Informatica Grafica

Gianluigi Ciocca, Simone Bianco

F1801Q120

Ray Tracing Avanzati

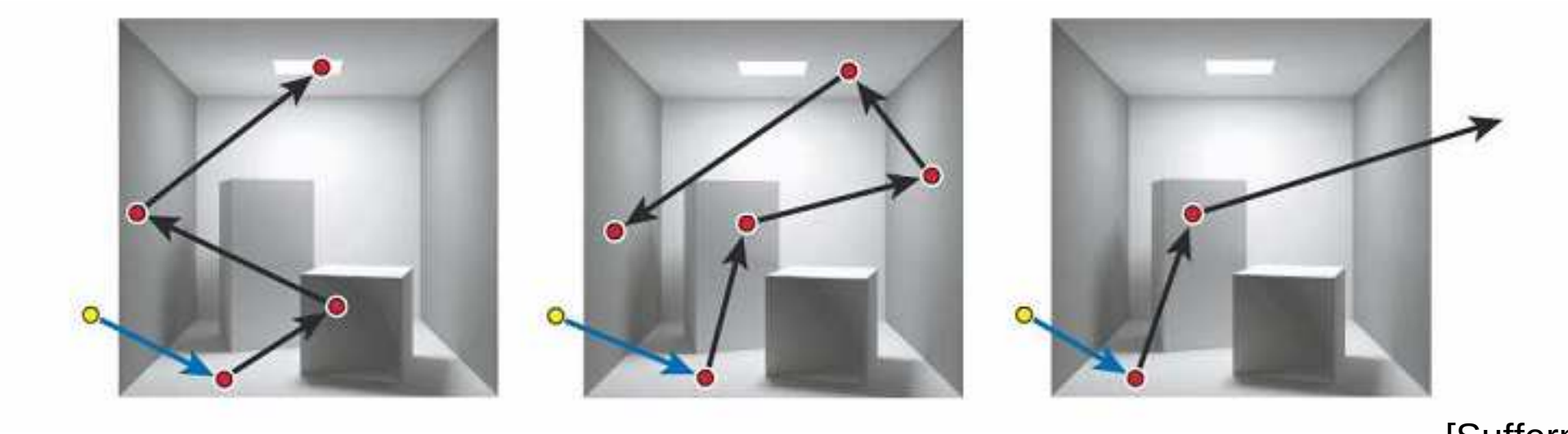
- Altre tecniche di rendering avanzate
  - Path Tracing
  - Bidirectional Path Tracing
  - Photon Map
  - Radiosity

Path Tracing
Idea (1)

- Viene seguito 1 solo raggio alla volta
  - Superficie->Luce
- Si lanciano più raggi per pixel
  - I contributi dei raggi sono accumulati e mediati
- Ad ogni punto di iterazione vengono fatte scelte randomiche
  - Basato su un approccio Monte Carlo
  - Scelte randomiche che MEDIAMENTE danno il risultato corretto

- Meno calcoli ma risultati garantiti rispetto al Distributed Ray Tracing

Path Tracing
Idea (2)



[Suffern]

- Accumulando evidenze (random path) in modo corretto, è garantita la convergenza del risultato
  - Considerare la probabilità di interazione luce/materiale lungo una certa direzione

Path Tracing
Idea (3)

- La luce che vediamo lungo una direzione ϖ_o in un punto p della scena è data da
  - Luce emessa dal punto p (0 per non luci) +
  - Luce riflessa ricevuta da una generica direzione ϖ_i

- La luce riflessa è pesata per tener conto delle caratteristiche del materiale (ci torneremo più avanti)

$$weight = \frac{f_r(x, \varpi_i, \varpi_o)\cos\theta_i}{p(\varpi_i)}$$

- f_r: luce che da ϖ_i arriva verso ϖ_o
- θ_i: angolo tra la normale e ϖ_i
- $p(\varpi_i)$: probabilità di scegliere la direzione ϖ_i

Path Tracing
Algoritmo base (1)

```
Raytrace() // top level function
  for each pixel(x,y)
    pixel(x,y).color = Trace(ray_through_pixel(x,y))
  end for
```

```
Trace(rayO) // fire a ray, returns radiance
  object_point = closest_intersection(ray)
  if object_point
    if object_point is Light
      return object_point.ke
    else
      rayI = Compute_random_ray(rayO,object_point)
      weight = Compute_weight(rayO,rayI,object_point)
      return obj.emiss + weight * Trace(rayI)
```

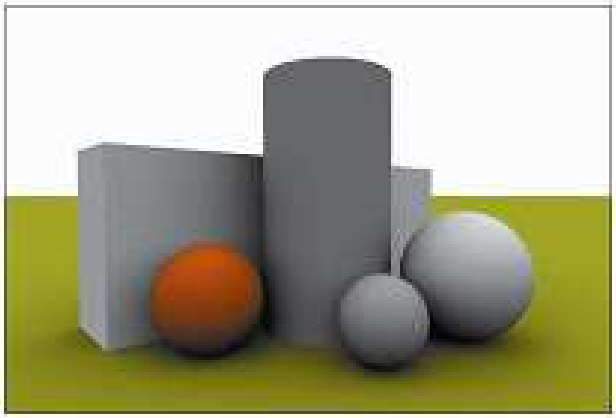
Path Tracing
Algoritmo base (2)

```
Compute_random_ray(ray,point) // returns a new ray
  // Depends on material
  // Depends on light
```

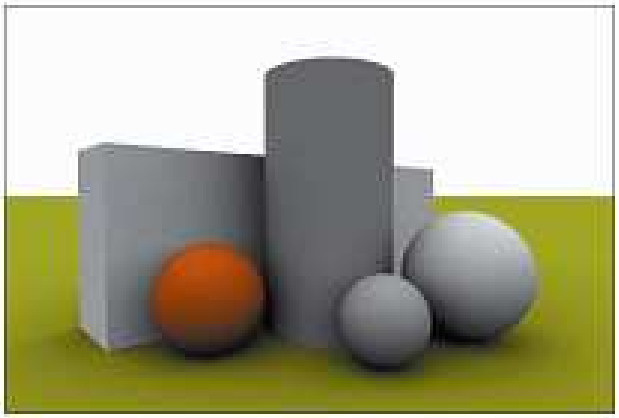
```
Compute_weight(wo,wi) // returns a reflectance weight
  // Use equation
  // Depends on material
  // Depends on light
```

- E i vari coefficienti delle luci (k_r,k_d,...)?
  - La funzione f_r considera tutte le proprietà del materiale
  - Modella completamente il comportamento della luce
  - Ci torneremo più avanti

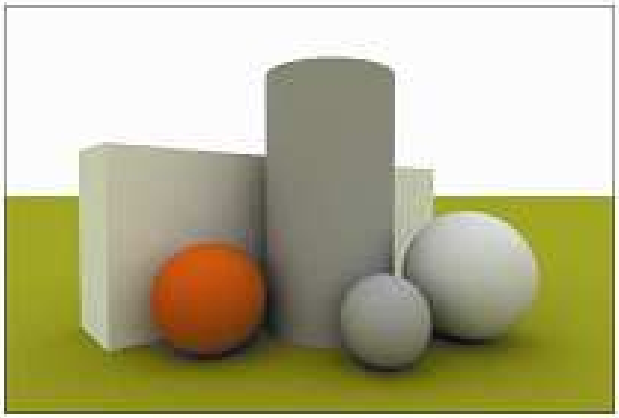
Path Tracing
Risultati (1)



direct illumination



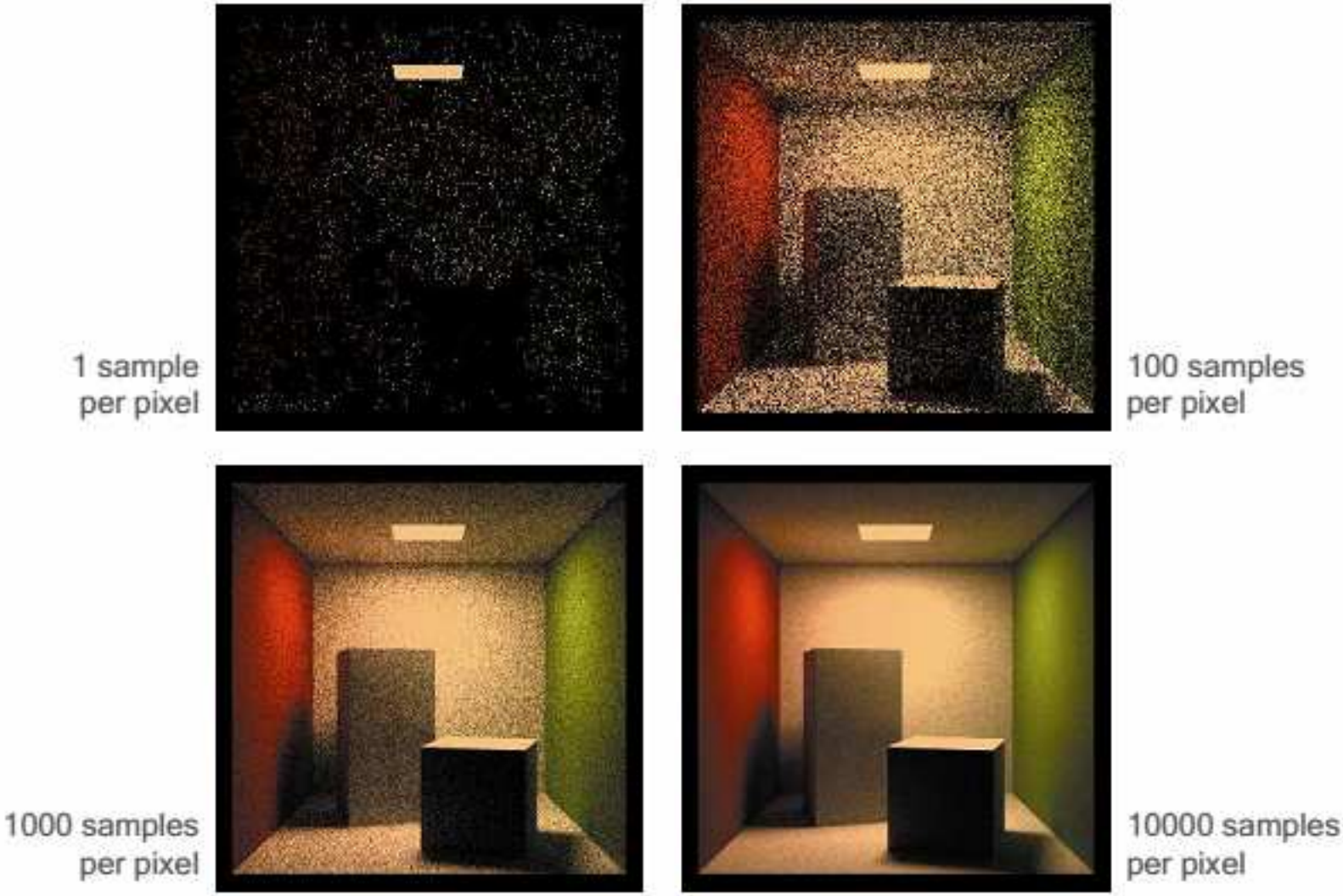
path tracing
one bounce



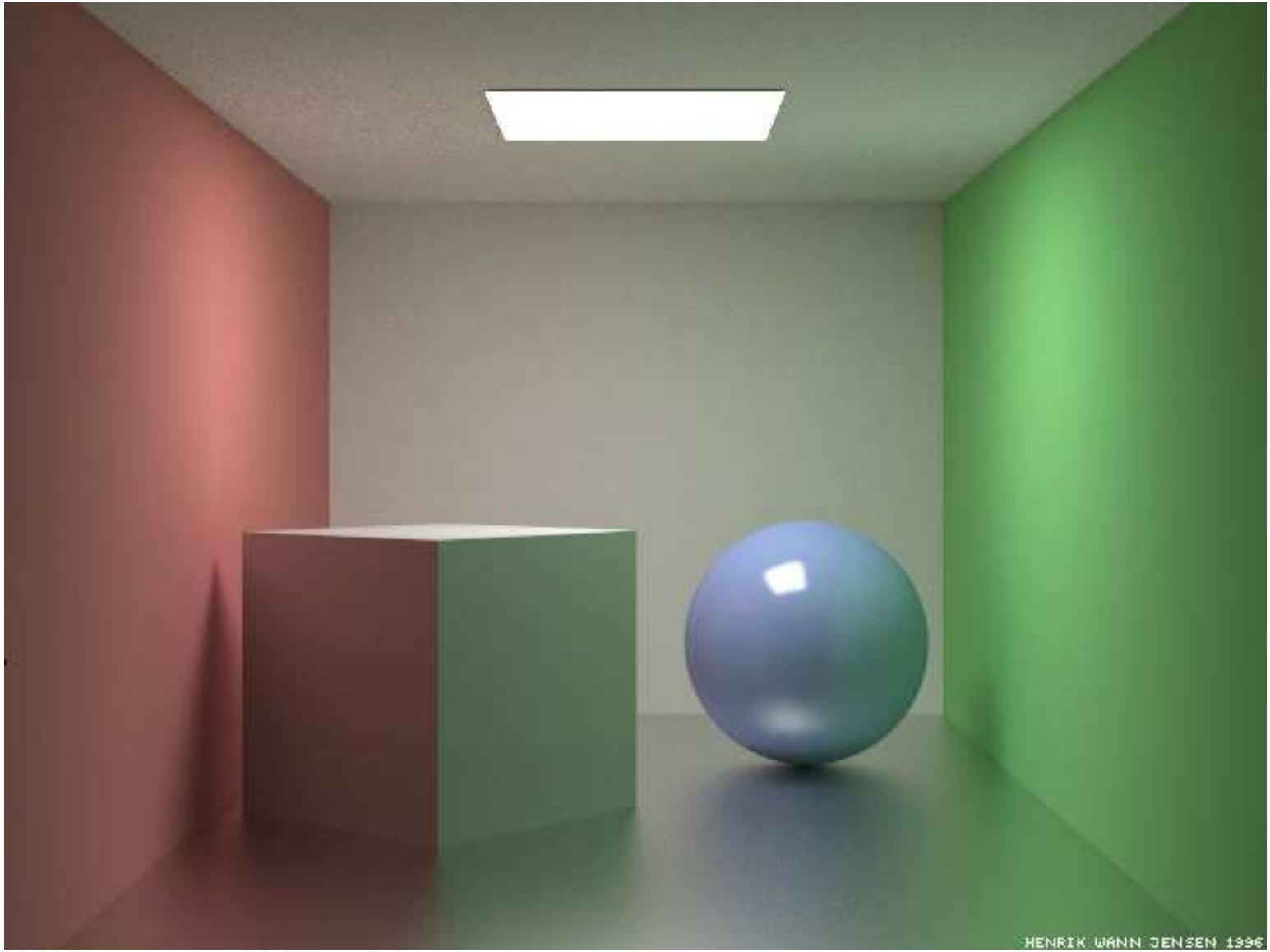
path tracing
five bounces

[Suffern]

Path Tracing
Risultati (2)



Path Tracing
Risultati (3)



Path Tracing
Importance sampling

- L'algoritmo è molto inefficiente
  - I raggi secondari sono scelti senza nessun criterio particolare
    - Molti vanno "persi"
    - Possono analizzare parti "poco rilevanti" della scena

- Scelta dei raggi usando un concetto di importanza
  - Importanza dal punto di vista del contributo di luce
  - Es. La luce diretta contribuisce di più di quella indiretta
  - Es. In una superficie a specchio c'e' una direzione preferenziale
  - Es. Un certo materiale può riflettere maggiormente in certe condizioni e assorbire in altre
  - ...

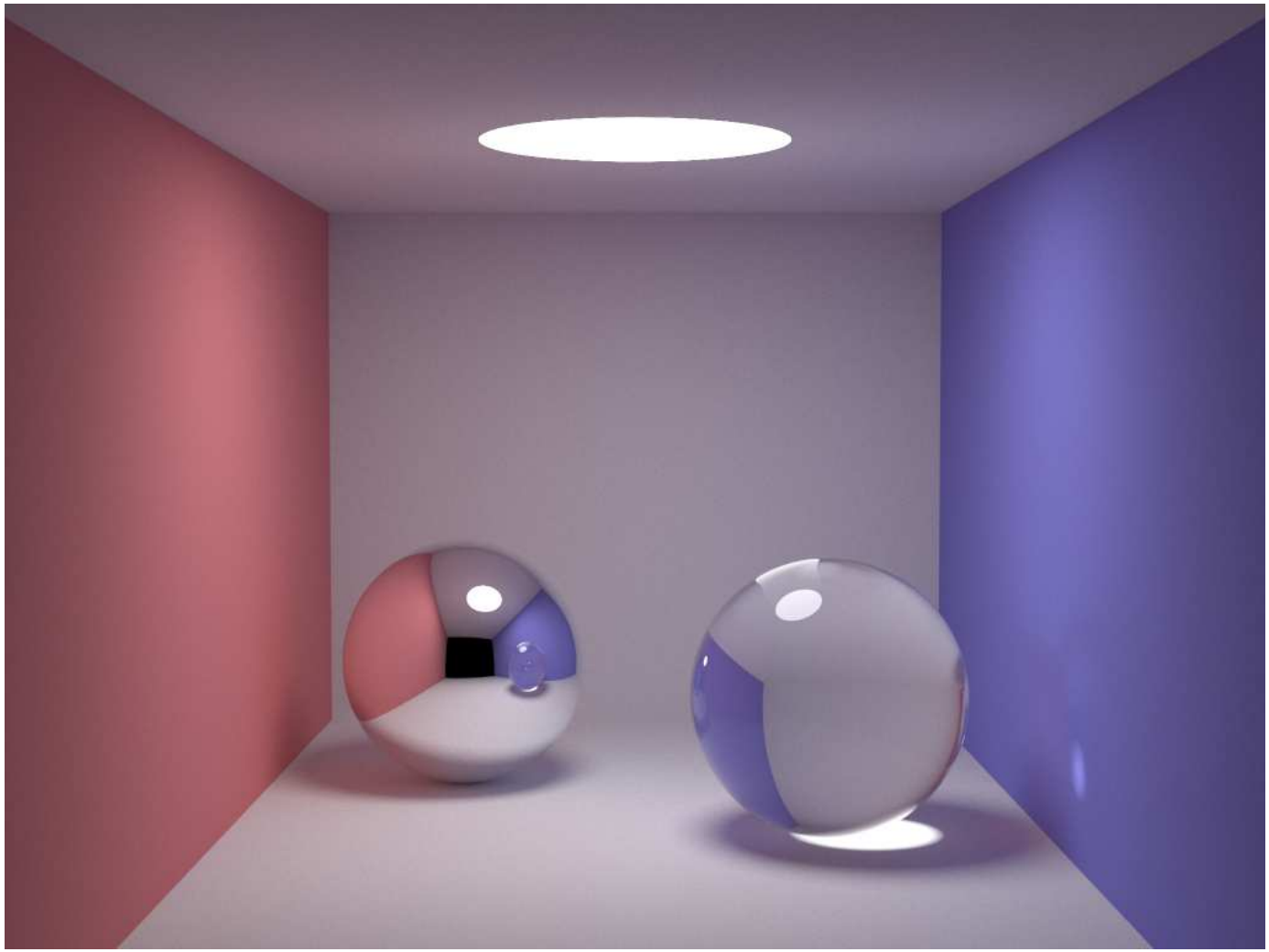
Path Tracing
Russian Roulette

- Termina alcuni path prematuramente
  - Usata una soglia randomicamente scelta su certi valori
    - Peso accumulato lungo il path
    - Valore di riflettanza
    - ...

- Usata anche per scegliere se seguire un raggio di Riflettanza O Trasmittanza nel caso di materiali semi-trasparenti
  - Dimezza i raggi analizzati
  - I pesi sono adattati per tenere conto di questo e avere un risultato corretto

Path Tracing

SmallPt: Un path tracer in 99 righe di codice C++



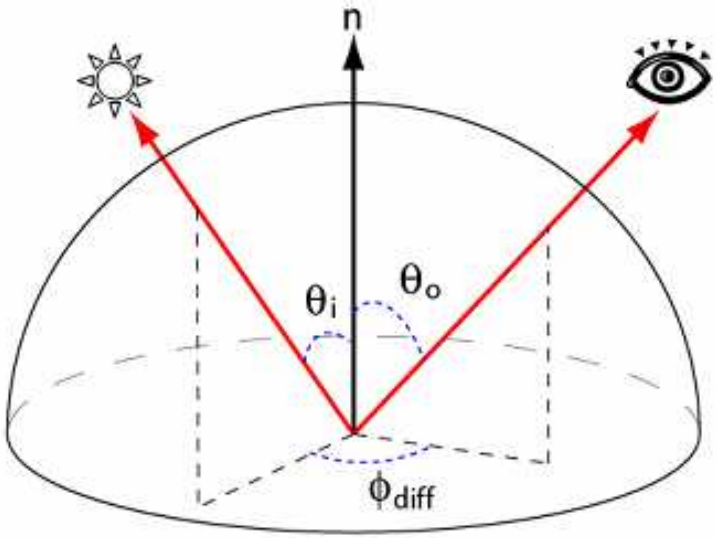
- Supporta
  - Importance Sampling
  - Russian Roulette

http://www.kevinbeason.com/smallpt/

BRDF (1)

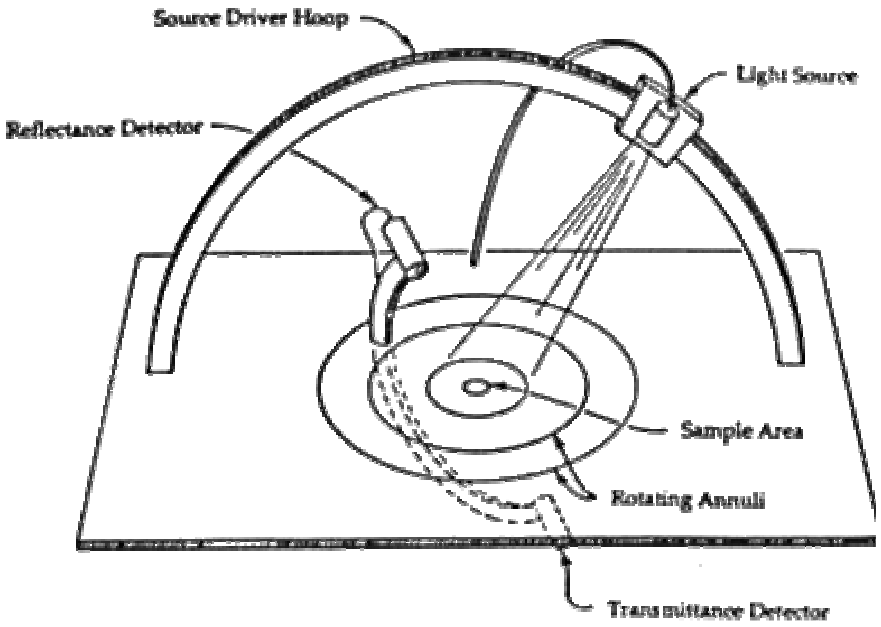
- Che cosa è la funzione $f_r(p,\varpi_o,\varpi_i)$?
  - E' detta **Bidirectional Reflectance Distribution Function** (BRDF)
  - Dice quanta luce, da una direzione ϖ_i viene vista in direzione ϖ_o rispetto ad un punto p
    - E' una funzione a 4 dimensioni (usando coordinate sferiche) e dipende dalla lunghezza d'onda della luce:

$$BRDF_\lambda(\theta_i,\phi_i,\theta_o,\phi_o)$$



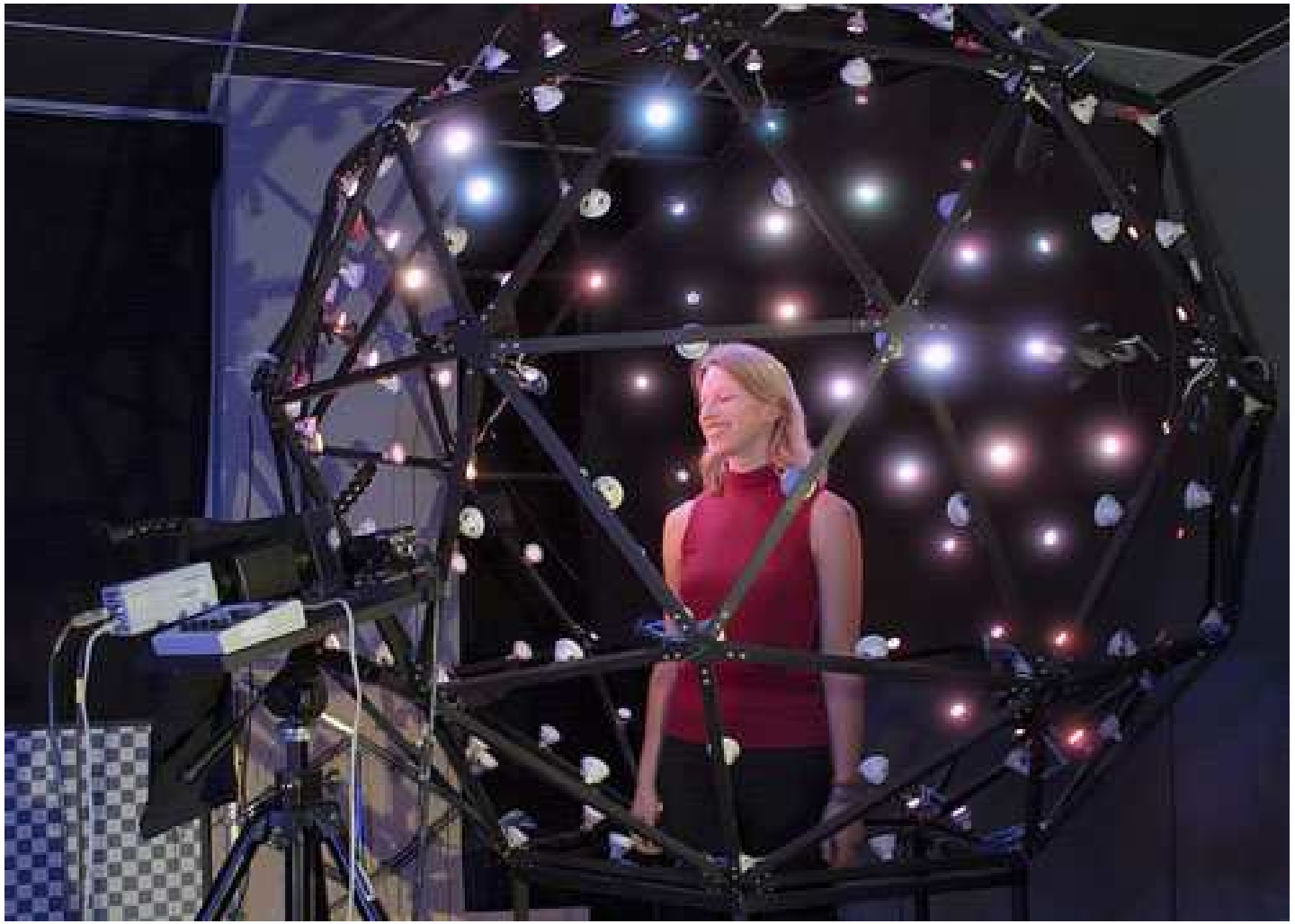
BRDF (2)

- Si può misurare: Spettro Goniometro

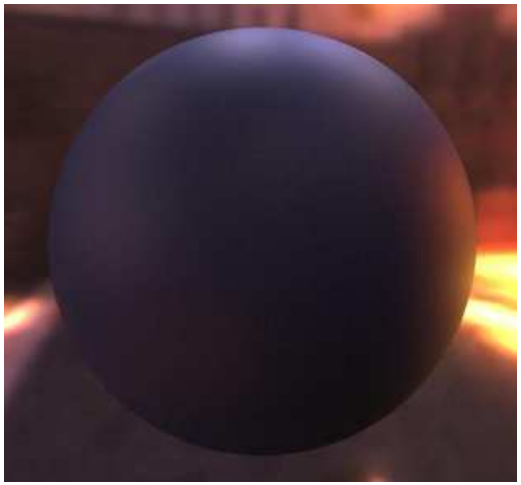


BRDF (3)

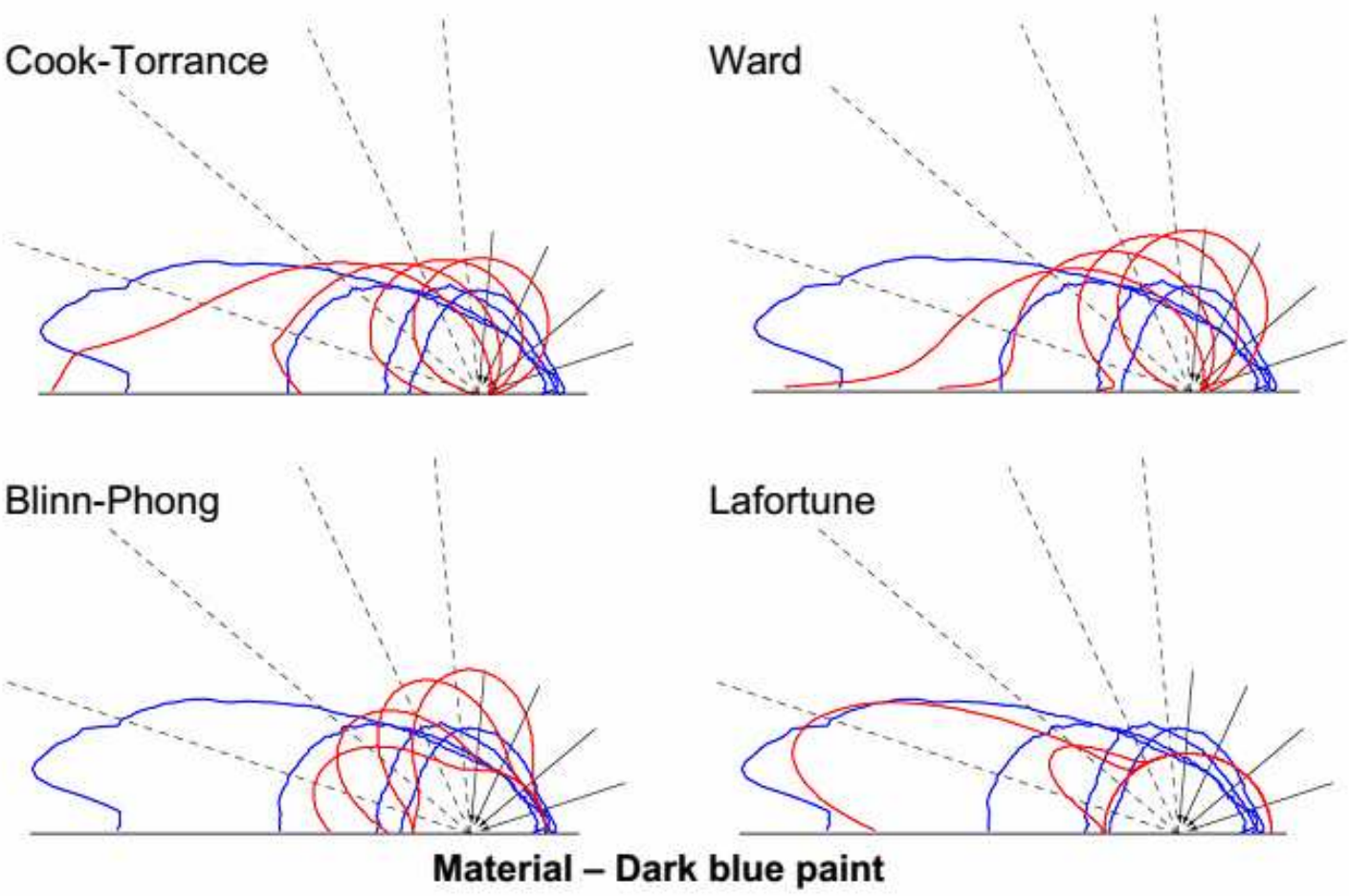
- Nel cinema...



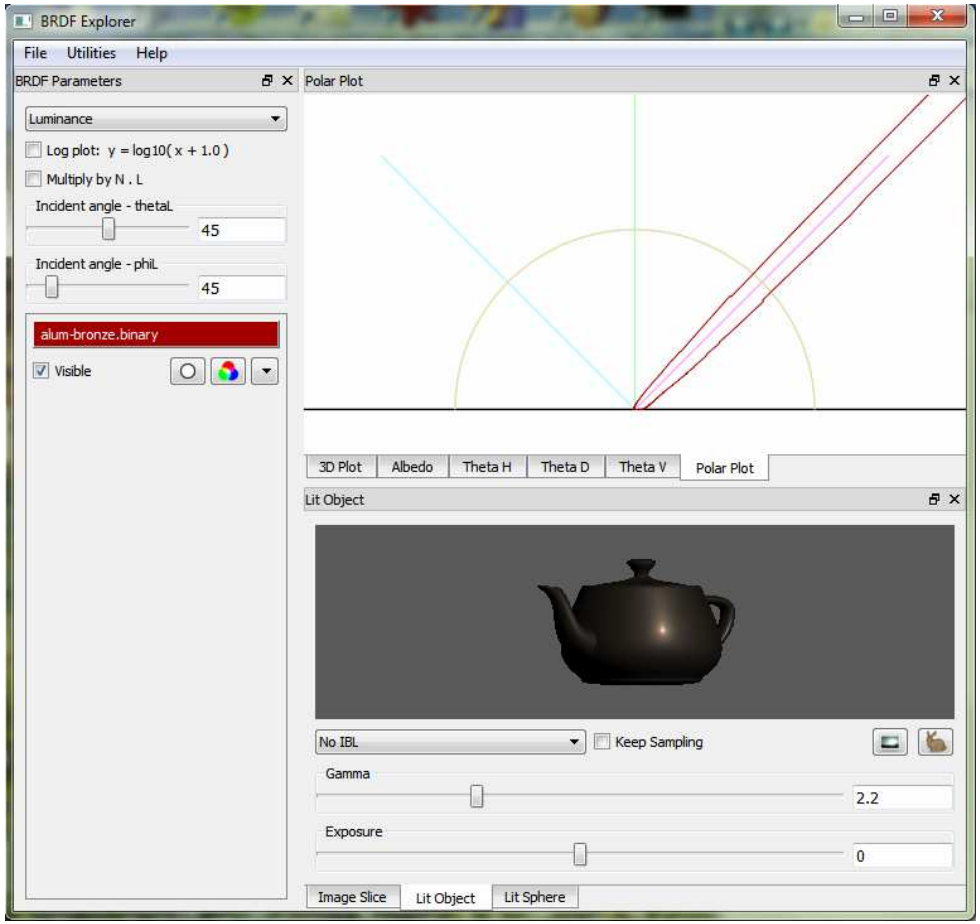
BRDF (4)




Fitted model
Data


Cook-Torrance
Ward
Blinn-Phong
Lafortune
Material – Dark blue paint

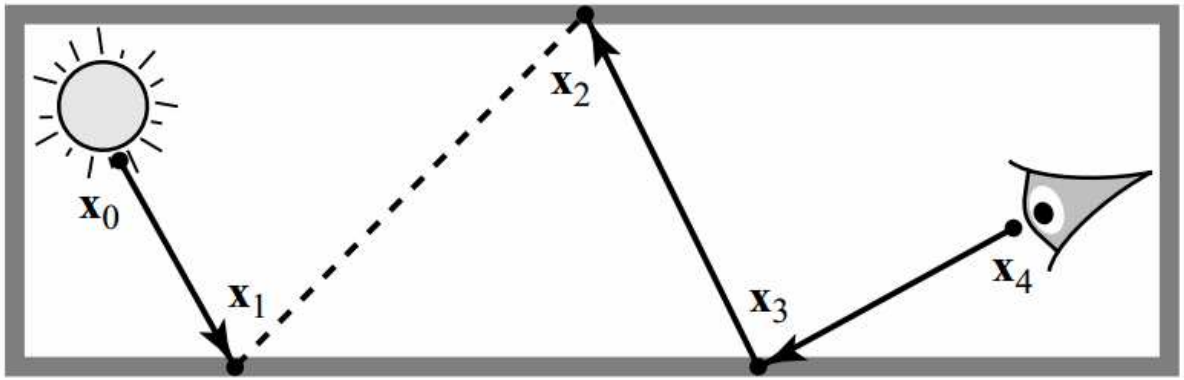
BRDF (5)



MERL BRDF Database: http://www.merl.com/brdf/
Disney's BRDF Explorer: http://www.disneyanimation.com/technology/brdf.html
A. Ngan, F. Durand, W. Matusik "Experimental Validation of Analytical BRDF Models":
*http://people.csail.mit.edu/addy/research/ngan-04-**brdfmodels**.pdf*

Bidirectional Path Tracing (1)

- Idea
  - Viene generato un **subpath** a partire dal punto di vista
  - Viene generato un **subpath** a partire dalla sorgente di luce
  - I due subpath si connettono (se i punti terminali sono reciprocamente visibili)

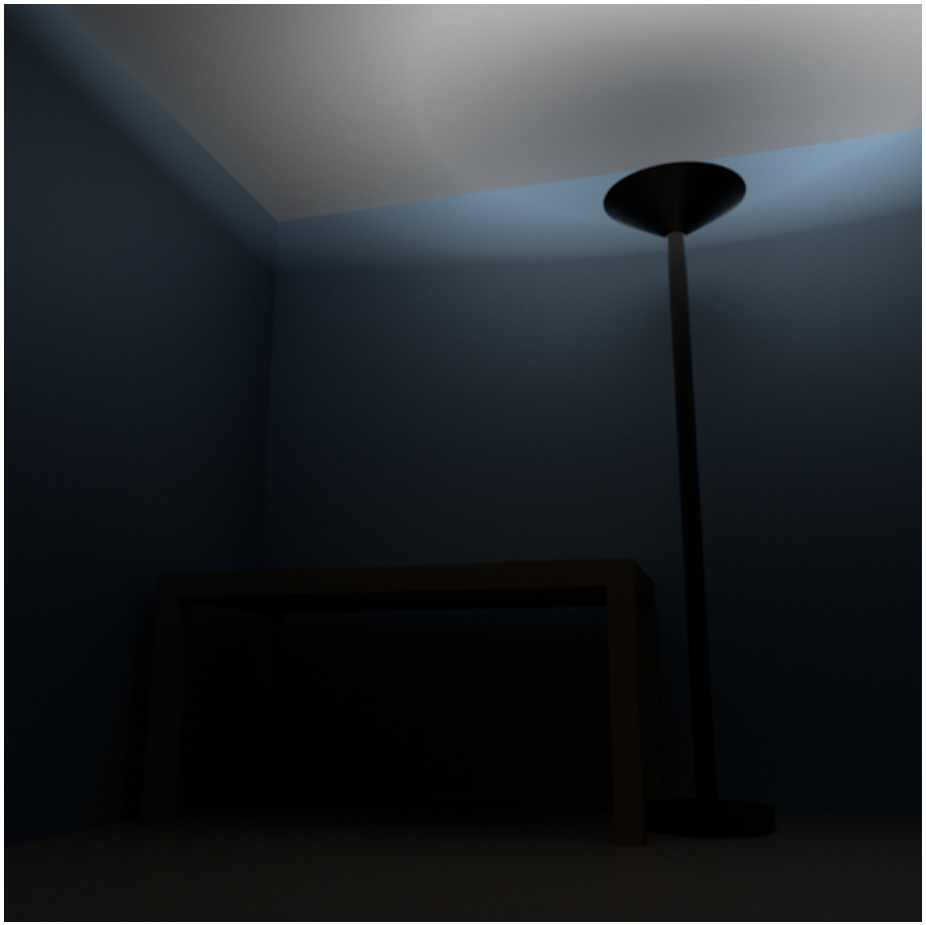


Bidirectional Path Tracing (2)

Bidirectional Path Tracing

Path Tracing

Bidirectional Path Tracing (3)



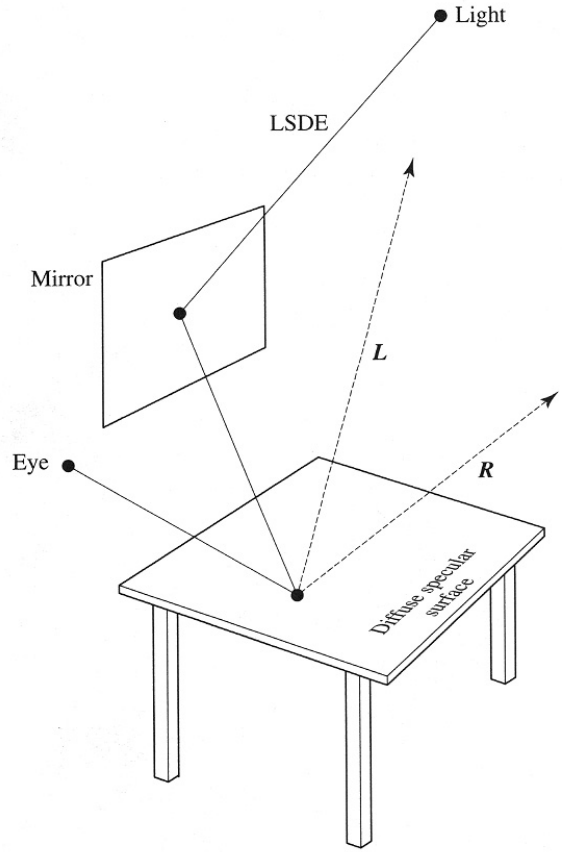
Path Tracing

Bidirectional Path Tracing

http://bptmlt.blogspot.it/

Photon Mapping (1)

- Simula il percorso dei raggi lungo il percorso LSDE: Light-Specular-Diffuse-Eye
  - Passo 1
    - Si seguono diversi raggi dalla luce verso ciascuna superficie riflettente
    - Quando un raggio riflesso colpisce una superficie diffusa in un punto P, il contributo viene memorizzato in una Photon Map indicizzata sul punto
  - Passo 2
    - Ray tracing "normale" ma
    - Quando un raggio incide sul punto P viene usata la Photon Map per calcolare il contributo di luce diretta



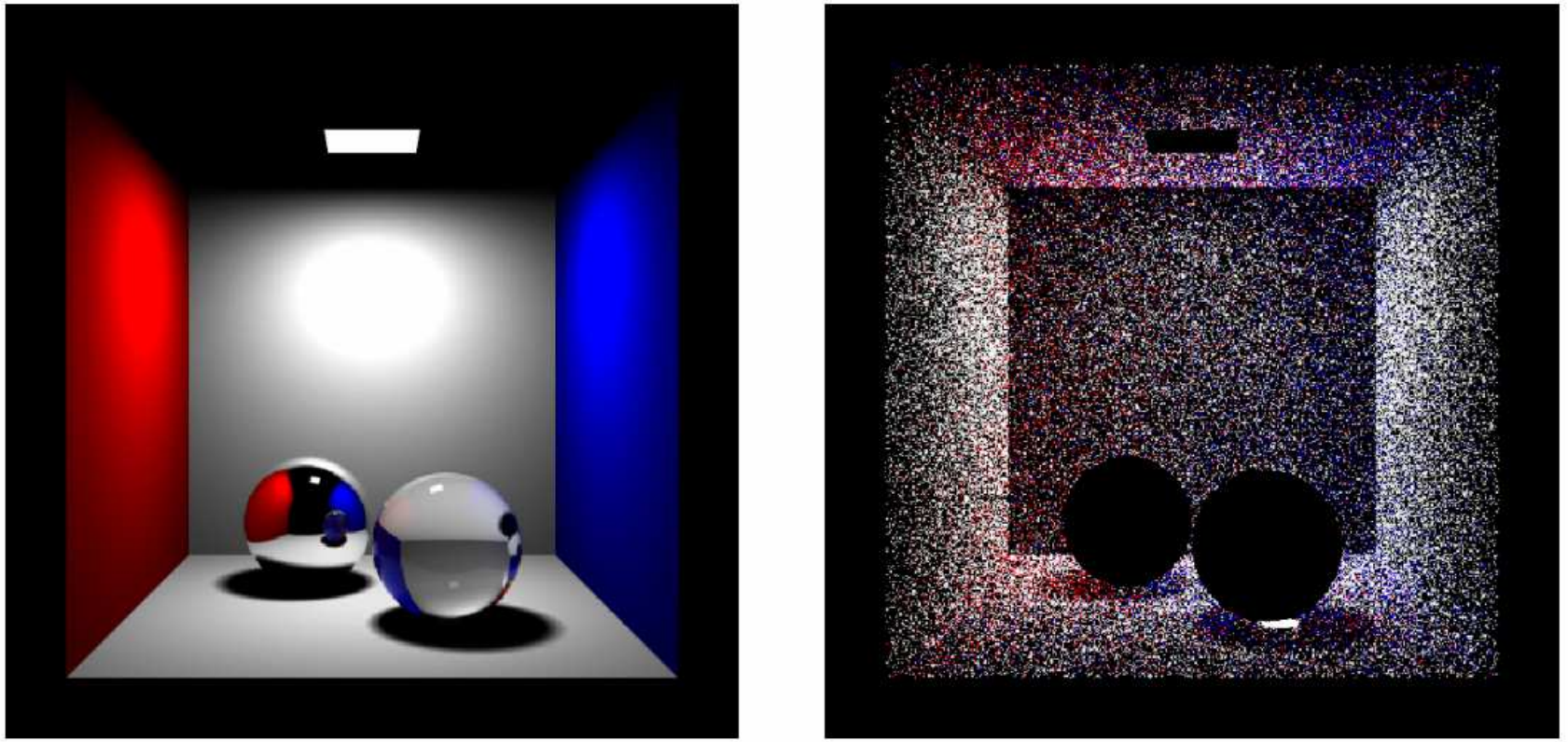
Photon Mapping (2)

- Necessaria una struttura dati per memorizzare la Photon Map
  - Ad ogni intersezione si usano i k punti più vicini nella photon map per stimare la luce
  - k-nearest neighbour search su kd-tree data structure

```
struct photon {
    float x, y, z;        // position ( 3 x 32 bit floats )
    char p[4];            // power(rgb) packed as 4 chars
    char phi, theta;      // compressed incident direction
    short flag;           // flag used for kd-tree
}
```

Jensen, Henrik W., Realistic Image Synthesis Using Photon Mapping, A K Peters, Ltd., Massachusetts, 2001

Photon Mapping (3)



(a) (b)

Figure 2.4: "Cornell box" with glass and chrome spheres: (a) ray traced image (direct illumination and specular reflection and transmission), (b) the photons in the corresponding photon map.

Photon Mapping (4)

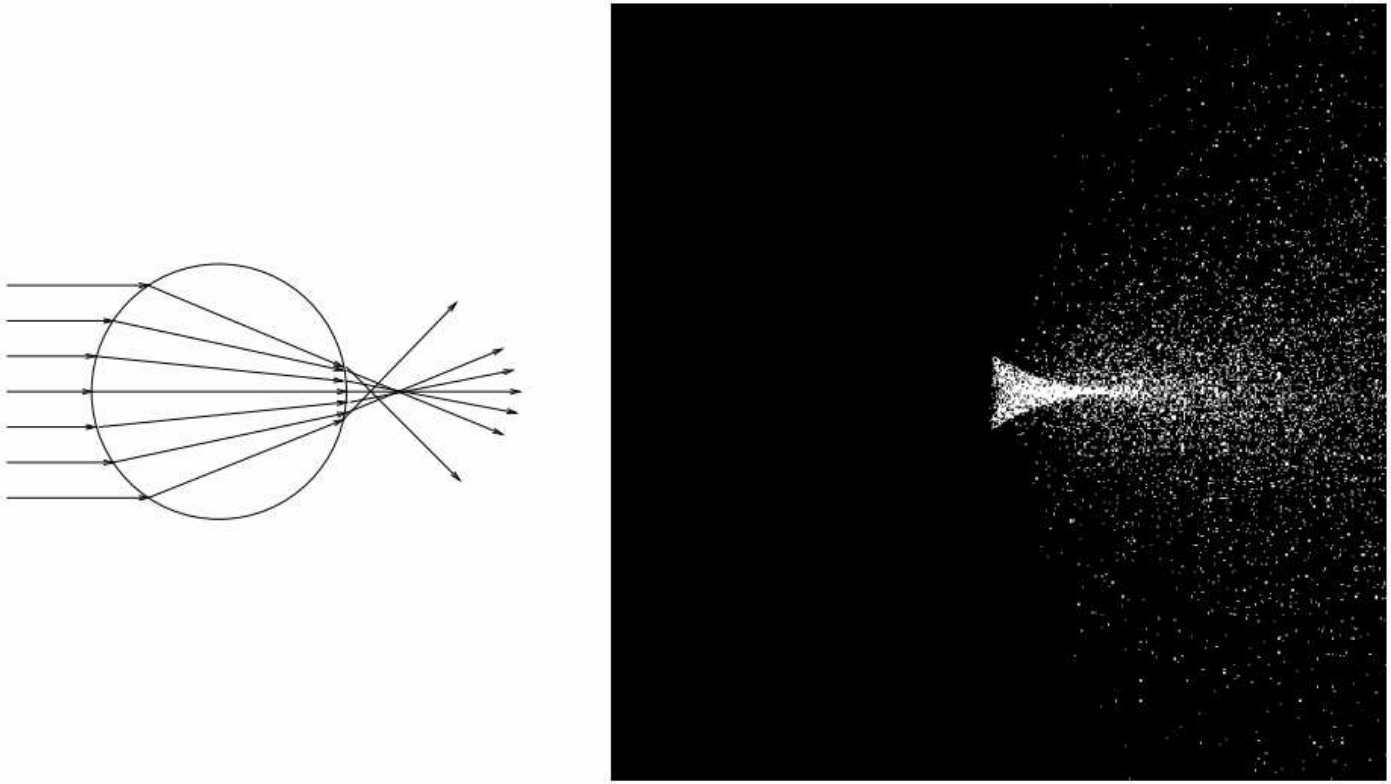
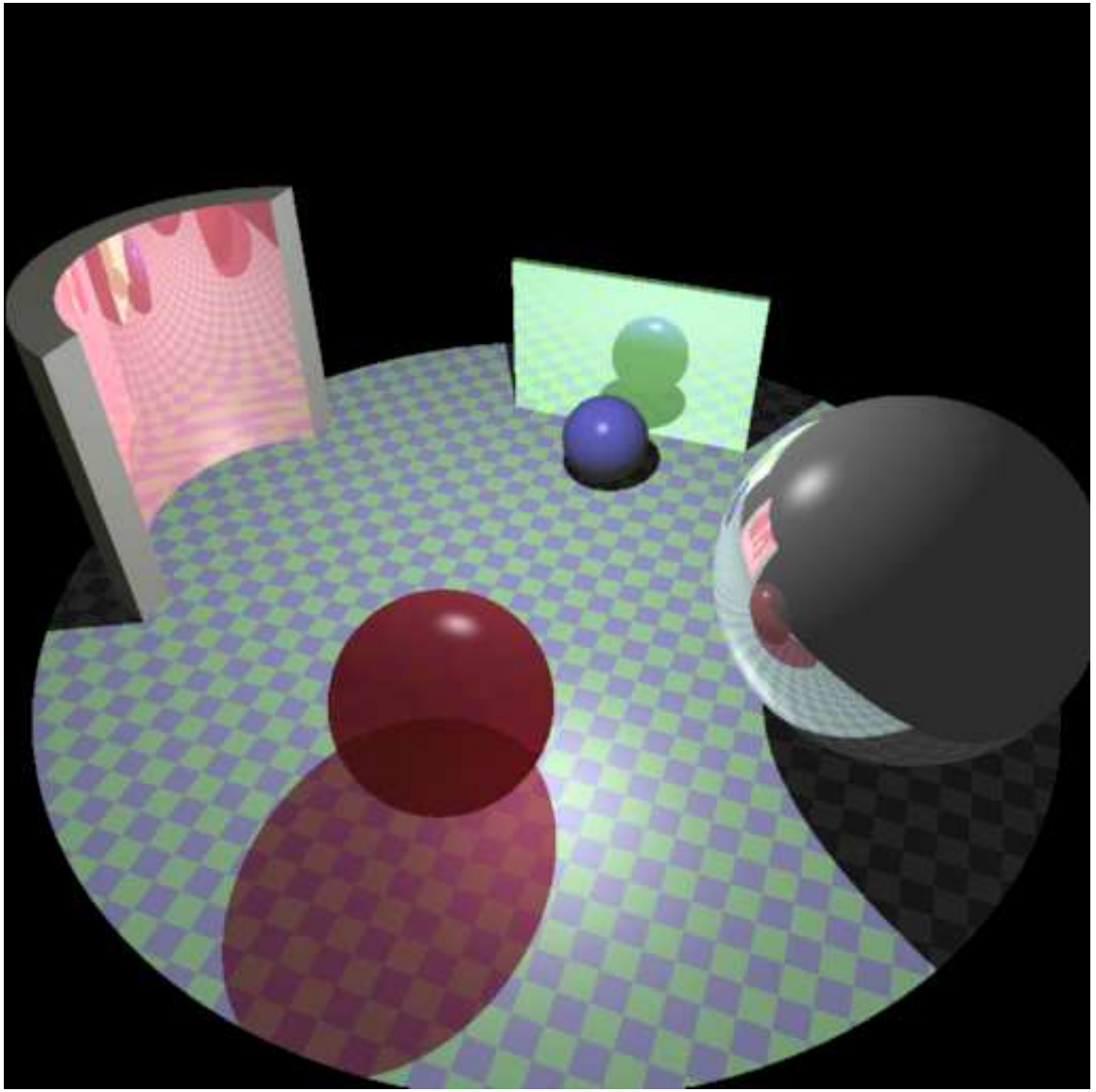


Figure 2.5: Sphere in fog: (a) schematic diagram of light paths, (b) the caustic photons in the photon map.

Photon Mapping
Esempio (1)

- Esempio

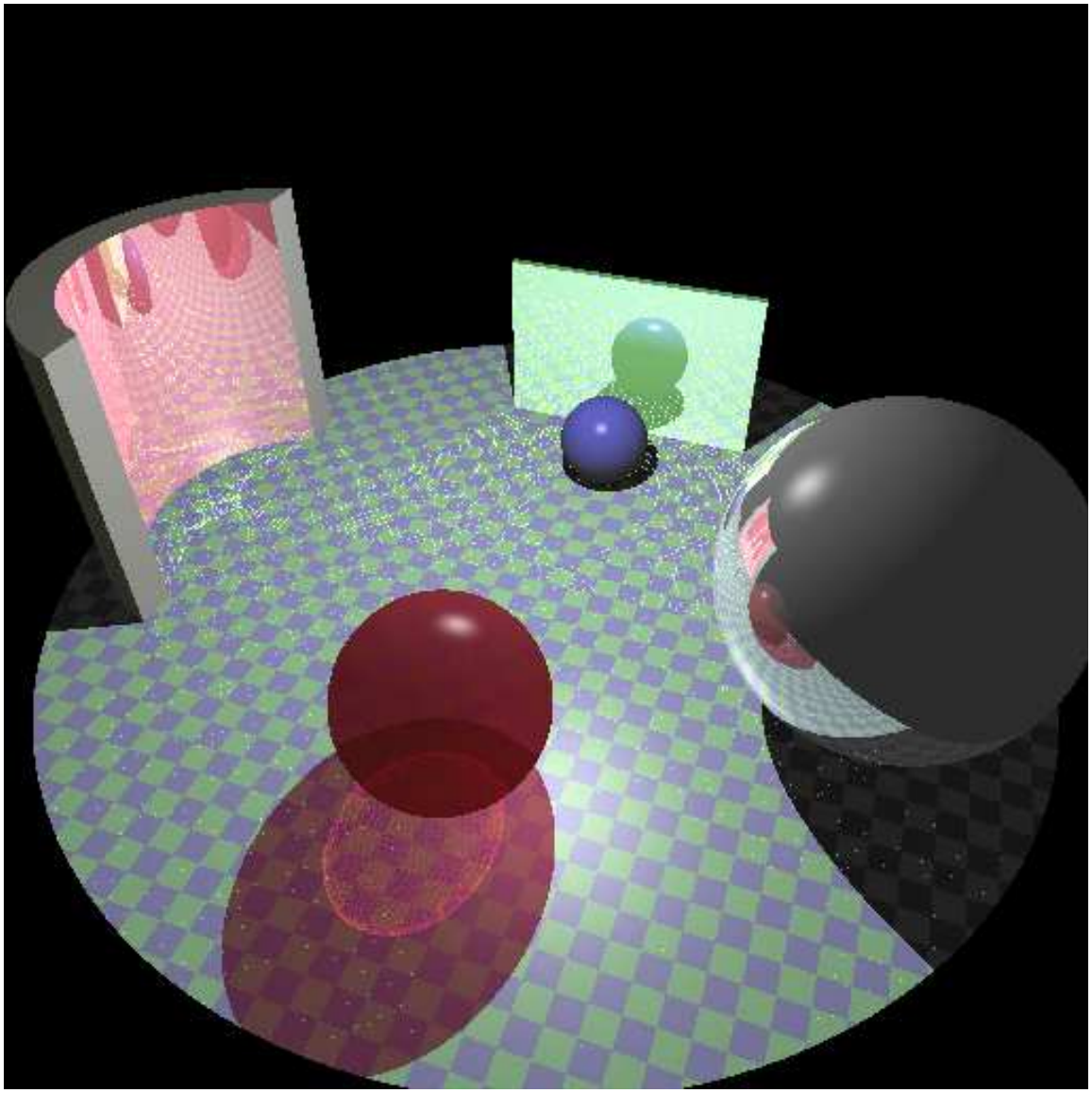


Ray tracing convenzionale

Roberto Pirrone

Photon Mapping
Esempio (2)

- Esempio

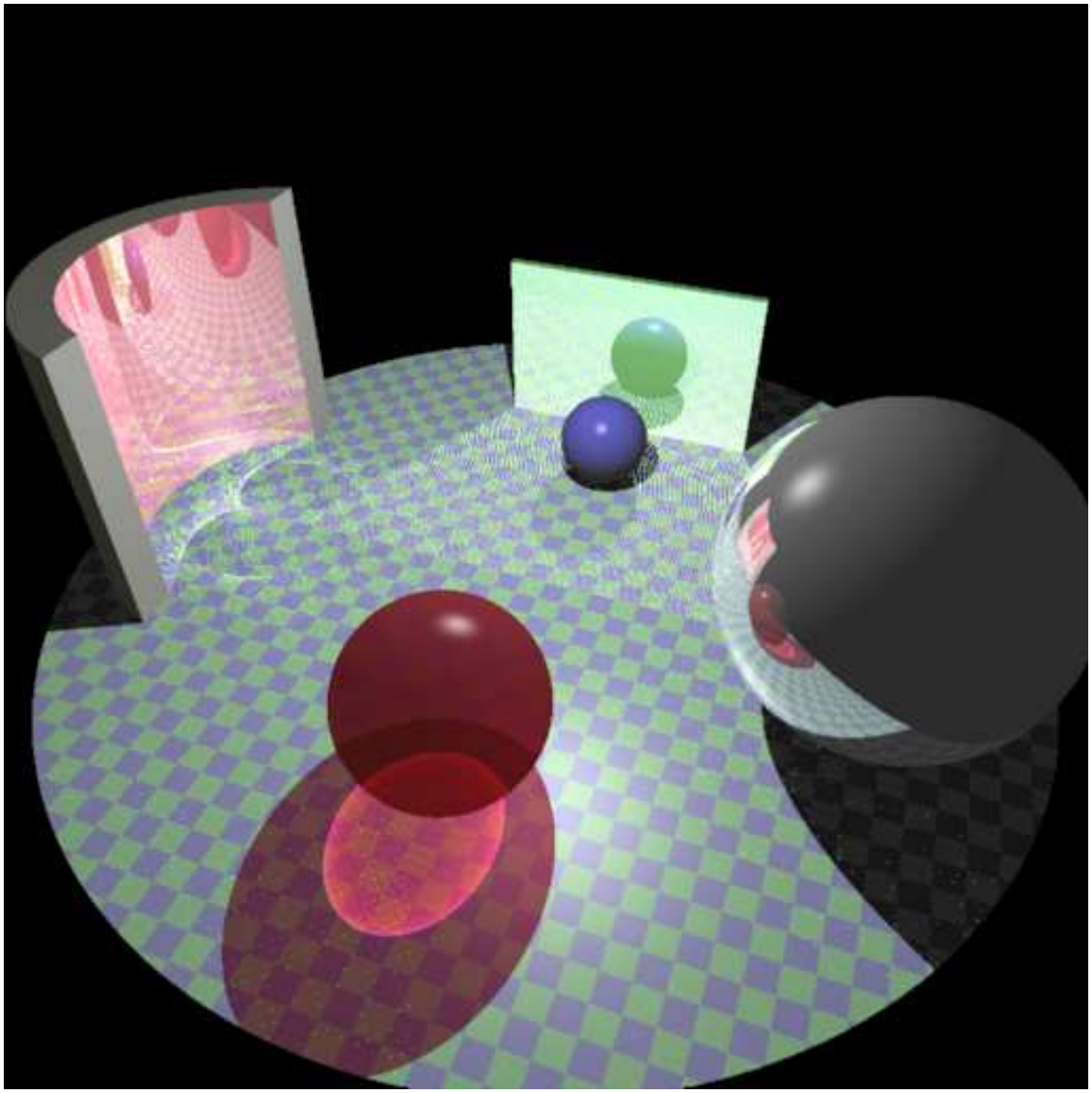


Two Pass Ray tracing: 200 raggi emessi dalla sorgente

Roberto Pirrone

Photon Mapping
Esempio (3)

- Esempio

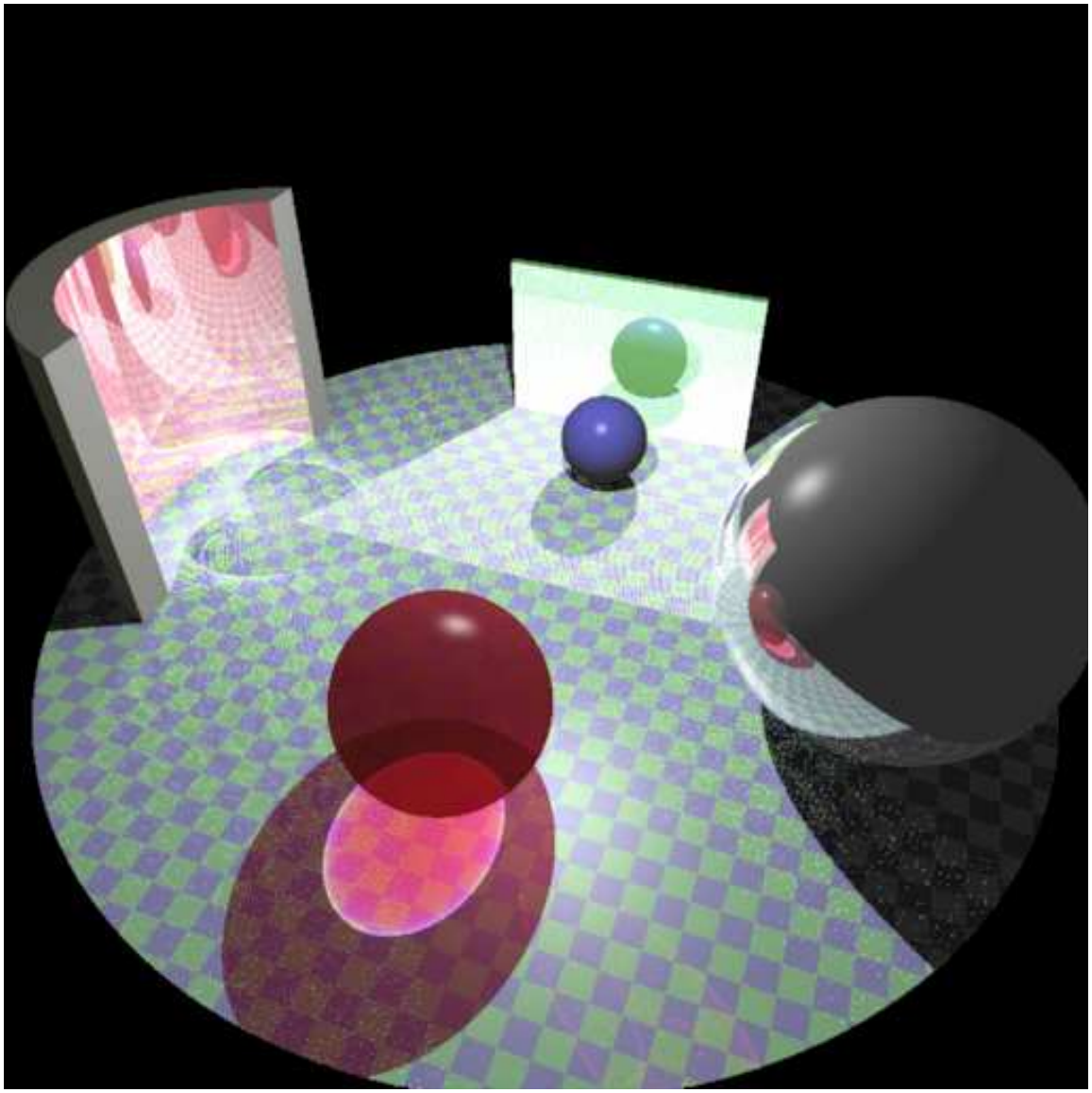


Two Pass Ray tracing: 400 raggi emessi dalla sorgente

Roberto Pirrone

Photon Mapping
Esempio (4)

- Esempio

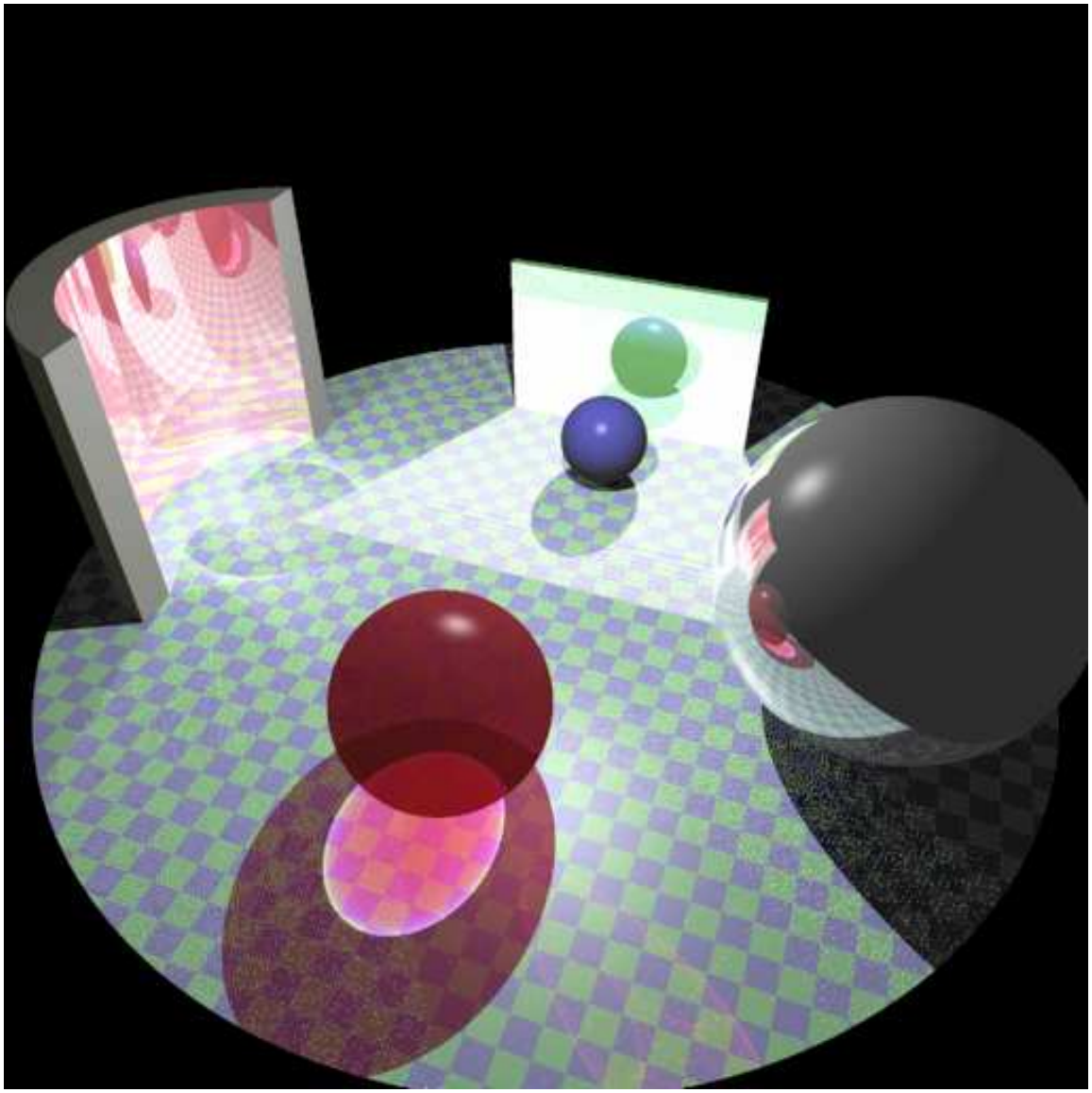


Two Pass Ray tracing: 800 raggi emessi dalla sorgente

Roberto Pirrone

Photon Mapping
Esempio (5)

- Esempio



Two Pass Ray tracing: 1000 raggi emessi dalla sorgente

Roberto Pirrone

Photon Mapping (5)

Photon Mapping (6)

- Subsurface Scattering

Rendering Equation (1)

- James Kajiya (1986)
  - E' alla base di tutti i metodi di rendering
  - Descrive il trasporto della luce da una superficie ad un'altra

$$L(x,\varpi_o) = L_e(x,\varpi_o) + L_r(x,\varpi_o)$$

$$L_r(x,\varpi_o) = \int_{\varpi_i} f_r(x,\varpi_i,\varpi_o) L_i(x,\varpi_i) \left|\cos\theta_i\right| d\varpi_i$$

  - x : punto sulla superficie
  - ϖ_o : direzione di vista
  - L_e : Radianza emessa
  - L_r : Radianza riflessa

Rendering Equation (2)

$$L_r(x,\varpi_o) = \int_{\varpi_i} f_r(x,\varpi_i,\varpi_o) L_i(x,\varpi_i) \cos\theta_i d\varpi_i$$

- E' la BRDF

$$f_r(x,\varpi_i,\varpi_o) = \frac{dL_r(\varpi_o)}{dE_i(\varpi_i)} = \frac{dL_r(\varpi_o)}{L(\varpi_i)\cos\theta_i d\varpi_i}$$

  - Deve soddisfare la conservazione dell'energia
  - Deve valere il principio di reciprocità

$$\int_{\varpi_i} f_r(x,\varpi_i,\varpi_o)\cos\theta_i d\varpi_i \le 1 \qquad f_r(x,\varpi_i,\varpi_o) = f_r(x,\varpi_o,\varpi_i)$$

Rendering Equation (3)

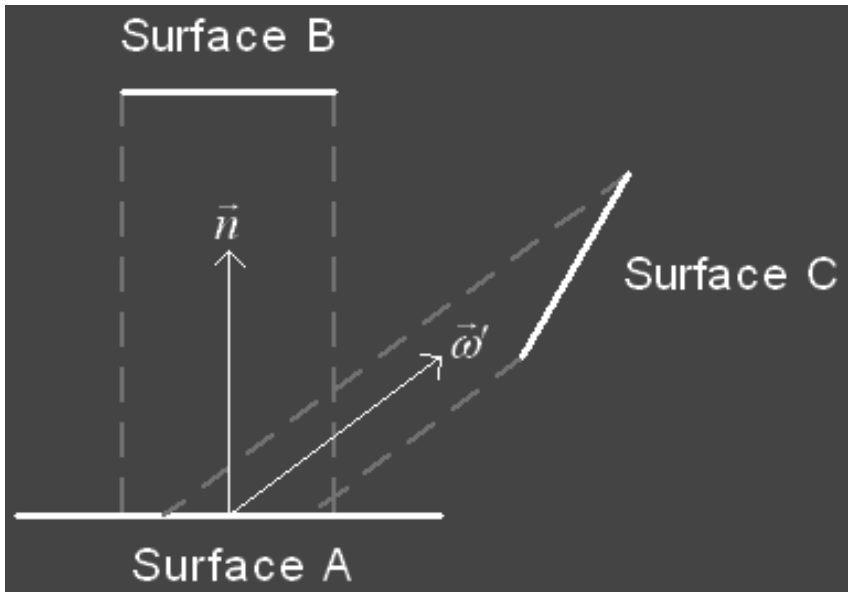
$$L_r(x,\varpi_o) = \int_{\varpi_i} f_r(x,\varpi_i,\varpi_o) L_i(x,\varpi_i) \cos\theta_i d\varpi_i$$

- Radianza incidente dalla direzione ϖ_i
  - A sua volta è un integrale

Rendering Equation (4)

$$L_r(x,\varpi_o) = \int_{\varpi_i} f_r(x,\varpi_i,\varpi_o) L_i(x,\varpi_i) \cos\theta_i d\varpi_i$$

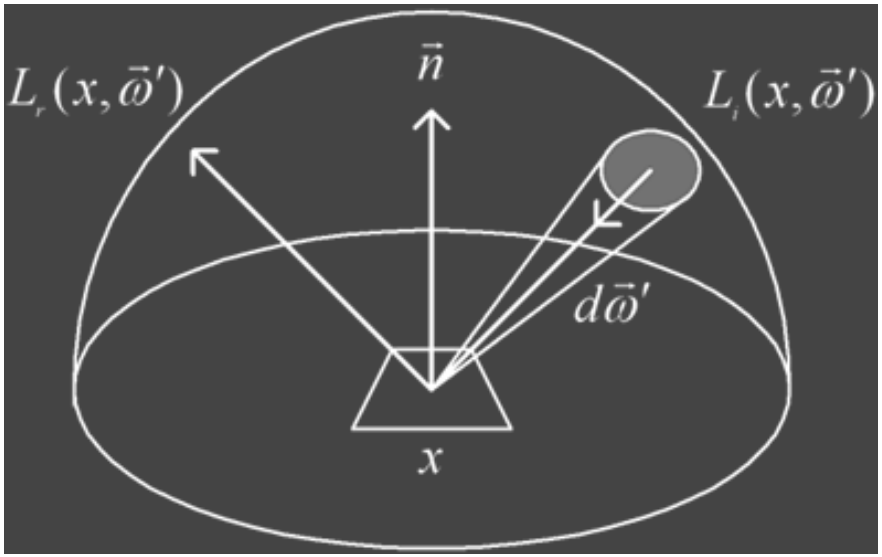
- Legge del coseno



Rendering Equation (4)

$$L_r(x,\varpi_o) = \int_{\varpi_i} f_r(x,\varpi_i,\varpi_o) L_i(x,\varpi_i) \cos\theta_i d\varpi_i$$

- Angolo solido differenziale



Rendering Equation (5)

- Il problema degli algoritmi di ray tracing
  - Stochastic ray tracing
  - Path tracing
  - Photon Mapping
  - Radiosity
  - Monte Carlo ray tracing
  - ...

- Trovare un modo efficiente per calcolare/approssimare la l'integrale dell'equazione

Monte Carlo (1)

- Perché lanciare raggi "a caso" risolve l'integrale?
  - L'approccio Monte Carlo dà una soluzione corretta all'integrale **IN MEDIA**

- Esempio
  - Dato l'integrale $\int_a^b f(x)dx$
  - Date variabili causali uniformemente distribuite $X_i \in [a,b]$
  - Lo stimatore Monte Carlo dell'integrale è $F_N = \frac{b-a}{N}\sum_{i=1}^{N} f(X_i)$

Monte Carlo (2)

- Esempio
  - Dato lo stimatore

$$F_N = \frac{b-a}{N}\sum_{i=1}^{N} f(X_i)$$

  - Il valore atteso dello stimatore ("MEDIA") è dato da

$$E[F_N] = E\left[\frac{b-a}{N}\sum_{i=1}^{N} f(X_i)\right] = \frac{b-a}{N}\sum_{i=1}^{N} E[f(X_i)]$$

  - Il valore atteso di una funzione è dato da

$$E[f(x)] = \int_a^b f(x)\,p(x)\,dx$$

    - Con p(x) probabilità di avere il valore x

Monte Carlo (3)

- Esempio
  - Il valore atteso diventa

$$E[F_N] = \frac{b-a}{N} \sum_{i=1}^{N} \int_b^a f(x)\, p(x) dx$$

  - Dato che i campioni sono estratti in modo uniforme

$$E[F_N] = \frac{b-a}{N} \frac{1}{b-a} \sum_{i=1}^{N} \int_b^a f(x) dx = \frac{1}{N} \sum_{i=1}^{N} \int_b^a f(x) dx = \int_b^a f(x) dx$$

$$E[F_N] = \int_b^a f(x) dx$$

Monte Carlo (4)

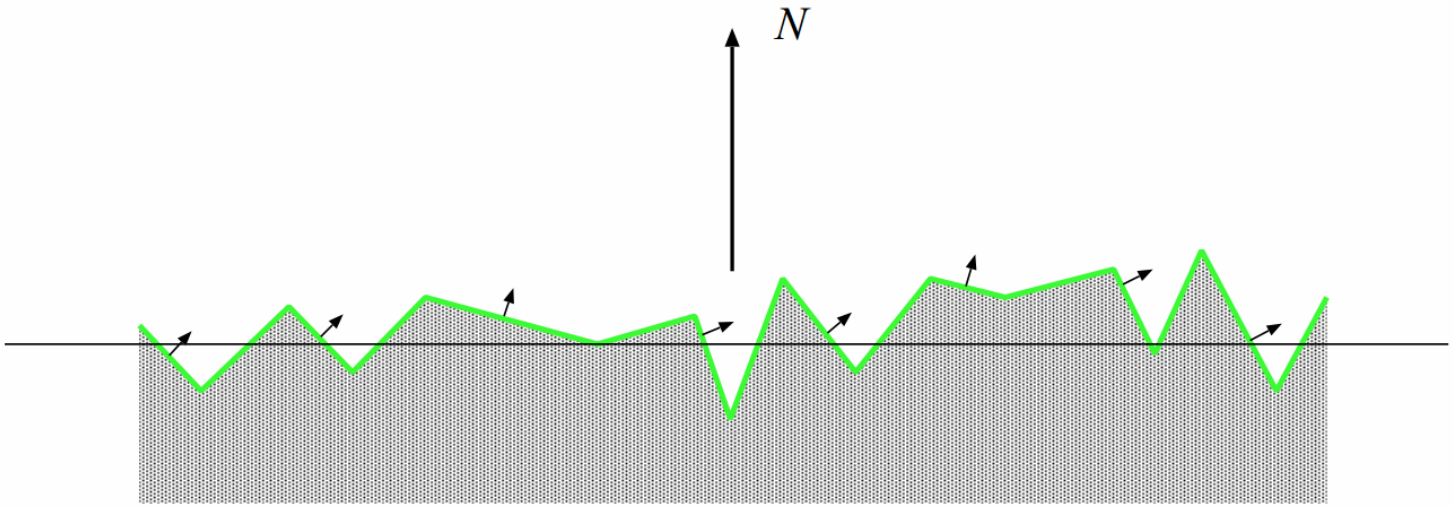
- In generale, se le variabili casuali X_i sono estratte da un funzione di densità di probabilità p(x) si ha:

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}$$

- Quindi l'integrale dell'equazione di rendering può essere risolto accumulando evidenze di path randomici
  - Pesati per la probabilità di loro occorrenza

Modelli di illuminamento avanzati
Cook-Torrance (1)

- Cook-Torrance (’82)
  - Derivato da un modello di Blinn del 77
  - Sono introdotte delle microfacce (microfacet) per modellare la componente speculare
  - Ogni microfaccia è considerata uno specchio perfetto



Modelli di illuminamento avanzati
Cook-Torrance (2)

- Cook-Torrance (’82)

$$I = I_a \cdot k_a + I_l \cdot \left[k_d \cdot (\mathbf{N} \cdot \mathbf{L}) + k_s \cdot \frac{D \cdot G \cdot F}{(\mathbf{N} \cdot \mathbf{V})(\mathbf{N} \cdot \mathbf{L})} \right]$$

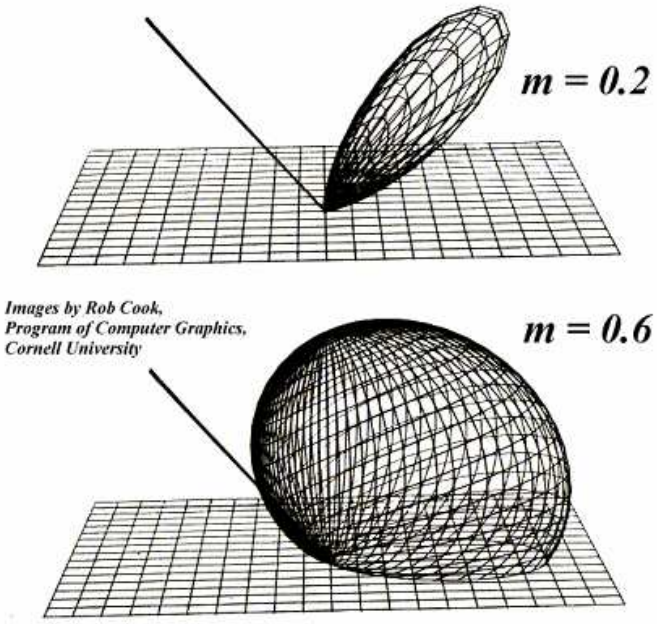
  - L’equazione è simile al modello di Blinn
    - Cambia la parte speculare
  - N: Media delle normali slla superficie
  - L: Direzione della luce
  - V: Direzione di vista

Modelli di illuminamento avanzati
Cook-Torrance (3)

- D è la funzione di distribuzione delle direzioni delle microfacce

$$D = \frac{e^{-\left(\frac{\tan\alpha}{m}\right)^2}}{4m^2\cos^4\alpha}$$

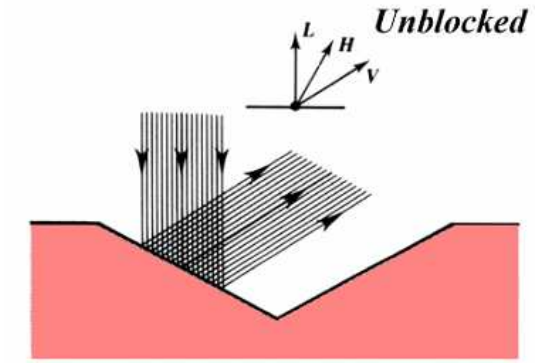
  - α è l'angolo tra **H** e **N**
  - m controlla la rugosità della superficie



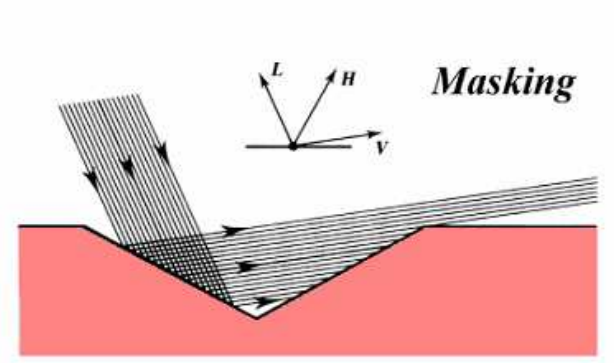
Images by Rob Cook, Program of Computer Graphics, Cornell University

Modelli di illuminamento avanzati
Cook-Torrance (4)

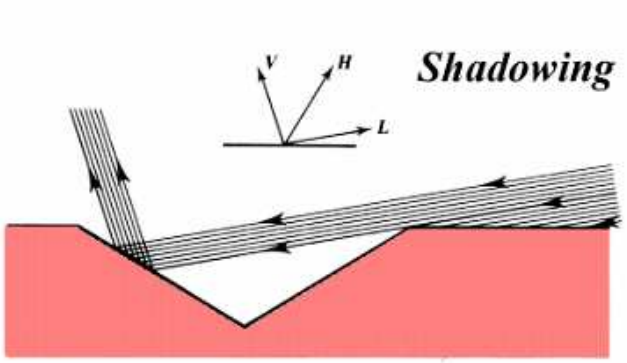
- G è un fattore di attenuazione dovuto alla geometria delle microfacce



$$G_1 = 1$$



$$G_2 = \frac{2 \cdot (\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})}{\mathbf{V} \cdot \mathbf{H}}$$



$$G_3 = \frac{2 \cdot (\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{L})}{\mathbf{L} \cdot \mathbf{H}}$$

$$G = \min\left\{1, \frac{2 \cdot (\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{V})}{\mathbf{V} \cdot \mathbf{H}}, \frac{2 \cdot (\mathbf{N} \cdot \mathbf{H})(\mathbf{N} \cdot \mathbf{L})}{\mathbf{L} \cdot \mathbf{H}}\right\}$$

Modelli di illuminamento avanzati
Cook-Torrance (5)

- F è il termine di Fresnel
  - Modella il comportamento della luce in materiali differenti
  - Dipende dalla lunghezza d'onda della luce λ e dalla direzione di luce incidente θ_i

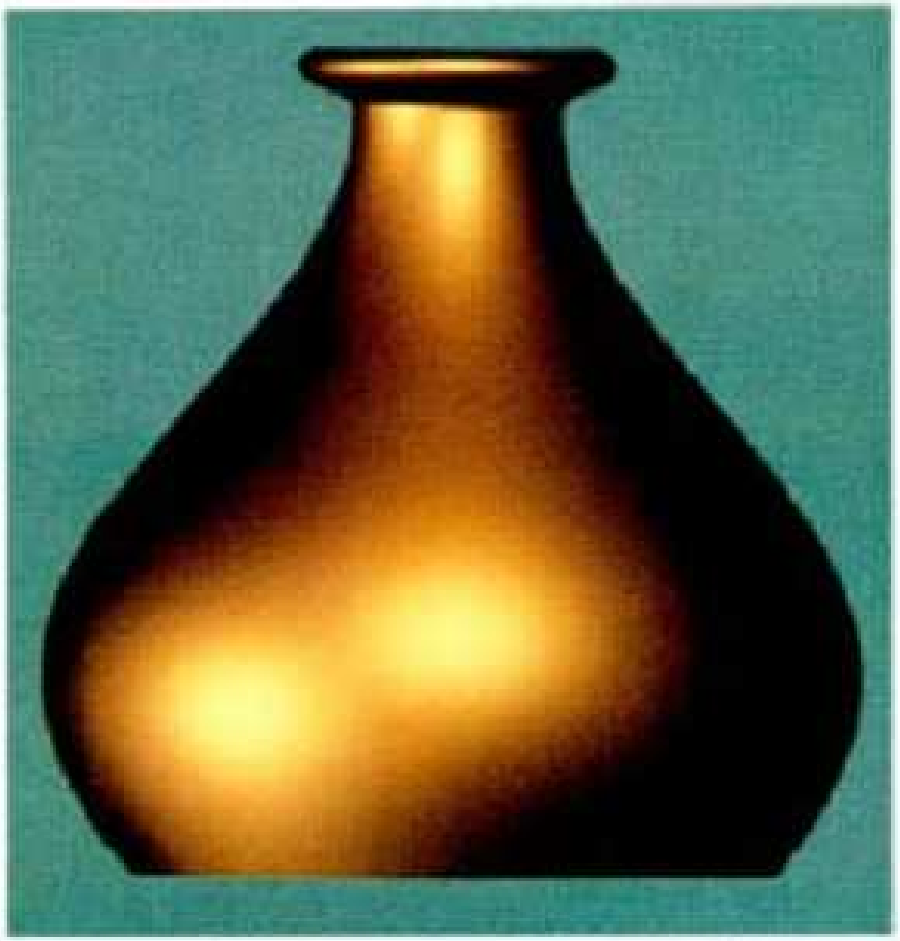
$$F_\lambda(\theta_i) = \frac{1}{2}\frac{(g-c)^2}{(g+c)^2}\left(1+\frac{\left(c(g+c)-1\right)^2}{\left(c(g-c)+1\right)^2}\right)$$

$$c = \cos\theta_i = \hat{\mathbf{L}}\cdot\hat{\mathbf{H}}$$

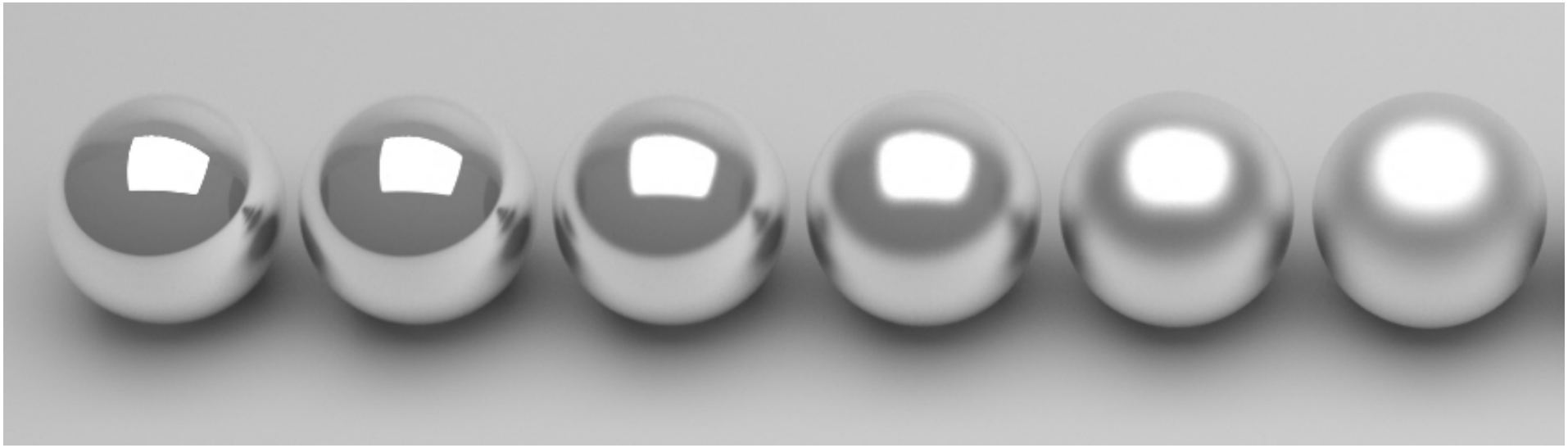
$$g = \sqrt{\eta^2 + c^2 - 1}$$

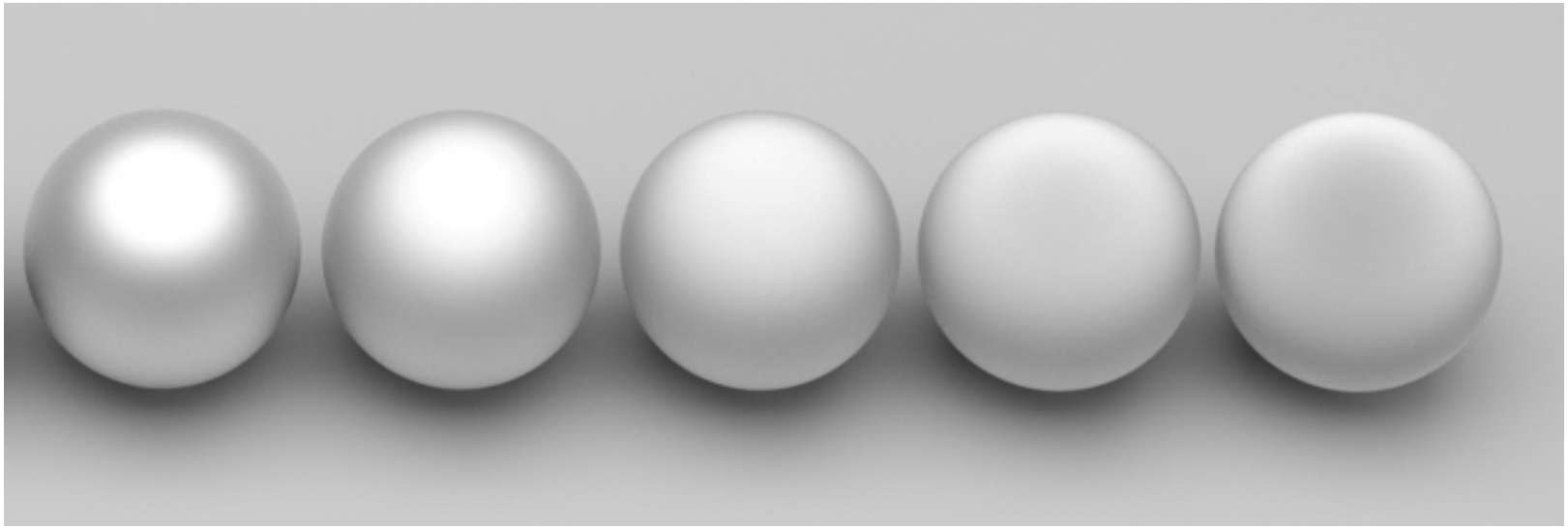
  - η è l'indice di rifrazione del materiale

Modelli di illuminamento avanzati
Cook-Torrance (6)



Modelli di illuminamento avanzati
Cook-Torrance (7)



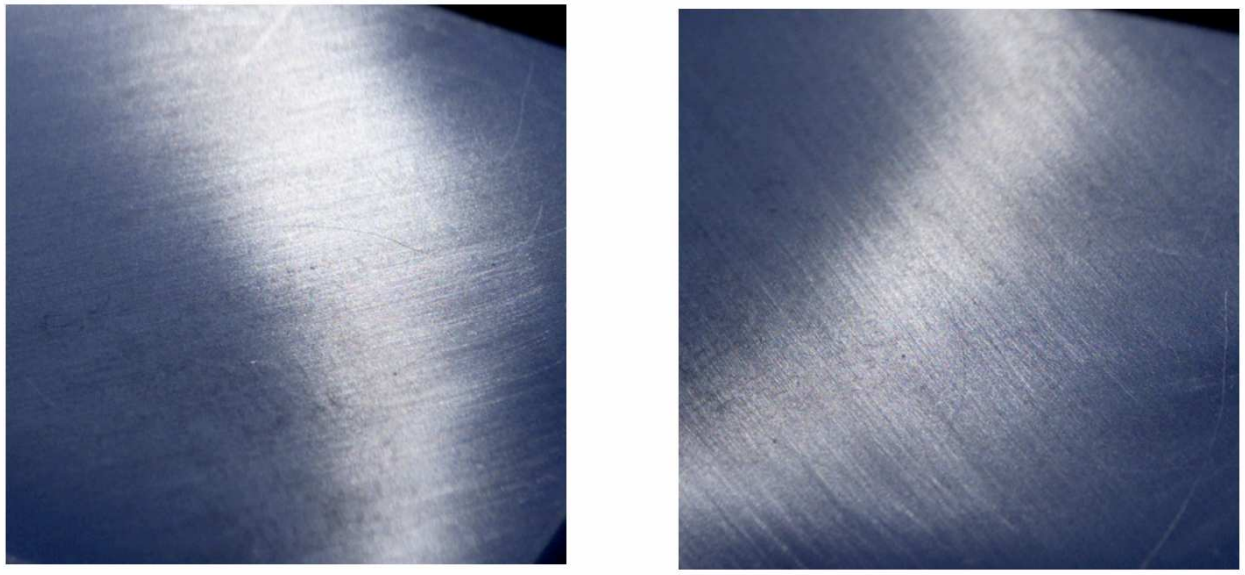


◆Roughness

Modelli di illuminamento avanzati
Ward (1)

- I modelli visti considerano il materiale isotropico
  - Ruotando l'oggetto, lo shading non cambia

- Molti materiali in natura sono anisotropici
  - Le microsuperfici hanno una orientazione preferenziale



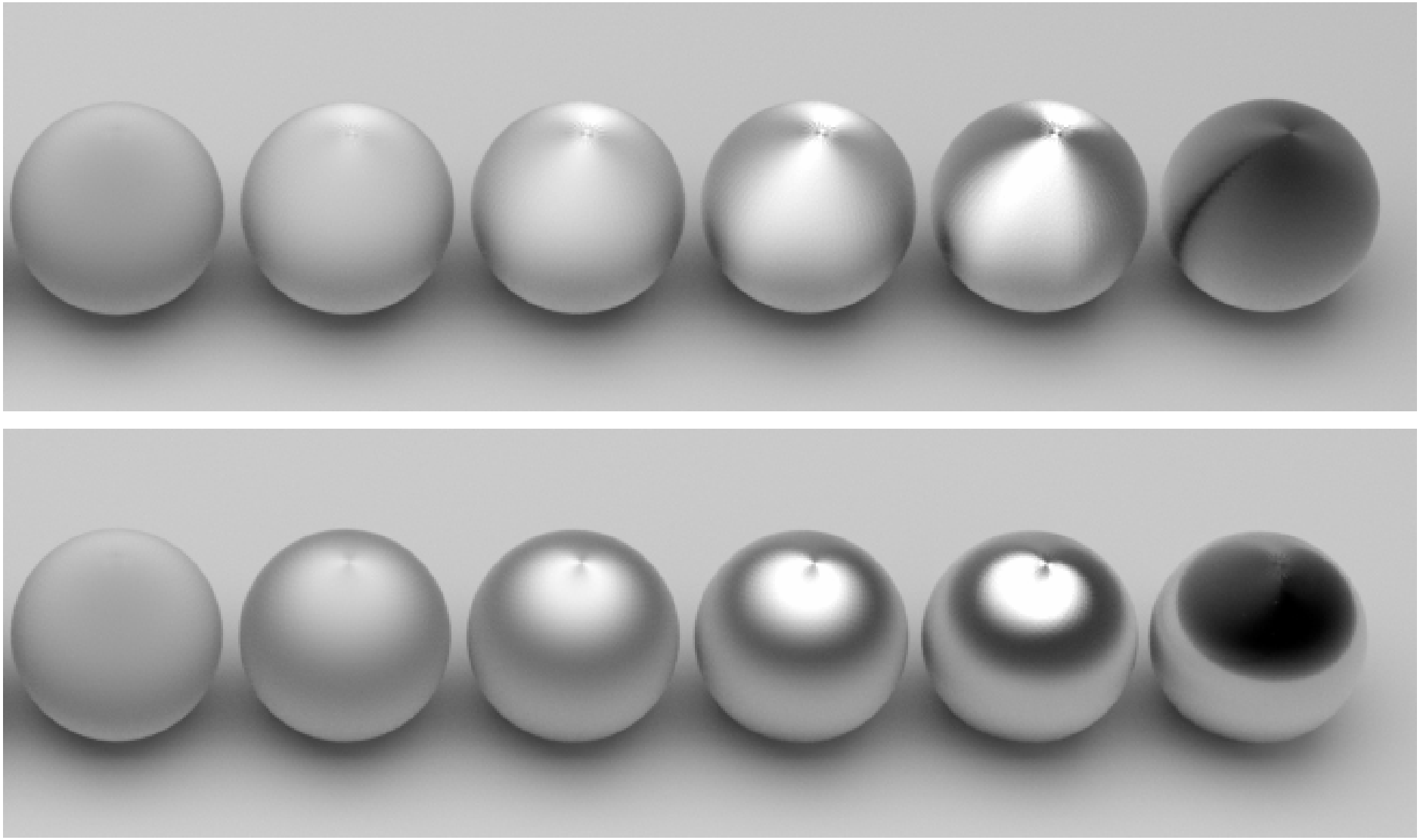
Modelli di illuminamento avanzati
Ward (2)

- Modello di Ward (’92)
  - La componente speculare è definita come

$$k_s = \frac{1}{\sqrt{(\mathbf{N}\cdot\mathbf{L})(\mathbf{N}\cdot\mathbf{V})}} \frac{\mathbf{N}\cdot\mathbf{L}}{4\pi\alpha_x\alpha_y} e^{\left(-2\frac{\left(\frac{\mathbf{H}\cdot\mathbf{X}}{\alpha_x}\right)^2 + \left(\frac{\mathbf{H}\cdot\mathbf{Y}}{\alpha_y}\right)^2}{1+(\mathbf{H}\cdot\mathbf{N})} \right)}$$

  - **X** e **Y** sono due vettori di uno spazio perpendicolare alla normale che definiscono le direzioni di anisotropia
  - α_x e α_y controllano l’entità dell’anistropia nelle due direzioni

Modelli di illuminamento avanzati
Ward (3)

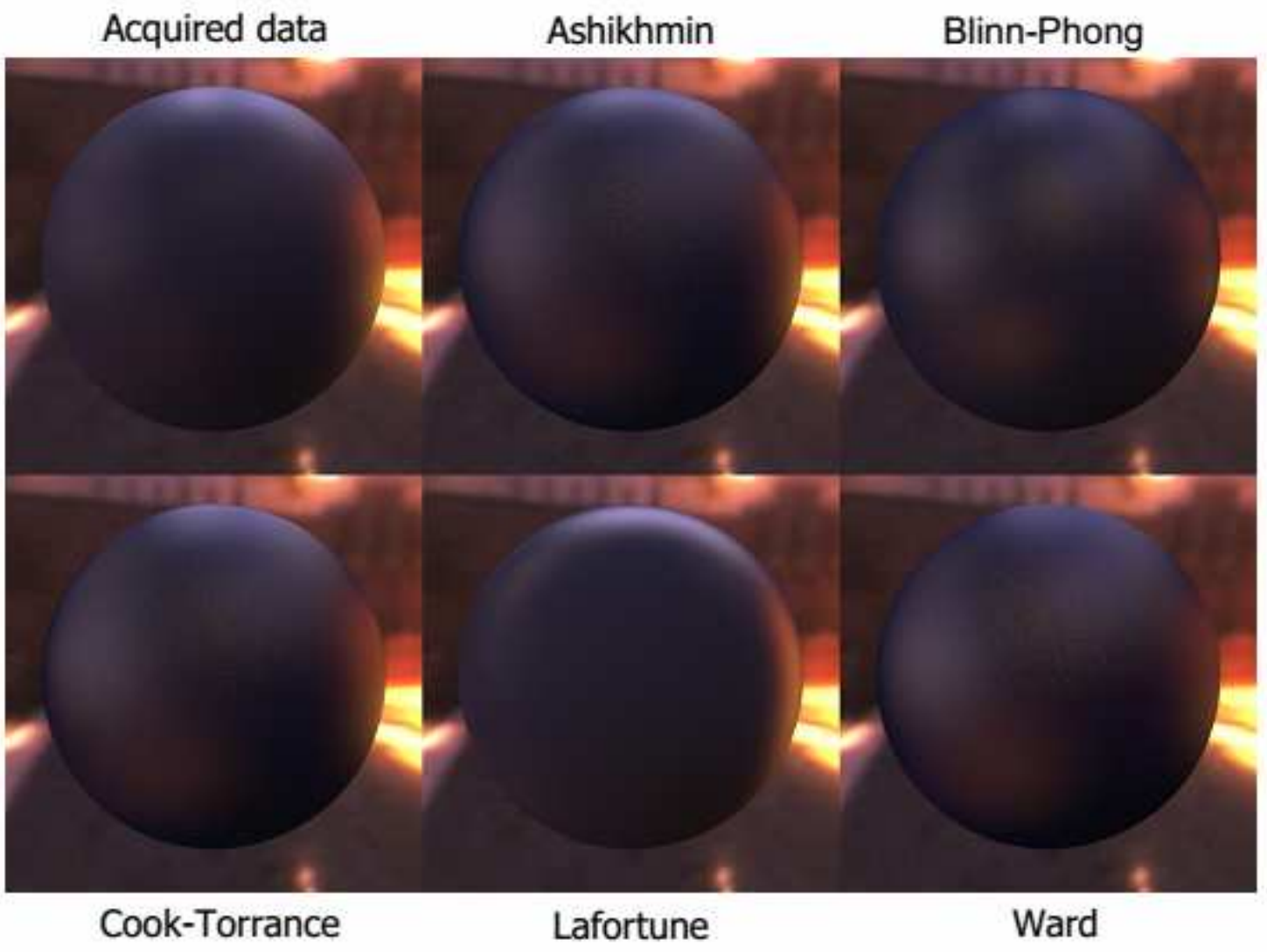


Anisotropy

Modelli di illuminamento avanzati
Ward (4)

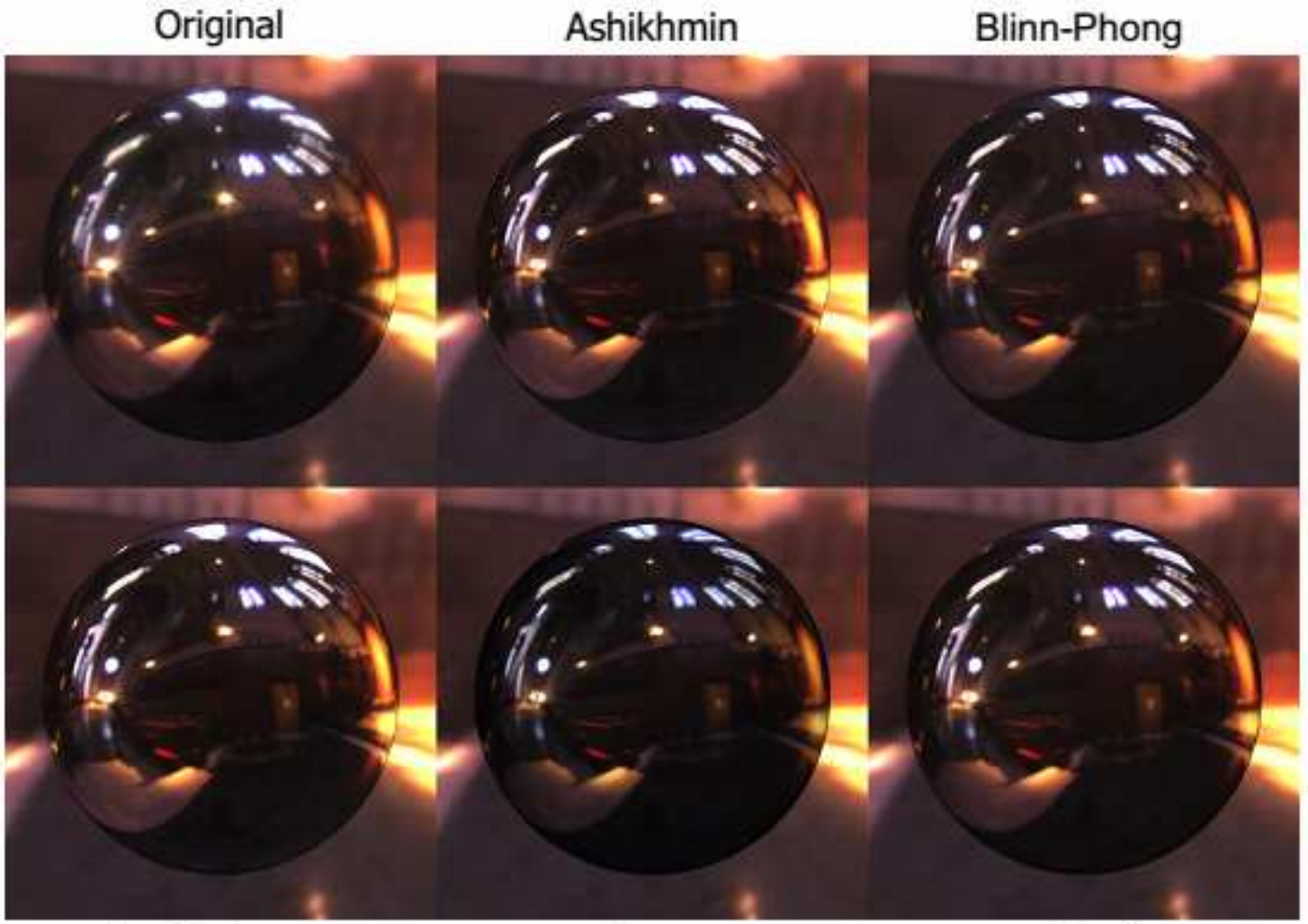
Anisotropy

Modelli di illuminamento avanzati



Material – Dark blue paint

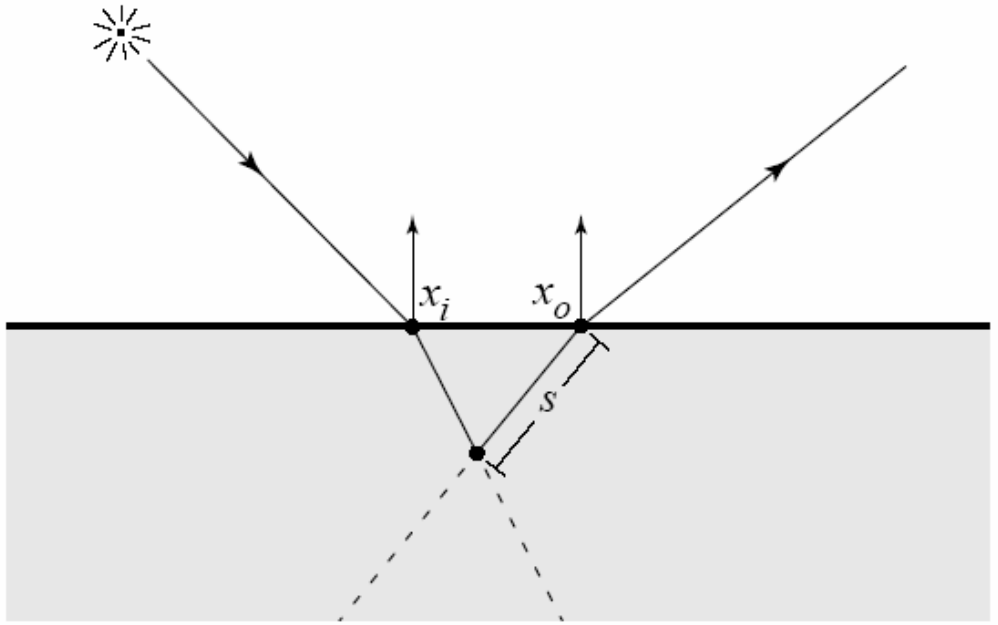
Modelli di illuminamento avanzati



Material – Chrome

BSSRDF (1)

- Bidirectional Surface Scattering Reflectance Distribution Function
  - La luce attraversa un materiale e viene riflessa da un punto diverso



Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy and Pat Hanrahan:
"A Practical Model for Subsurface Light Transport". Proceedings of SIGGRAPH'2001.

BSSRDF (2)

BRDF

BSSRDF

BSSRDF (3)

BRDF

BSSRDF

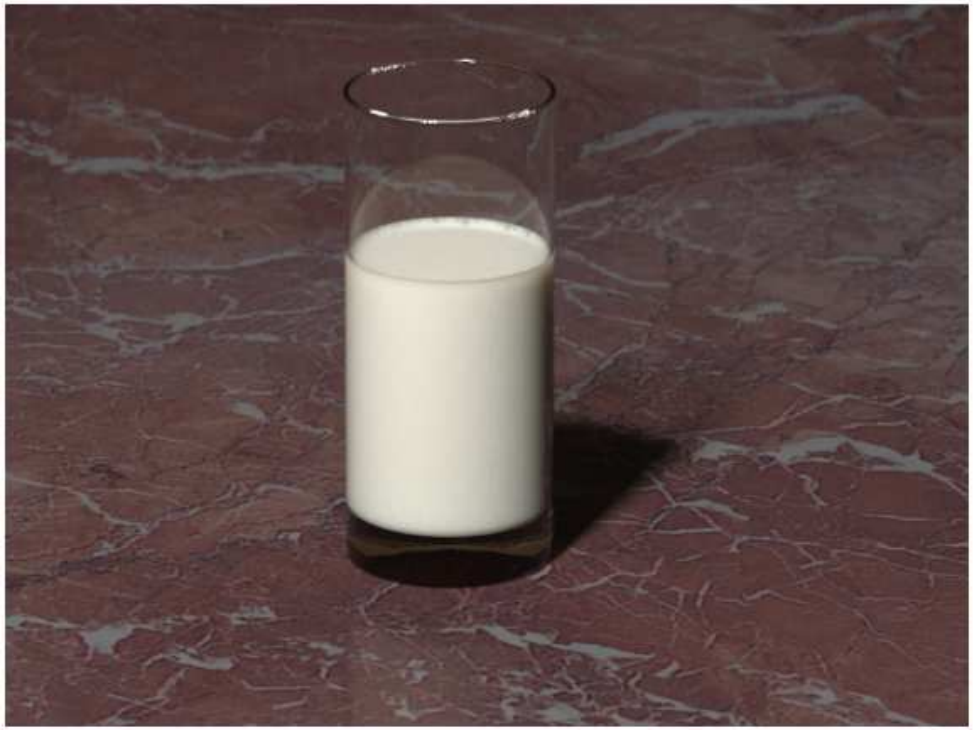
BSSRDF (4)

Photon Mapping

BSSRDF

BSSRDF (5)

BRDF



BSSRDF