

---

# Fondamenti di Grafica Tridimensionale

Paolo Cignoni

p.cignoni@isti.cnr.it

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

# 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]*

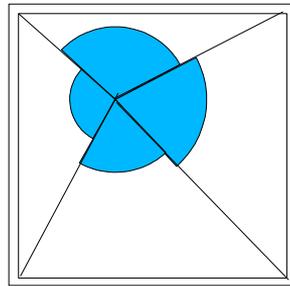
$$I(x, x') = g(x, x') \left[ \varepsilon(x, x') + \int_s \rho(x, x', x'') I(x', x'') dx'' \right]$$

# R.E. e Radianza

- ❖ La Radianza  $L$  e' una delle 8 unità fisiche radiometriche del SI.
- ❖ Misura l'intensità di un raggio luminoso definita come la potenza per angolo solido unitario per area proiettata.
- ❖ Si misura in  $Watt/(sr \cdot m^2)$ 
  - ❖ Sr sono steradiani

# Radianza e Irradianza

- ❖ Consideriamo funzione di distribuzione della radianza per un punto
  - ❖ Generalmente molto discontinua



- ❖ Irradianza e' l'integrale della radianza su tutte le direzioni (pesato con il cos della direzione)

$$E = \int_{\Omega} L(\omega) \cos \theta d \omega$$

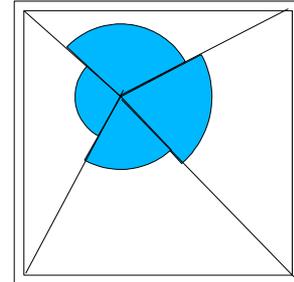
# Radianza e Irradianza

- ❖ Irradianza e' l'integrale della radianza su tutte le direzioni (pesato con il cos della direzione)

$$E = \int_{\Omega} L_{in}(\omega) \cos \theta d\omega$$

$\theta$  è l'angolo tra la normale alla sup e  $\omega$ .

$L_{in}(\omega)$  è la radianza in ingresso da  $\omega$



# Radianza e Radiosity

---

## ❖ Radianza

units: power / (area  $\times$  solid angle)

nel vuoto la radianza lungo un raggio e' costante

❖ **Radiosity** energia per unita' di area che lascia una superficie per unità di tempo

❖ units: power / area

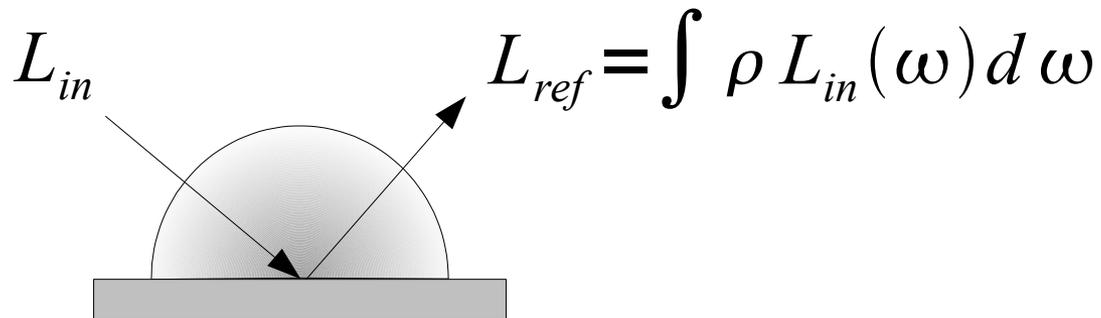
# R.E. e Radianza

- ❖ RE ridotta al minimo:

$$L_{ref} = \int_{\Omega} \rho L_{in}(\omega) d\omega$$

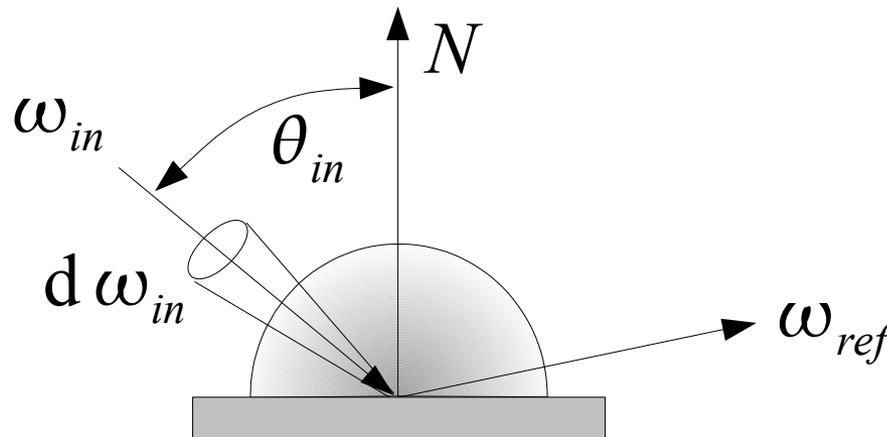
- ❖  $\rho$  È il termine di scattering.

Rappresenta la quantità di energia (luce) che rimbalza da una direzione verso un'altra.



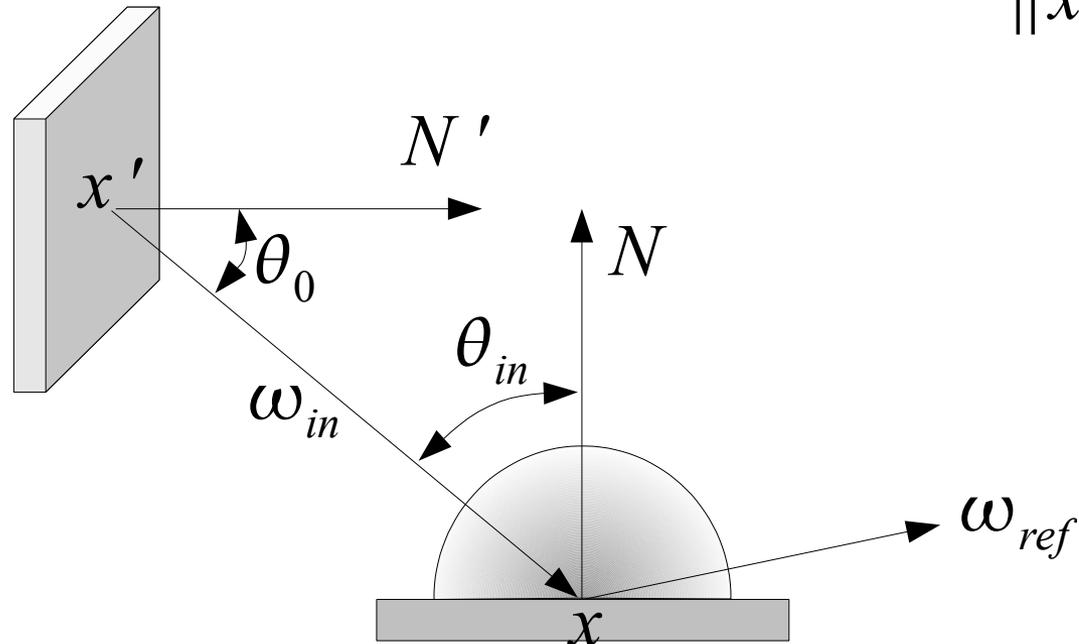
# Con variazione direzionale

$$L_{ref}(x, \omega_{ref}) = L_e(x, \omega_{ref}) + \int_{\Omega} \rho(x, \omega_{in} \rightarrow \omega_{out}) L_{in}(x, \omega_{in}) \cos \theta d\omega_{in}$$



# Con occlusione

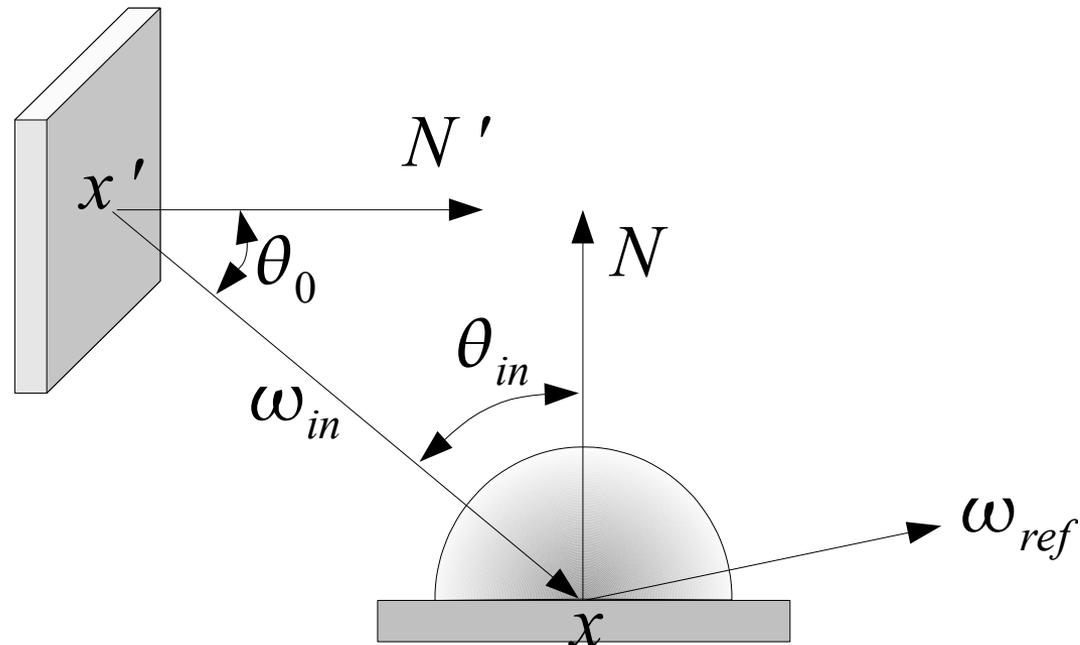
$$L_{ref}(x, \omega_{ref}) = L_e(x, \omega_{ref}) + \int_S \rho(x, \omega_{in} \rightarrow \omega_{out}) L_{in}(x, \omega_{in}) g(x, x') \cos \theta_{in} \frac{\cos \theta_0 dA}{\|x - x'\|^2}$$



# Con occlusione 2

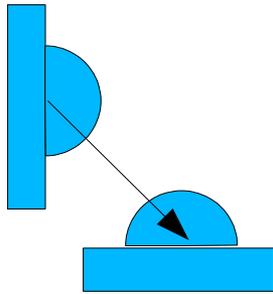
❖ Da notare il cambio 
$$d\omega = \frac{\cos\theta_0 dA}{\|x - x'\|^2}$$

che permette di integrare sulle sup della scena anziché sugli angoli

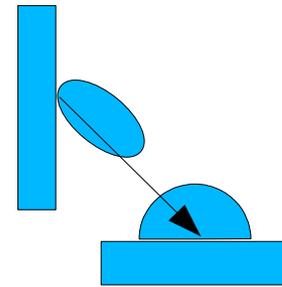


# Caratterizzazione Light Path

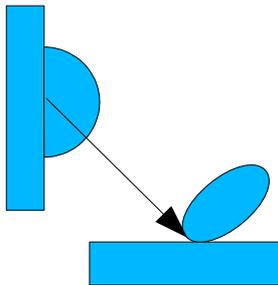
## ❖ Diffuse Diffuse



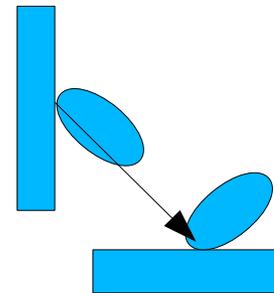
## ❖ Specular Diffuse



## ❖ Diffuse Specular



## ❖ Specular Specular



# Light Paths

---

- ❖ Utile per caratterizzare i vari alg di Global Illumination
- ❖ Si considera il path dalla luce all'eye e tutte le interazioni con le superfici che incontra
- ❖ Ogni path si puo' scrivere come
  - ❖  $L(S|D)^*E$

# Light Paths



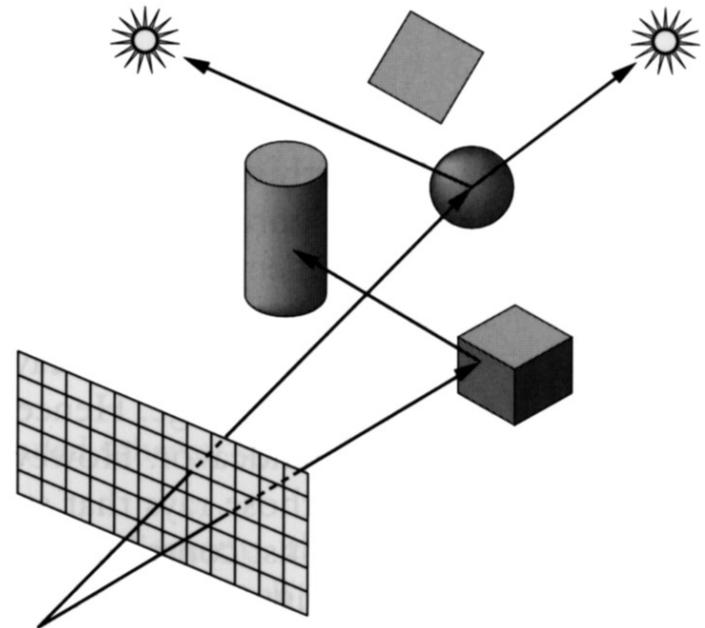
# Light Paths

- ❖ LDE
- ❖ LDDDE
- ❖ ecc.



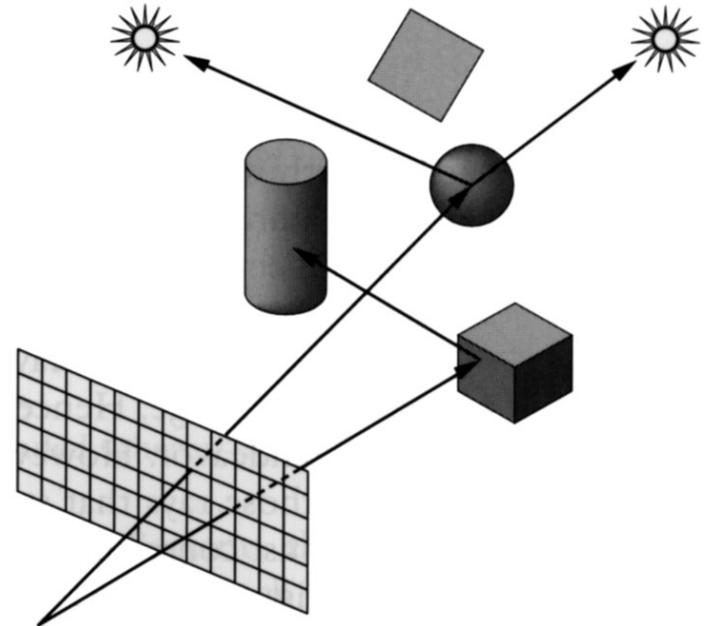
# GI History: Ray Tracing

- ❖ Ray Tracing classico
  - ❖ noto come Whitted Ray-Tracing
- ❖ Si tracciano light ray in direz. contraria
  - ❖ From eye back into the scene toward the light source
  - ❖ Da un punto di vista global illumination, modella solo la riflessione speculare pura



# GI: Ray Tracing

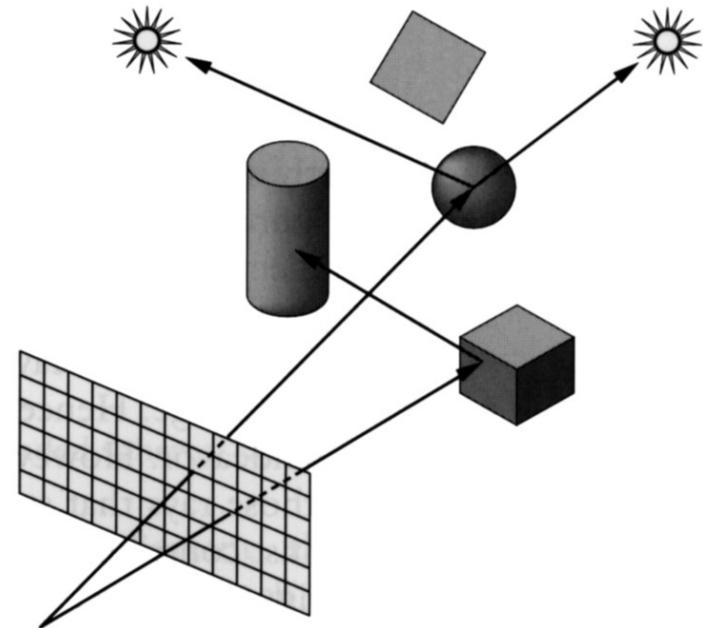
- ❖ Ricorsivo, si genera nuovi raggi per ogni sup speculare che si incontra
- ❖
- ❖ Simula perfettamente
  - ❖ LDS\*E, LS\*E
- ❖ Ma non
  - ❖ DD (diffuse inter reflection)



# GI: Ray Tracing

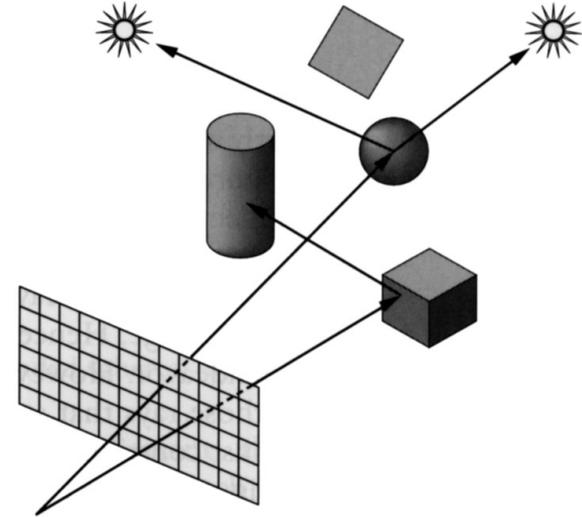
## ❖ In termini di Rendering Equation:

- ❖ La scattering function  $e'$  è ridotta alla legge della riflessione (rifrazione) perfetta
- ❖ L'integrale sulla scena  $S$  si riduce quindi a calcolare un singolo raggio uscente per ogni riflessione + il raggio verso la luce per approssimare la componente diffusa



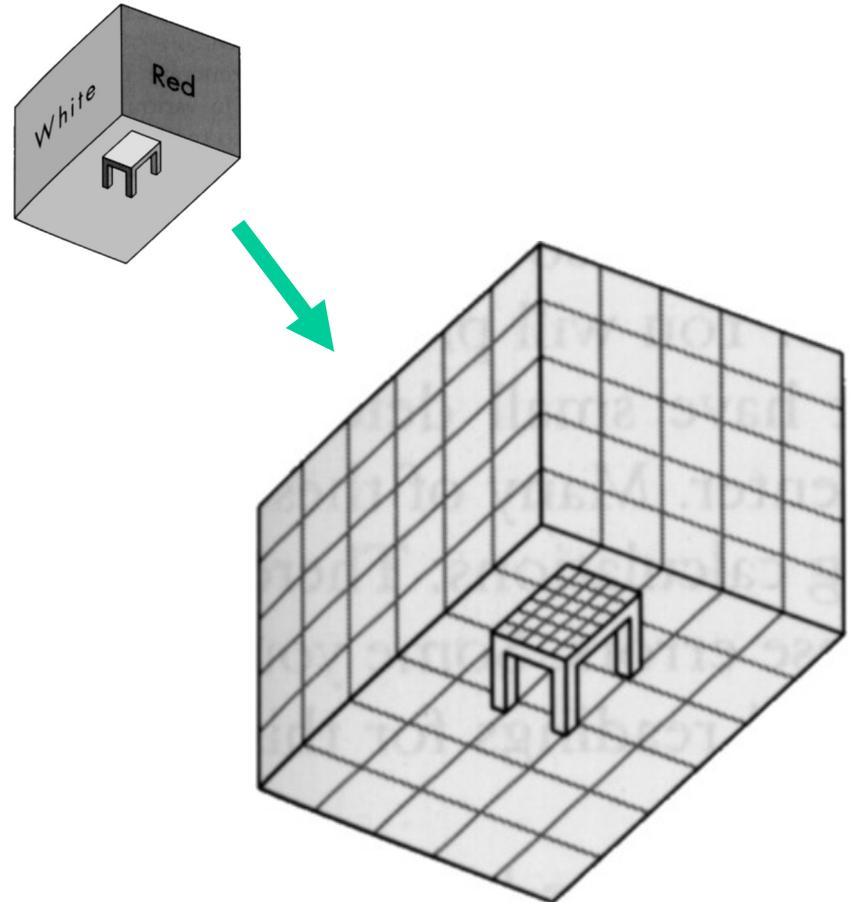
# GI: Ray Tracing

- ❖ In termini di Rendering Equation:
  - ❖ L'integrale sulla scena  $S$ , rimane ricorsivo, ma si riduce a una somma di tre componenti
    - ❖ Diffuse
    - ❖ reflective
    - ❖ transmittive



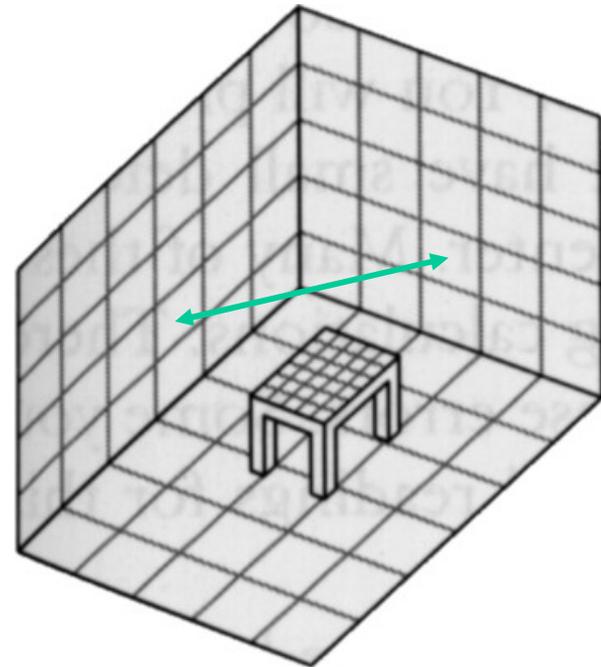
# Radiosity

- ❖ Nel metodo radiosity la scena viene suddivisa in *pezze* (*patches*), ovvero in molti poligoni piatti e di dimensioni limitate, ciascuno dei quali è considerato perfettamente diffusivo



# Radiosity

- ❖ Il metodo prevede due passi per determinare le gradazioni di colore da assegnare alle varie pezze
- ❖ Il primo passo consiste nel determinare, per ogni coppia di pezze, i **fattori di forma (form factor)**, che descrivono come la luce che lascia una pezza influenza l'altra

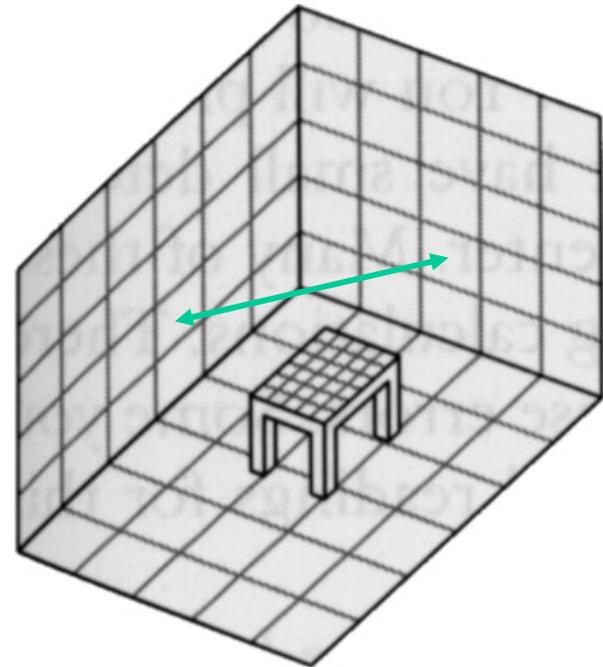


# Form Factor

- ❖ In pratica I form factor definiscono quanta parte dell'energia che esce da una patch arriva su un'altra patch, tenendo in considerazione di occlusioni, orientamento delle patch, distanza ecc.
- ❖ Il calcolo dei form factor delle patch di una scena sarebbe inerentemente quadratico, ma la maggior parte dei ff sono praticamente nulli.
  - ❖ Patch lontane non si influenzano.

# Radiosity

- ❖ Una volta determinati i fattori di forma, sapendo quali patch emettono luce, capire come si distribuisce la luce all'interno di una scena corrisponde ad un sistema di equazioni lineari:
- ❖ La somma delle quantità di luce (radianza) che arrivano su una patch deve essere uguale alla radianza che esce più la luce assorbita dalla patch stessa



# GI: Radiosity

- ❖ Considera solo DD
  - ❖  $LD * E$
- ❖ Invece di seguire raggi considera l'interazione tra porzioni della scena
- ❖ Soluzione view independent
  - ❖ Il rendering del risultato e' a parte.
  - ❖ La scena deve essere discretizzata a priori
    - ❖ Mentre nel ray tracing la discretizzazione era implicita nel pixelsampling

# GI: Radiosity

- ❖ In termini di L.E.

$$B(x') dx' = \varepsilon(x') dx' + \rho(x') \int_S B(x) F(x, x') dx$$

Espressa come radiosity  $B(x)$

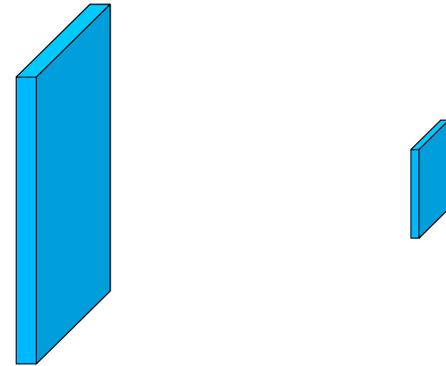
- ❖ energia emessa per unita' di tempo per unita' di superficie
- ❖ Form Factor  $F(x, x')$ 
  - ❖ Coefficiente funzione solo della relazione spaziale tra  $x$  e  $x'$  e determina la frazione di  $B(x')$  che arriva su  $x$ .

# GI: Radiosity: form Factor

## ❖ Principio di reciprocità

$$F(i, j) A_i = F(j, i) A_j$$

$$F(i, j) = F(j, i) \frac{A_j}{A_i}$$



## ❖ I form factor includono anche occlusione

# GI: Radiosity

❖ In termini di L.E.

$$B(x') dx' = \varepsilon(x') dx' + \rho(x') \int_S B(x) F(x, x') dx$$

La soluzione viene trovata discretizzando l'integrale in una sommatoria

❖ Patches discrete invece di porzioni infinitesime della superficie

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

# GI: Radiosity

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

❖ Sfruttando il principio di reciprocità dei FF, divido per  $A(p_i)$

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

# GI: Radiosity

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

- ❖ I tre problemi della radiosity
  - ❖ Trovare la discretizzazione in patches
    - ❖ View independent
    - ❖ Solution dependent
  - ❖ Calcolare  $F(i,j)$  per tutte le coppie di patch
  - ❖ Risolvere il sistema

# GI: Radiosity

$$B(p_i) = \varepsilon(p_i) + \rho(p_i) \sum_{j=1}^n B(p_j) F(i, j)$$
$$B_i = E_i + R_i \sum_{j=1}^n B_j F_{ij}$$

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

# Radiosity

- ❖ Gauss-Siedel converge velocemente
  - ❖ La somma dei ff su ogni riga  $< 1$  per def.
  - ❖ (ma in ogni caso ho bisogno di tutti i FF...)

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

# Progressive Radiosity

---

- ❖ Approccio iterativo: Due obiettivi
  - ❖ Evitare di calcolarsi e memorizzarsi tutti i FF subito
  - ❖ Avere una soluzione approx subito che poi migliora
  
- ❖ Foreach iteration
  - ❖ Select a patch  $i$
  - ❖ Calculate  $F_{ij}$  for all patch  $j$
  - ❖ For each patch  $j$ 
    - ❖ Update radiosity of patch  $j$
    - ❖ Update emission of patch  $j$
  - ❖ Set emission of patch  $j$  to zero

# Progressive Radiosity

---

## ❖ Shooting

- ❖ Ad ogni iterazione ogni patch contiene una quantità di unshot radiosity
  - ❖ Differenza tra una iterazione e la successiva.
- ❖ Si sceglie sempre la patch con il max di tale valore

# Progressive Radiosity

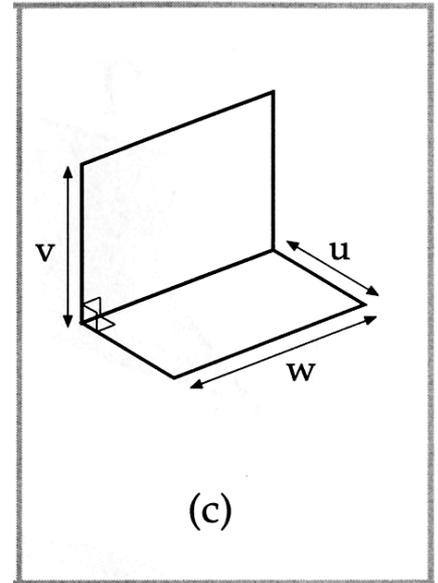


## PROGRESSIVE SOLUTION

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

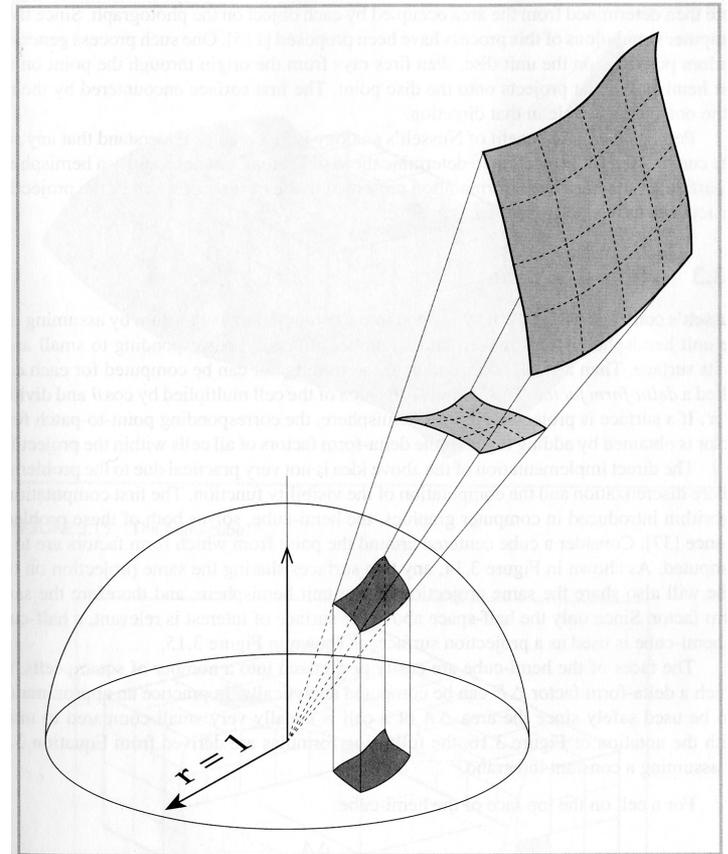
# Form Factor

- ❖ In casi molto semplici esiste una forma chiusa
  - ❖ Caso rect rect che si vedono perfettamente
  - ❖ In generale sono risultati solo teorici...



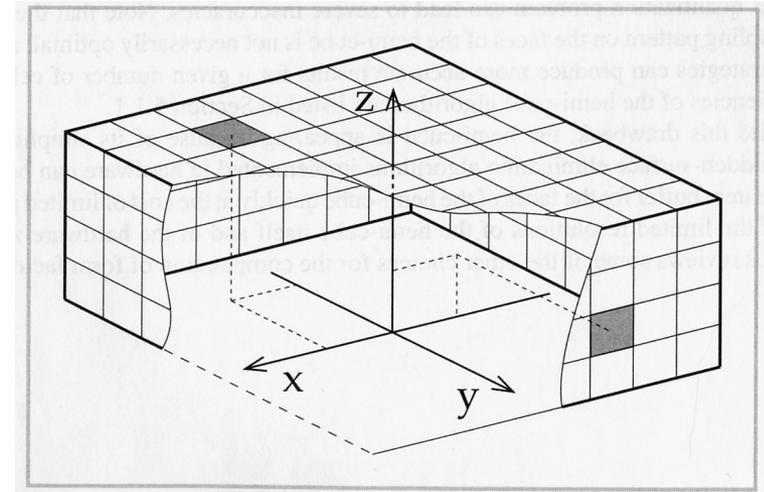
# Form Factor

- ❖ 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 e' la somma pesata del numero di raggi che colpisce quella patch

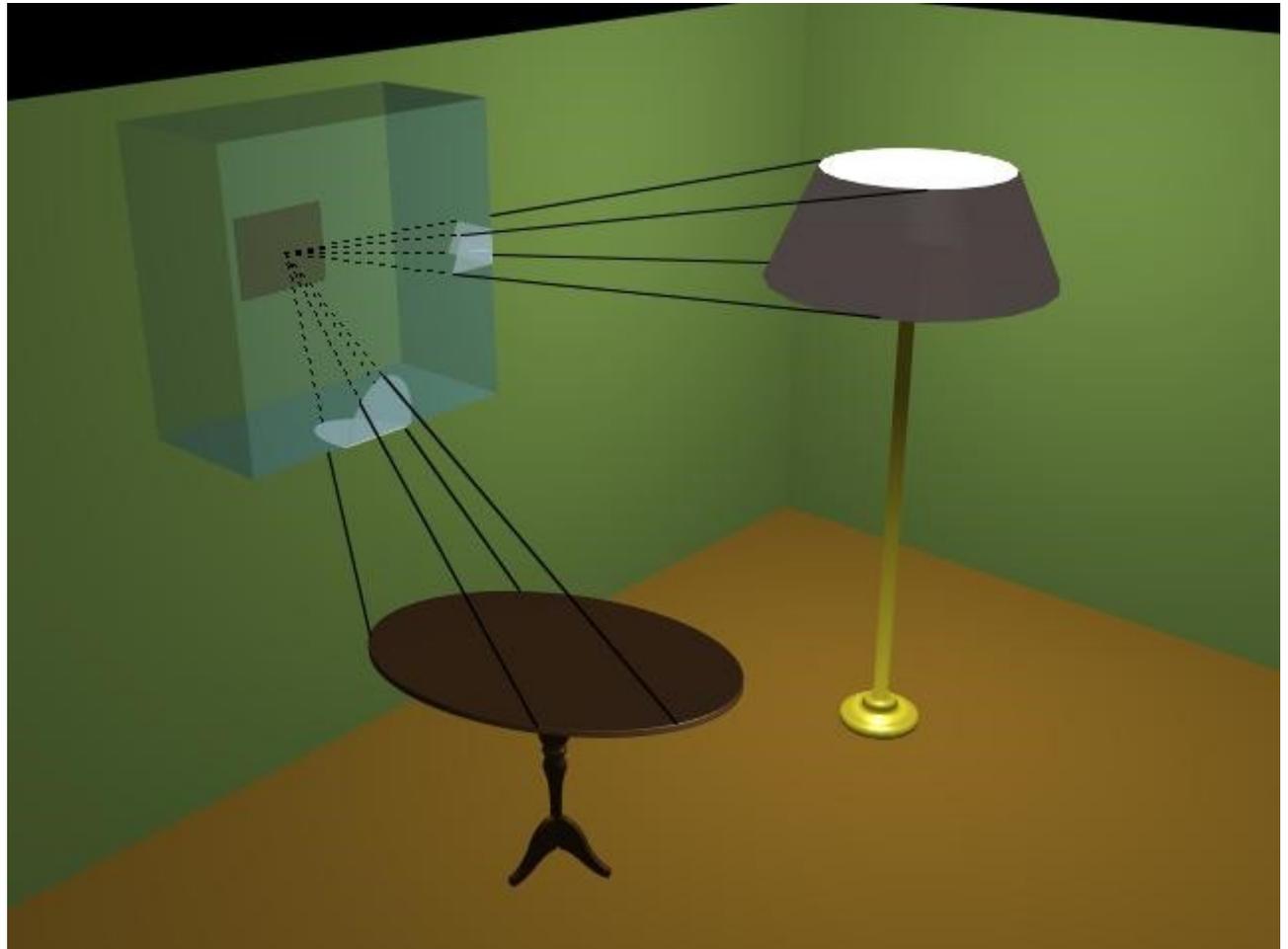


# FF: Hemicube

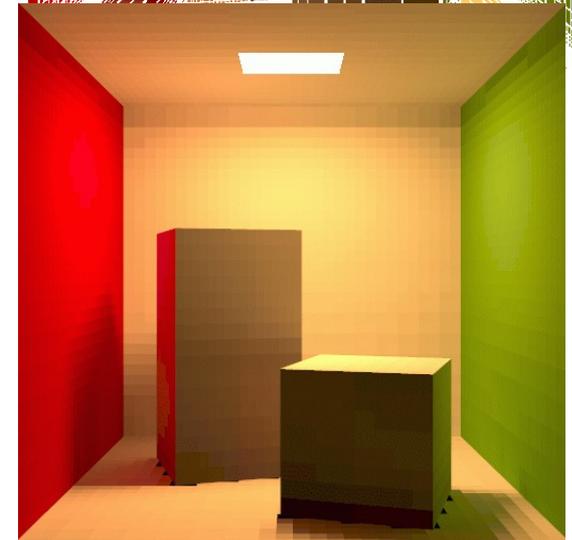
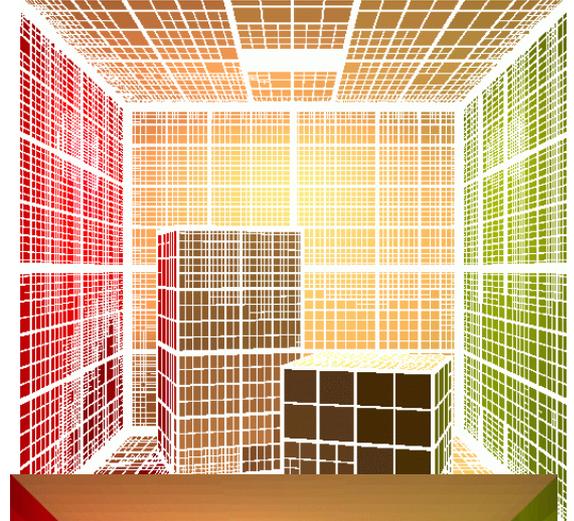
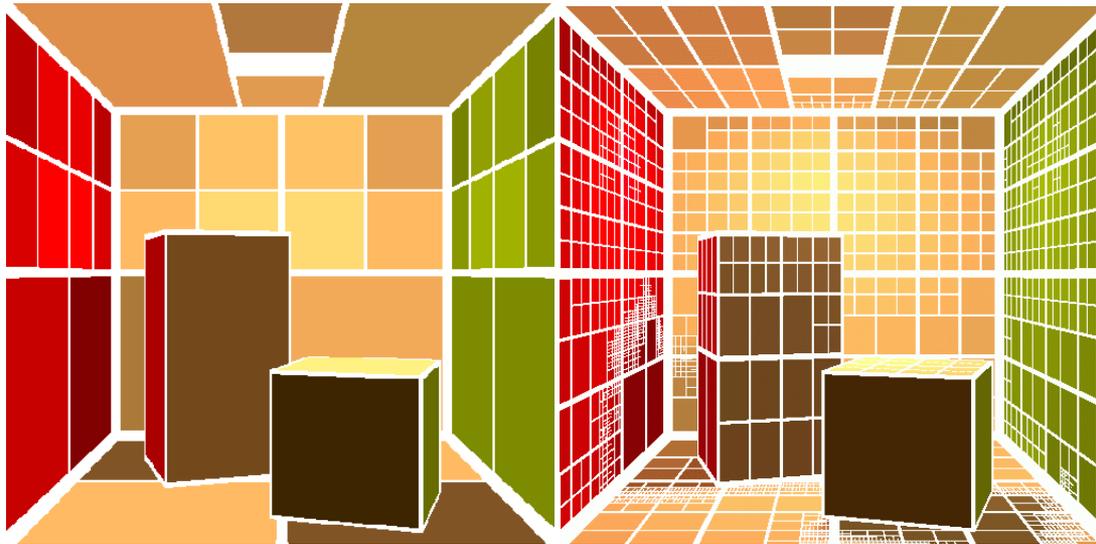
- ❖ Usare HW grafico per non fare raycasting.
- ❖ Si considera il semi cubo,
- ❖ Ogni pixel ha un dato FF.
- ❖ Rendering proiettivo (con COP sul centro della patch)



# The Hemicube in action

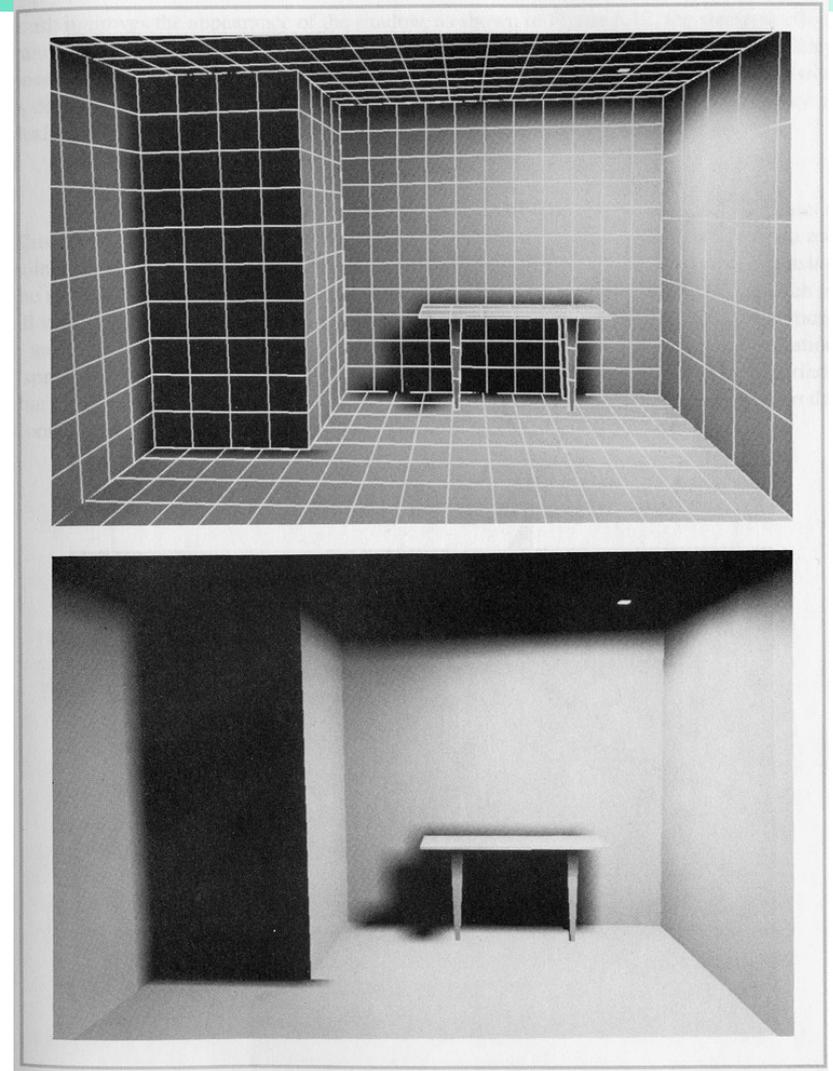


# Mesh Refinement



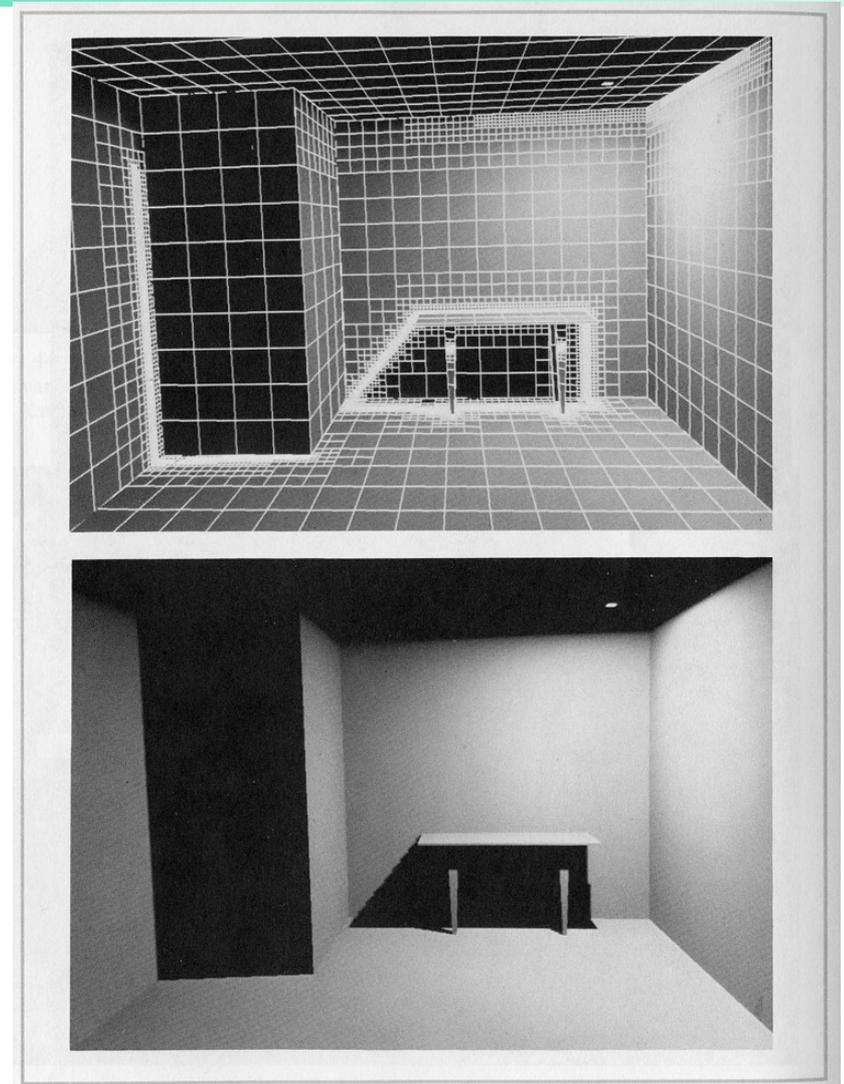
# Meshing Artifacts

- ❖ StairCase
- ❖ Shadow Leaks
- ❖ La soluzione e' di solito:
  - ❖ Finer Meshing



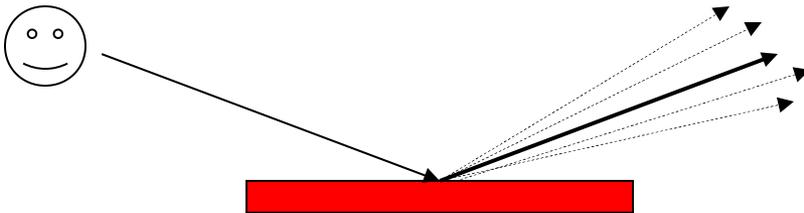
# Meshing Artifacts

- ❖ StairCase
- ❖ Shadow Leaks
- ❖ La soluzione e' di solito:
  - ❖ Finer Meshing



# Distributed Ray Tracing

- ❖ Estensione del Raytracing per la gestione di superfici non speculari/trasmissive perfette
  - ❖ Rimane sempre  $LDS * E$ ,  $LS * E$
- ❖ Solito meccanismo multiple rays according to the distribution of Rho function



# Two Pass Ray tracing (bidirectional RT)

---

- ❖ Incorpora SD transfer.
- ❖ Prima passata calcolare path  $LS*D$  e memorizzarli in una illumination map
  - ❖  $LS*DS*E$
  - ❖ View independent
  - ❖ Necessita di parametrizzazione
- ❖ Seconda passata
  - ❖ Eye trace (ray tracing standard)

# Two Pass Ray tracing (bidirectional RT)

---

- ❖ Se invece di memorizzare i contributi dei path  $LS^*D$  sugli oggetti stessi memorizzo direttamente con una struttura adattiva, i punti di impatto di tali raggi
- ❖ Photon Mapping.
- ❖ Non necessita di parametrizzazione