendering equation
 - Ad ogni intersezione genera n raggi secondari \rightarrow esplosione esponenziale
- Monte Carlo Path Tracing è una ottimizzazione:
 - Emettiamo più raggi per pixel
 - Ogni intersezione genera solo 1 raggio secondario

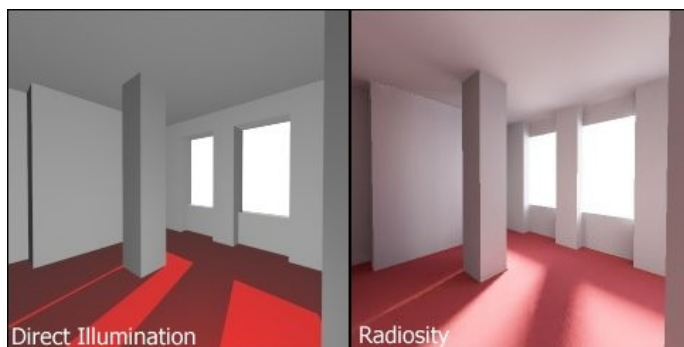


- Numero di raggi limitato a priori
- Antialiasing come effetto collaterale!

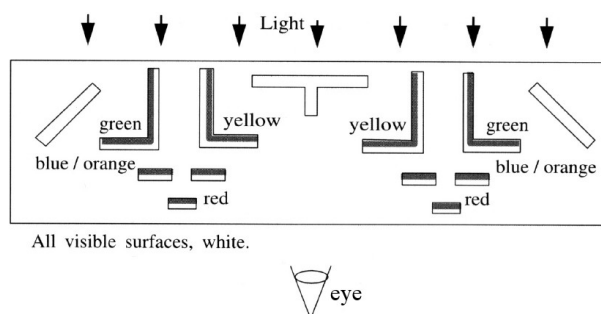
16

Radiosity

- Radiosity cerca di risolvere il problema del rendering come soluzione del trasporto energetico della luce
- Affronta il problema delle interriflessioni multiple tra superfici come equilibrio tra radiazione entrante ed uscente



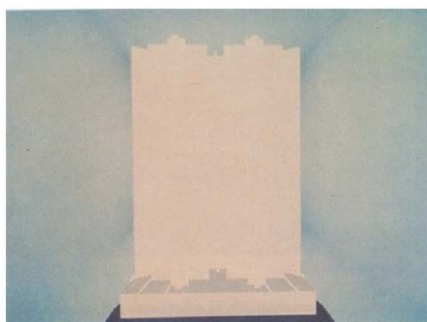
- Inter-riflessioni multiple possono dare effetti cromatici importanti
 - Diffuse Panels (John Ferren)



Radiosity Vs. Ray Tracing



Original sculpture by John Ferren lit by daylight from behind.



Ray traced image. A standard ray tracer cannot simulate the interreflection of light between diffuse surfaces.

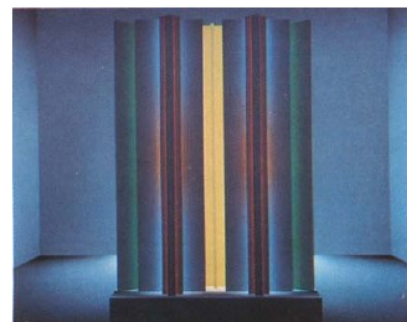


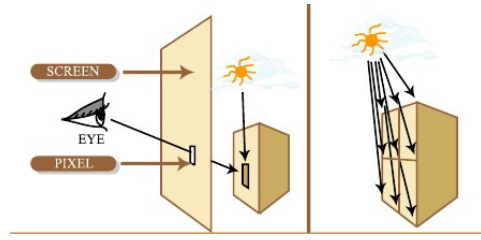
Image rendered with radiosity. note color bleeding effects.

- Anche Ray Tracing usa componente ambientale per simulare interazioni multiple che non è in grado di calcolare direttamente



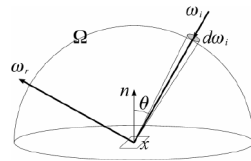
Radiosity e Rendering Equation

- Radiosity è un algoritmo object-space che calcola esplicitamente le interazioni energetiche tra le primitive

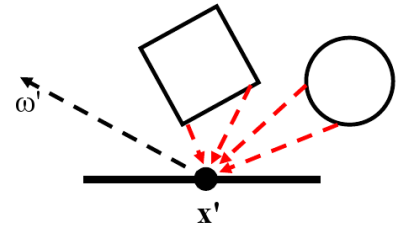


- Finora abbiamo espresso la Rendering Equation come un integrale su tutte le direzioni del contributo energetico

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



- Per Radiosity rendiamo esplicita l'interazione tra le primitive
- Accumuliamo dalle superfici anziché dalle direzioni



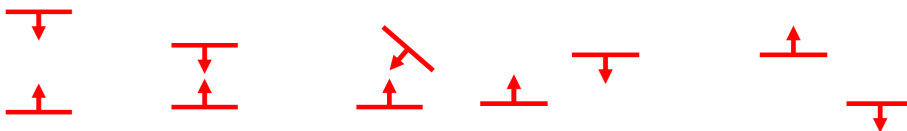
19

Radiosity e Rendering Equation

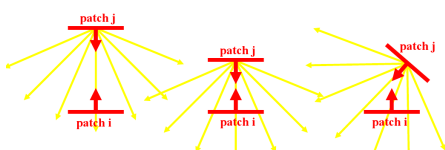
- Assumiamo n primitive A_1, \dots, A_n
- L'equazione di rendering può essere riscritta come

$$L_o(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \sum_{i=1}^n \int_{A_i} f_r \left(x, \frac{y-x}{\|y-x\|}, \vec{\omega}_r \right) L(x, y) G(x, y) V(x, y) dy$$

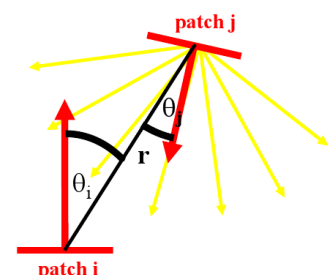
- $L(x, y)$ = radianza da y verso x
- $V(x, y)$: termine di visibilità (1 se x è visibile da y, 0 se è occluso).
- $G(x, y)$: relazione geometrica tra le superfici in x e y
- Quale distribuzione di superfici dà il maggiore trasferimento di energia luminosa e perché?



- Dobbiamo calcolare quanta energia emessa da una patch colpisce l'altra



- Con un po' di calcoli abbiamo $G(x, y) = \frac{\cos \theta_i \cos \theta_j}{\pi r^2} dA_i dA_j$



20

Assunzioni ed Approssimazioni

- Radiosity per poter risolvere l'equazione di rendering fa alcune approssimazioni

1) Superfici perfettamente Lambertiane:

- BRDF è costante
- termine $L(x,y)$ di radianza da y a x dipende solo dalla Radiosità $B(y)$ emessa da y

$$L_o(x, \vec{\omega}_r) = L_e(x, \vec{\omega}_r) + \sum_{i=1}^n \int_{A_i} \rho(x) L(x, y) G(x, y) V(x, y) dy$$

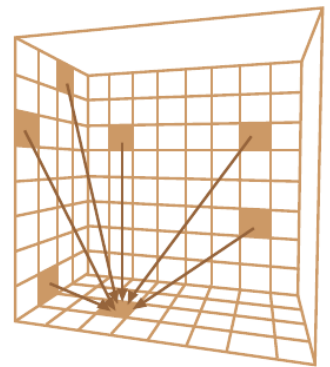
$$B(x) = E(x) + \rho(x) \sum_{i=1}^n \int_{A_i} B(y) \underbrace{G(x, y) V(x, y)}_{F(x, y)} dy$$

Form factor

- 2) Approssimiamo la funzione di radianza come una funzione costante per ogni patch di superficie (approssimazione agli elementi finiti)

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

- Ridotto ad una equazione lineare nei valori di radiosità B_i



21

Soluzione dell'equazione di radiosità

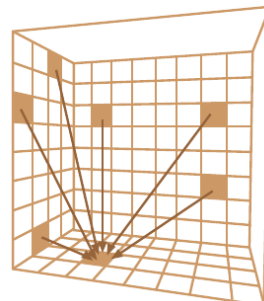
- La radiosità delle patch corrisponde alla soluzione del sistema lineare

$$\begin{bmatrix} 1 - \rho_1 F_{11} & -\rho_1 F_{12} & \cdots & -\rho_1 F_{1n} \\ -\rho_2 F_{21} & 1 - \rho_2 F_{22} & & \\ \vdots & & \ddots & \\ -\rho_n F_{n1} & \cdots & \cdots & 1 - \rho_n F_{nn} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_n \end{bmatrix}$$

- Dimensione molto grande → approcci iterativi

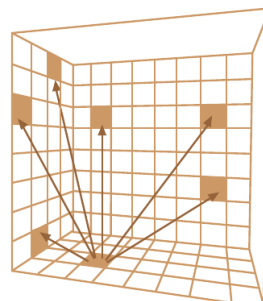
- 1) Raccogliamo la radianza per ogni singola patch (equivalente a rilassamento di Gauss-Siedel)

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_i \\ \vdots \\ E_n \end{bmatrix} + \begin{bmatrix} \rho_1 F_{1i} & \rho_1 F_{1i} & \cdots & \rho_1 F_{1i} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix}$$

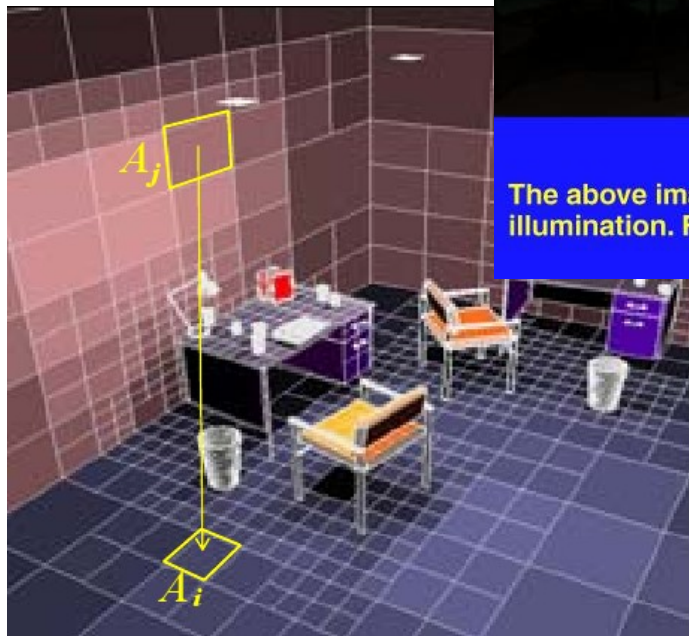


- 2) Raffinamento progressivo: distribuiamo la radianza da ogni singolo patch (equivalente a rilassamento di southwell)

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix} + \begin{bmatrix} \cdots & \rho_1 F_{1i} & \cdots \\ \cdots & \rho_2 F_{2i} & \cdots \\ \vdots & \vdots & \ddots \\ \cdots & \rho_n F_{ni} & \cdots \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_i \\ \vdots \\ B_n \end{bmatrix}$$



22



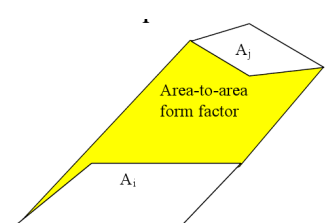
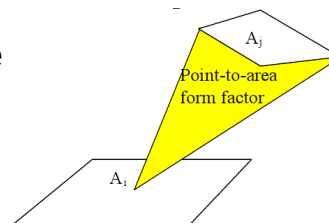
PROGRESSIVE SOLUTION

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

23

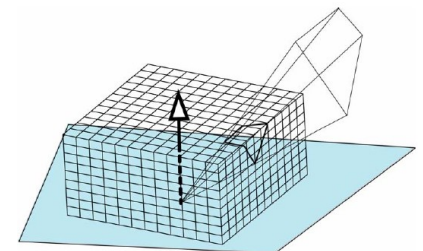
Calcolo delle Form Factor

- Le form factor possono essere calcolate da punto a superficie o da superficie a superficie



Algoritmo del semicubo

- Un semicubo è costruito attorno alla patch
- Le facce del semicubo vengono divise in "pixel" con le loro form factor pre-calcolate
- Tutte le altre patch vengono rasterizzate su queste facce
- La form factor per la patch è la somma delle form factor dei pixel su cui viene proiettata
- La visibilità viene risolta per mezzo dell'algoritmo z-buffer

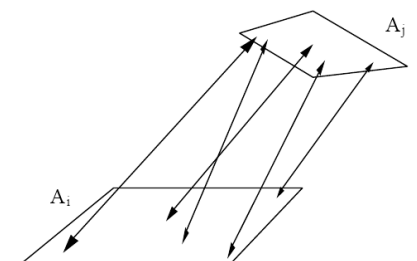


Algoritmo della semisfera

- Si utilizza una proiezione particolare che proietta su di una sfera invece che su di un semicubo

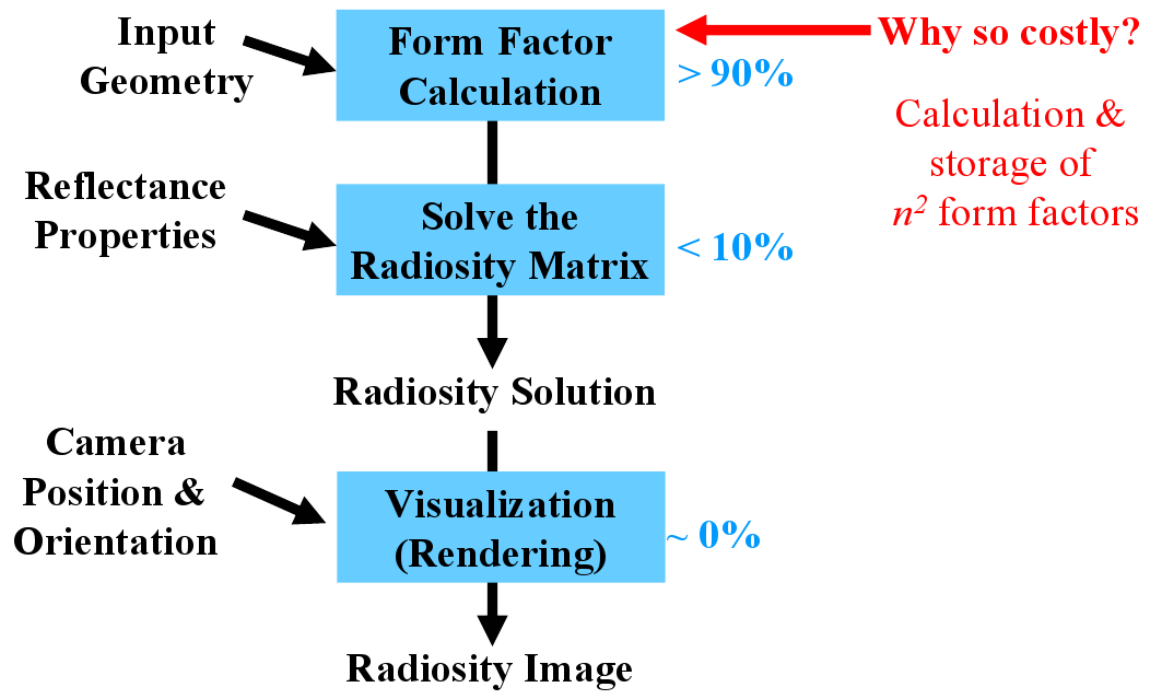
Approccio Ray Casting

- Si emettono n raggi da una superficie e l'altra
 - Per ciascuno si calcola la visibilità
 - Si integrano le form factor ottenute da ciascun raggio
- Permette il calcolo delle form factor da superficie a superficie



24

Stadi nel calcolo della Radiosità

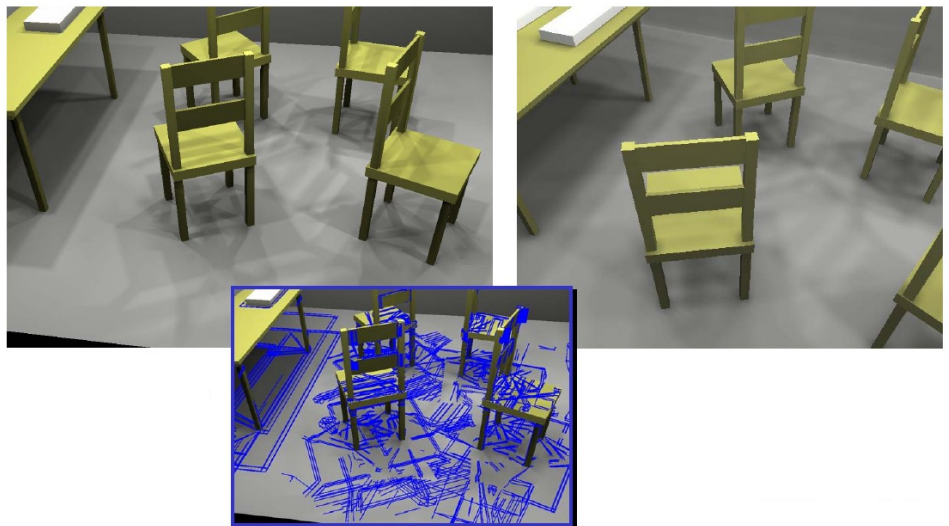
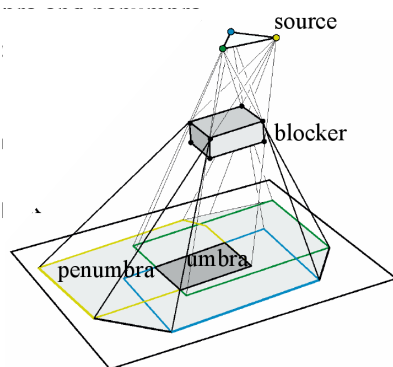


-
- Caratteristica fondamentale di radiosity è che è invariante dal punto di vista: Data una scena possiamo pre-computare la radianza di ogni patch e poi generare immagini da nuovi punti di vista in tempo reale
- Molto utilizzato per walkthrough architettonici

25

Limiti ed Estensioni

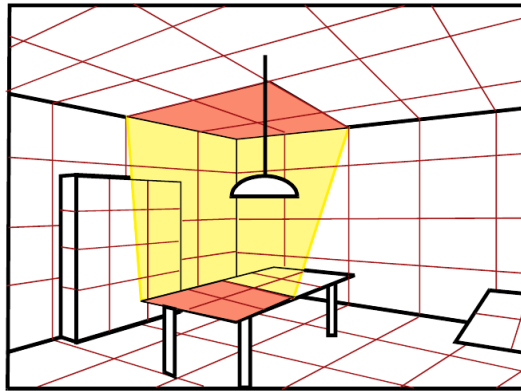
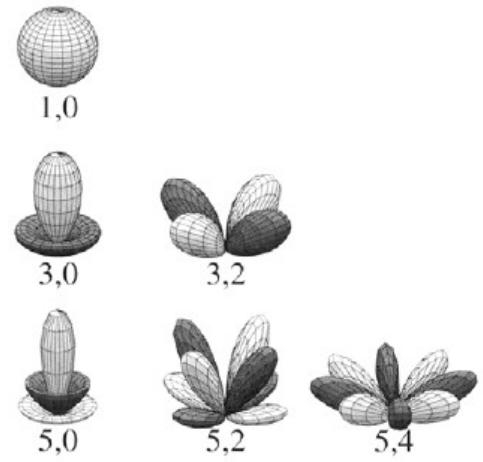
- Radiosity offre solo componente lambertiana
- Ha problemi con cambiamenti rapidi di illuminazione (ombre) dovuto alla divisione in patch
- Quest'ultimo può essere affrontato con le tecniche di discontinuity meshing che cercano di dividere le mesh lungo le linee di discontinuità dell'illuminazione
 - Aumentano le patch e complessità cresce con il quadrato



26

Estensioni

- Le componenti speculari diffuse si possono gestire associando ad ogni patch non una costante di radiosità, ma una funzione angolare
 - Uso di armoniche sferiche come basi funzionali
- Approcci gerarchici vengono usati per ridurre la complessità
 - Raggruppa patch per cui lo scambio di luce non è importante



27

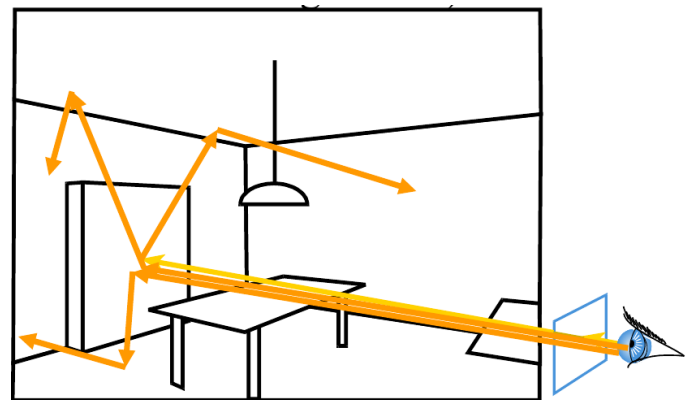
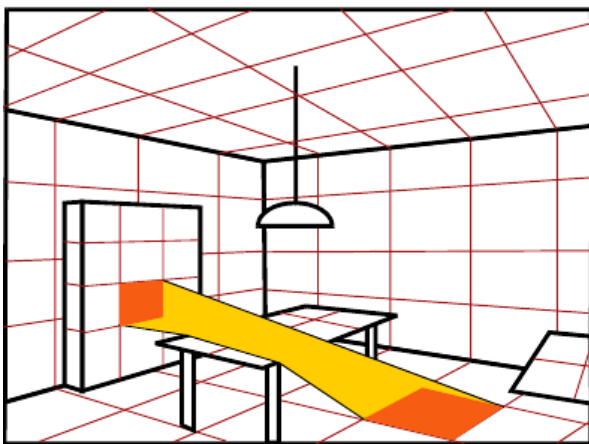
Radiosity Vs. Ray/Path Tracing

Radiosity

- Elementi finiti (proietta su patch discrete)
- Soluzione attraverso un sistema lineare
- Costruzione sistema costoso
- Indipendente al punto di vista (niente informazione angolare)

Path Tracing

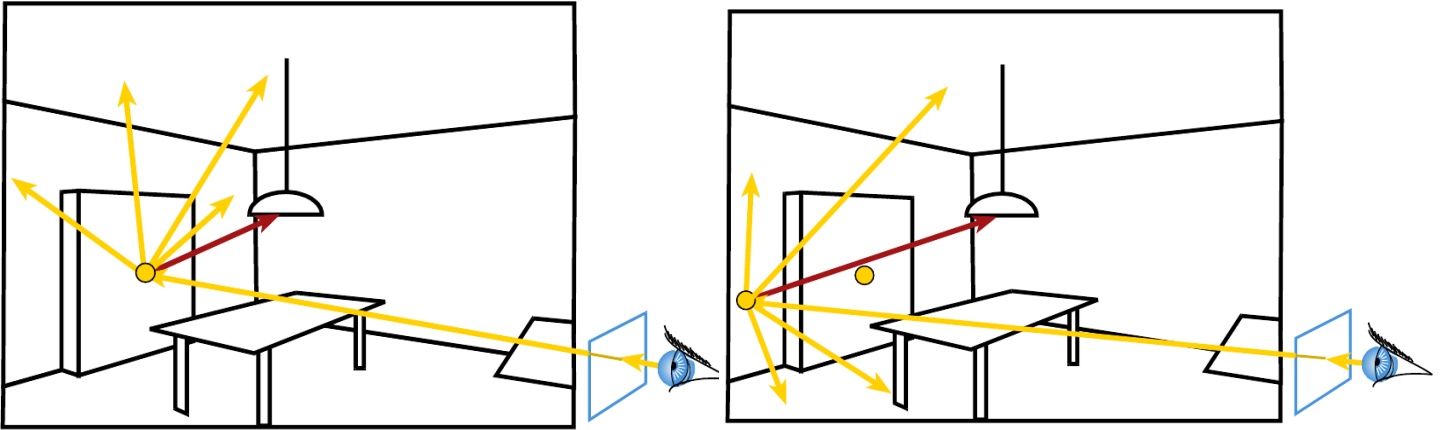
- Campionamento stocastico
- Integrazione Monte Carlo
- Convergenza lenta (richiede molti raggi)
- Dipende dal punto di vista



28

Perché Radiosity funziona bene?

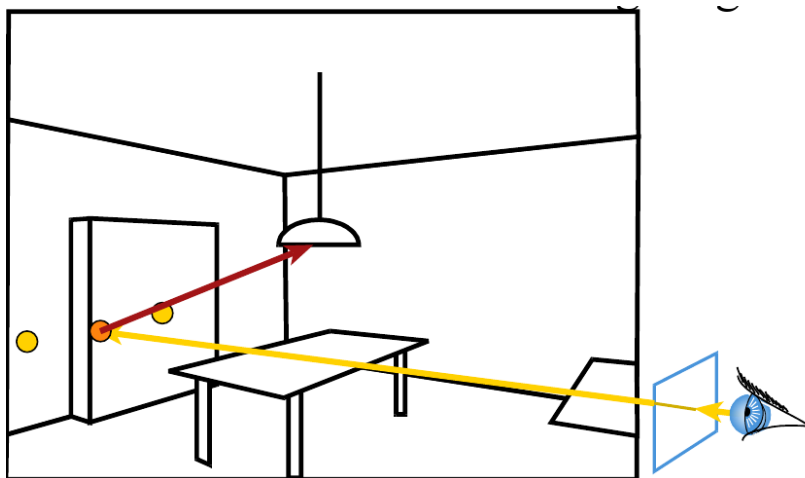
- Il limite di risoluzione delle patch sembra molto forte. Eppure le immagini prodotte da Radiosity sono molto realistiche
- La illuminazione indiretta è molto “smooth”: cambia poco tra posizioni vicine perché queste accumulano sostanzialmente dalle stesse sorgenti



29

Irradiance cache

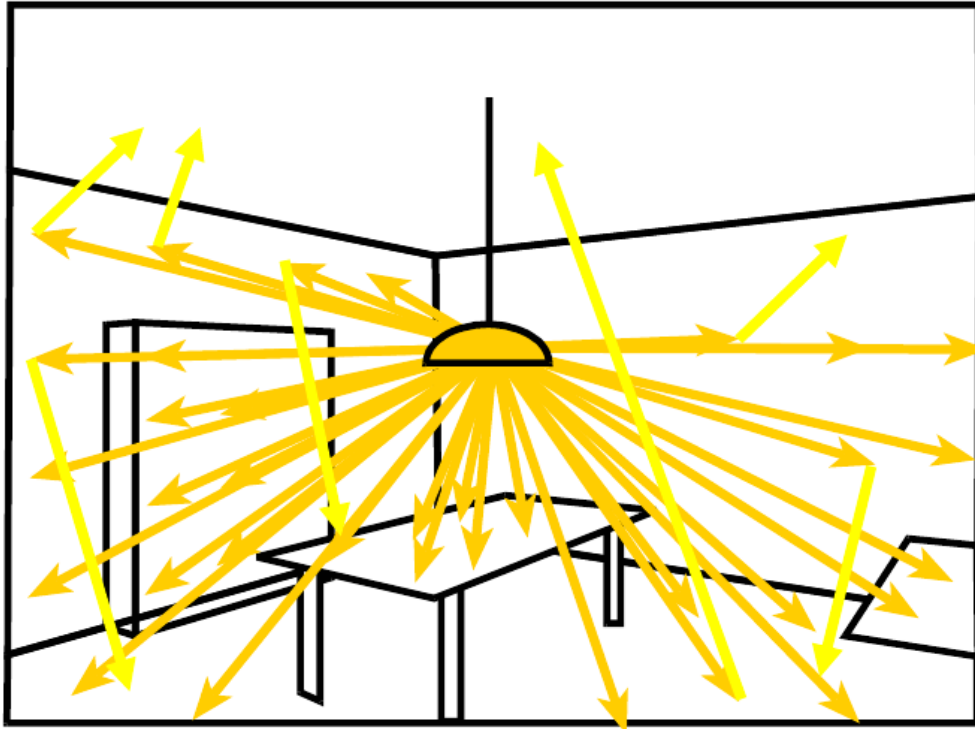
- Campioniamo la illuminazione indiretta
- Interpoliamo la illuminazione indiretta dai cache di campioni
- Facciamo la computazione diretta per la illuminazione diretta (e riflessa/rifratta ideale)



30

Photon Mapping: Preprocessamento

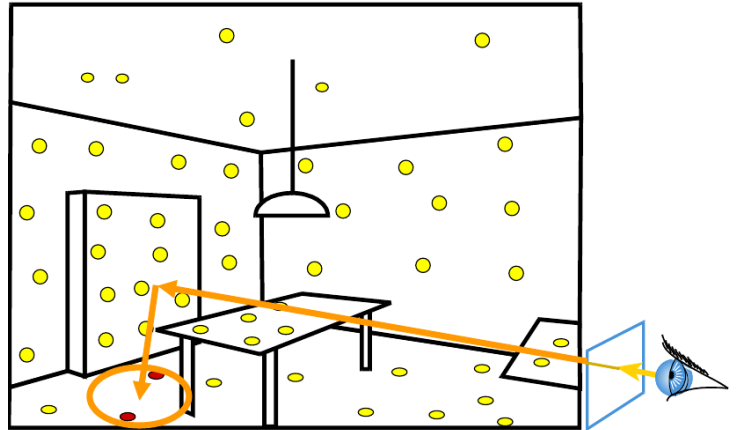
- Emettiamo raggi dalle sorgenti luminose
- Li salviamo in una mappa dei fotoni (photon map)
 - (posizione + potenza + direzione d'arrivo)



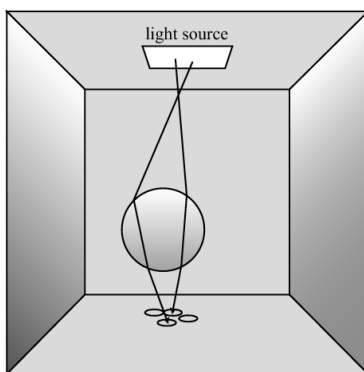
31

Photon Mapping : Rendering

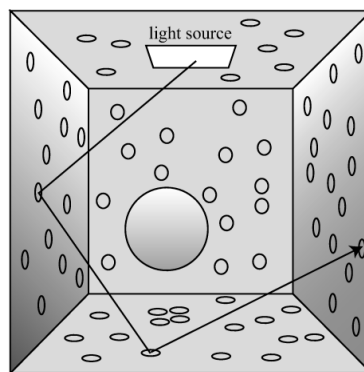
- Emettiamo raggi primari
- Per i raggi secondari
 - Ricostruiamo la radianza usando i fotoni salvati vicini
 - Prendiamo i k fotoni più vicini



- Caustics
- Manteniamo una mappa speciale per elementi rifrangenti/riflettenti



Caustic Map



Global Photon Map

32

