

Image Based Lighting

Francesco Banterle

francesco.banterle@isti.cnr.it

Image Based Lighting

- **IDEA:** Illuminare oggetti virtuali utilizzando fotografie
- Perché?
 - Le sorgente luminose sono complesse da modellare:
 - Informazione colore
 - Distribuzione dell'intensità
 - Questa tecnica è una buona approssimazione della Global Illumination ed è molto utilizzata nei VFX

IBL: equazione

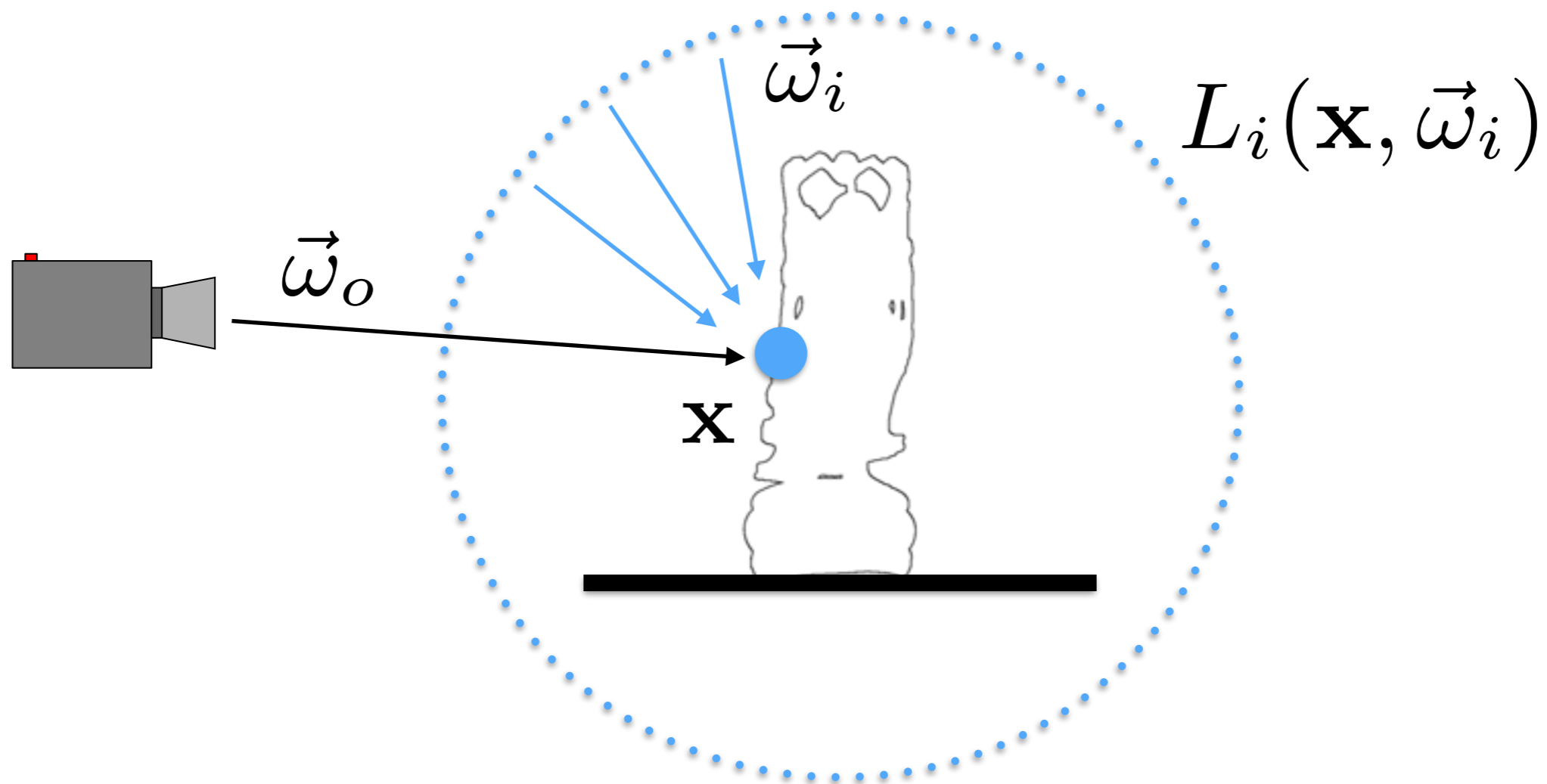
$$L_o(\mathbf{x}, \vec{\omega}_o) = L_e(\mathbf{x}, \vec{\omega}_o) + \int_{\Omega^+} L_i(\mathbf{x}, \vec{\omega}_i) f_r(\mathbf{x}, \vec{\omega}_i, \vec{\omega}_o) \langle \vec{n}, \vec{\omega}_i \rangle d\vec{\omega}_i$$



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

IBL: cattura delle sorgenti

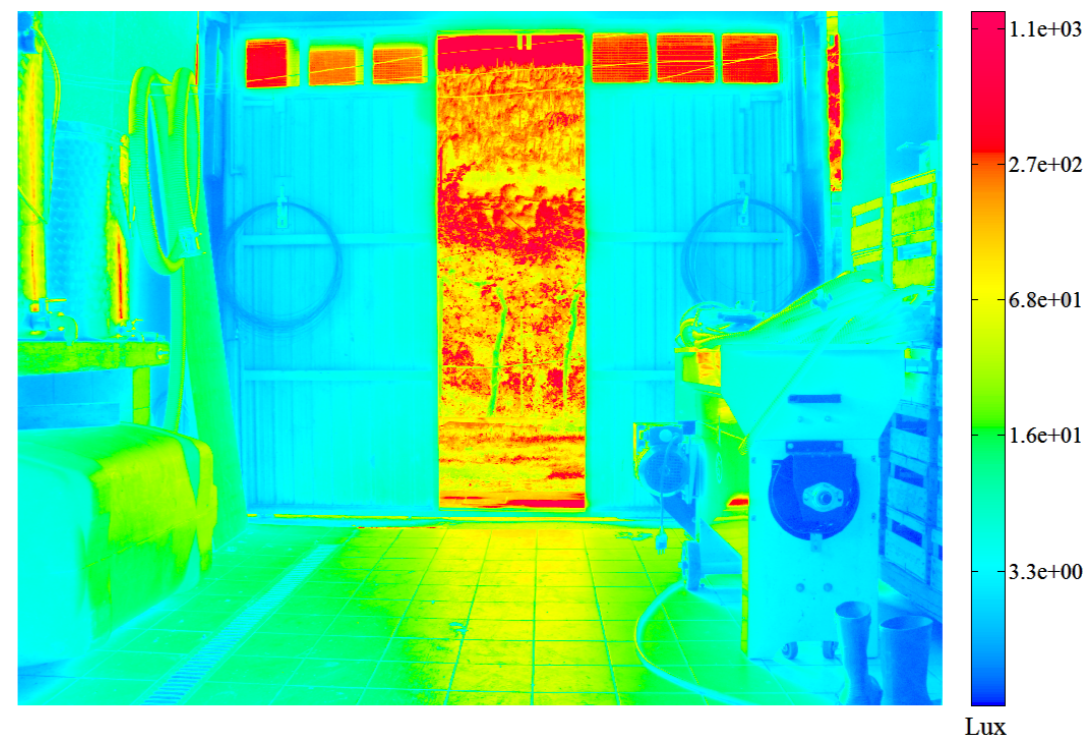
- Per l'IBL serve catturare una fotografia a 360 gradi che abbia l'informazione di radianza:



IBL: immagini HDR

- Le immagini panoramiche devono essere catturate utilizzando la tecnologia High Dynamic Range (HDR)
- Perché?
 - Catturare valori fisici espressi in $[\text{cd}/\text{m}^2]$
 - Non avere zone dell'immagine sovraesposte; dove c'è l'informazione sulle sorgenti luminose!!!

IBL: immagini HDR



IBL: cattura

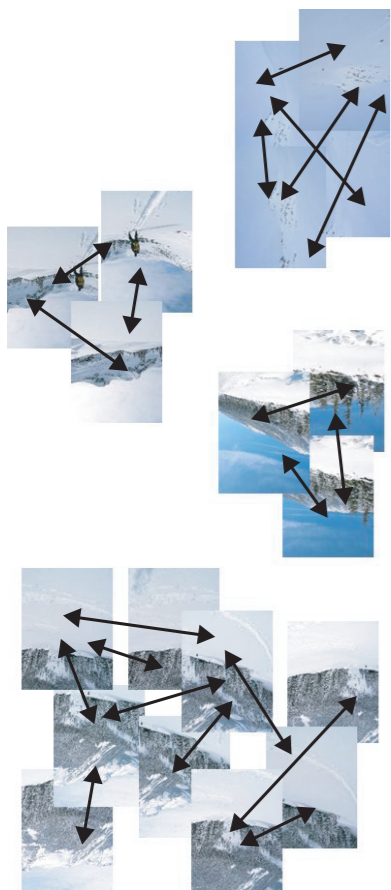


Fotografare
una sfera cromata



Fotografare con una
lente fisheye

IBL: cattura



[Brown and Lowe 2007]

IBL: cattura



SpheronVR HDR

- Macchina fotografica specifica per panorami
- 50Mpix
- 26 f-stops
- Allineamento automatico
- Molto costosa ~ 48mila Euro

IBL: parametrizzazione



Sphere Mapping:

Pro: compatto, pixel hanno area costante

Contro: basso sampling agli edge!



Angular Mapping:

Pro: compatto, sampling uniforme

Contro: i pixel non hanno area costante

IBL: parametrizzazione



Longitude-latitude mapping:

Pro: Facile da implementare e capire

Contro: distorsione ai poli; scalare per il coseno dell'angolo dall'altezza

Cube mapping:

Pro: supporto GPU

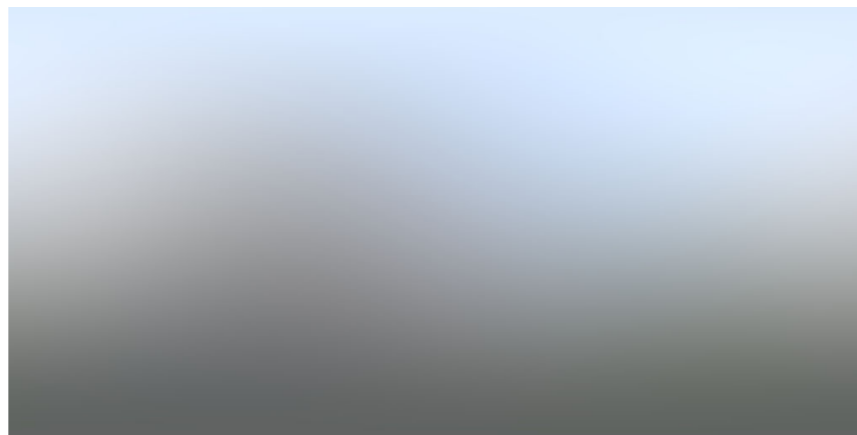
Contro: i pixel non hanno area costante, vanno compensati

IBL: rendering



$$L_o(\vec{\omega}_o) = \frac{\rho}{\pi} \int_{\Omega^+} L_i(\vec{\omega}_i) \langle \vec{n}, \vec{\omega}_i \rangle d\vec{\omega}_i$$

Convoluzione



IBL: rendering



BRDF costante
(look-up alla texture
convoluta)

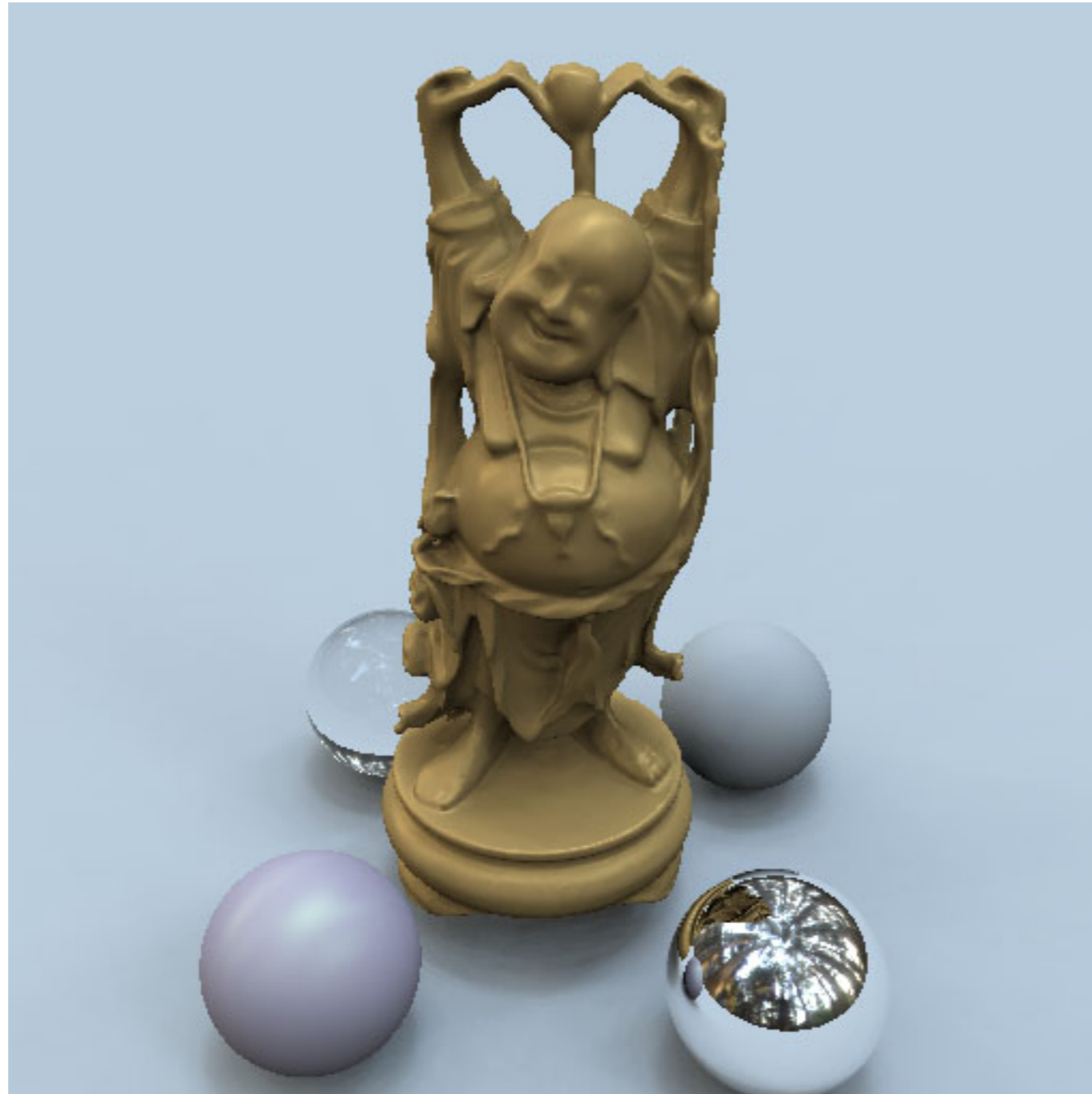


BRDF speculare:
riflessione pura (look-up
alla texture di sfondo)



BRDF speculare:
rifrazione pura (look-up
alla texture di sfondo)

IBL: rendering



IBL: rendering

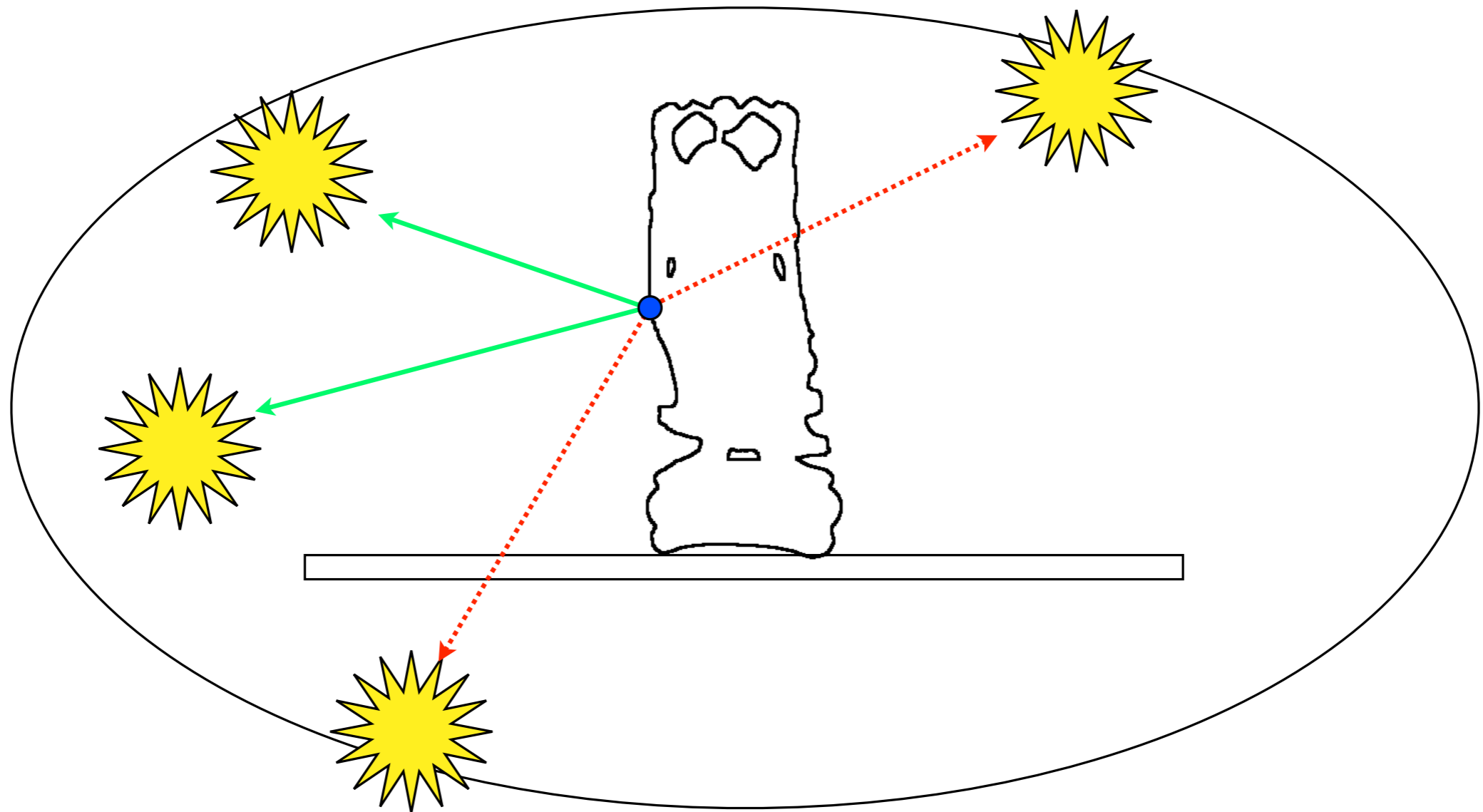
- L'equazione del rendering deve essere valutata in modo corretto, tenendo conto:
 - visibilità
 - BRDF

IBL: rendering

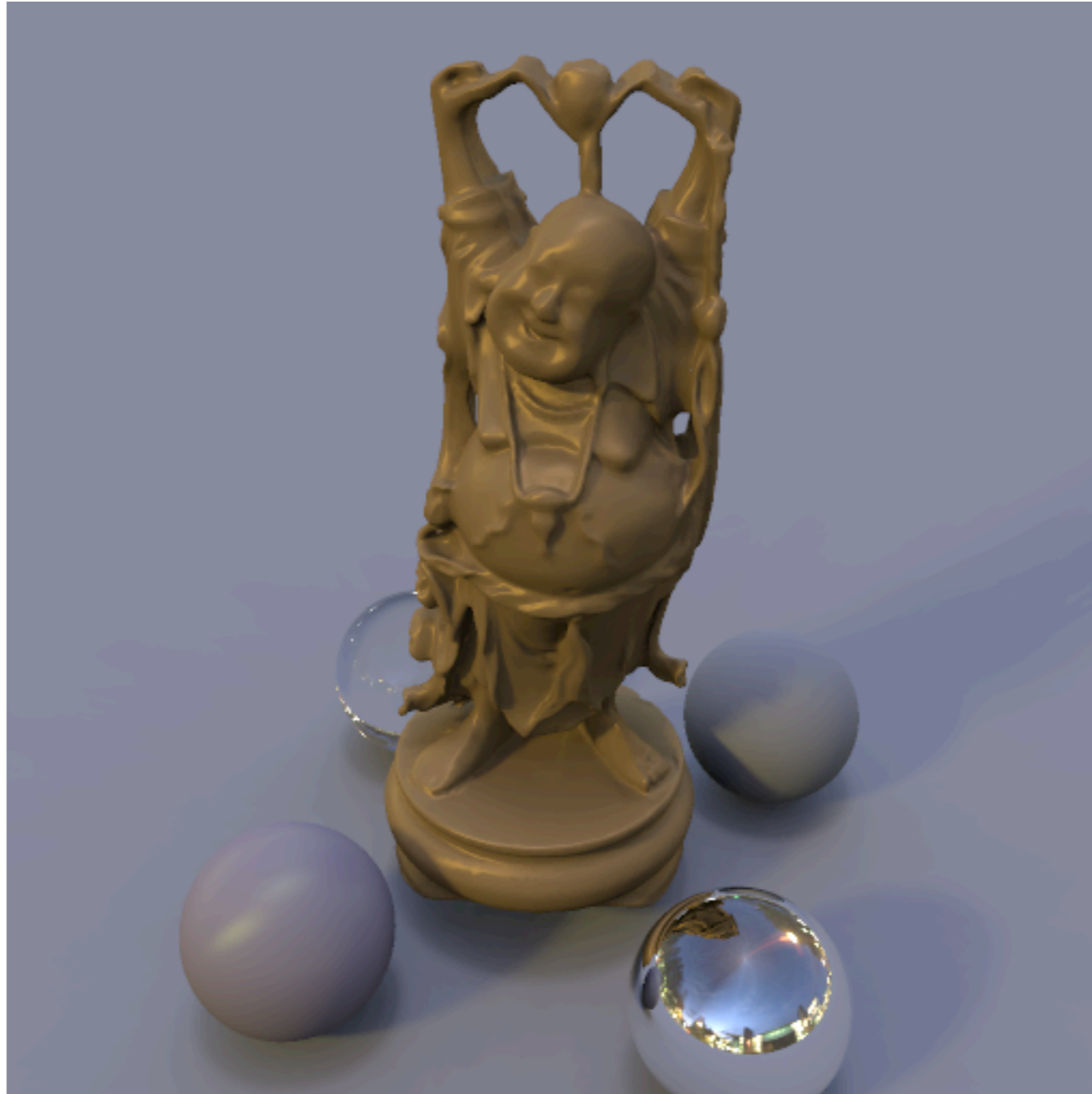
- Ci sono due approcci tipicamente per il calcolo dell'IBL:
 - **Discreto**: l'environment map viene convertita in sorgenti luminose; e.g. luci direzionali
 - **Continuo**: l'environment map viene campionata come nel caso delle *area lights* utilizzando opportune tecniche come l'importance sampling

IBL: approccio discreto

- Creazione di un'array di luci direzionali



IBL: approccio discreto



IBL: approccio discreto

- Algoritmo a campionamento uniforme:
 - **Input:** numero di luci; n
 - **Output:** n luci direzionali; direzione, colore e intensità
 - L'environment map viene suddivisa regolarmente in n patch
 - Per ogni i -esima patch:
 - viene calcolato il centroide della patch; media pesata utilizzando la luminaria; le coordinate (i, j) del centroide viene convertito in una direzione (x, y, z) ; la direzione della i -esima luce
 - viene calcolata la somma dei valori di colore e di intensità all'interno della patch e sono assegnati alla i -esima luce

IBL: approccio discreto

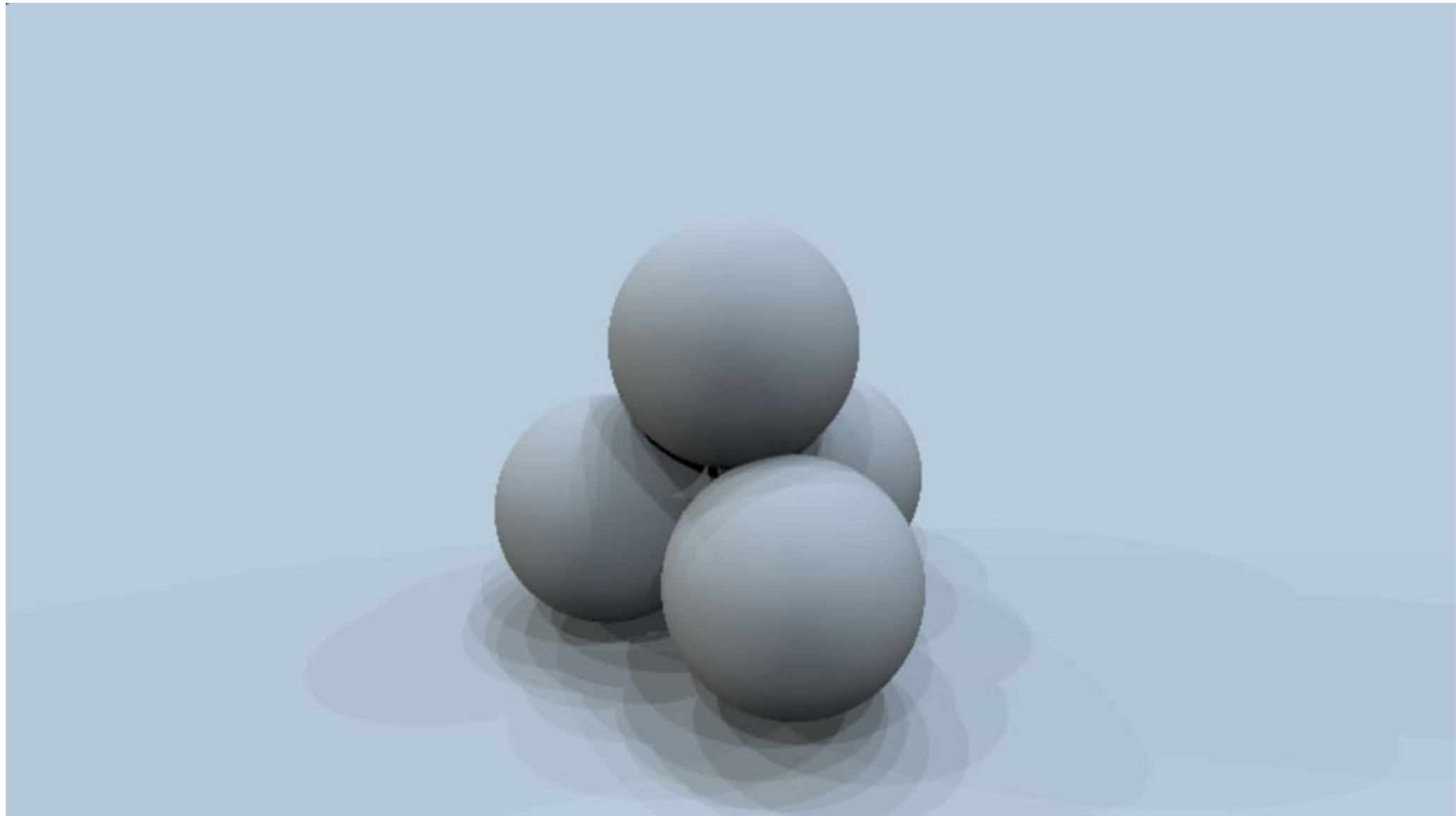
- Algoritmo “funziona”; però:
 - *sottocampiona* le zone chiare — con tanta informazione
 - *sovracampiona* le zone scure — poca informazione

IBL: approccio discreto



1024 luci direzionali

IBL: approccio discreto



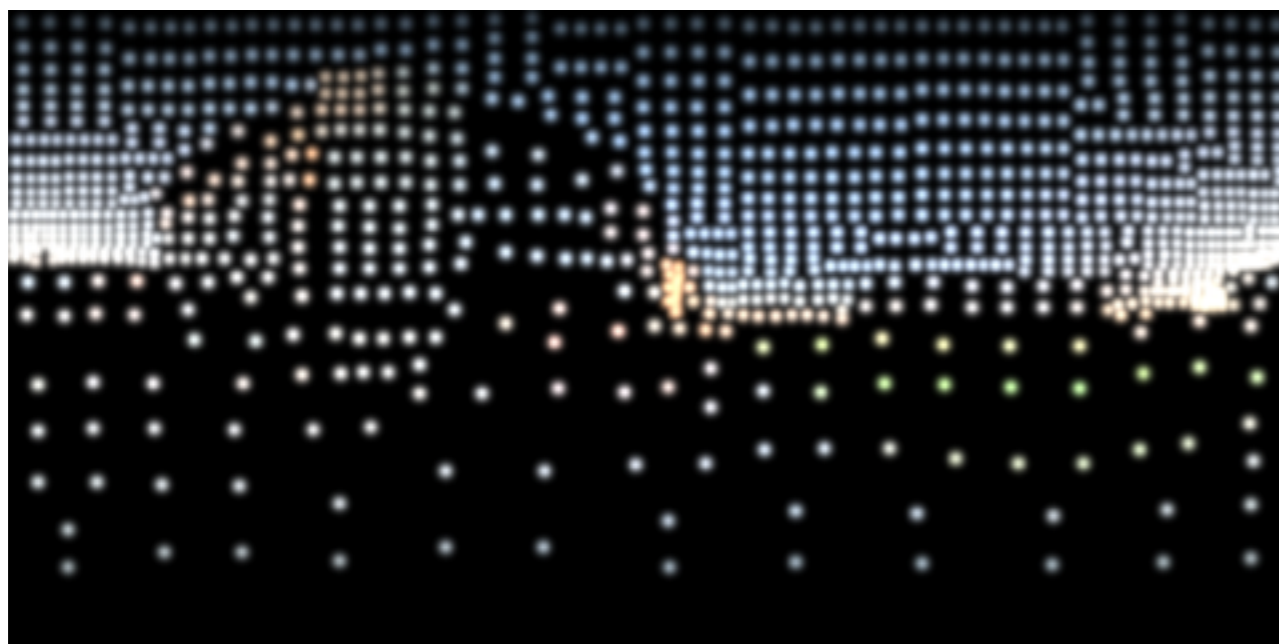
IBL: approccio discreto

- Algoritmo a campionamento *median-cut*:
 - **Input**: numero di luci; n
 - **Output**: n luci direzionali; direzione, colore e intensità
 - Per $\log_2 n$ iterazioni
 - il punto di taglio della dimensione più lunga (x o y) viene calcolato tale che l'energia dell'environment map è suddivisa in modo equo
 - All'ultima iterazione, una regione foglia viene creata
 - Per ogni regione foglia creata; viene calcolata una direzione, colore ed intensità come per l'algoritmo uniforme

IBL: approccio discreto

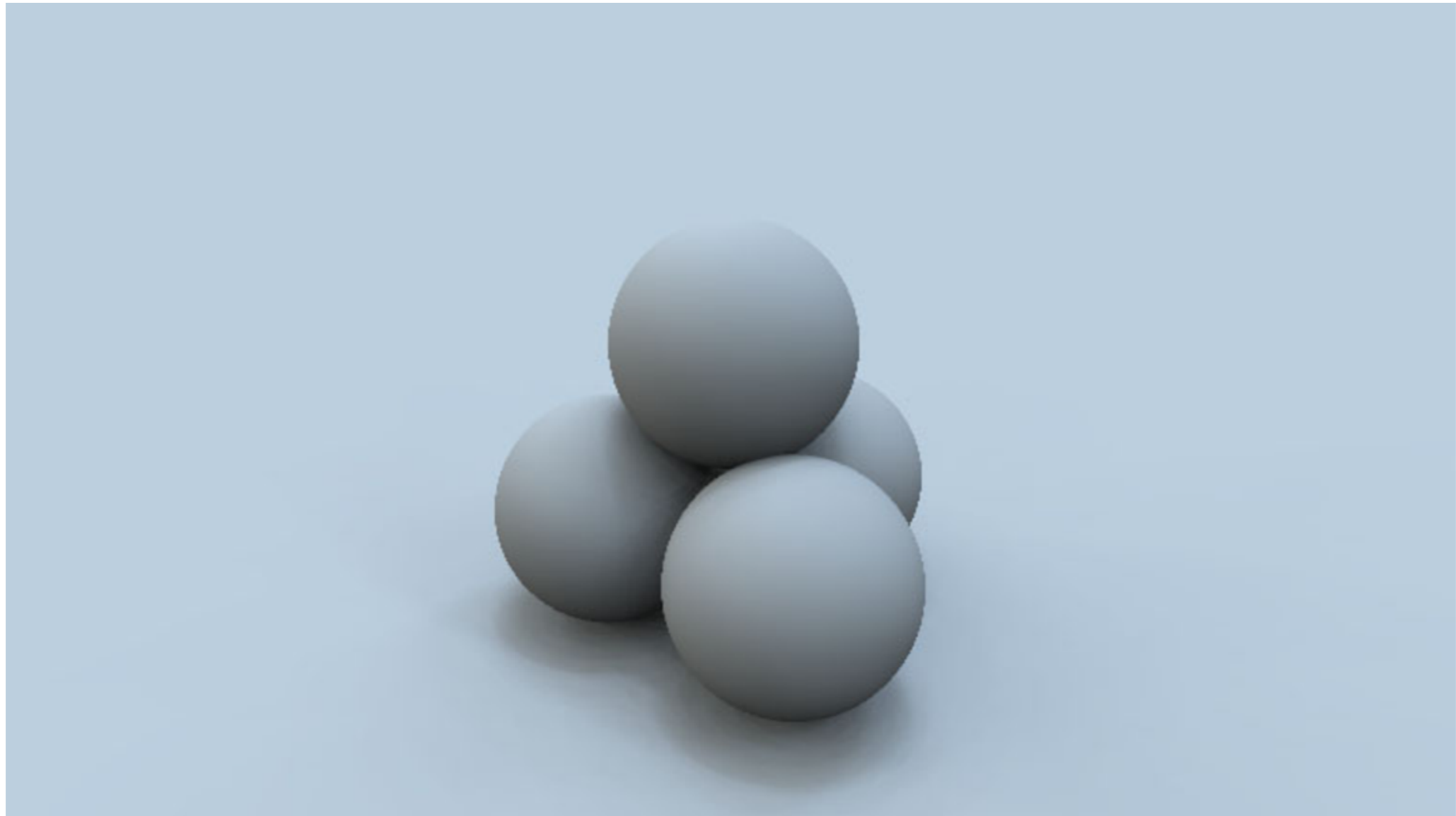


IBL: approccio discreto



1024 luci direzionali

IBL: approccio discreto



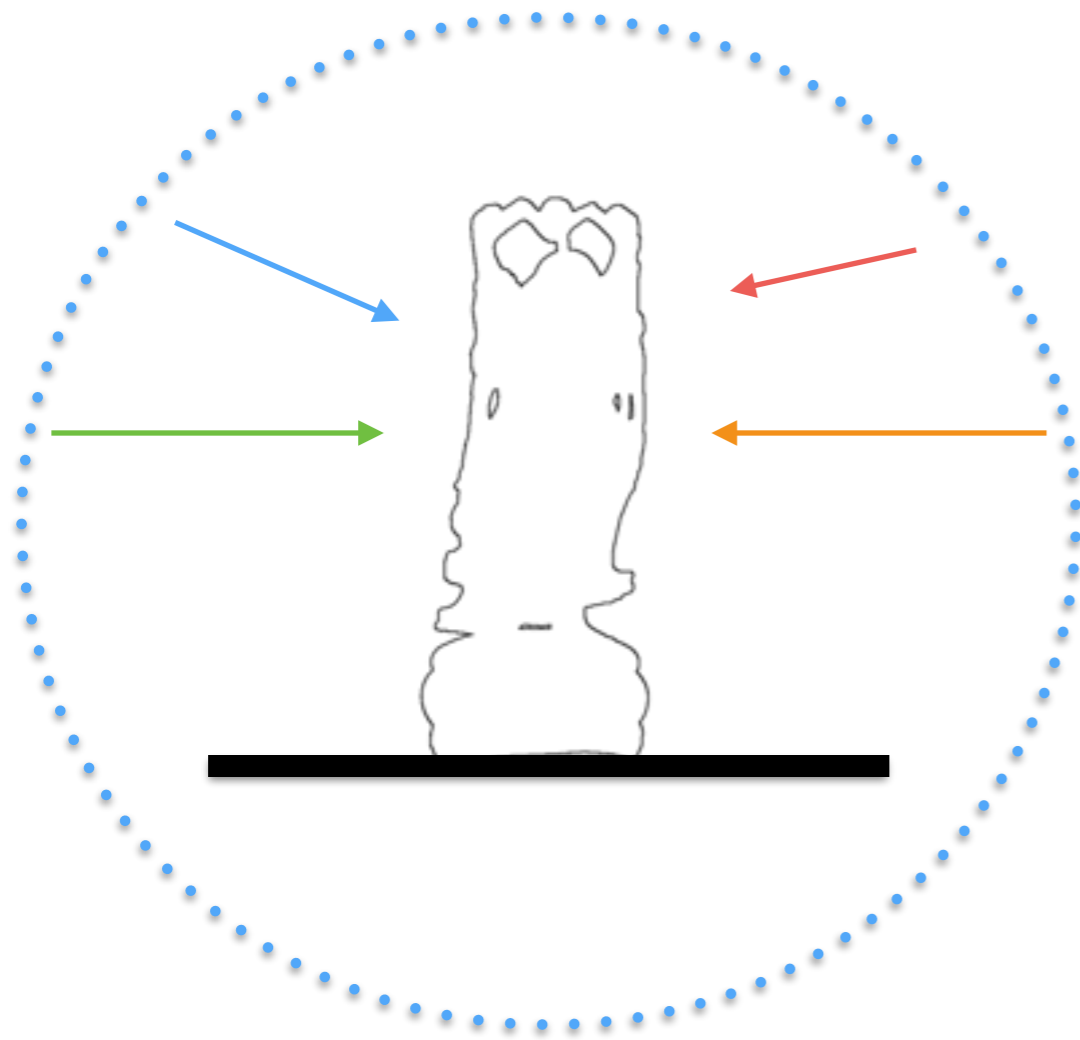
IBL: approccio discreto

- L'approccio discreto può generare migliaia di luci:
 - per poter avere un'elevata qualità
 - per limitare il bias; aumentare la convergenza all'integrale
- Tante luci \longrightarrow tanto tempo!

IBL: approccio discreto

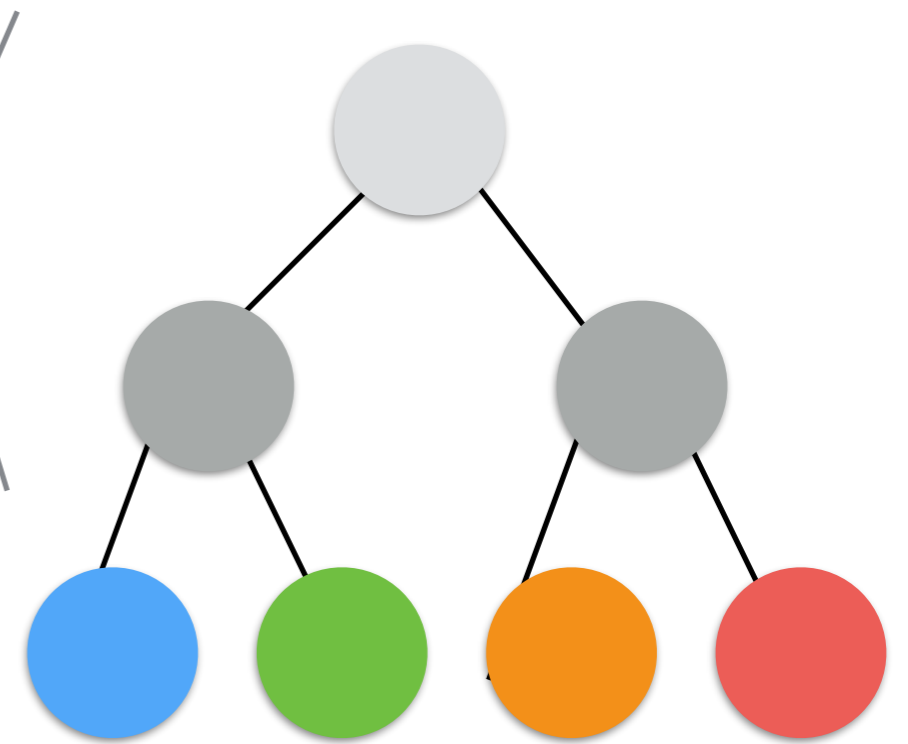
- Ci sono vari approcci per velocizzare il calcolo in presenza di tante luci:
 - Irradiance cache o Instant Caching: questo metodo funziona bene però funziona solo per materiali con BRDF costante!
 - Light-cuts: creare un albero delle luci ed effettuare dei tagli

IBL: approccio discreto

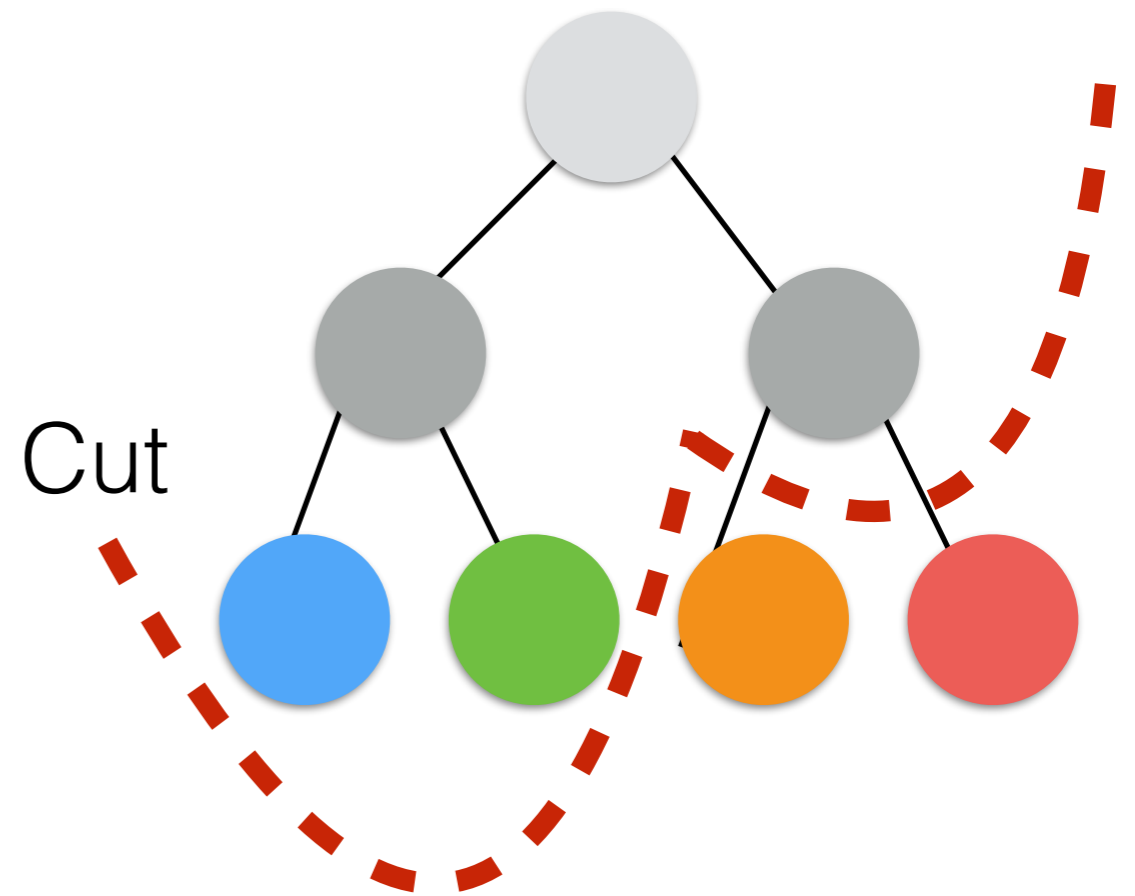
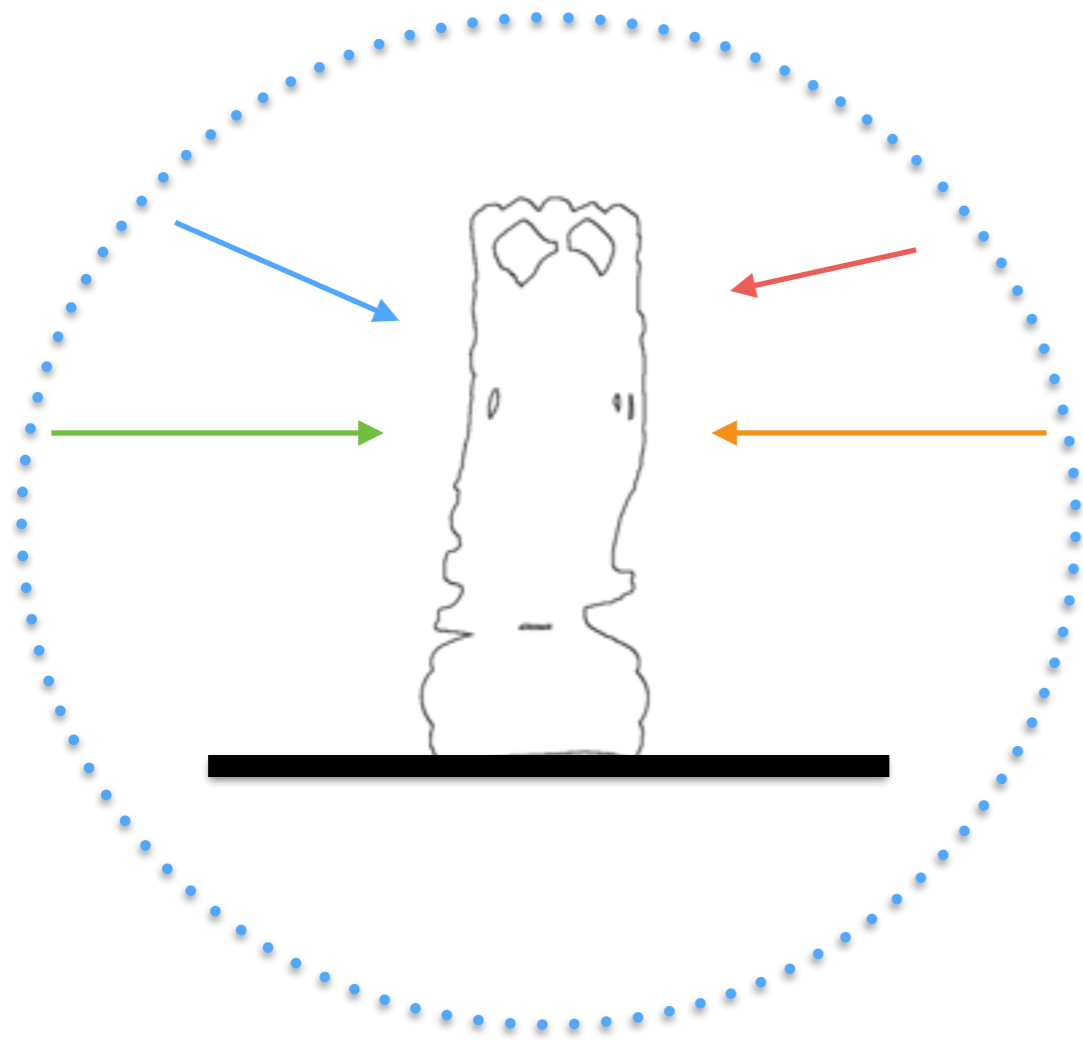


Cluster

Luci



IBL: approccio discreto



IBL: approccio continuo

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

IBL: integrazione montecarlo

$$F = \int_a^b f(x) dx$$

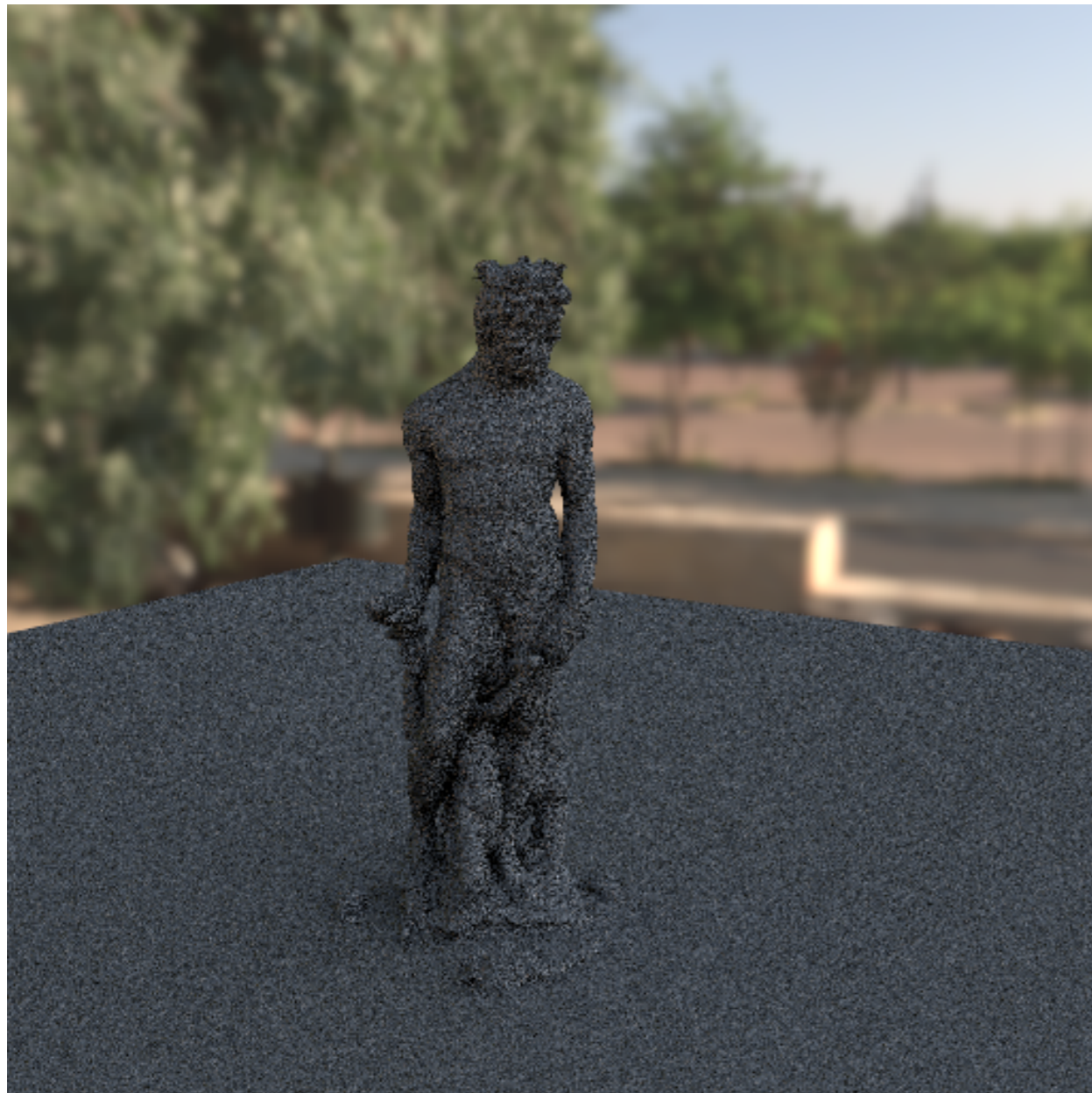
$$F_N = \frac{b-a}{N} = \sum_{i=1}^N f(X_i) \quad X_i \in [a, b]$$

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

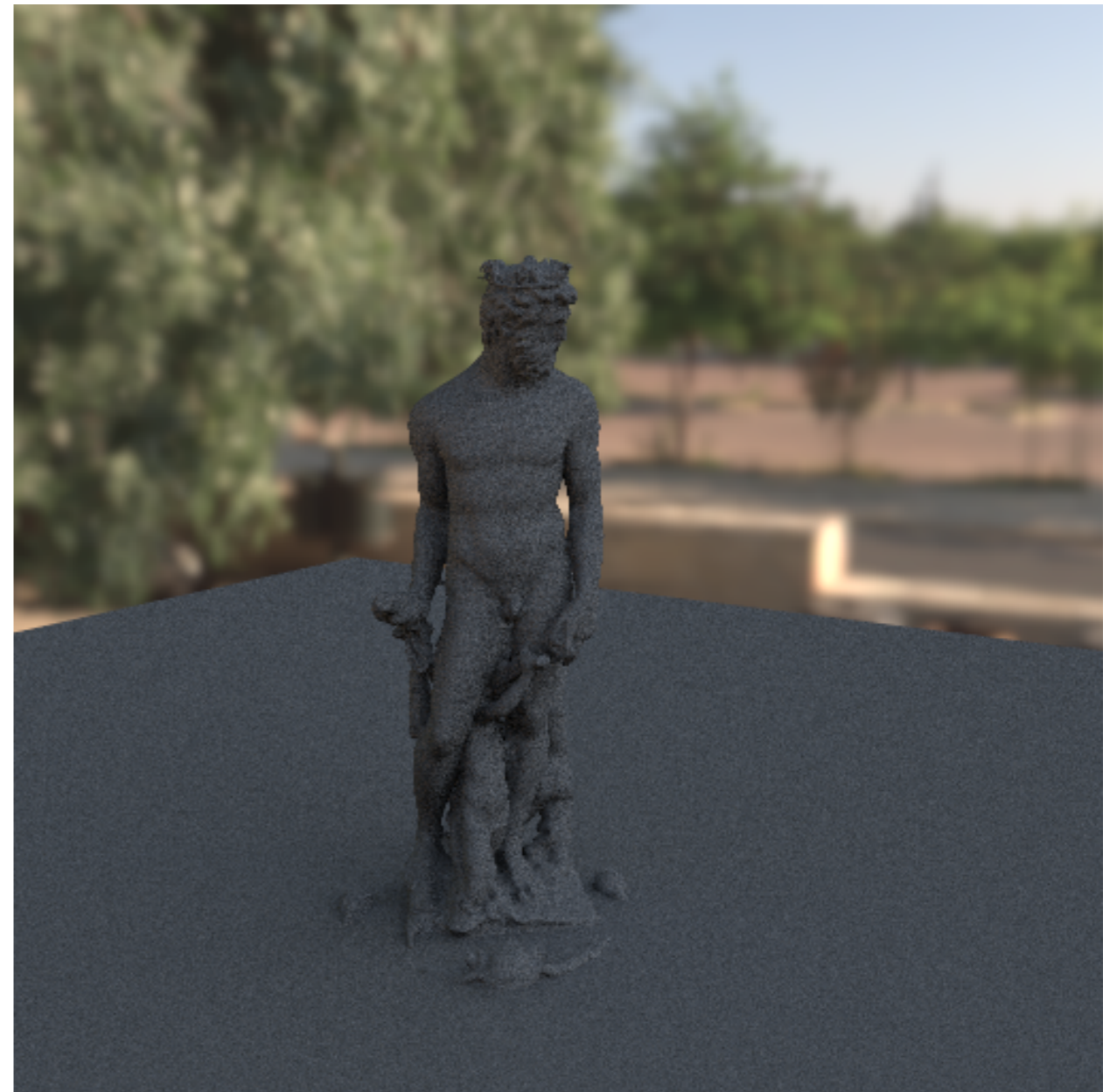
IBL: integrazione montecarlo

- Algoritmo base:
 - **Assume:** $p(X)$ uniforme
 - Quindi viene generata una direzione casuale per ogni direzione e si valuta l'integrale!

IBL: integrazione montecarlo



16 samples



128 samples

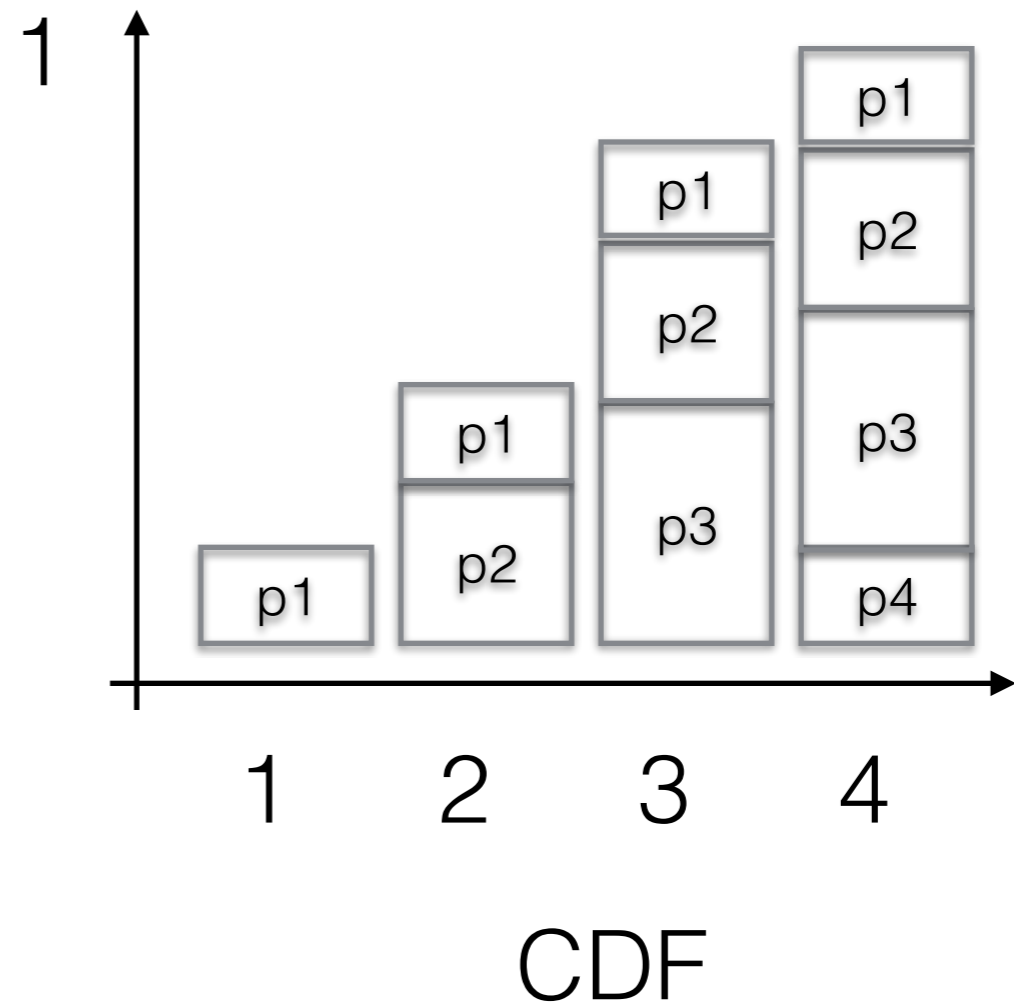
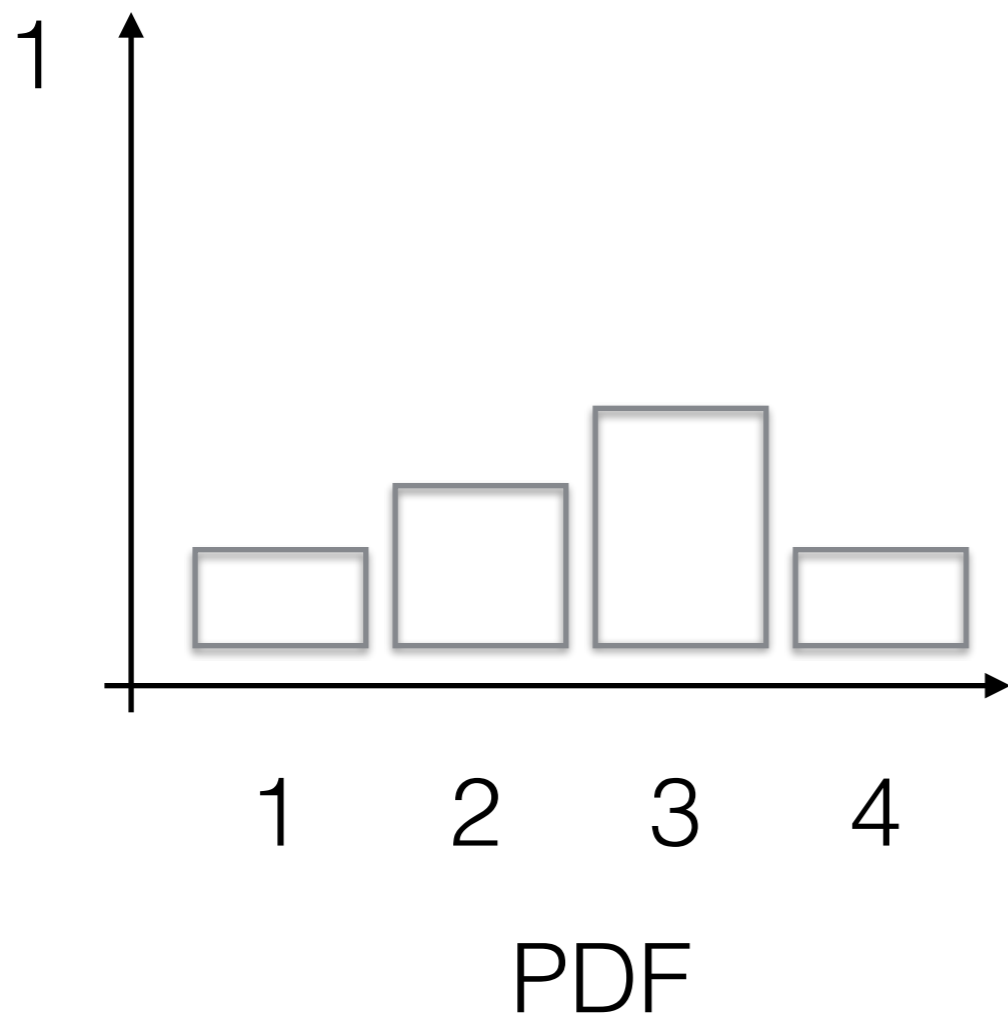
IBL: integrazione montecarlo

- Possiamo utilizzare come PDF (probability density function o funzione di densità di probabilità), $p(X)$, l'illuminazione:
 - l'immagine panoramica, L_i

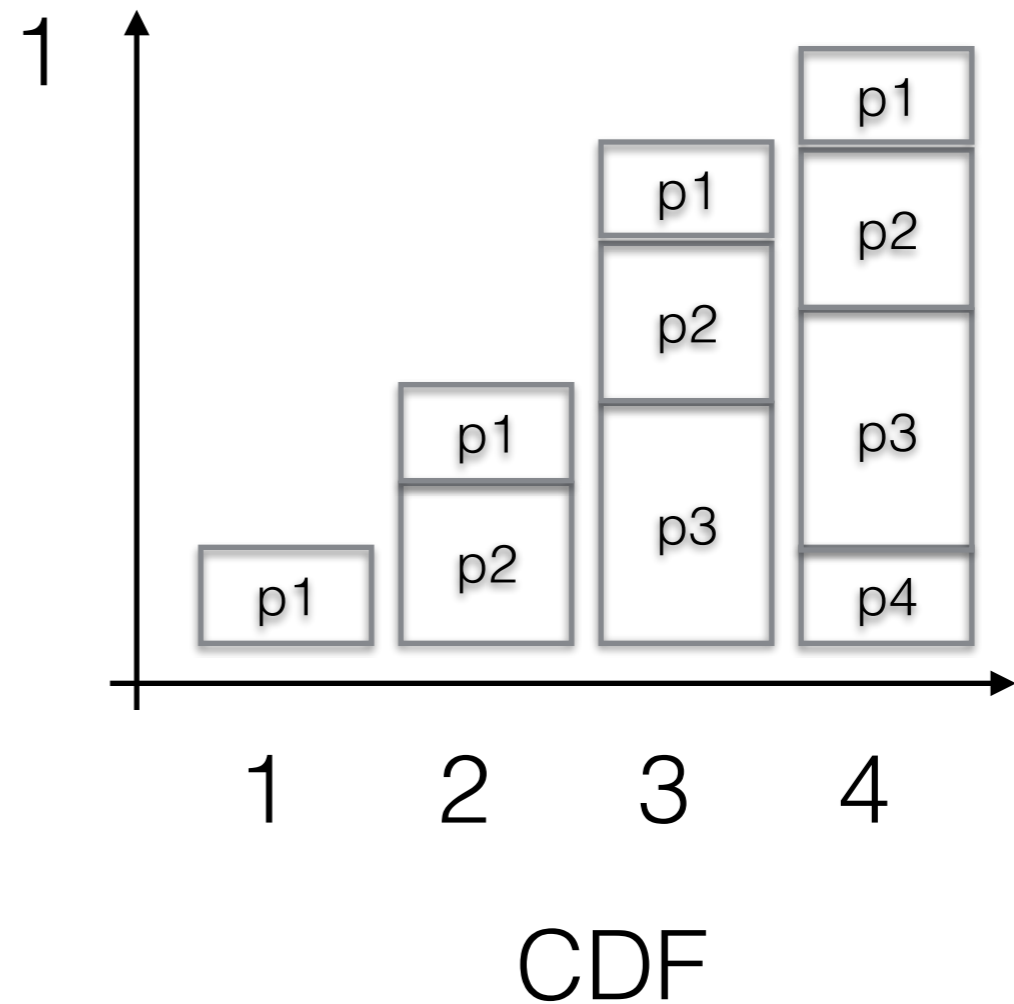
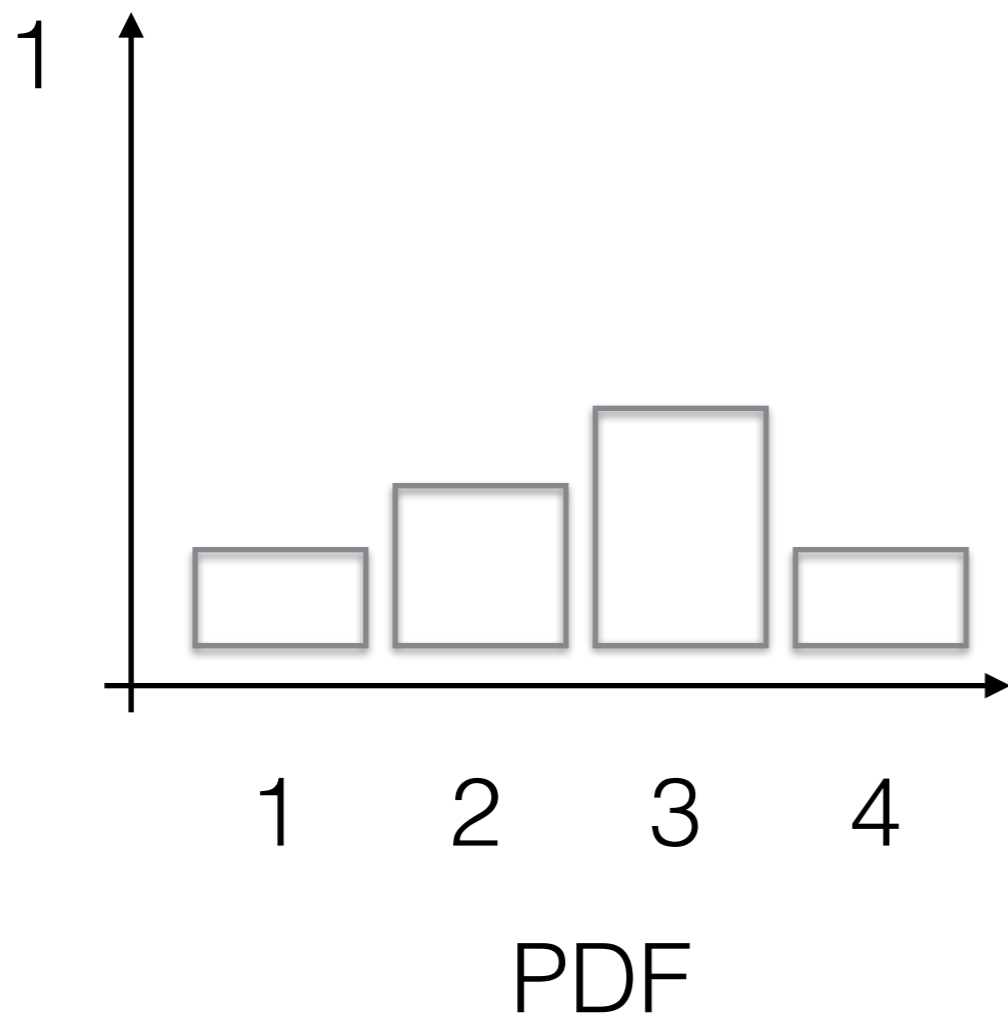
IBL: integrazione montecarlo

- Algoritmo di importance sampling 1D:
 - Calcolo della CDF (cumulative distribution function o funzione di probabilità cumulata), $P(x)$, di $p(x)$
 - Calcolo dell'inversa $F = P^{-1}(x)$
 - Calcolo di un numero casuale uniforme, e
 - Calcolo di $X_i = F(e)$

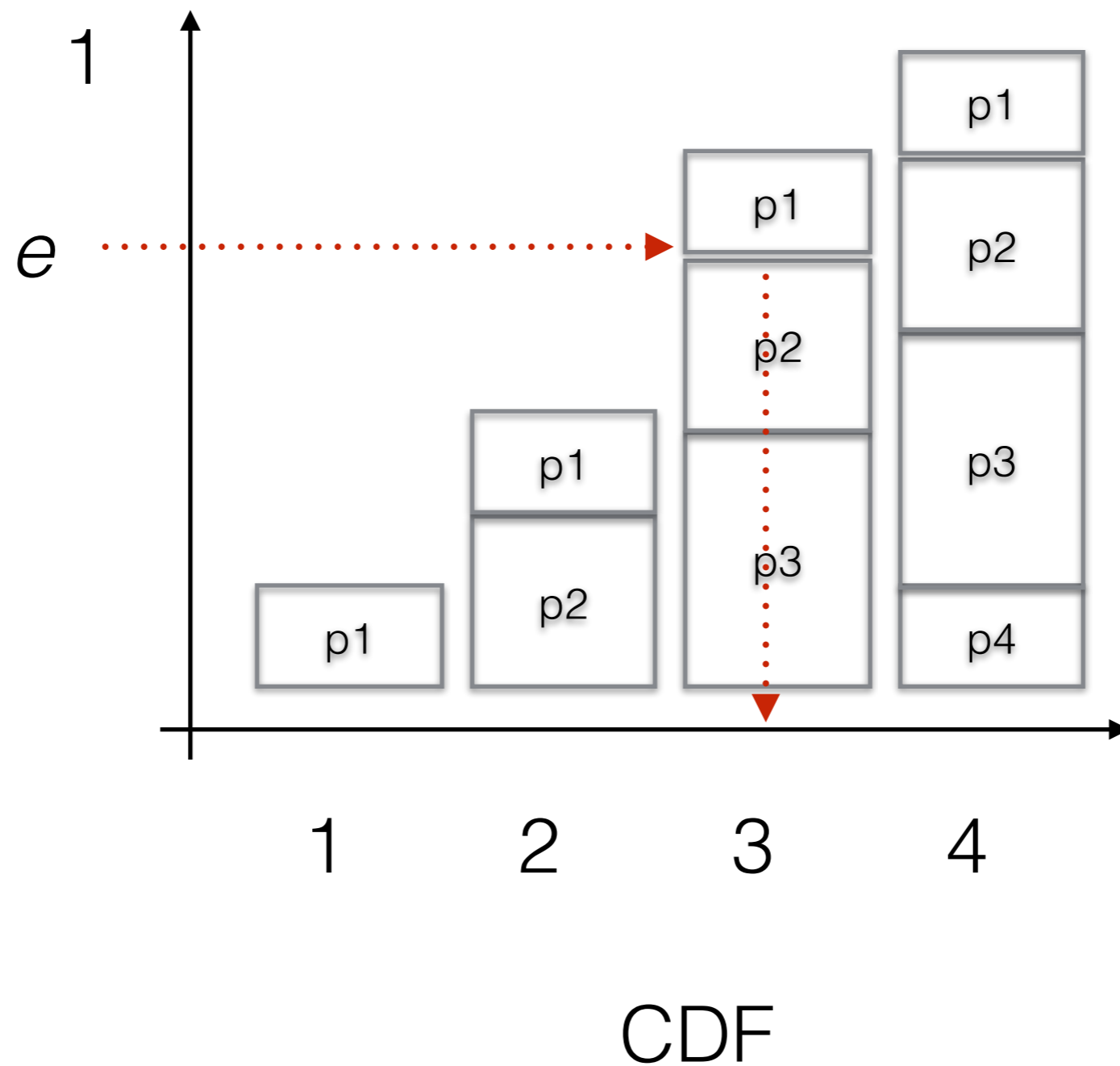
IBL: integrazione montecarlo



IBL: integrazione montecarlo



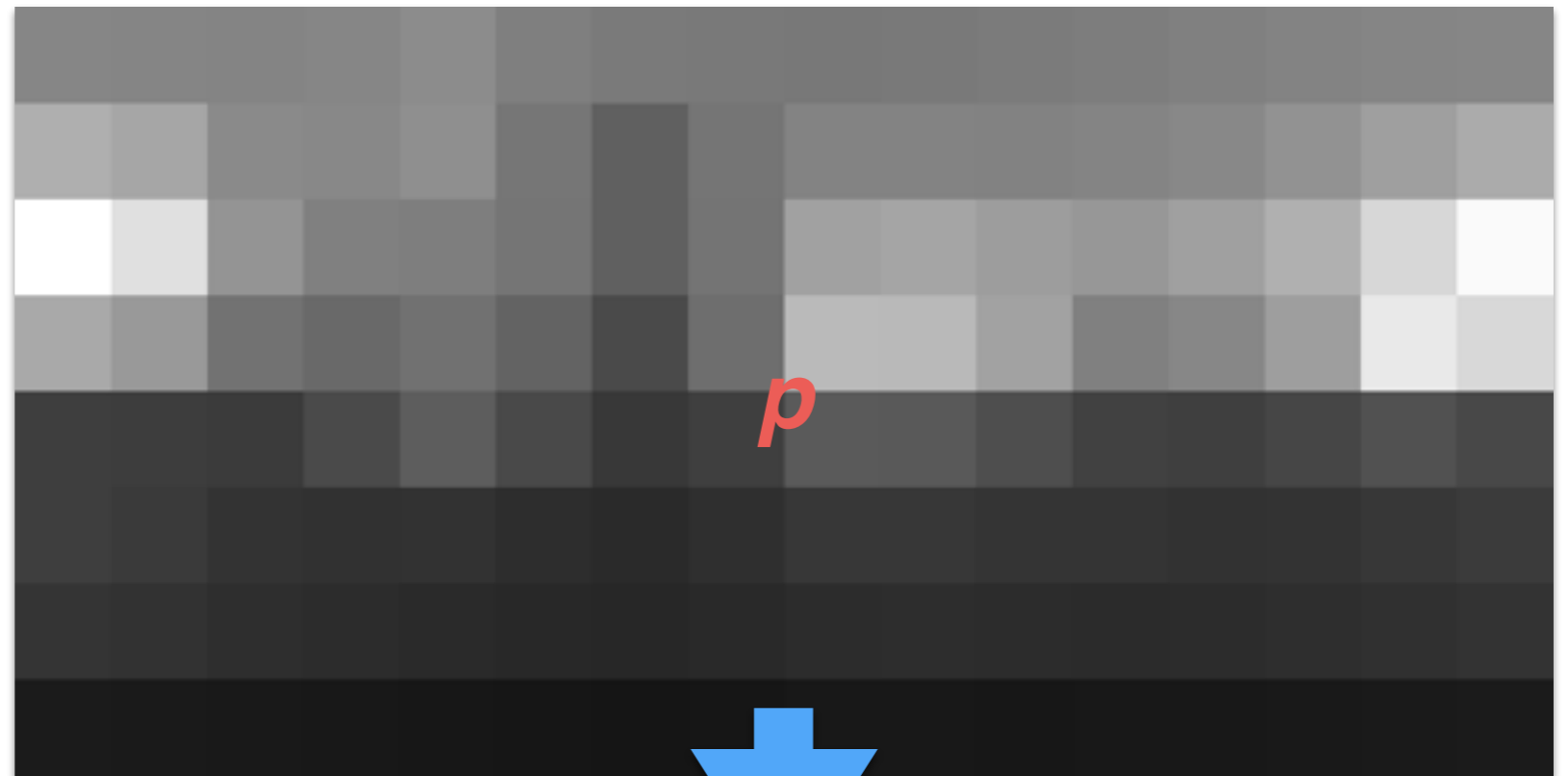
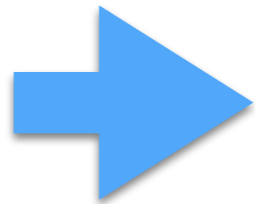
IBL: integrazione montecarlo



IBL: integrazione montecarlo

- Algoritmo di importance sampling in 2D:
 - Calcolo della PDF, p , della luminanza, L , dell'immagine: dividere L per la somma dei suoi valori
 - Calcolo di una PDF, p_{cols} , delle colonne:
 - creazione di un'array 1D; il valore di ogni indice è la somma dei valori della colonna
 - divisione dei valori dell'array per la somma dei valori dell'array
 - Calcolo della CDF, P_{cols}

IBL: integrazione montecarlo



IBL: integrazione montecarlo

- Per ogni colonna si calcola una PDF, p_c :
 - creazione di un'array 1D; copia della colonna L_c
 - divisione dei valori dell'array per la somma dei valori dell'array
- Calcolo per ogni p_c della CDF, P_c

IBL: integrazione montecarlo

- Algoritmo di importance sampling 2D:
 - vengono generati due numeri casuali; (e_1, e_2)
 - Calcolo di $u = P_{cols}^{-1}(e_1) \rightarrow$ determino quale colonna
 - Calcolo di $v = P_u^{-1}(e_2) \rightarrow$ determino quale riga
 - Il campione è quindi $\rightarrow X_i = (u, v)$
 - La PDF di X_i è $p(X_i) = p_{cols}(u) p_u(v)$

IBL: integrazione montecarlo



1024 campioni

IBL: approccio continuo



16 samples



128 samples

IBL: approccio continuo



16 samples



128 samples

IBL: rendering

- Entrambi gli approcci; continuo e discreto, creano delle immagini di qualità. Che approccio scegliere?
- Tipicamente si ha che:
 - noise free velocemente \rightarrow approccio discreto
 - bias free velocemente \rightarrow approccio continuo

IBL: differential rendering

- Approccio per inserire oggetti virtuali in fotografie:
 - Fotografia calibrata; conoscenza della telecamera
 - Cattura di una fotografia panoramica della scena

IBL: differential rendering

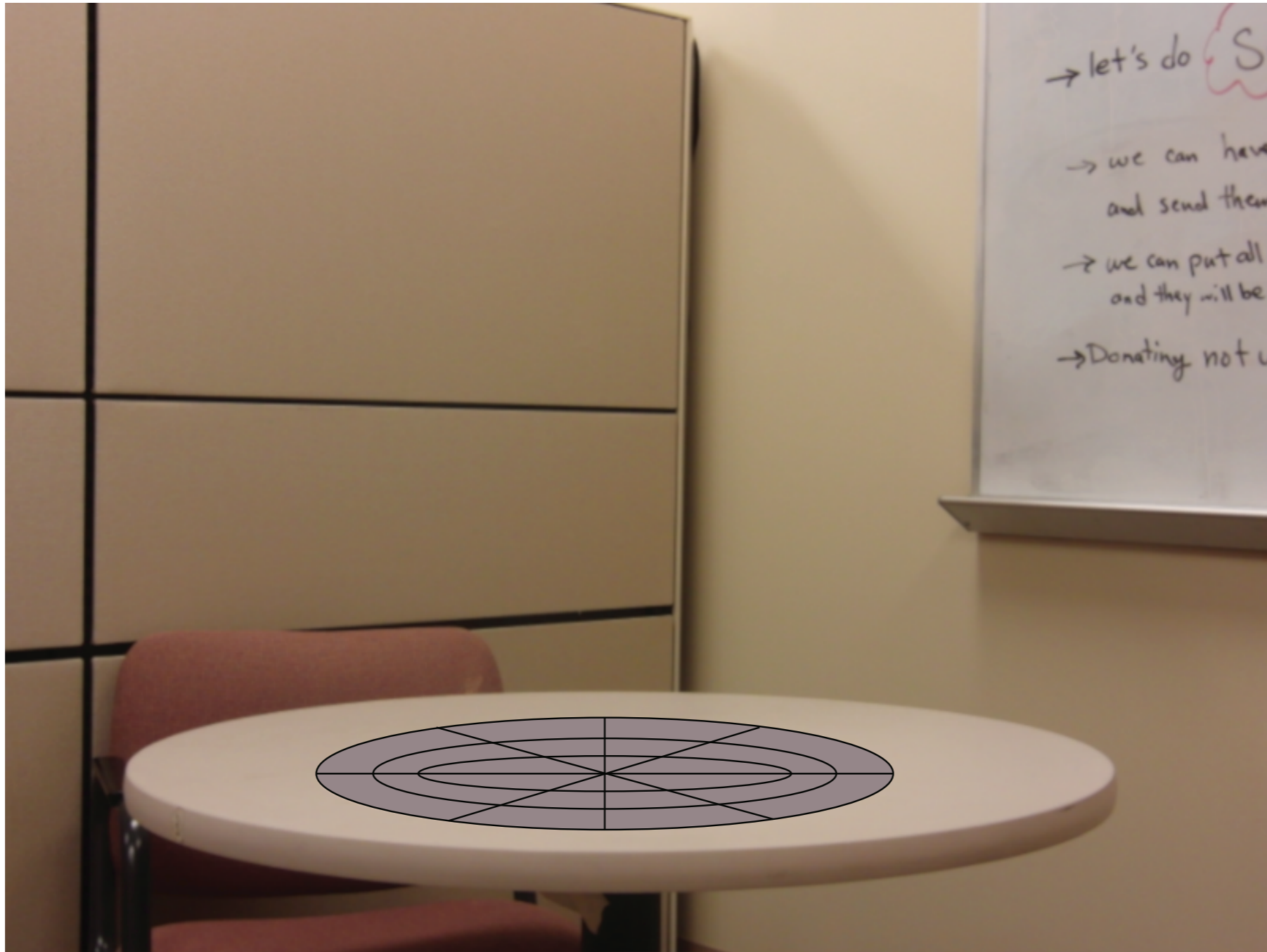


Fotografia di inserimento



Panorama catturato al centro del tavolo

IBL: differential rendering



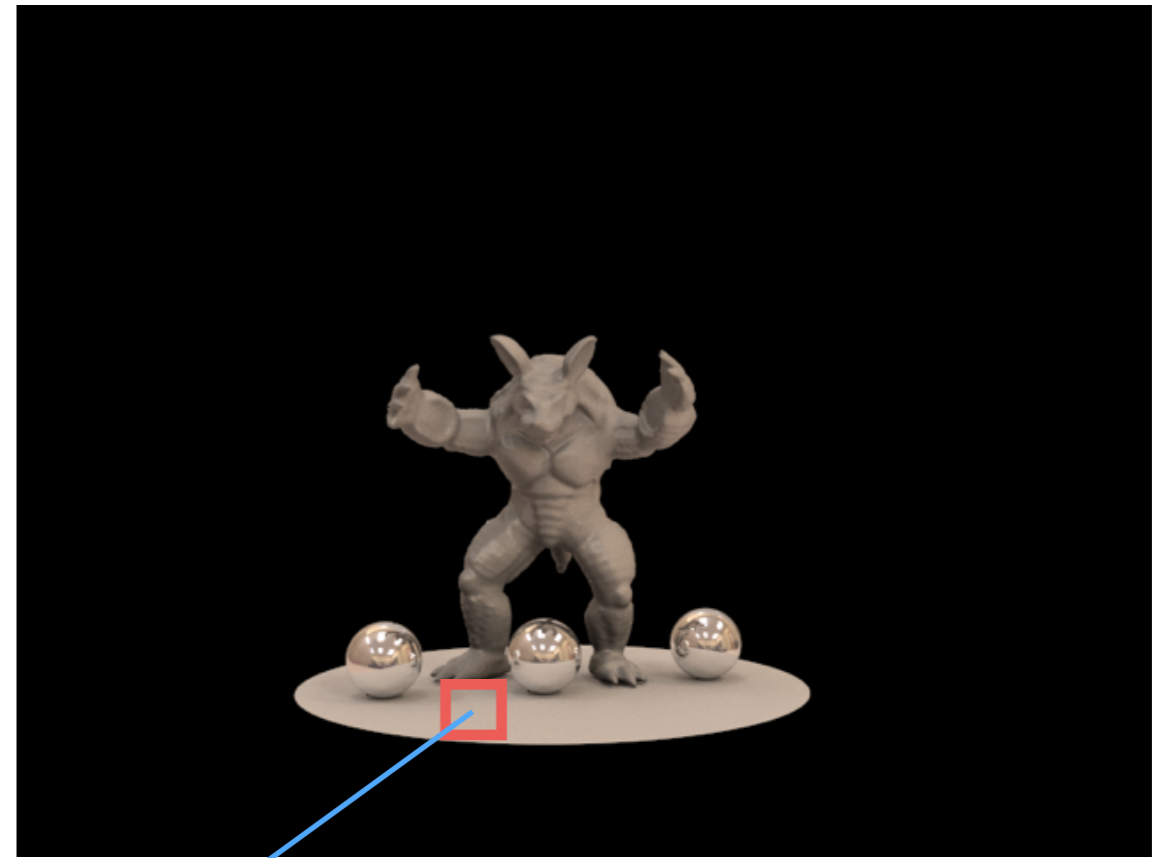
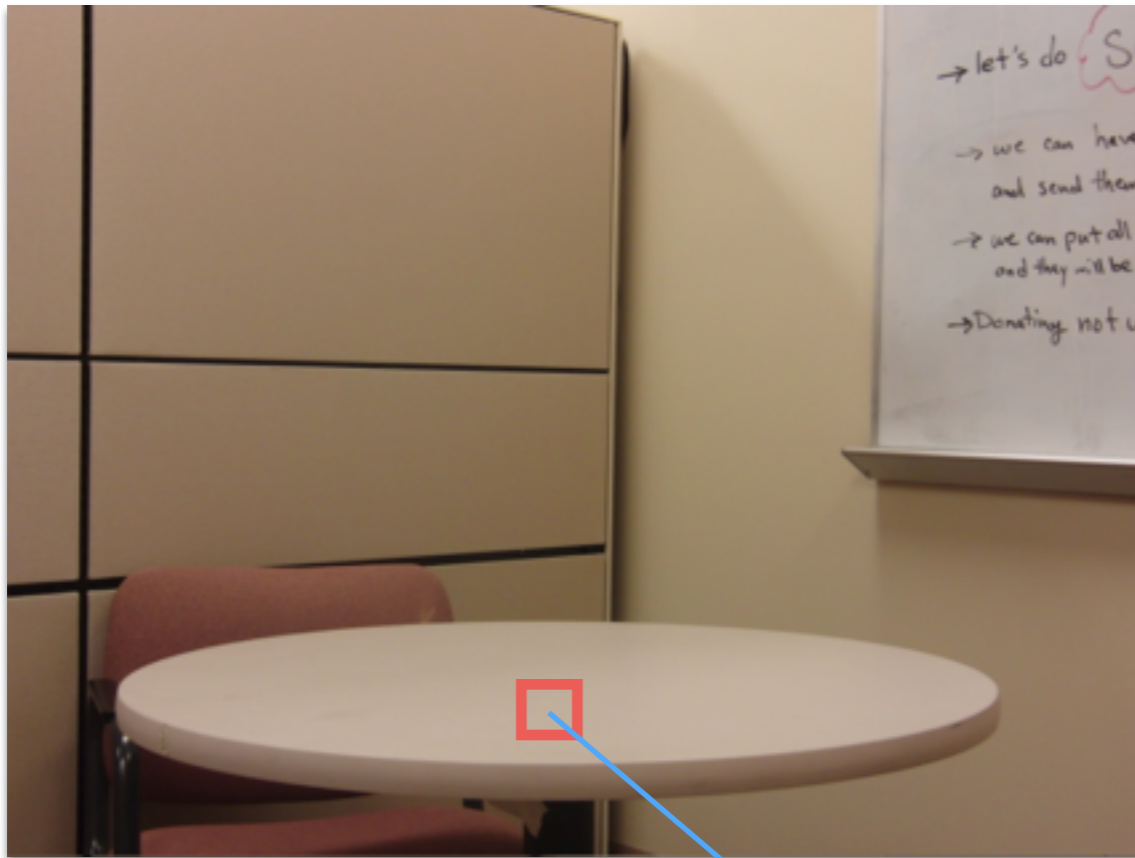
Calibrazione

IBL: differential rendering



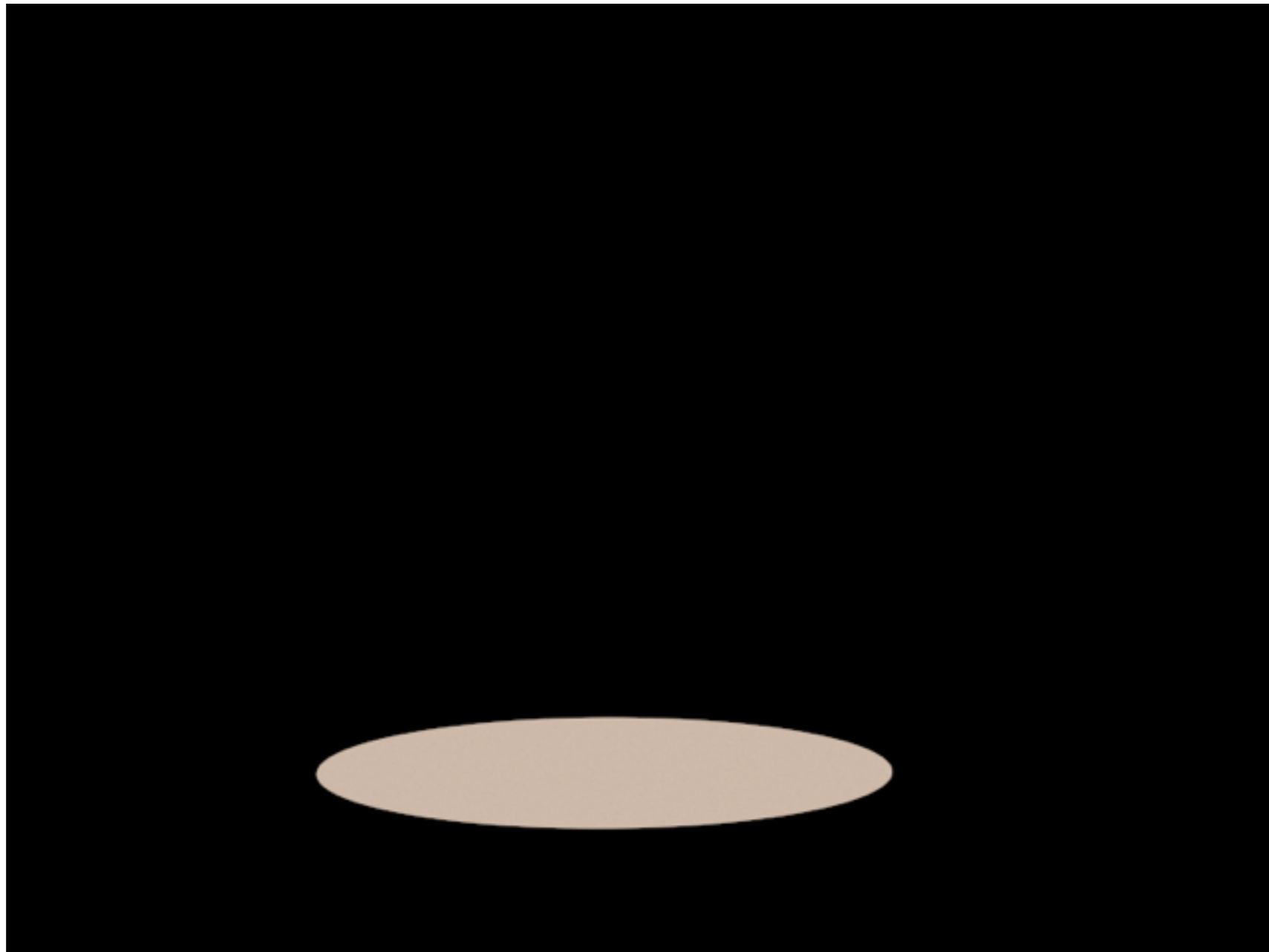
Rendering degli oggetti virtuali con superficie d'appoggio

IBL: differential rendering



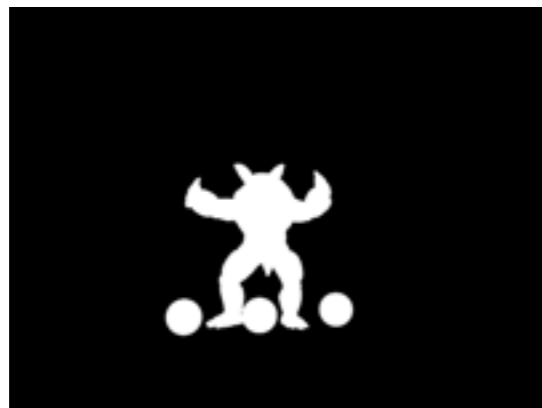
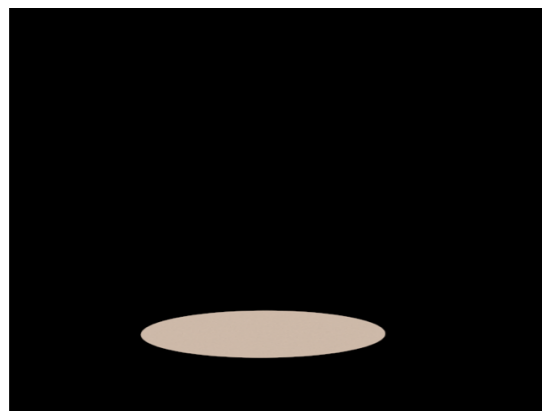
Diversi!!!

IBL: differential rendering



Superficie d'appoggio

IBL: differential rendering



IBL: differential rendering



Rendering finale

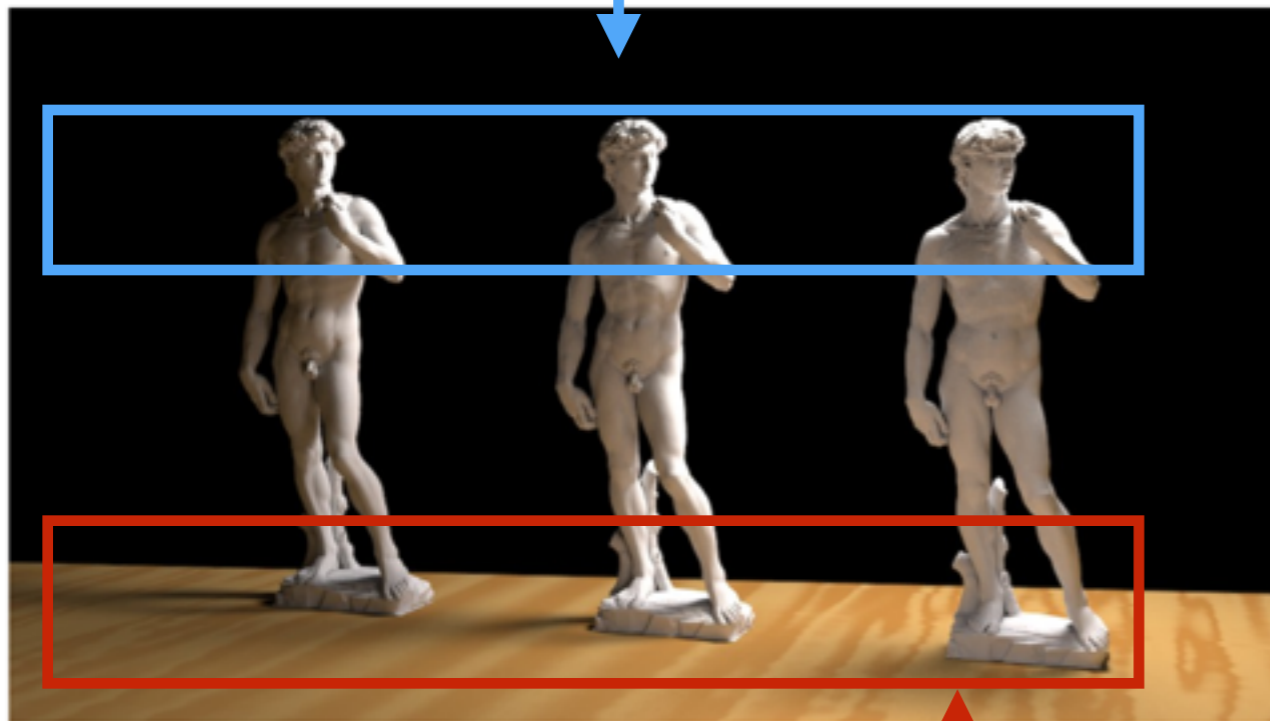
Spatial IBL

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

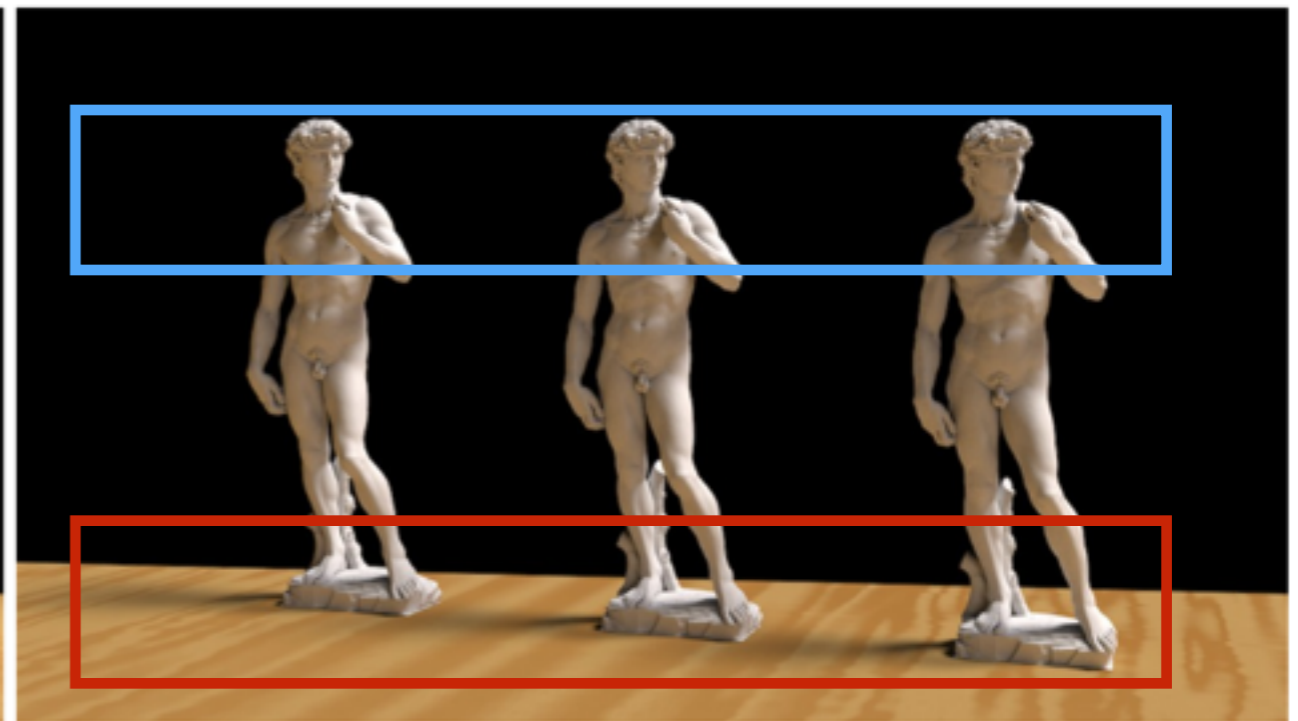
Più immagini panoramiche per catturare le variazioni spaziali!

Spatial IBL

Variazione dell'illuminazione



Spatial IBL



IBL

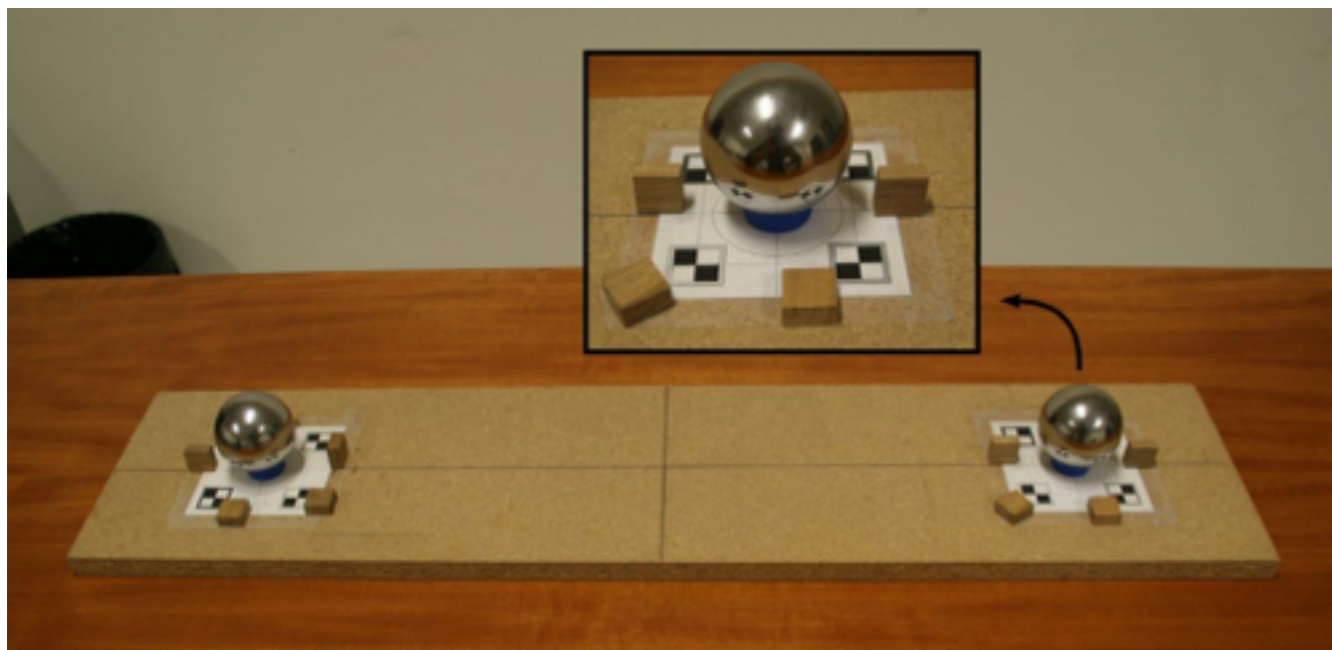
Variazione delle ombre



Spatial IBL: stereo lightprobe

- **Idea:**
 - Catturare due environment map
 - Calibrare le environment map; posizione spaziale (x,y,z) di cattura
 - Triangolare i punti

SIBL: stereo lightprobe



Setup

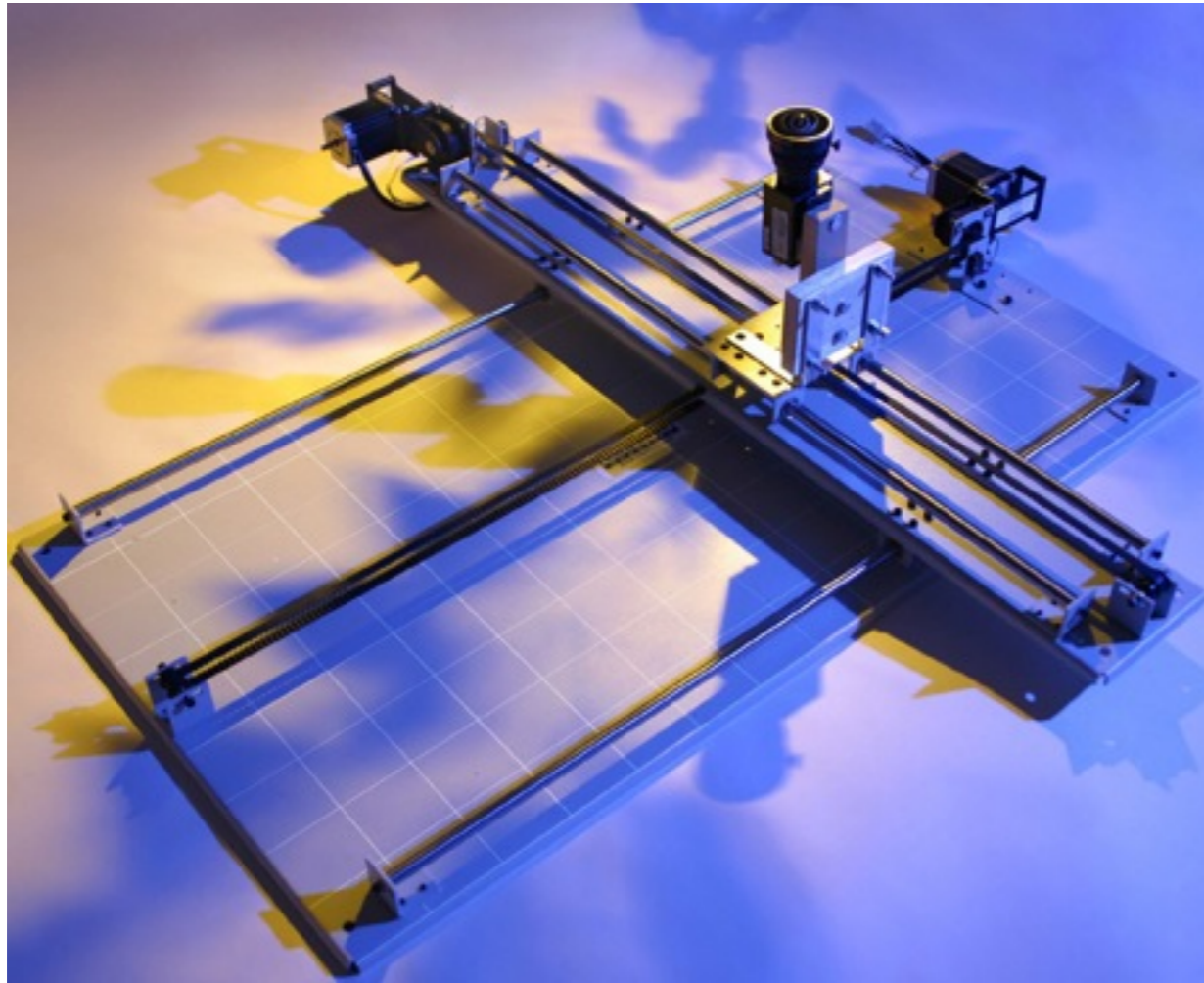


Rendering

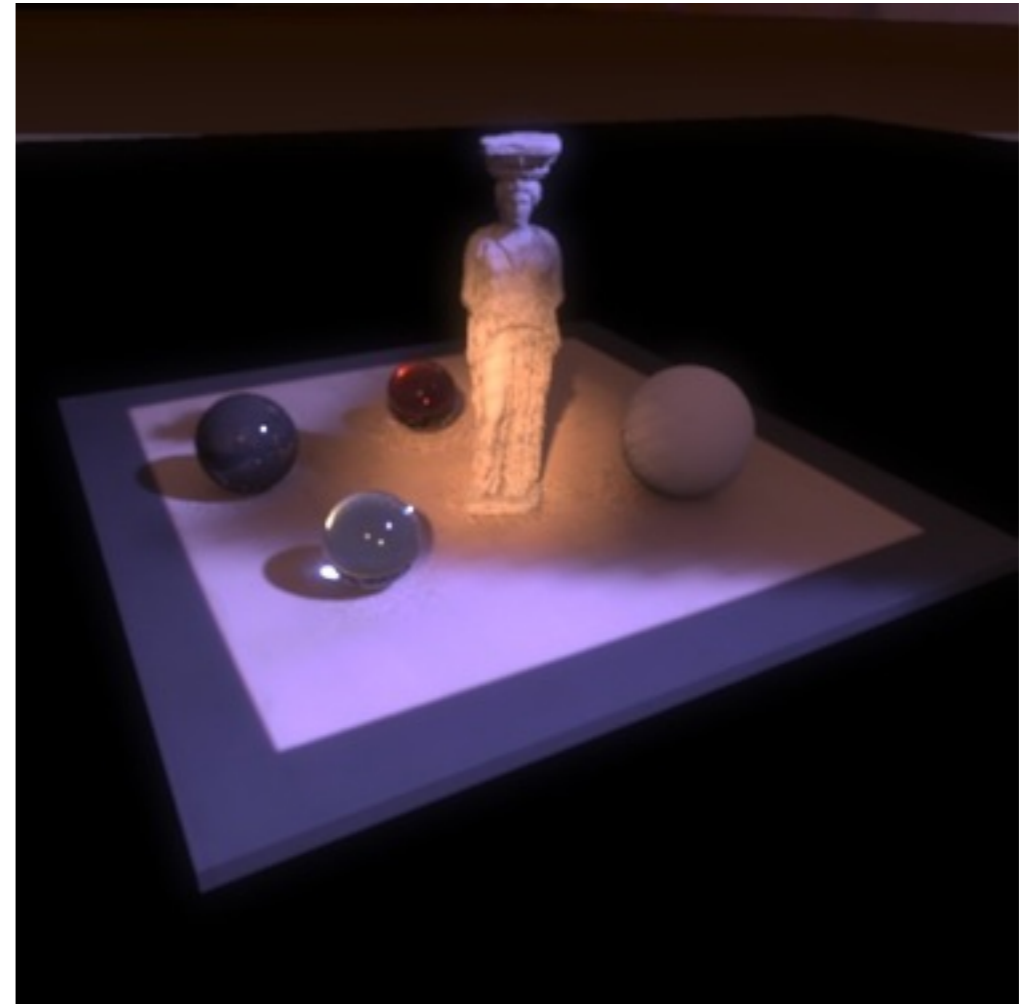
Spatial IBL: incident lightfield

- **Idea:**
 - Catturare più environment map
 - Calibrare le environment map; posizione spaziale (x,y,z) di cattura
 - Triangolare i punti

Spatial IBL: incident lightfield



Setup



Rendering