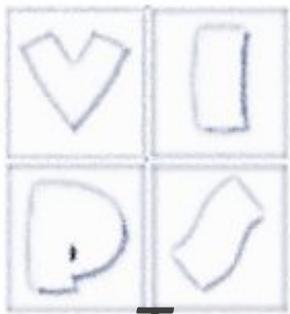


Grafica al calcolatore - Computer Graphics

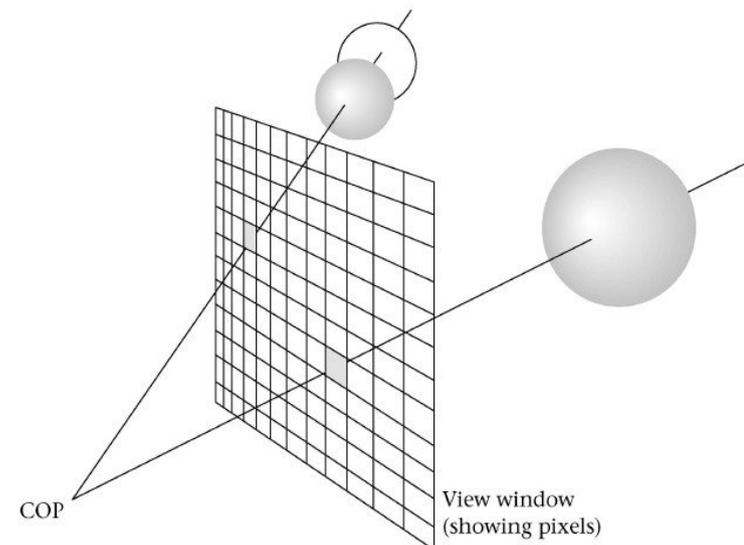
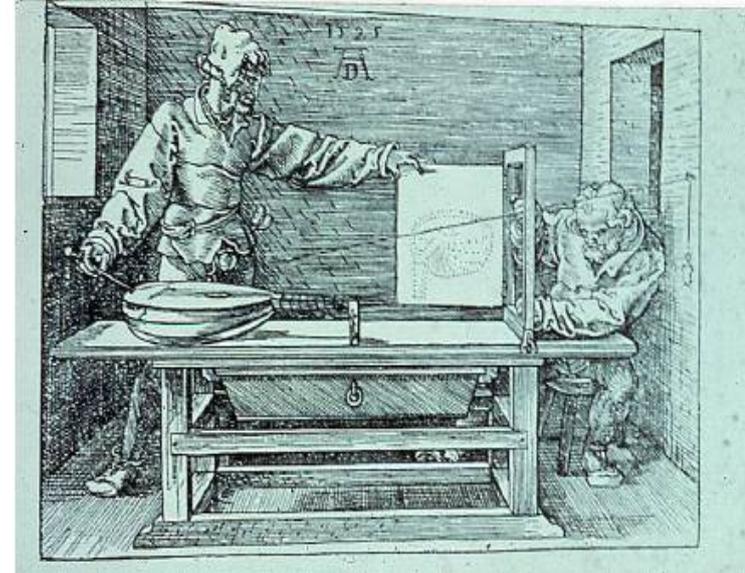


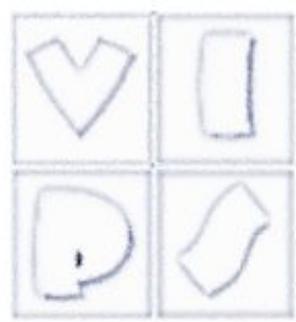
6 – Modelli di illuminazione

Ray casting e colore



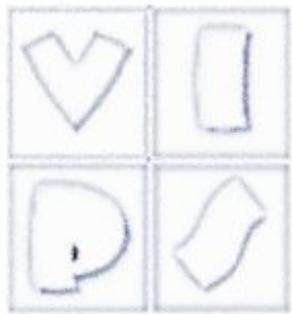
- Tornando al problema della formazione dell'immagine, possiamo creare le immagini se sappiamo assegnare il colore dipendente dall'illuminazione dei punti incontrati dal raggio (shading)
- Abbiamo visto che è complicato risolvere la simulazione fisica
- Ma possiamo fare approssimazioni anche drastiche che forniscono soluzioni per certi versi accettabili





Modelli di illuminazione approssimati

- La cosa è più semplice usando un modello di illuminazione locale, che tratta l'interazione tra una sorgente ed una singola superficie, buttando via del tutto la ricorsione
- Nei modelli locali ciascun punto viene trattato indipendentemente dal resto della scena (no interriflessioni, no ombre, no riflessioni speculari)
- Il modello che vedremo (Phong) è semplice, abbastanza veloce da funzionare in tempo reale e produce risultati accettabili, per scene semplici.
- Complicheremo poi il ray casting mediante il ray tracing, un metodo globale che tratta le sole riflessioni speculari.



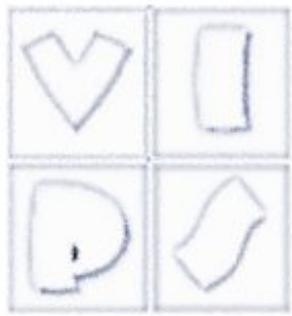
Il modello di Phong

- Le formule che seguono sono in un'unica **banda cromatica**: servono per la produzione di un'immagine a diversi livelli di intensità (toni di grigio) piuttosto che diversi colori
- Quando si utilizza una rappresentazione a colori RGB l'equazione viene calcolata in modo indipendente per ciascuna delle tre componenti cromatiche



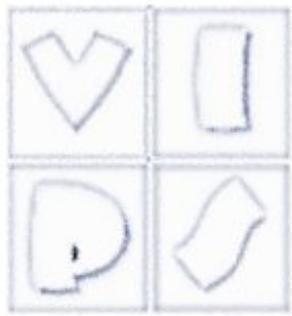
Modello di Phong

- La luce viene considerata composta da tre componenti cromatiche R,G,B. Si calcola l'intensità indipendentemente per ogni componente cromatica, ottenendo un colore.
 - Nel seguito faremo riferimento ad una singola componente.
- Confondiamo "l'intensità luminosa" I con la radianza. La I è il valore che viene assegnato al raggio.
- Sia P il punto della superficie di cui si vuole calcolare il colore. Questo equivale a calcolare l'intensità luminosa I_{out} lungo la direzione v che congiunge P con il COP



Il modello di Phong

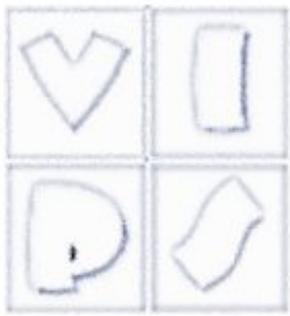
- Modello dovuto a Phong Bui-Tran, prima metà degli anni '70
- Semplifica lo schema fisico di interazione:
 - Solo sorgenti puntiformi
 - No inter-riflessioni
 - Calcolo locale dell'equazione di illuminazione
 - Approssimazione con due costanti della funzione di riflessione



Il modello di Phong

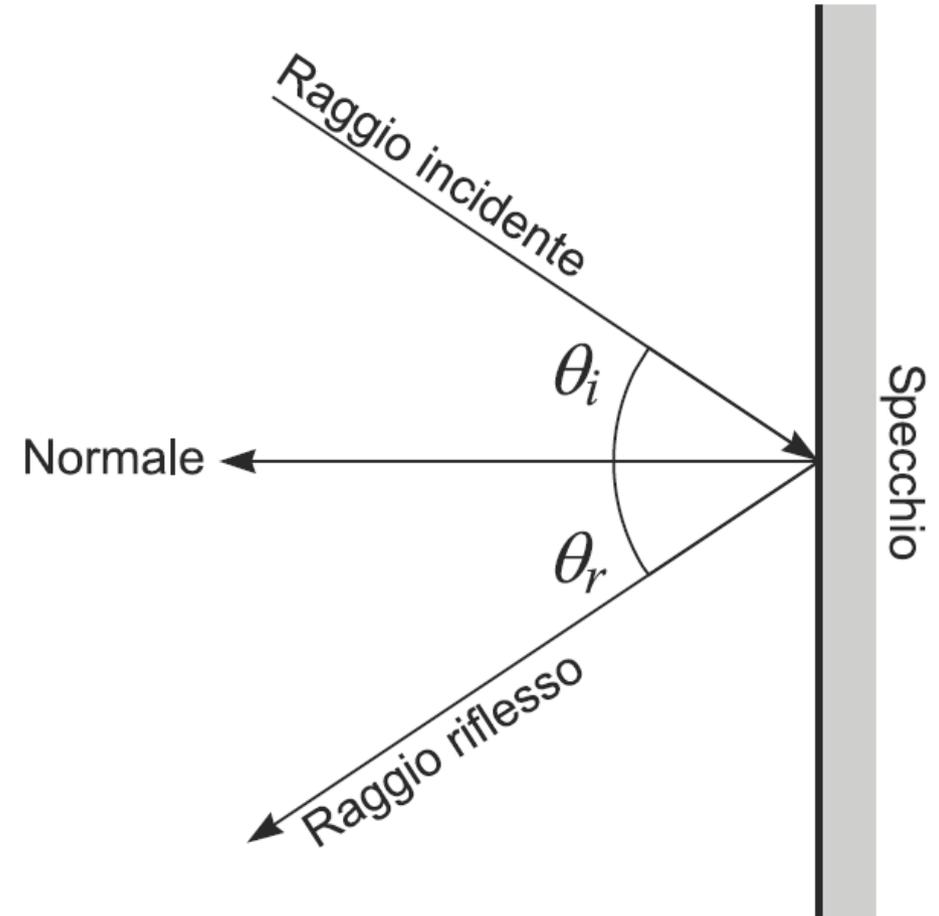
- Simula il comportamento di materiali opachi
- Non modella la rifrazione: no materiali trasparenti o semi-trasparenti
- Non si considerano le ombre

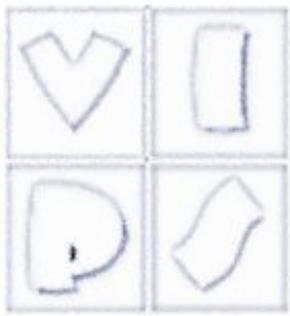
Metodo: semplificazione del fenomeno della riflessione usando le leggi della fisica che regolano la riflessione speculare (Fresnel) e la riflessione diffusa (Lambert)



Legge di Fresnel

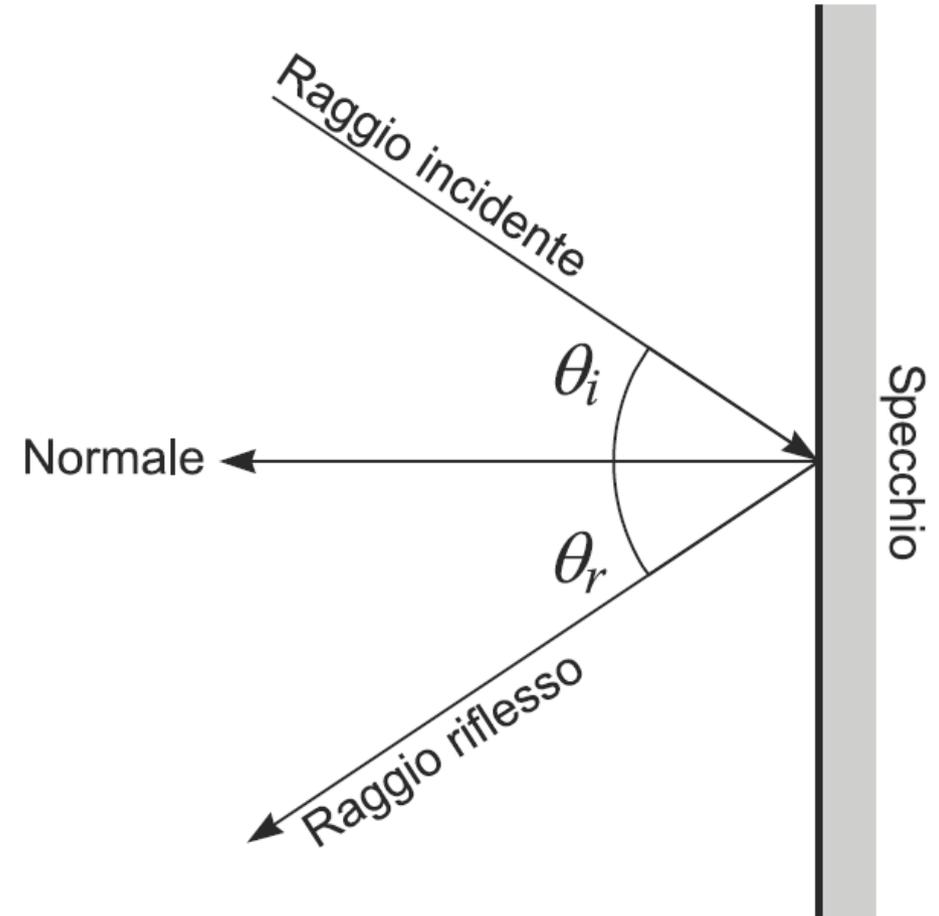
- Quando un raggio di luce passa da un mezzo ad un altro con diverso indice di rifrazione raggiunta la superficie di separazione parte del raggio viene riflessa e parte trasmessa
- La somma delle energie dei due raggi è uguale all'energia del raggio originale

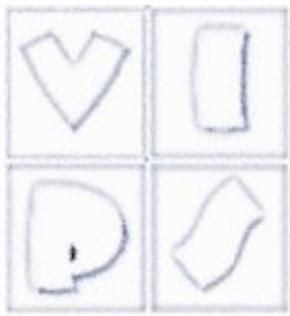




Legge di Fresnel

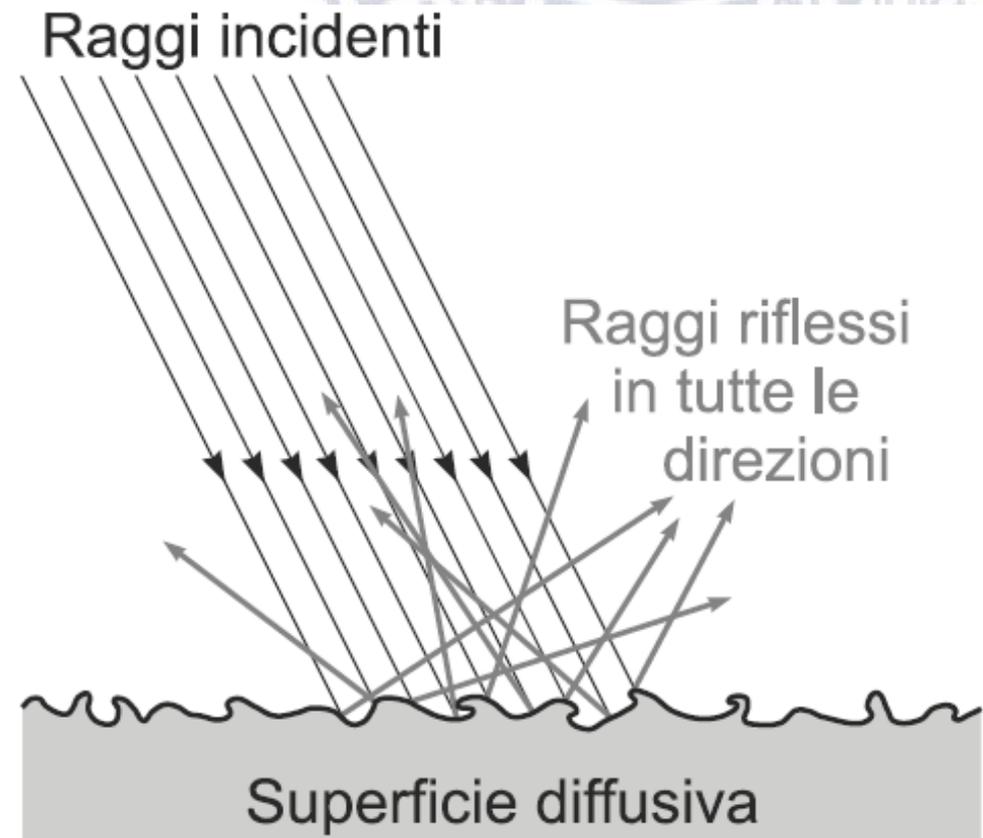
- Se da aria a corpo solido non c'è rifrazione si ha solo riflessione
- L'angolo di incidenza è uguale all'angolo di riflessione
- Vale per materiali molto lisci e lucidi

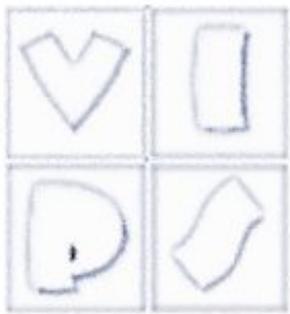




Legge di Lambert

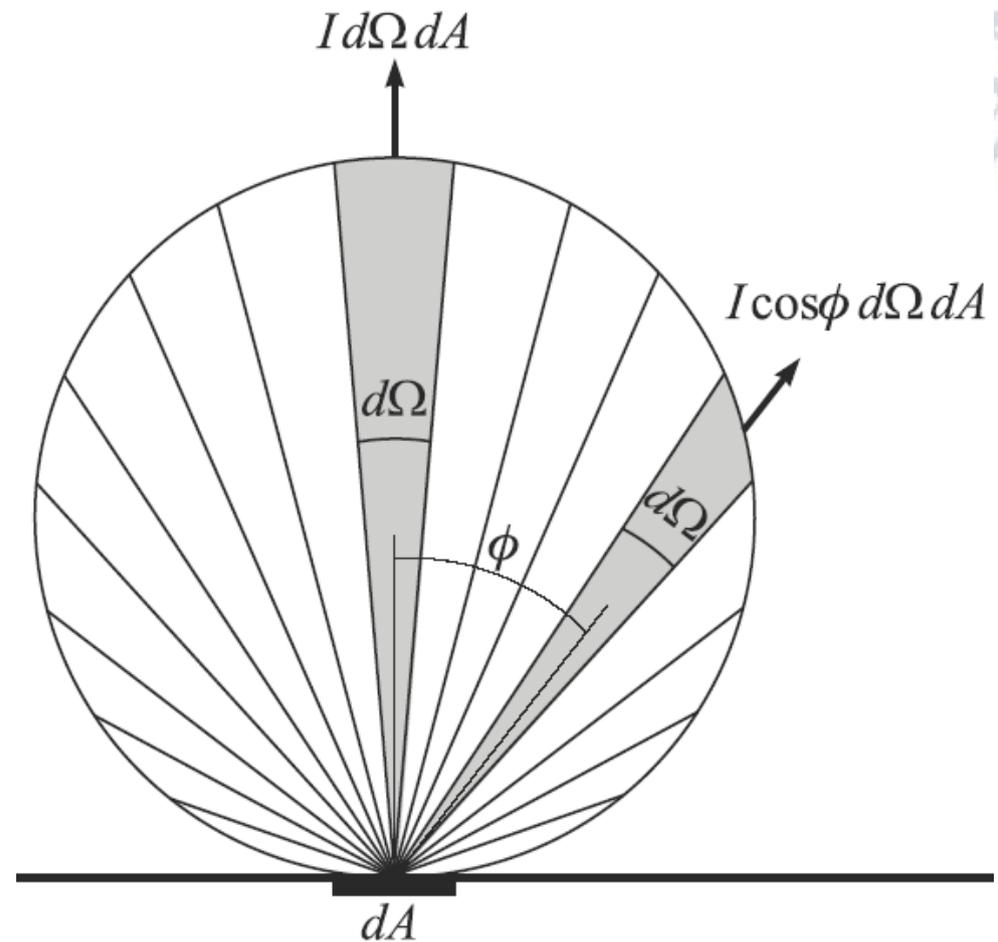
- Materiali molto opachi (es. gesso e legno) hanno una superficie che, a livello microscopico, ha piccole sfaccettature che riflettono la luce in una direzione casuale





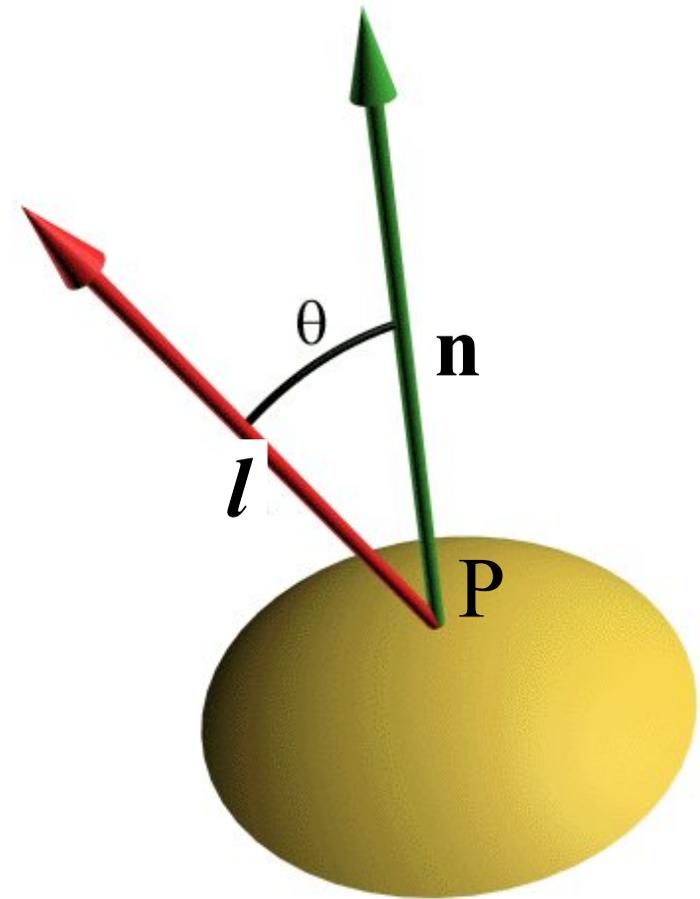
Legge di Lambert

- Integrando su scala macroscopica: la luce si riflette uniformemente verso tutte le direzioni, con intensità proporzionale al rapporto tra la direzione del raggio incidente e la normale alla superficie in quel punto

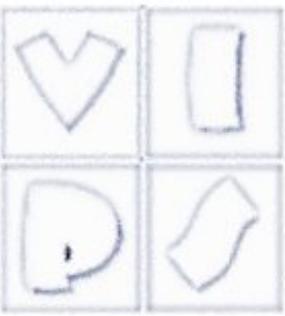


Modellazione della riflessione diffusiva

- Sorgenti luminose puntiformi:
 - posizione nella scena
 - intensità della luce emessa
- Dipendenza solo dall'angolo tra la direzione della luce vista da P , ovvero l e la normale in P , che indichiamo con n e supponiamo di norma unitaria.
 - $\cos(\Theta) = l \cdot n$

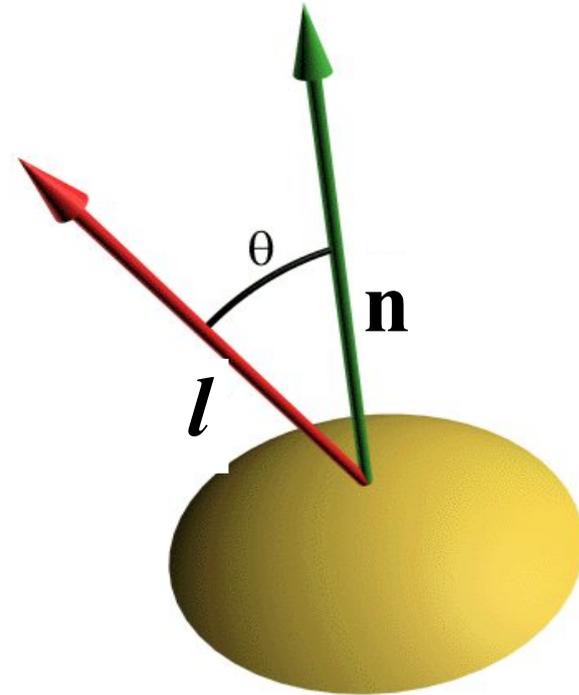


Componente diffusiva



- Si approssima la funzione di riflessione diffusa della superficie come una costante k_d dipendente dal materiale
- Equazione di illuminazione (solo diffusiva)

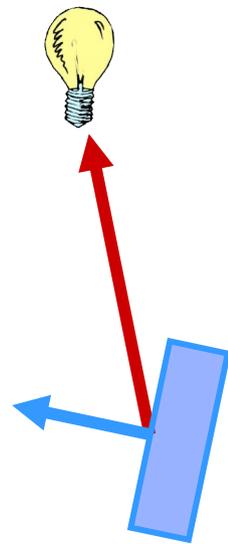
$$I_d^{out} = I k_d \cos \theta = I k_d (\mathbf{n} \cdot \mathbf{l})$$



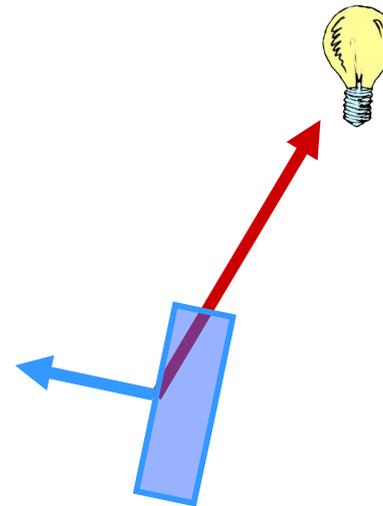
- Simula il comportamento di alcuni materiali (per esempio il gesso, o il coccio) i quali riflettono la luce che ricevono in maniera uniforme in tutte le direzioni

Componente diffusiva

Si considera solo per valori di θ compresi tra 0 e $\pi/2$

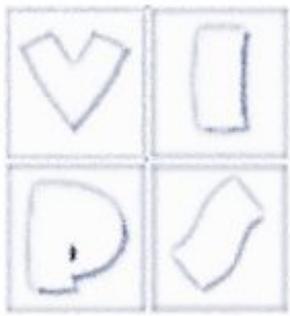


OK

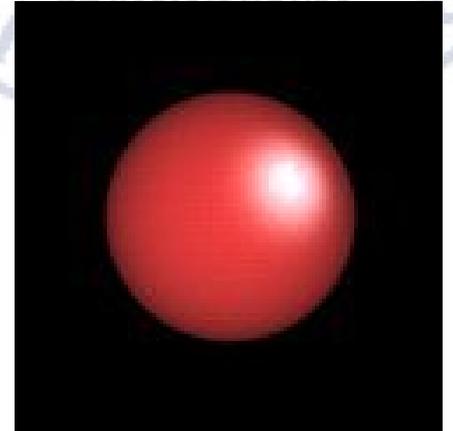
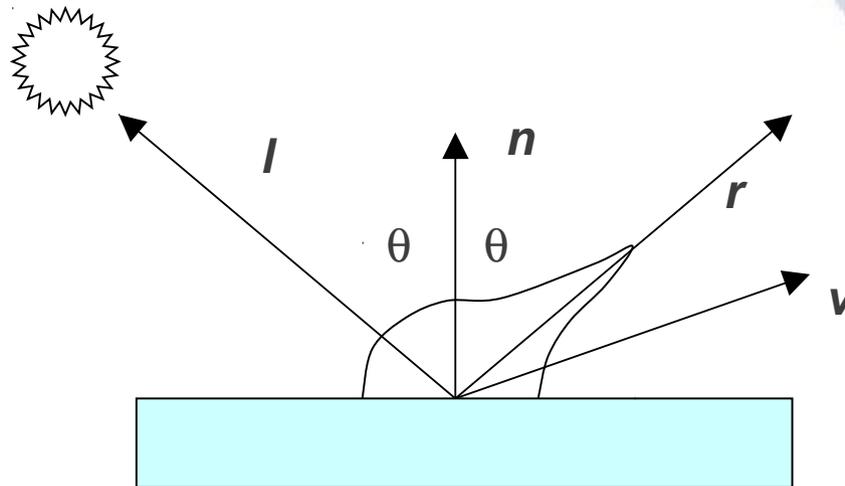
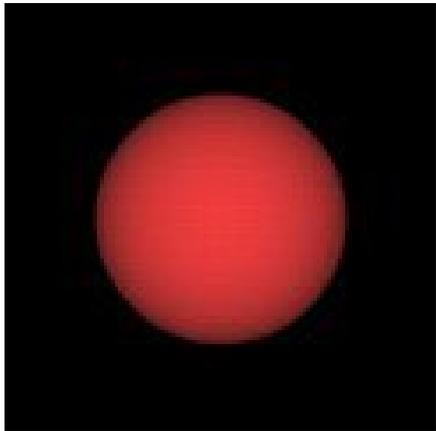


NO

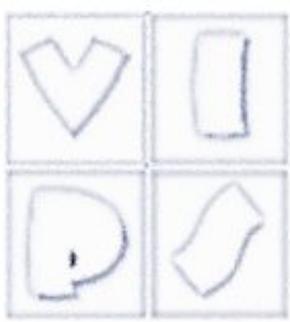
Riflessione speculare



- Novità sostanziale: **riflettore non perfetto**

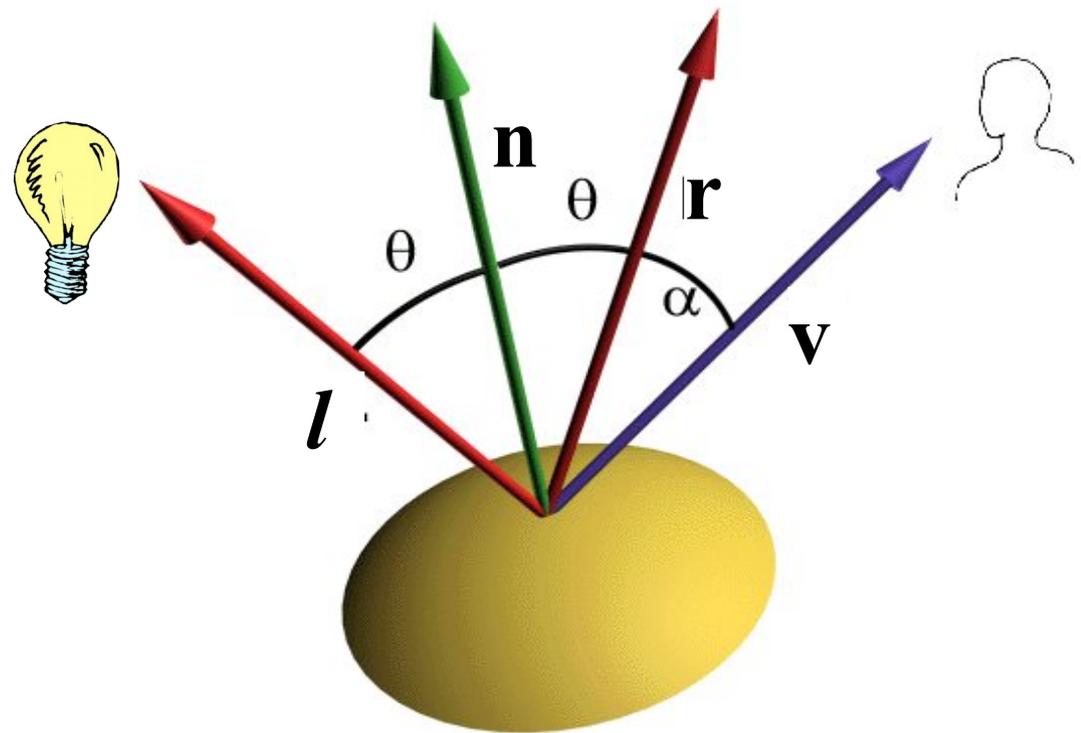


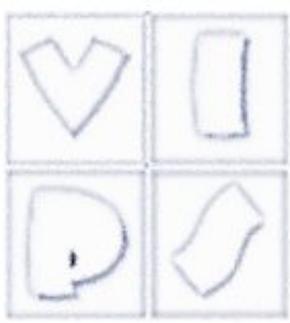
- Approssimazione empirica di una riflessione più realistica rispetto alla legge di Fresnel
- Conseguenza: **specular highlight**. Simula superfici lucide in generale



Modellazione della riflessione speculare

- Dipendenza dall'angolo α compreso tra la direzione di riflessione ideale e la direzione di vista
- Riflessione massima per $\alpha = 0$
- Decadimento più o meno rapido all'aumentare di α





Modellazione della riflessione speculare

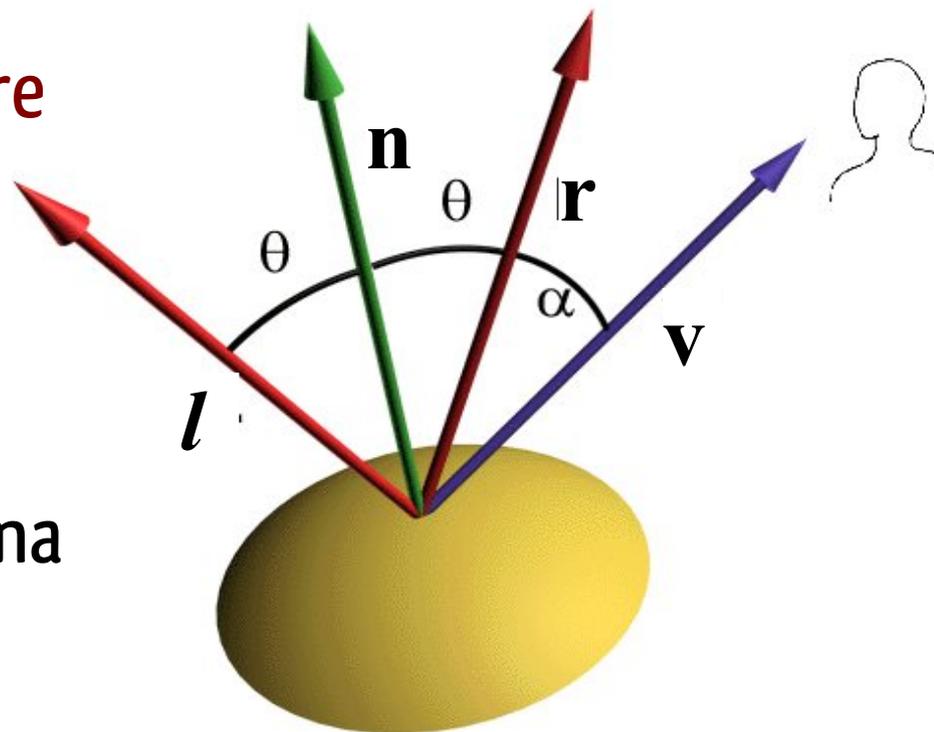
- Phong (1975) introduce il seguente modello empirico per la componente speculare

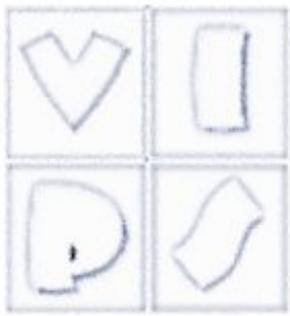
$$I_s^{out} = I k_s (\mathbf{r} \cdot \mathbf{v})^n$$

- ove k_s è il **coefficiente di riflessione speculare** e n l'**esponente di riflessione speculare** del materiale.

- n modula la lucidità della superficie.

- Per n che tende ad infinito si ha una riflessione speculare perfetta.



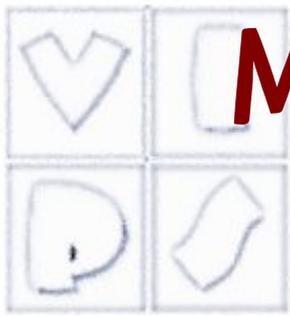


Modellazione della riflessione speculare

- La formulazione di Blinn (1977) è leggermente diversa:

$$I_s^{out} = I k_s (\mathbf{n} \cdot \mathbf{h})^n \quad \text{dove} \quad \mathbf{h} = \frac{\mathbf{l} + \mathbf{v}}{\|\mathbf{l} + \mathbf{v}\|}$$

- \mathbf{h} è detto “halfway vector”, ed è il vettore che biseca l’angolo formato da \mathbf{l} e \mathbf{v} .
- Il vantaggio della formulazione di Phong-Blinn è che gestisce bene il caso in cui l’angolo tra \mathbf{r} e \mathbf{v} sia $> \pi/2$
 - l’angolo tra \mathbf{n} ed \mathbf{h} non è lo stesso che c’è tra \mathbf{r} e \mathbf{v} : lo stesso esponente non produce lo stesso effetto.
- entrambi sono modelli empirici, privi di significato fisico, che hanno il merito di simulare credibilmente superfici lucide.

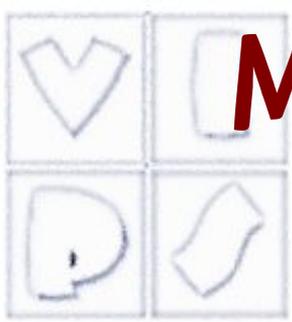


Modellazione della componente ambientale

- Le inter-riflessioni tra oggetti diversi nella scena non sono modellate in modo accurato dal modello di Phong
- Sono approssimate dalla componente:

$$I^{\text{out}} = I_a k_a$$

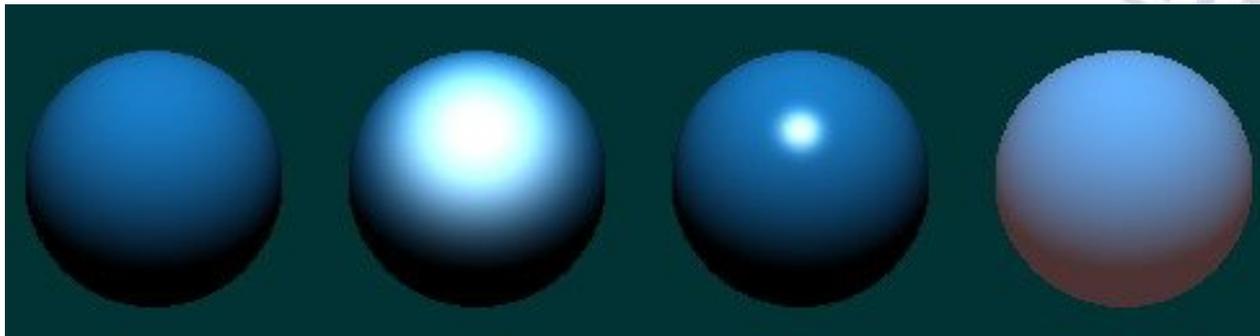
- I_a modella la radiazione luminosa totale emessa nella scena
- k_a modella la riflettività del materiale
- I_a è costante per tutti i punti di tutti gli oggetti



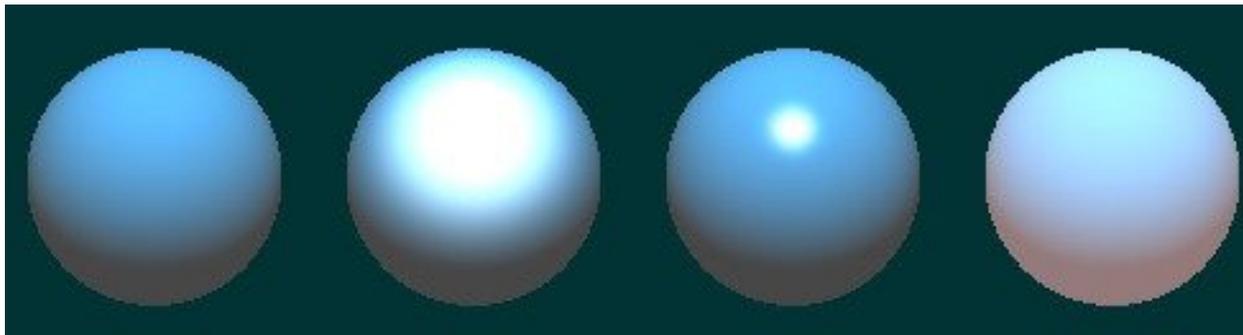
Modellazione della componente ambientale

- La componente ambientale aggiunge realismo alla scena

Senza



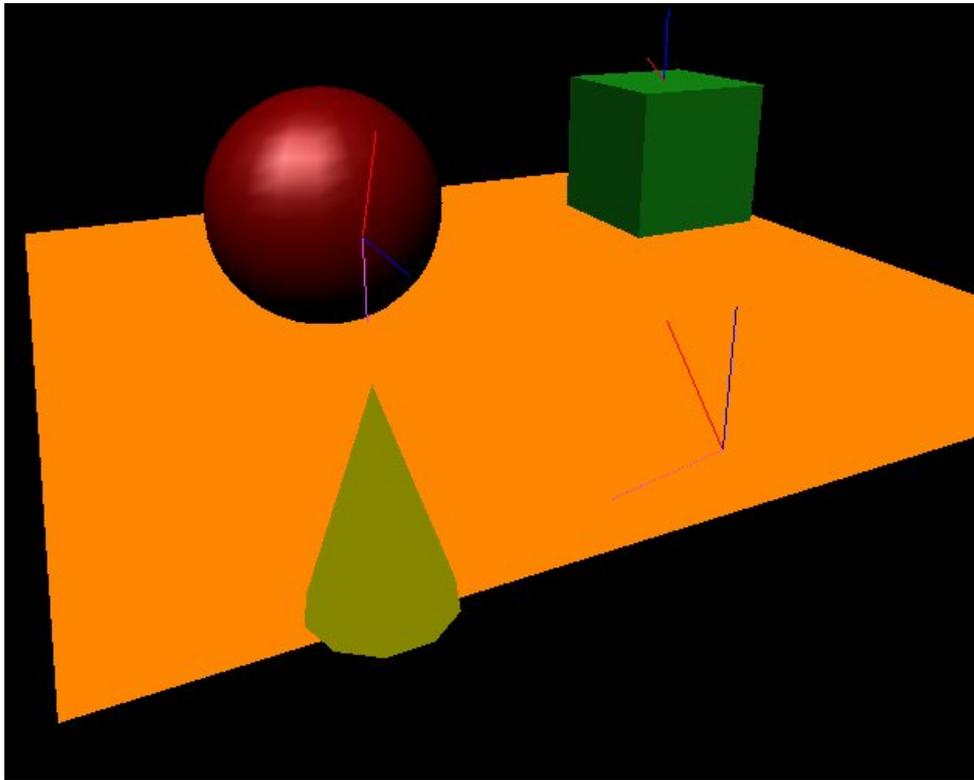
Con



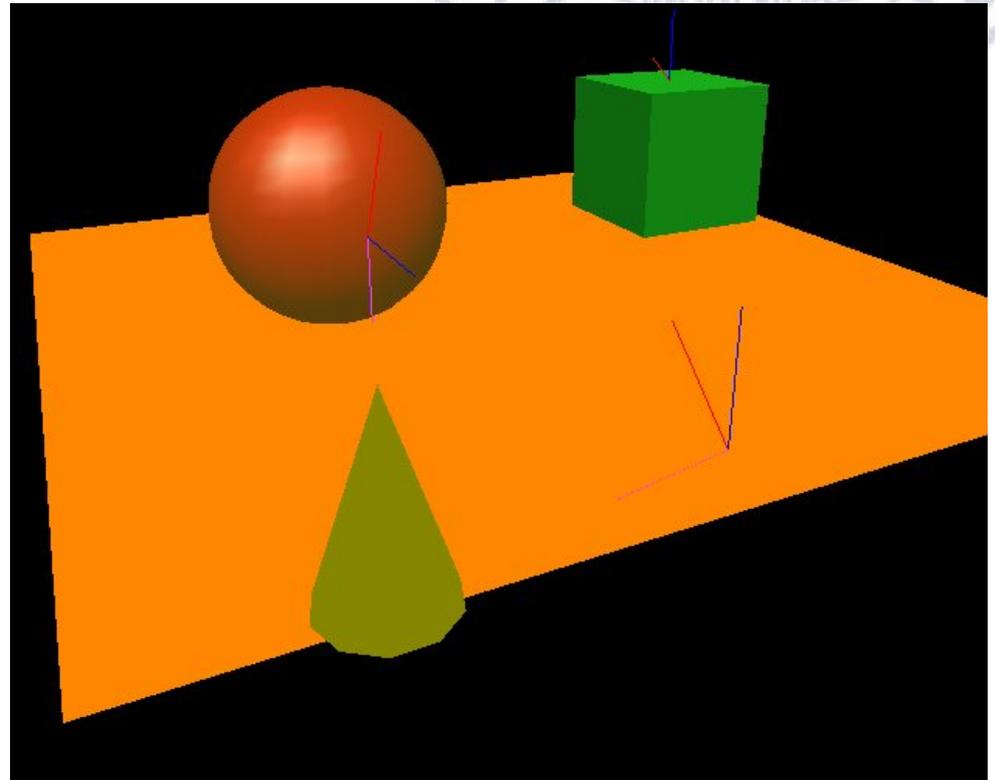


Modellazione della componente ambientale

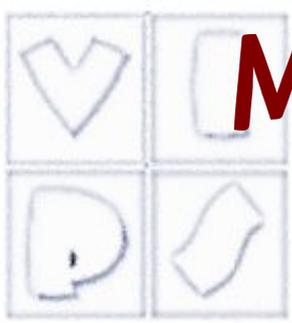
- La componente ambientale aggiunge realismo alla scena



Senza



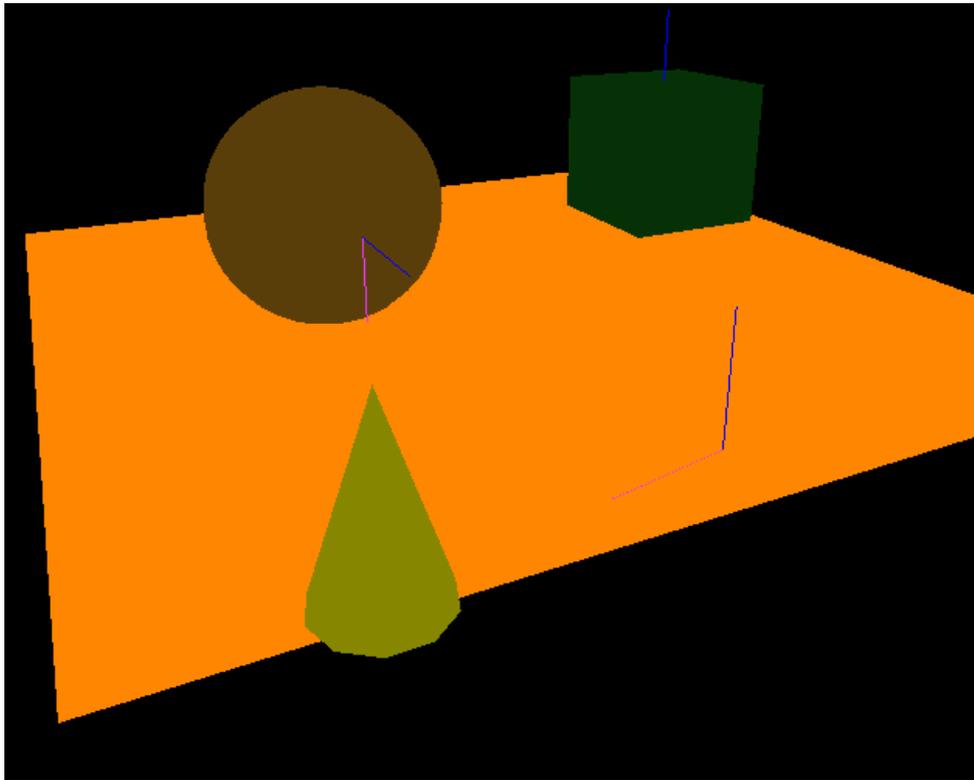
Con



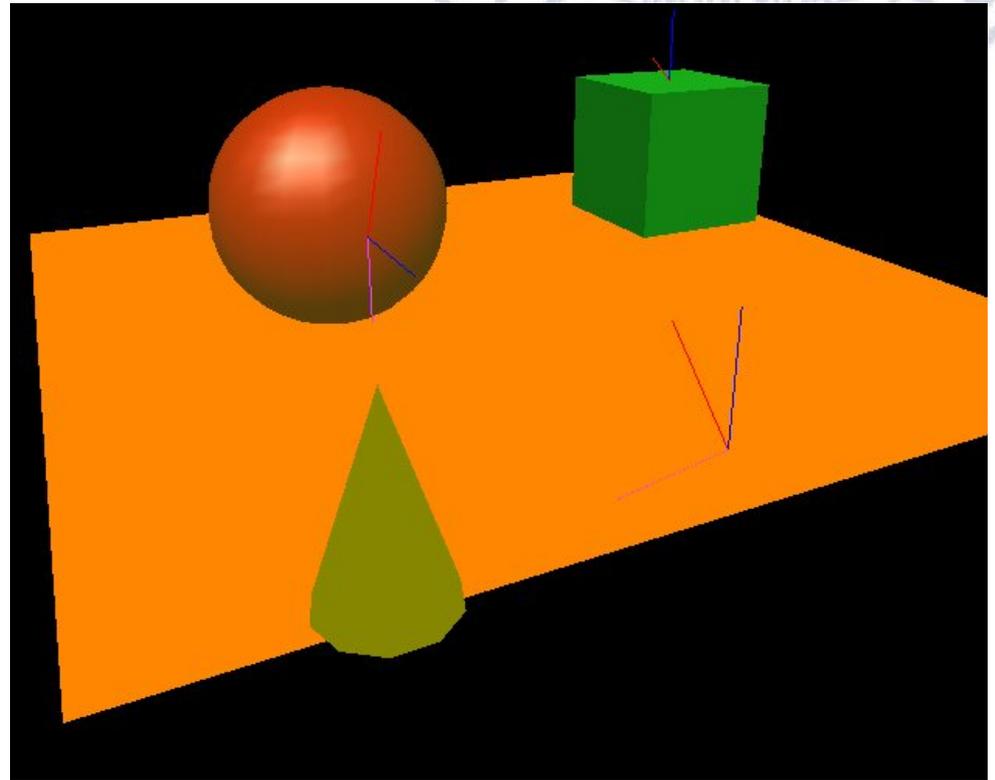
Modellazione della componente ambientale



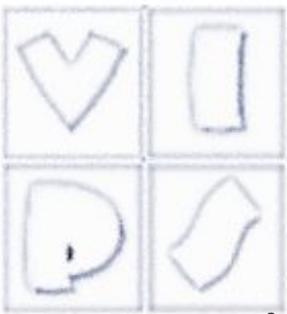
- Ma da sola non basta!



Solo ambientale



Con riflessioni



Mettere tutto insieme

- Tutti i contributi descritti si vanno a sommare per calcolare l'illuminazione

$$I^{out} = I_a k_a + I (k_d(\mathbf{n} \cdot \mathbf{l}) + k_s(\mathbf{n} \cdot \mathbf{h})^n)$$

- Se si stanno considerando i colori, allora sia le intensità della luce che i coefficienti del materiale vanno definiti per ogni componente (r,g,b)

$$I^{r,out} = I_a^r k_a^r + I^r (k_d^r(\mathbf{n} \cdot \mathbf{l}) + k_s^r(\mathbf{n} \cdot \mathbf{h})^n)$$

$$I^{g,out} = I_a^g k_a^g + I^g (k_d^g(\mathbf{n} \cdot \mathbf{l}) + k_s^g(\mathbf{n} \cdot \mathbf{h})^n)$$

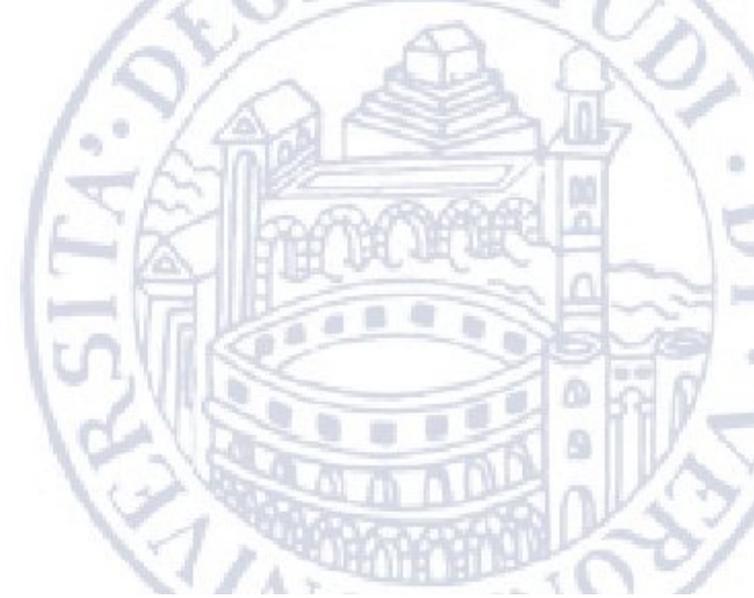
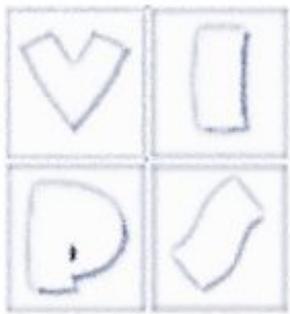
$$I^{b,out} = I_a^b k_a^b + I^b (k_d^b(\mathbf{n} \cdot \mathbf{l}) + k_s^b(\mathbf{n} \cdot \mathbf{h})^n)$$

- I coefficienti di diffusione e ambientali di solito sono uguali
 - La superficie appare del colore specificato dalla terna di coefficienti quando illuminata da luce bianca.
 - Le riflessioni speculari (highlights) invece sono di solito del colore della luce ($k_s^r = k_s^g = k_s^b = 1$)



Note

- Si parla di modello di Phong, anche se, a rigore, il contributo di Phong riguarda il solo termine speculare.
- Nella forma più generale, prevede tre componenti separate: luce speculare I^s (riflessa specularmente dalle superfici), luce diffusa I^d (diffusa dalle superfici, e luce ambientale I^a
- Nella trattazione sopra abbiamo assunto $I^s = I^d = I$.
- Il fatto che una sorgente di luce non emetta luce e basta, ma luce di tre tipi diversi (!?) non ha un significato fisico, ma può servire a rendere il modello più flessibile (per simulare effetti globali). Per esempio, la luce ambientale la potrà in generale avere un colore diverso dalla I , rendendo magari più realistico l'effetto della mutua illuminazione tra le superfici: in una stanza con le pareti rosse, la luce ambientale è rossa
- Inoltre, la luce ambientale è associata ad ogni sorgente e non è un termine globale unico, così quando una luce viene spenta (o accesa) l'illuminazione ambientale diminuisce (o aumenta)



(1) Ambientale

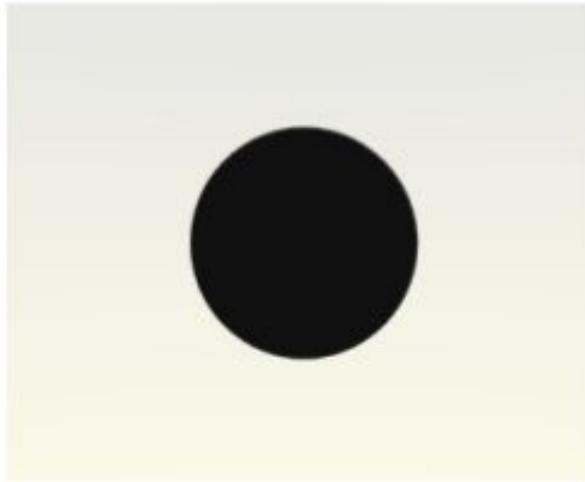
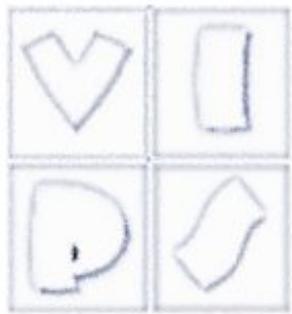


(2) Ambientale + Diffusa

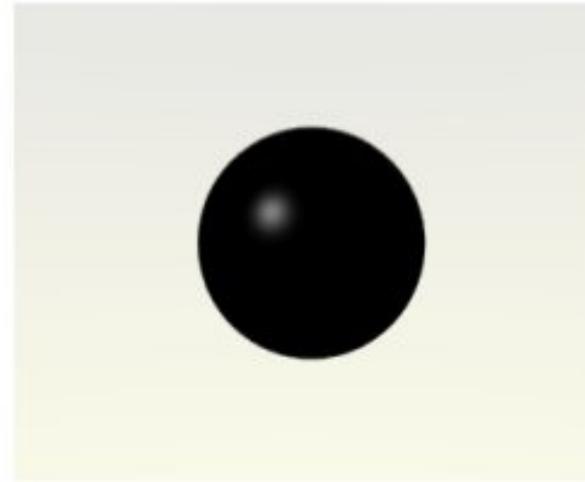


(3) Ambientale + Diffusa + Speculare

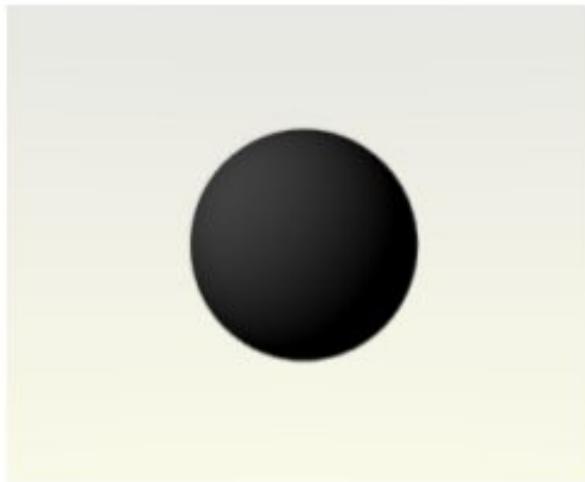
©Alan Watt



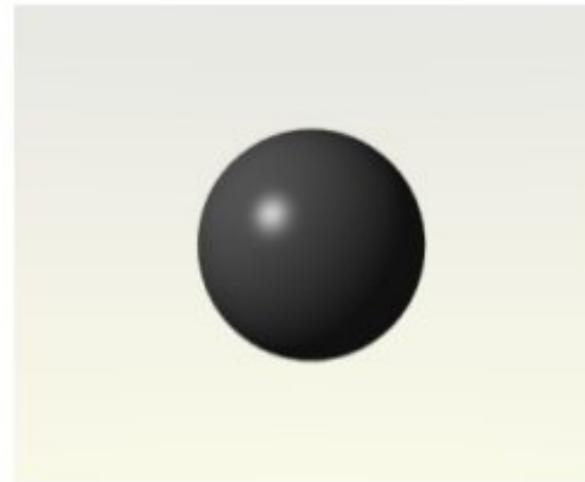
(4) Ambientale



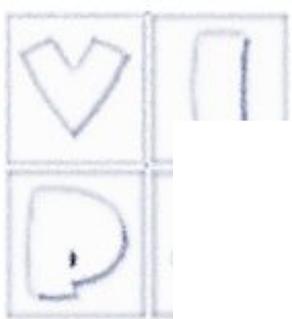
(5) Speculare



(6) Diffusa



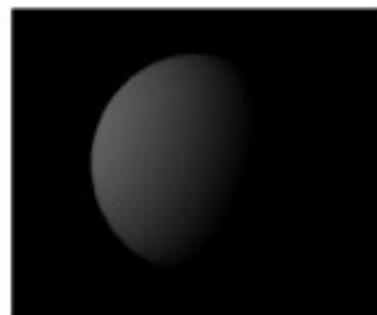
(7) Totale



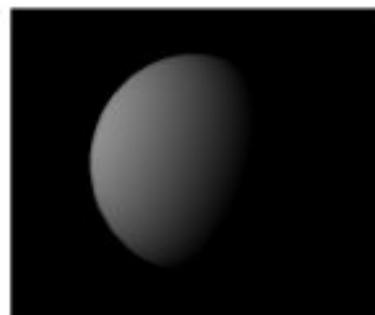
$I_a = 0$



(8) $k_d = 0.3$



(9) $k_d = 0.6$



(10) $k_d = 0.9$

$I_a = 0.1$



(11) $k_d = 0.3$



(12) $k_d = 0.6$



(13) $k_d = 0.9$

$I_a = 0.3$



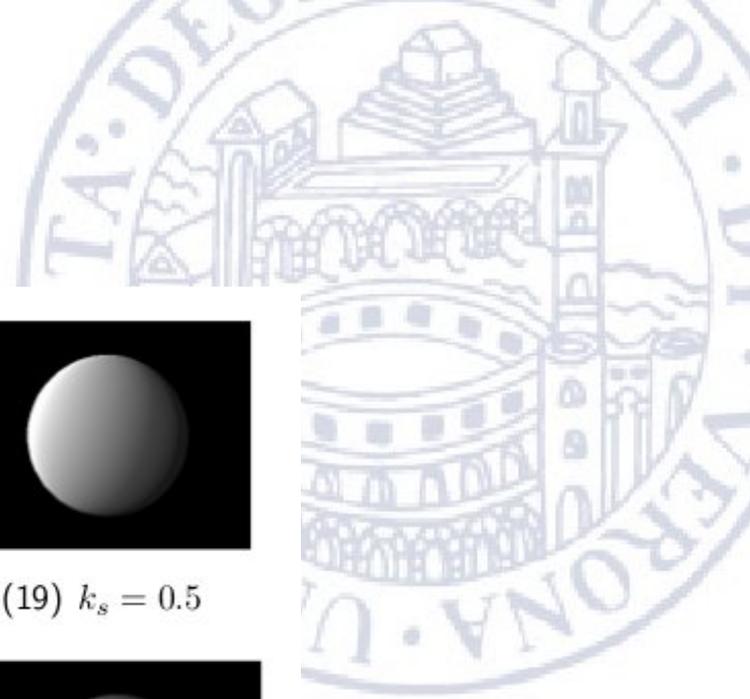
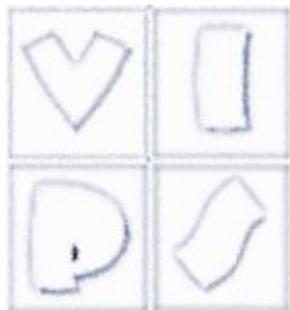
(14) $k_d = 0.3$



(15) $k_d = 0.6$



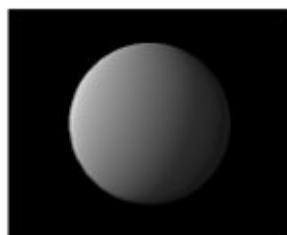
(16) $k_d = 0.9$



$n = 1$



(17) $k_s = 0.1$

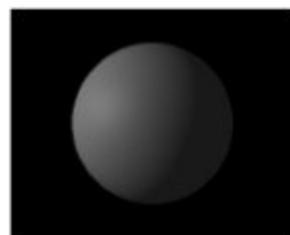


(18) $k_s = 0.25$



(19) $k_s = 0.5$

$n = 20$



(20) $k_s = 0.1$

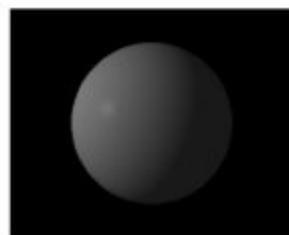


(21) $k_s = 0.25$

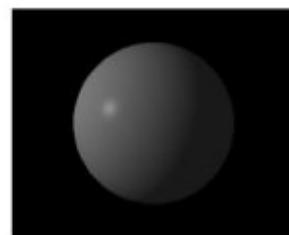


(22) $k_s = 0.5$

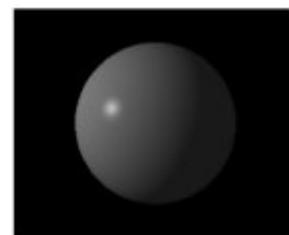
$n = 200$



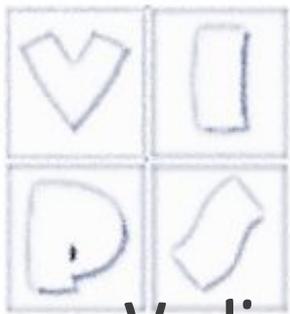
(23) $k_s = 0.1$



(24) $k_s = 0.25$



(25) $k_s = 0.5$



BRDF di Phong

- Vediamo come connettere il modello locale di Phong con l'equazione della radianza, per capire che tipo di approssimazioni sono state fatte.
- Abbiamo introdotto le sorgenti di luce puntuali, che sono distinte dagli oggetti della scena, dunque per tutte le superfici della scena $L_e = 0$.
- Per determinare la “intensità” di un punto si deve calcolarne la radianza uscente nella direzione ω che lo unisce al COP:
- • Dobbiamo quindi calcolare

$$L(x, \omega) = \int_{\Omega} L(x, \omega_i) \rho(x, \omega_i, \omega) (\omega_i \cdot \mathbf{n}) d\omega_i$$



- Ma esiste una sola direzione lungo la quale il contributo all'integrale è diverso da zero: la direzione ω_L che punta alla sorgente luminosa. Otteniamo dunque:

$$L(x, \omega) = L(x, \omega_L) \rho(x, \omega_L, \omega) (\omega_L \cdot \mathbf{n}) d\omega$$

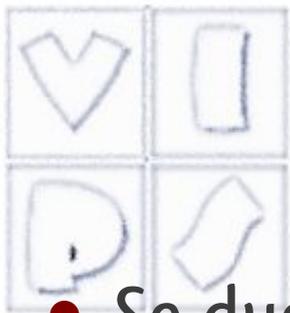
- Confrontando con il modello di Phong (senza il termine ambientale)

$$I^{out} = I \left(k_d + k_s \frac{(\mathbf{n} \cdot \mathbf{h})^n}{(\mathbf{n} \cdot \mathbf{l})} \right) (\mathbf{n} \cdot \mathbf{l})$$

- e confondendo la radianza uscente $L(x, \omega)$ con la intensità I^{out} e l'irradianza $L(x, \omega_L) d\omega$ con I si vede che la BRDF di

Phong è:

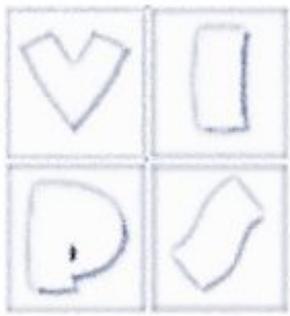
$$\rho(x, \omega_i, \omega_r) = k_d + k_s \frac{(\mathbf{n} \cdot \mathbf{h})^n}{(\mathbf{n} \cdot \omega_i)} \quad \text{dove } \mathbf{h} = \frac{\omega_i + \omega_r}{\|\omega_i + \omega_r\|}$$



Attenuazione

- Se due superfici distinte e parallele nella scena, costituite dallo stesso materiale, si sovrappongono nell'immagine, è impossibile distinguere dove finisce una e comincia l'altra, poiché avranno esattamente lo stesso colore, in base al modello locale.
- Per ottenere risultati più realistici, in genere si introduce un coefficiente di attenuazione della luce, f_{att} , che moltiplica la luce incidente I , per tener conto che l'intensità emessa da un punto che raggiunge una superficie decresce con l'inverso del quadrato della distanza:

$$f_{att} = \begin{cases} \frac{1}{c_1 + c_2 d_L + c_3 d_L^2} & \text{se } \frac{1}{c_1 + c_2 d_L + c_3 d_L^2} < 1 \\ 1 & \text{altrimenti} \end{cases}$$



Emissione

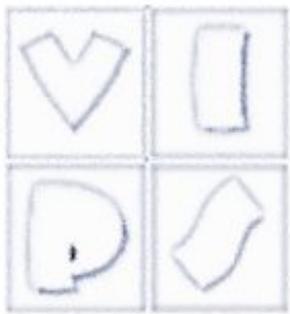
- Si può associare ad un oggetto una intensità di emissione I_e
- da aggiungere all'intensità calcolata con la formula di Phong.
- L'effetto di tale termine è che l'oggetto emette un proprio colore oltre alla luce riflessa dalla sorgente luminosa.
- Da notare che siccome stiamo lavorando con un modello locale, la luce emessa in questo modo da un oggetto influenza l'apparenza solo di quell'oggetto e non l'apparenza degli oggetti vicini (ovviamente poco realistico).
- Ovvero: un corpo emissivo non si comporta come una sorgente luminosa, in questo modello locale

Modello più fisicamente motivato

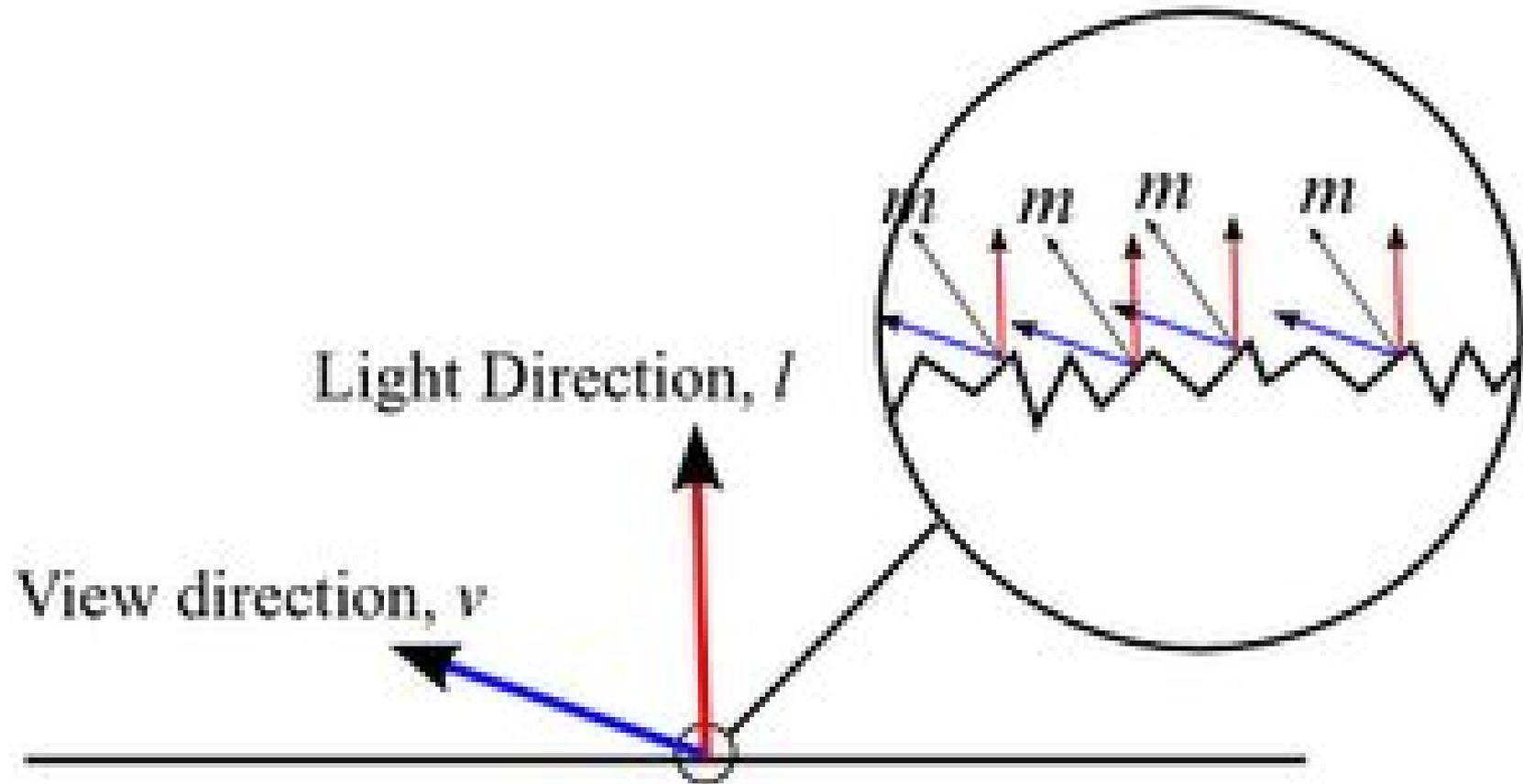
- Un modello maggiormente motivato dal punto di vista fisico potrebbe essere quello di Cook-Torrance
- basato sul modello microfacets della superficie, che descrive la superficie come formata da piccole facce disposte in modo variabile.
- Il modello di Cook-Torrance differisce da quello di Phong nella componente speculare.

$$I_s^{out} = I_s k_s \frac{\mathbf{n} \cdot \mathbf{l}}{\mathbf{n} \cdot \mathbf{v}} FGD$$

- dove $D(\mathbf{l}, \mathbf{v})$ misura la frazione di microfacette orientate in modo da dare riflessione speculare da \mathbf{l} a \mathbf{v} , $G(\mathbf{l}, \mathbf{v})$ misura la diminuzione di luce riflessa a causa dell'occlusione da parte di altre microfacette, $F(\mathbf{l}, \mathbf{v}, \lambda)$ è il coefficiente di Fresnel che fornisce la frazione di luce incidente che viene riflessa.



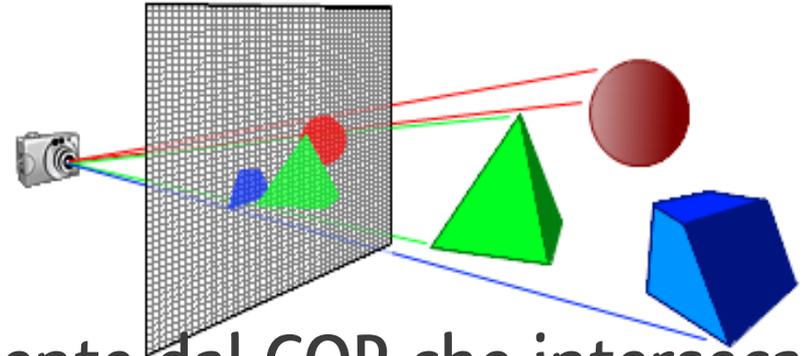
Microfacets



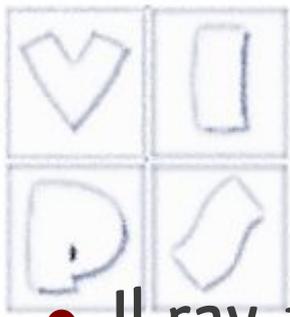
Quindi: ray casting

● Dovendo assegnare un colore ad ogni pixel, consideriamo il raggio ottico uscente da ciascun pixel.

● Un raggio ottico è una semiretta uscente dal COP che interseca il piano vista.



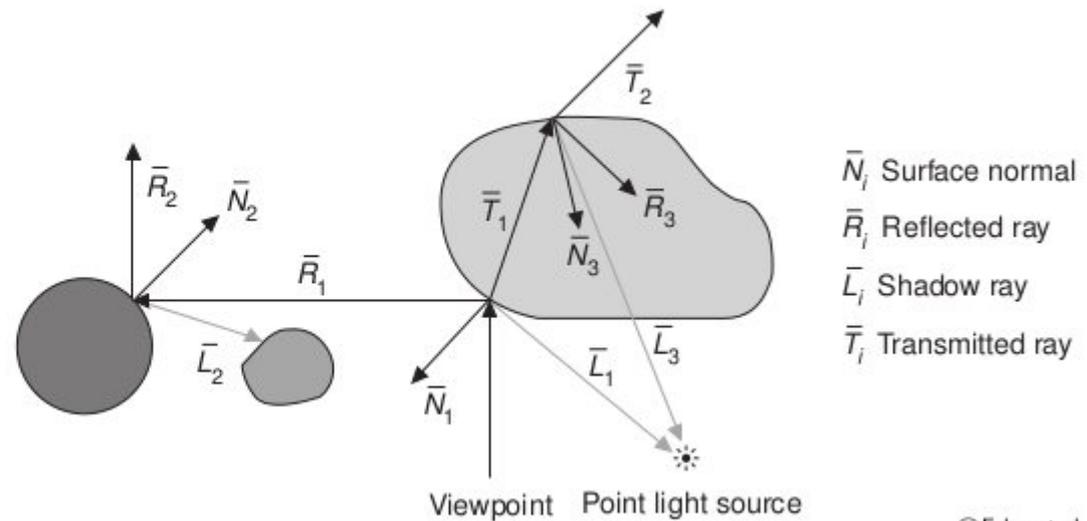
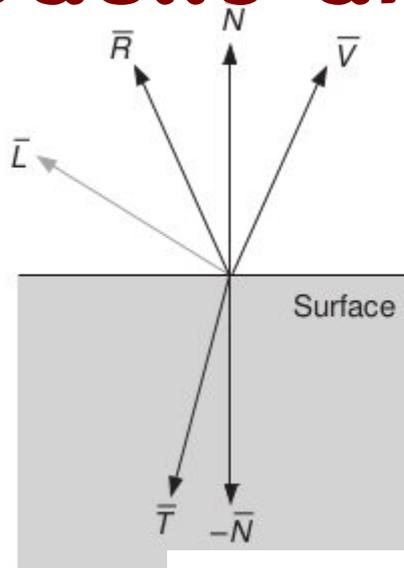
- Se il raggio non interseca alcun oggetto della scena allora gli assegno il colore di sfondo.
- Se il raggio interseca un oggetto, allora devo calcolare l'illuminazione I_{out} (il colore) nel punto di intersezione ed assegnarlo al pixel.
- Per calcolare il colore applico il modello locale (di Phong).
- Computazionalmente pesante trovare le intersezioni raggio-oggetti.
- Si possono facilmente aggiungere le ombre tracciando il raggio che connette il punto d'intersezione con la sorgente luminosa (shadow ray): se esso interseca qualche oggetto allora il punto è in ombra.



Ray tracing

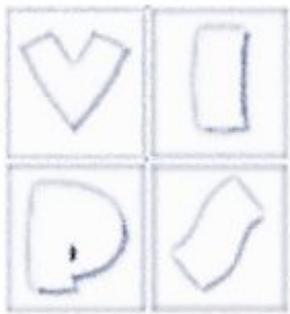
- Il ray-tracing (Whitted, 1980), usa tecnica ricorsiva e globale per calcolare l'illuminazione in un punto della superficie
 - modella bene per superfici trasparenti e riflettenti, ma non tiene conto della componente diffusiva della illuminazione indiretta.
- Si inizia come nel ray casting, tracciando il raggio ottico uscente da un pixel.
- Nel punto di intersezione del raggio con la superficie si applica il modello di illuminazione preferito (es. Phong) per le sorgenti luminose visibili da lì (si scoprono tracciando gli shadow rays).
- A partire dal punto di intersezione si tracciano altri due raggi:
 - uno raggio di riflessione ed uno di rifrazione, per raccogliere altri contributi
- Si applica il processo ricorsivamente, finché un raggio incontra una sorgente luminosa oppure si perde all'infinito.

Modello di illuminazione di Whitted



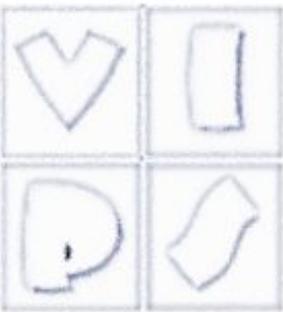
$$I^{out} = I_a k_a + I(k_d(\mathbf{n} \cdot \mathbf{l}) + k_s(\mathbf{n} \cdot \mathbf{h})^n) + k_s I_r + k_t I_t$$

- La prima parte coincide con il modello locale di Phong, illuminazione diretta da parte della sorgente luminosa.
- Gli ultimi due termini sono i contributi della luce riflessa specularmente e della luce trasmessa ricavate dalla legge di Snell per la rifrazione. Si calcolano ricorsivamente.



Calcolo delle intersezioni

- Si può fare con tutti i tipi di primitiva (e di modelli) visti
- Sfere, piani, poligoni, poliedri
- Computazionalmente pesante
 - Si possono usare strutture dati per semplificare il problema
 - Es. strutture dati gerarchiche, volumi di contenimento, octrees, kd trees



Intersezione retta piano

- L'eventuale punto di intersezione tra la retta $P = Q + tv$ ed il piano $(O - P) \cdot \mathbf{u} = 0$ deve soddisfare l'equazione:

$$(O - Q - tv) \cdot \mathbf{u} = 0$$

- Dunque si ottiene per t_0 la seguente soluzione

$$t_0 = (O - Q) \cdot \mathbf{u} / \mathbf{u} \cdot \mathbf{v}$$

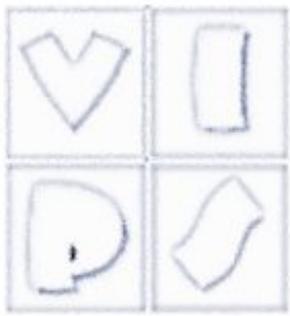
- che ha senso se $\mathbf{u} \cdot \mathbf{v} \neq 0$.

- Infatti se $\mathbf{u} \cdot \mathbf{v} = 0$ la retta è parallela al piano (e non lo interseca).



Intersezione retta-poligono

- Per prima cosa si fa un test per vedere se la retta interseca il piano contenente il poligono.
- Quest'ultimo è descrivibile semplicemente da uno dei vertici del poligono e dalla normale a questo
- Se la retta non interseca tale piano allora sicuramente non interseca nemmeno il poligono.
- Nel caso di intersezione bisogna determinare se il punto di intersezione giace o meno nel poligono.
- Assumiamo che il poligono P sia convesso. In tal caso un punto R giace all'interno di P se e solo se R si trova nel semipiano sinistro di tutte le rette orientate che contengono gli spigoli (la retta orientata da P_i a $P_{i+1} \forall i = 1 \dots n$).
 - Esercizio: trovare un modo per verificare che R giace a sinistra della retta orientata $P Q$



Riduzione della complessità

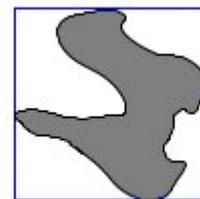
- Implementare un ray-tracer elementare è molto semplice.
 - Basta controllare l'intersezione di ciascun raggio con ogni primitiva (triangolo). Questo ha un costo lineare nel numero delle primitive ed è praticabile solo per modelli semplici.
- Per raggiungere costo sub-lineare sono necessarie strutture di indicizzazione spaziale, che consentano di limitare a-priori (senza fare il test) il numero delle primitive per le quali il test di intersezione viene effettuato.
- Queste tecniche di sfoltimento si chiamano anche pruning, culling o broad phase (contrapposto al narrow phase dove invece si calcolano le intersezioni (con test stringenti)).
- Tipicamente si considerano oggetti convessi. Se non lo sono, si spezzano in parti convesse.

Volumi di contenimento

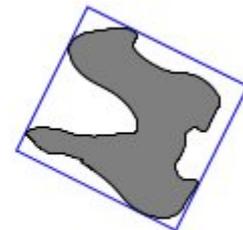
- Si racchiudono gli oggetti in solidi (“volumi”) che li contengono (bounding volume), con i quali sia facile testare l’intersezione: se non c’è intersezione con il bounding volume non c’è intersezione con l’oggetto racchiuso.
 - Questo non rende sub-lineare la complessità della ricerca delle intersezioni, ma semplifica le operazioni, e dunque sortisce nella pratica un miglioramento dei tempi.
- Tipici volumi di contenimento sono sfere, parallelepipedi con i lati paralleli agli assi cartesiani (AABB, da axis aligned bounding box), oppure parallelepipedi generici (OBB, da oriented bounding box)



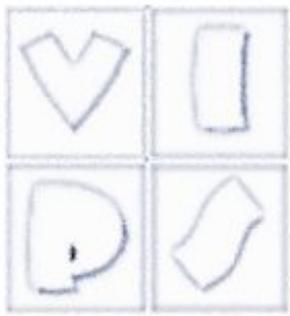
Sphere



AABB

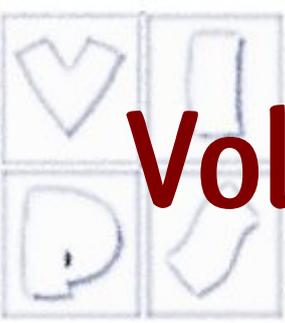


OBB



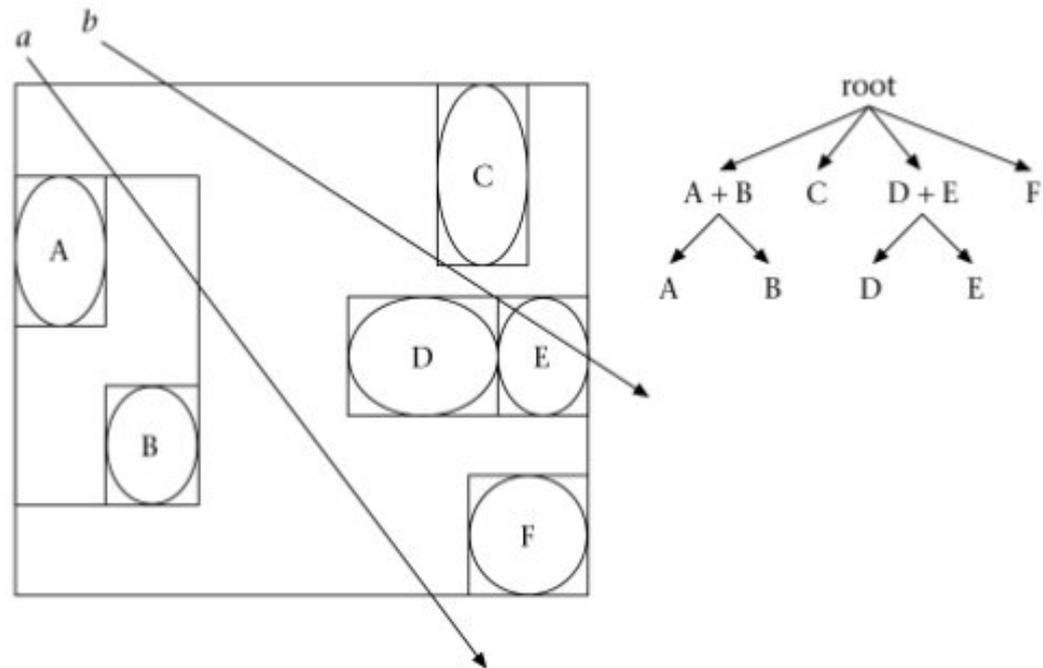
Partizionamento spaziale

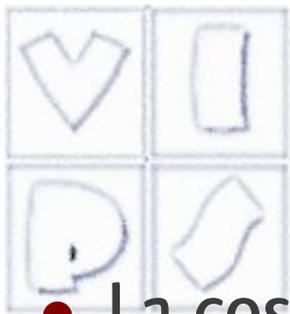
- Si suddivide lo spazio in celle, tipicamente cubiche (come una voxelizzazione grossolana).
- Si visitano le celle intersecate dal raggio (come Bresenham) e solo le primitive contenute in tali celle sono intersecati con il raggio.



Volumi di contenimento gerarchici

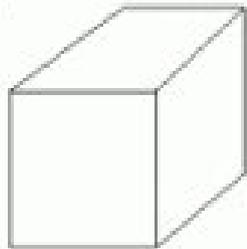
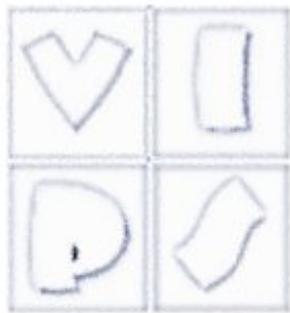
- Si costruisce una gerarchia di volumi di contenimento, dove al livello più alto si ha un volume che racchiude tutta la scena, ed al livello più basso si hanno volumi di contenimento per i singoli oggetti (o per un numero prefissato di primitive).
- E' una struttura organizzata bottom-up.
- E' stata introdotta proprio per il ray tracing



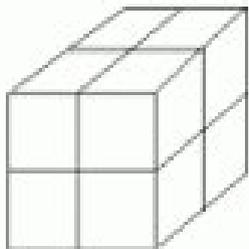


Octree

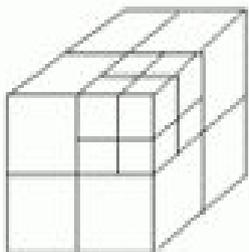
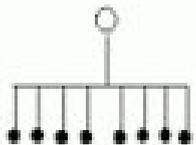
- La costruzione dell'Octree avviene come di consueto: una cella viene suddivisa ricorsivamente fino a che non contiene un numero prefissato di primitive.
- L'attraversamento (sequenziale) avviene nel seguente modo:
 - Interseca il raggio con il cubo che racchiude tutta la scena (nodo radice) e trova un punto P sul raggio "leggermente" all'interno.
 - Visita l'albero per scoprire a quale cella (foglia) appartiene P .
 - Interseca il raggio con le facce della cella per trovare la faccia di uscita.
 - La prossima cella è quella adiacente alla faccia di uscita.
- Come nel caso della partizione uniforme, solo le primitive contenute nelle celle visitate sono intersecati con il raggio.
 - Il vantaggio è che la divisione si adatta agli oggetti della scena invece di essere fissa ed uniforme



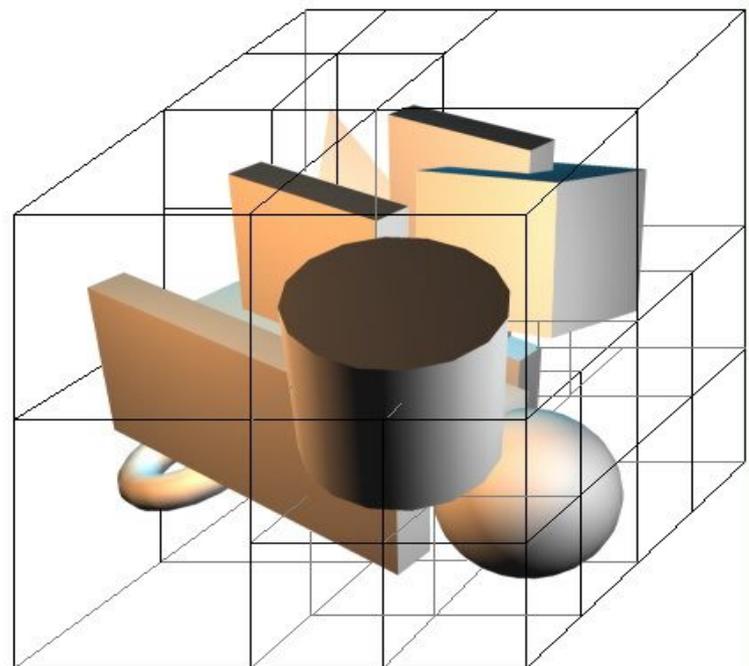
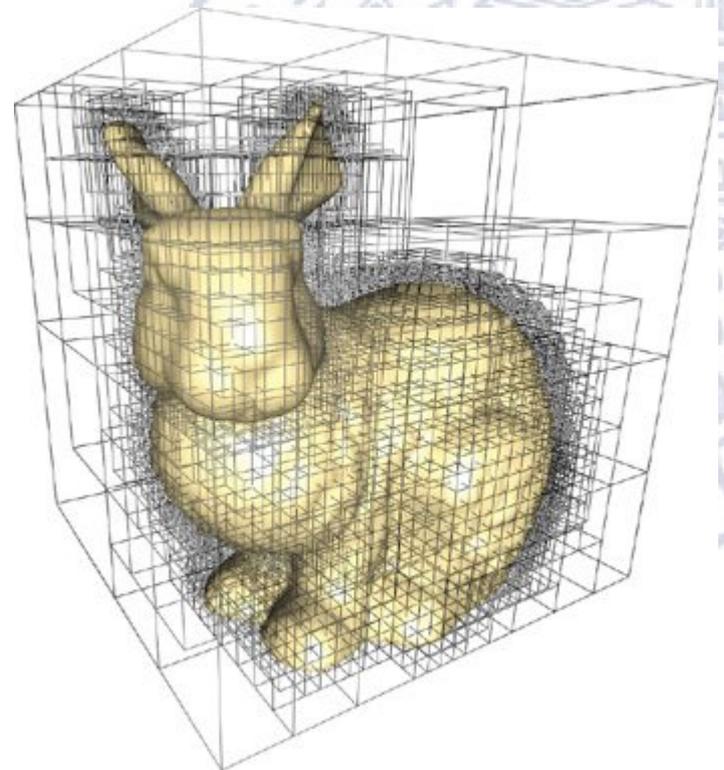
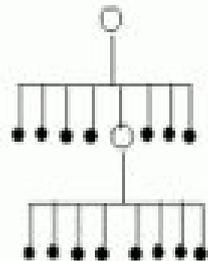
Radice

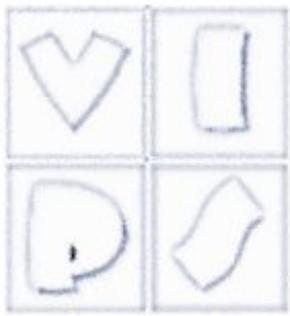


Livello 1



Livello 2

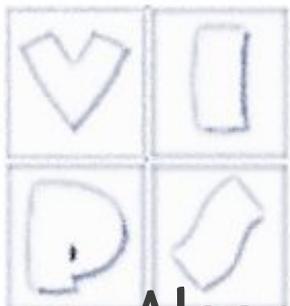




KD-Tree

- La costruzione avviene in modo che ogni piano divida a metà il numero di oggetti rimanenti.
- L'attraversamento può avvenire in modo sequenziale (come quello visto per l'Octree) oppure in modo ricorsivo, sfruttando la struttura dell'albero. Vediamo quest'ultimo.

Il raggio viene intersecato ricorsivamente con le regioni definite dall'albero. Visitando l'albero, il raggio viene intersecato con la regione corrente e ricorsivamente suddiviso in intervalli. Solo gli oggetti contenuti nelle foglie che alla fine intersecano un intervallo di raggio sono controllati.



Radiosity (Goral et al. 1984)

- Algoritmo globale che è complementare a ray-tracing: modella solo scene con superfici lambertiane (diffusori perfetti).
- Tutte le superfici della scena sono suddivise in frammenti chiamati “patch”.
- La radiosità di un patch (la sua “intensità”) è data dalla somma della (eventuale) radiosità emessa e di quella dovuta alla riflessione diffusa della luce incidente.
- La luce incidente proviene dalle sorgenti luminose (primarie) e da tutti gli altri patch, che, riflettendo diffusamente la luce, fungono da sorgenti secondarie.
- Pesante computazionalmente: ciascun patch dipende da tutti gli altri che sono da esso visibili, e per questo bisogna risolvere il problema della rimozione delle superfici nascoste dal punto di vista di ciascun patch della scena.



Radiosity

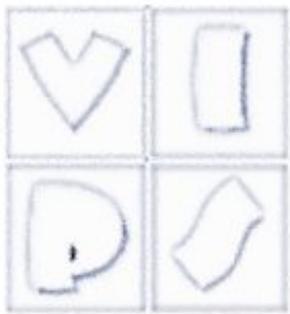
- Il Radiosity si basa sulla equazione della radiosità, che è la versione semplificata dell'equazione del rendering nel caso di superfici lambertiane.
- La BRDF è costante sull dominio angolare
- la radianza non dipende dalla direzione, e coincide con la radiosità (a meno di π): $B(x) = L(x) \int_{\Omega} \cos \theta d\omega = \pi L(x)$
- Riscriviamo dunque l'equazione del rendering (molt. per π):

$$\pi L(x) = \pi L_e(x) + \pi \rho(x) \int_{y \in S} L(y) G(x, y) dy$$

- Si ottiene così

l'equazione della radiosità:

$$B(x) = B_e(x) + \rho_d(x) \int_{y \in S} \frac{B(y)}{\pi} G(x, y) dy$$
$$= B_e(x) + \rho_d(x) \int_{y \in S} B(y) F(x, y) dy$$



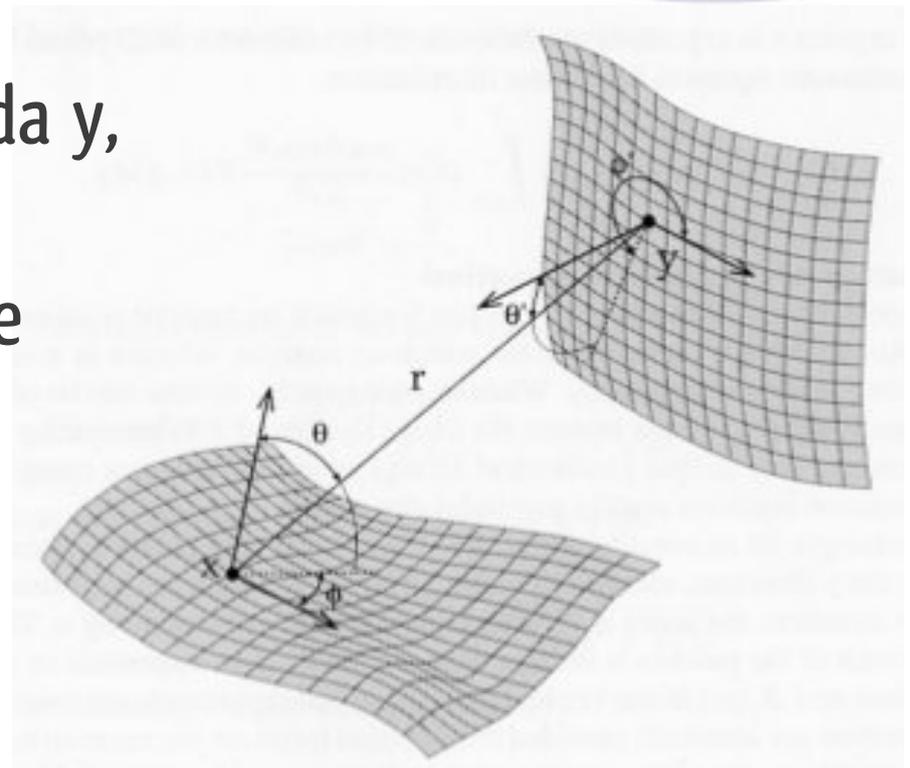
Fattore di forma

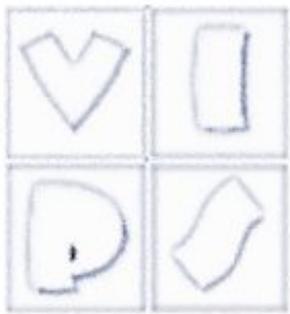
- Il termine

$$F(x, y) = \frac{G(x, y)}{\pi} = \frac{\cos \theta_{xy} \cos \theta_{yx}}{\pi \|x - y\|^2} V(x, y)$$

Prende il nome di fattore di forma e dipende solo dalla geometria della scena

- $V(x, y)$ vale 1 sse x è visibile da y ,
- θ_{xy} (θ_{yx}) è l'angolo formato dal raggio che congiunge x ed y con la normale in x (y)





Discretizzazione

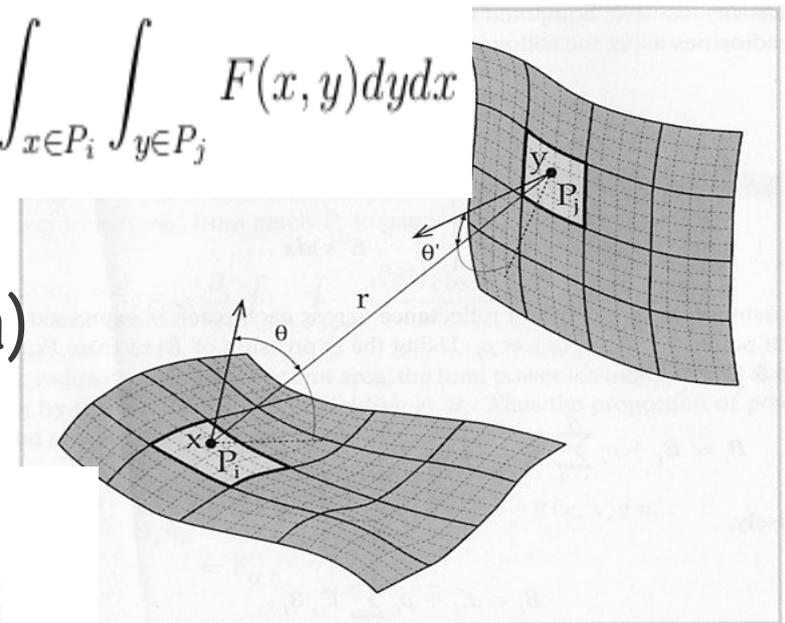
- Si può discretizzare e risolvere con elementi finiti
 - Suddividiamo la scena in N patch disgiunte P_j , $j = 1 \dots N$ ciascuna con area A_j .
 - Su ciascuna patch B , B_e e ρ_d sono costanti (un valore per patch).
 - Definiamo alcune quantità “per patch”:

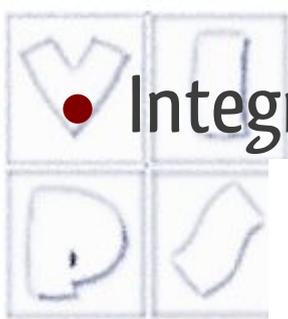
$$B_i = \frac{1}{A_i} \int_{x \in P_i} B(x) dx \quad E_i = \frac{1}{A_i} \int_{x \in P_i} B_e(x) dx \quad F_{ij} = \frac{1}{A_i} \int_{x \in P_i} \int_{y \in P_j} F(x, y) dy dx$$

(Attenzione: la E_i non c'entra con l'irradianza)

Approssimiamo l'integrale con una somma
sull

$$B(x) = B_e(x) + \rho_d(x) \sum_{j=1}^N \int_{y \in P_j} B(y) F(x, y) dy$$





- Integrando entrambi i membri

$$\frac{1}{A_i} \int_{x \in P_i} B(x) dx = \frac{1}{A_i} \int_{x \in P_i} B_e(x) dx + \frac{1}{A_i} \int_{x \in P_i} \rho_d(x) \sum_{j=1}^N \int_{y \in P_j} B(y) F(x, y) dy dx$$

- Usando le definizioni

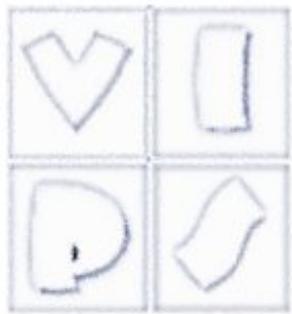
$$B_i = B_e + \sum_{j=1}^N \frac{1}{A_i} \int_{x \in P_i} \rho_d(x) \int_{y \in P_j} B(y) F(x, y) dy dx$$

- Essendo B_j e ρ_i costanti sulle patch ricaviamo la versione discreta dell'equazione della radiosità:

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

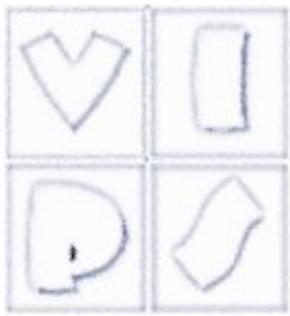
- In versione matriciale:

$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix} + \begin{bmatrix} \rho_1 F_{11} & \rho_1 F_{12} & \cdots & \rho_1 F_{1N} \\ \rho_2 F_{21} & \rho_2 F_{22} & \cdots & \rho_2 F_{2N} \\ \vdots & \ddots & & \vdots \\ \rho_N F_{N1} & \rho_N F_{N2} & \cdots & \rho_N F_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix}$$



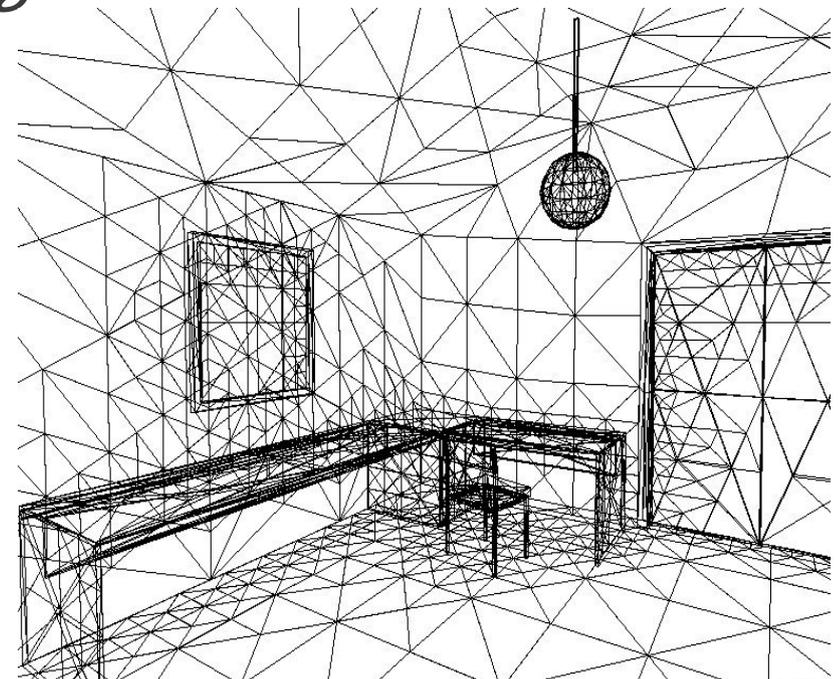
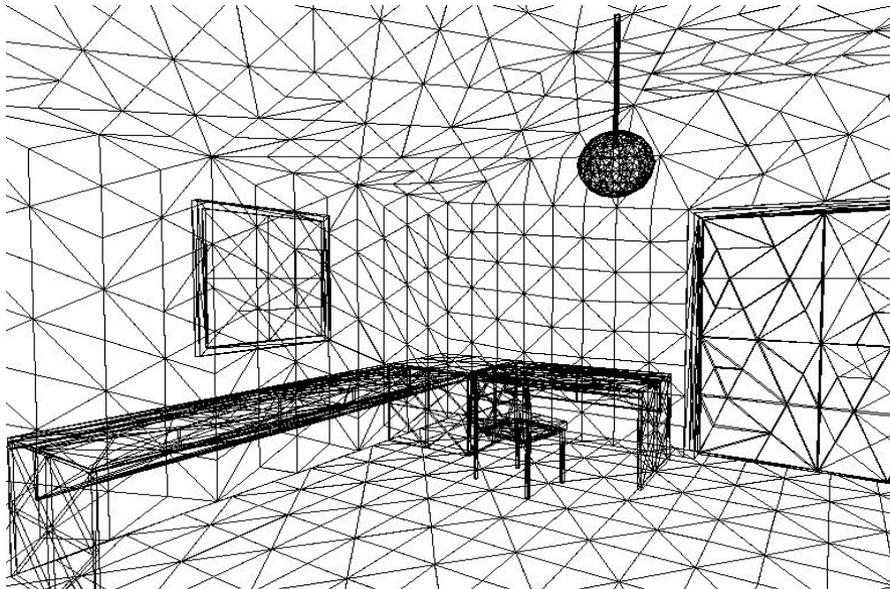
$$\begin{bmatrix} \rho_1 1 - F_{11} & \rho_1 F_{12} & \cdots & \rho_1 F_{1N} \\ \rho_2 F_{21} & 1 - \rho_2 F_{22} & \cdots & \rho_2 F_{2N} \\ \vdots & \ddots & \ddots & \vdots \\ \rho_N F_{N1} & \rho_N F_{N2} & \cdots & 1 - \rho_N F_{NN} \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}$$

- sistema lineare $N \times N$ denso e di grandi dimensioni (i patch sono solitamente più dei poligoni).
- La risoluzione fornisce la radiosità per tutti i patches, view-independent.
- Si può risolvere con un procedimento iterativo, es. Gauss-Seidel, aggiornando la radiosità di un singolo patch i impiegando la radiosità di tutti gli altri



Problemi

- Scelta dei patch (suddivisioni adattative)
- Soluzione del sistema lineare di grandi dimensioni (altri metodi iterativi)
- Calcolo dei fattori di forma: bisogna risolvere un problema di visibilità di tutti i patch con tutti gli altri.



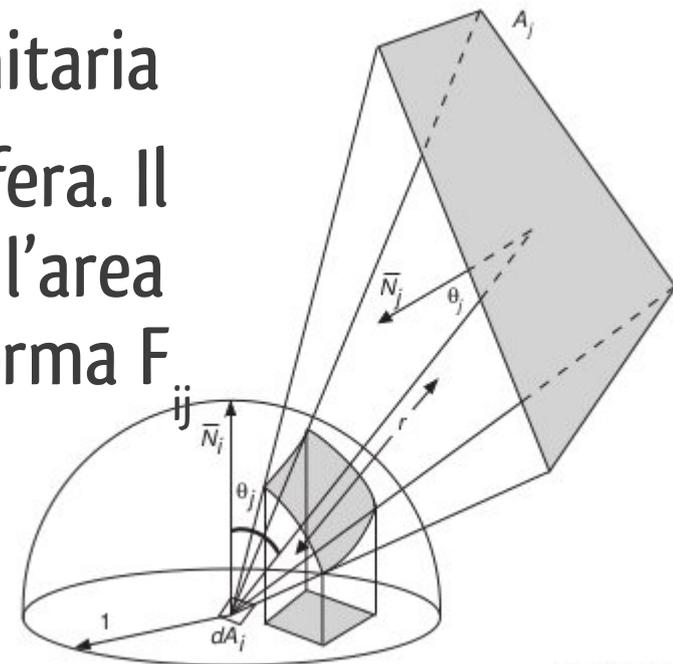


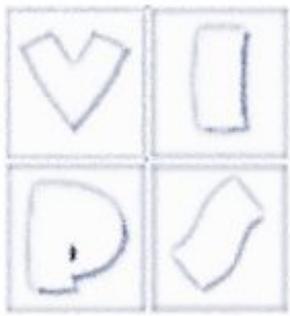
Fattore di forma

- Possiamo vedere il fattore di forma tra due patch F_{ij} come la media sul patch P_i della quantità F'_{ij}

$$F_{ij} = \frac{1}{A_i} \int_{x \in P_i} \underbrace{\left(\int_{y \in P_j} F(x, y) dy \right)}_{F'_{ij}} dx$$

- F'_{ij} è il fattore di forma tra un elemento di superficie infinitesimo $dx \in P_i$ ed il patch P_j
- Il metodo di Nusselt per calcolare F_{ij} consiste nel proiettare P_j sulla superficie della sfera unitaria centrata in \mathbf{x} , e poi sulla base della semisfera. Il rapporto tra l'area di questa proiezione e l'area della base della semisfera e il fattore di forma F_{ij}
 - Se assumiamo F_{ij} costante sulla patch, basta calcolarlo in un punto e si ha $F_{ij} = F'_{ij}$

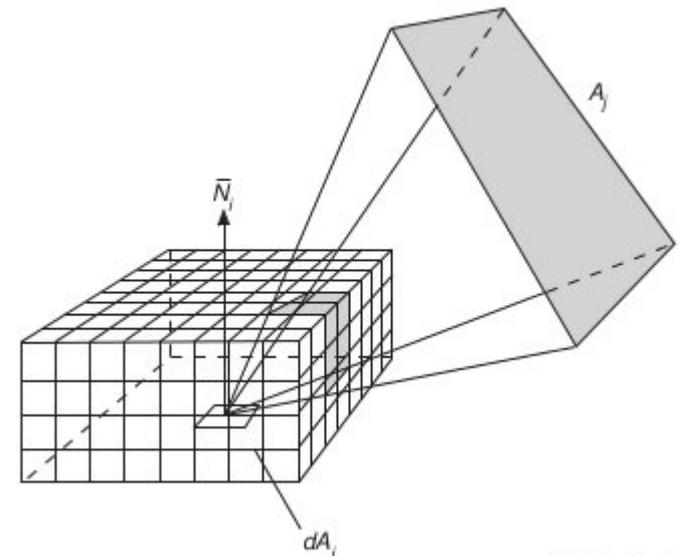


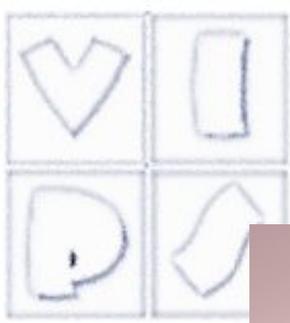


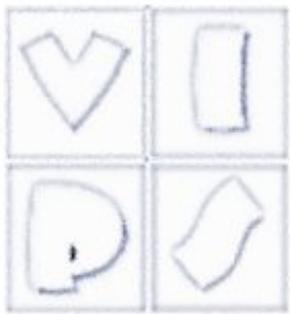
Metodo del semicubo

- Approssimazione del metodo di Nusselt che lavora in image-precision. Si usa un semicubo suddiviso in celle (pixel), invece che una semisfera, centrato sul punto x .
- Per ciascuna cella viene precalcolato il suo form factor, chiamato delta form factor.
- Il patch P_j viene proiettato sulle facce del cubo, sfruttando la pipeline di rasterizzazione
- Il form factor F_{ij} viene

approssimato come somma dei delta form factor delle celle del semicubo interessate dalla proiezione di P_j .







(32) ray-casting con Phong



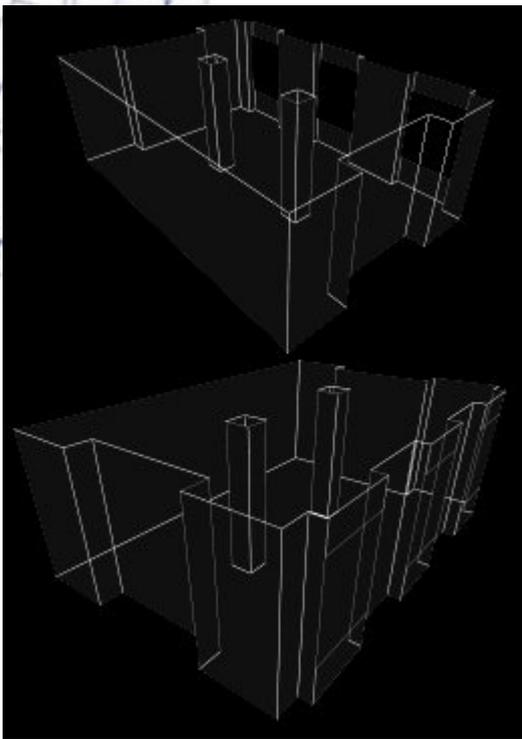
(33) ray-tracing



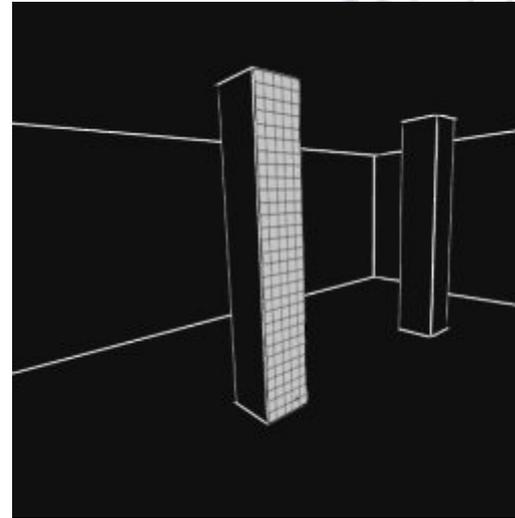
(34) radiosity + ray-tracing



(35) RADIANCE (Monte Carlo)

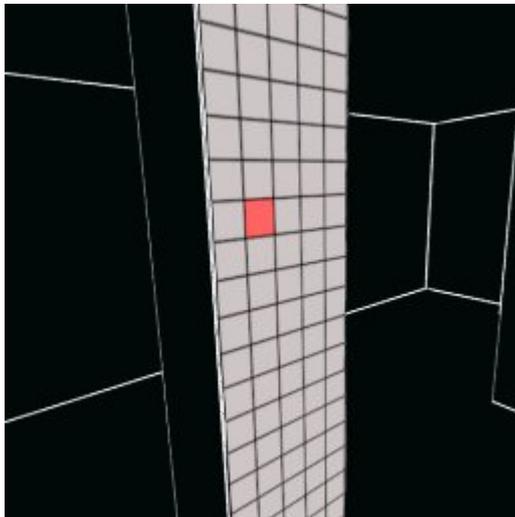


geometria

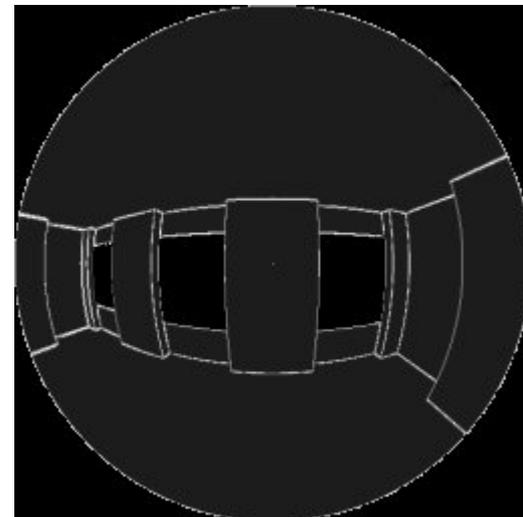


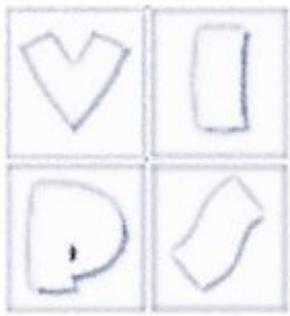
Patch su una colonna

Una patch



Vista dalla patch

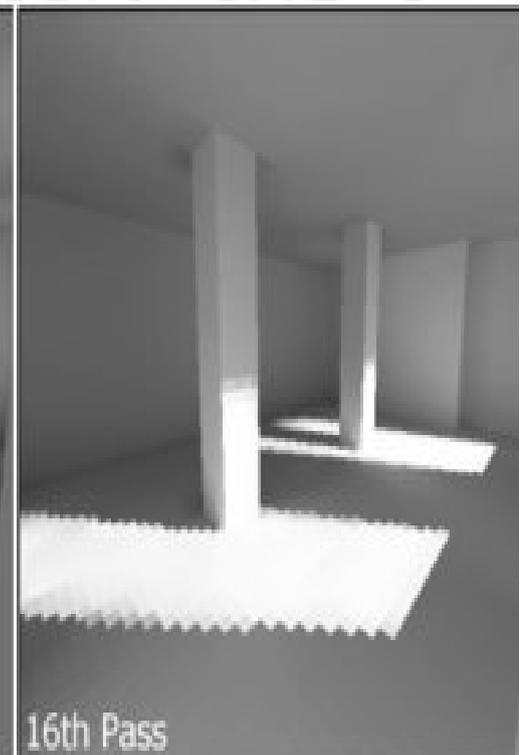
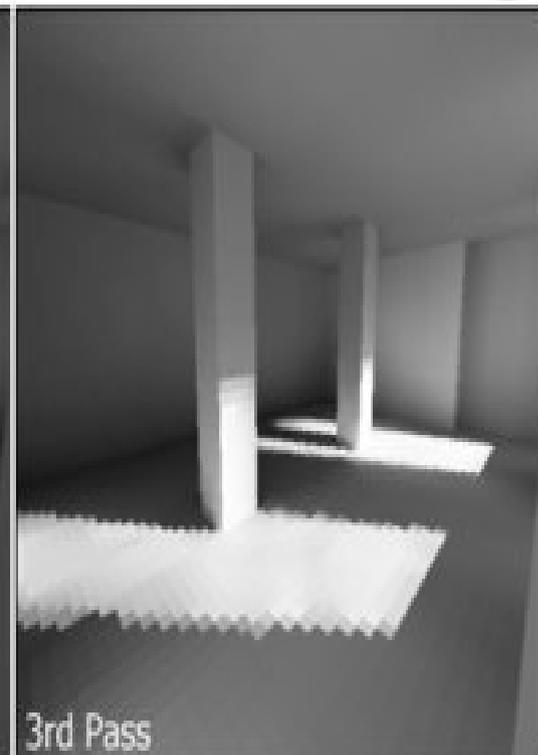
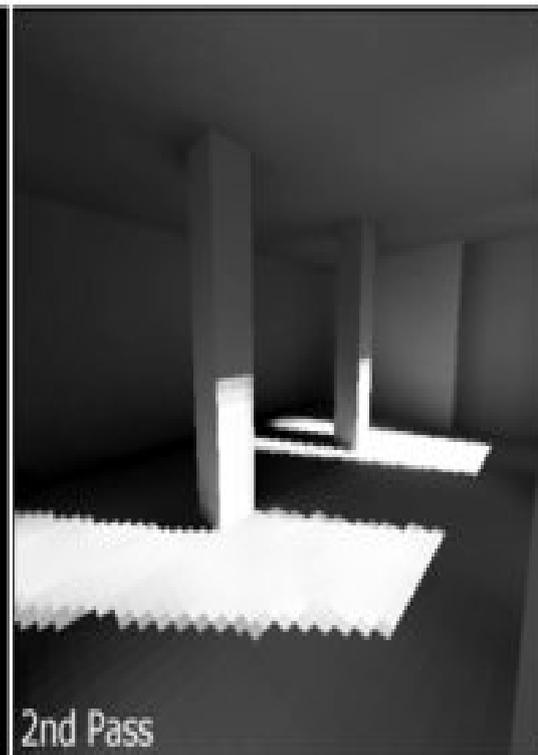
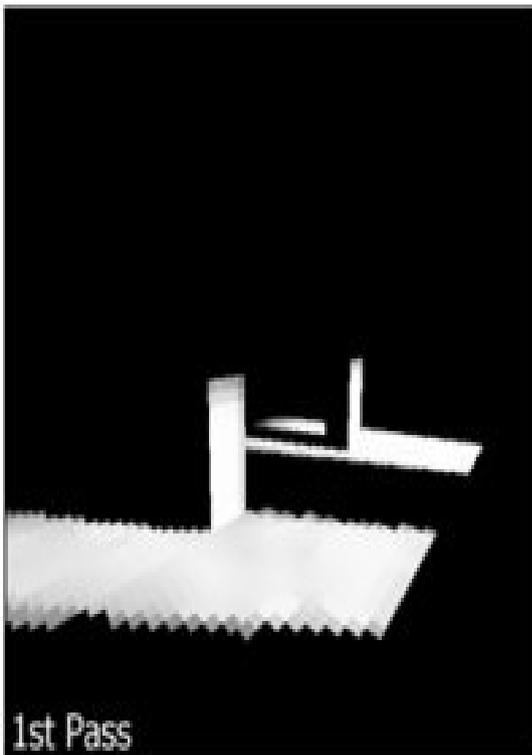


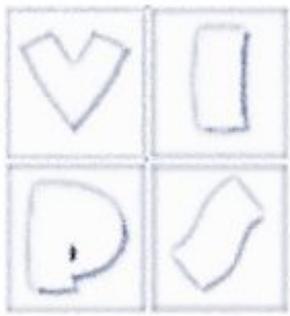


Radiosity



- Iterazioni successive





Riferimenti



- **Modelli di illuminazione**
 - Scateni (2005) cap. 6
 - Buss (2003) cap.3
 - Angel(2012) cap.5
- **RayTracing/Radiosity**
 - Scateni (2005) cap. 8
 - Buss cap. 9,10,11
 - Angel cap 11.1-11.5