

# Modelli di illuminazione



*Dove si discutono modelli per l'interazione della luce con le superfici.*

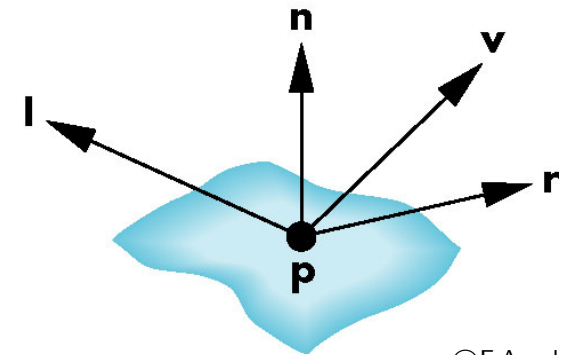
- **Introduzione**
- **Modello di Phong**
- **Modello di Cook-Torrance**
- **Tipi di luci**
- **Ray tracing**
- **Radiosity**

# Introduzione

- Un **modello di illuminazione** esprime i fattori che determinano il colore di una superficie in un determinato punto.
- Abbiamo introdotto il modello fisico di illuminazione (equazione del rendering).
- Vedremo in questa lezione quali modelli approssimati o empirici si usano in Grafica.
- Ci concentreremo prima su un modello di illuminazione **locale**, che tratta l'interazione tra una sorgente ed una singola superficie.
- Nei modelli locali ciascun punto viene trattato indipendentemente dal resto della scena (no interreflessioni, no ombre, no riflessioni speculari)
- Il modello che vedremo è semplice, abbastanza veloce da funzionare in tempo reale e produce risultati accettabili, per scene semplici.
- Riprenderemo poi il ray casting per introdurre il **ray tracing**, un metodo globale che tratta le riflessioni speculari.
- Discuteremo infine il metodo **radiosity**, in qualche senso ortogonale al ray tracing, che modella riflessioni tra diffusori perfetti.

# Modello locale di Phong

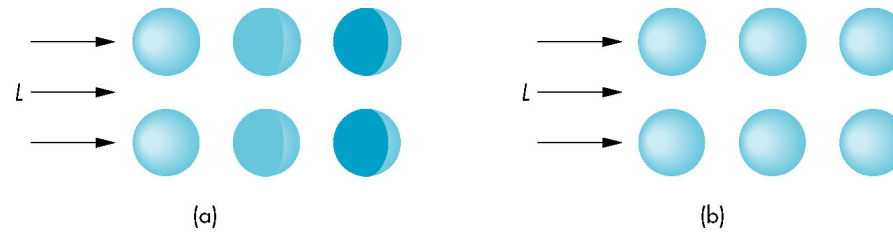
- La luce viene considerata composta da **tre componenti cromatiche** R,G,B. Si calcola l'intensità indipendentemente per ogni componente cromatica, ottenendo un colore.
- Nel seguito faremo riferimento ad una singola componente cromatica.
- Confondiamo "l'intensità luminosa"  $I$  con la radianza. La  $I$  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.
- In  $P$  incide un raggio luminoso da una sorgente puntiforme di intensità  $I$ . Sia  $l$  la direzione che congiunge  $P$  con la sorgente.
- Se sono presenti più sorgenti luminose, per il principio di sovrapposizione degli effetti, effettuiamo il calcolo indipendentemente per ogni sorgente e poi sommiamo.
- L'intensità  $I^{out}$  viene calcolata come somma di quattro componenti: luce ambientale, luce diffusa, luce riflessa e luce emessa:



©E.Angel

$$I^{out} = I_a^{out} + I_d^{out} + I_s^{out} + I_e^{out}$$

- Come vedremo, l'intensità luminosa  $I^{out}$  dipende da  $I$ , dalla orientazione della superficie in  $P$ ,  $\mathbf{n}$ , dalla posizione di sorgente luminosa,  $\mathbf{l}$ , ed osservatore,  $\mathbf{v}$  e dalle proprietà intrinseche del materiale che la compone.
- Nel modello locale conta la direzione  $\mathbf{l}$  sotto cui il punto vede la luce, non importa se il percorso che connette il punto alla sorgente è ostruito (no ombre).

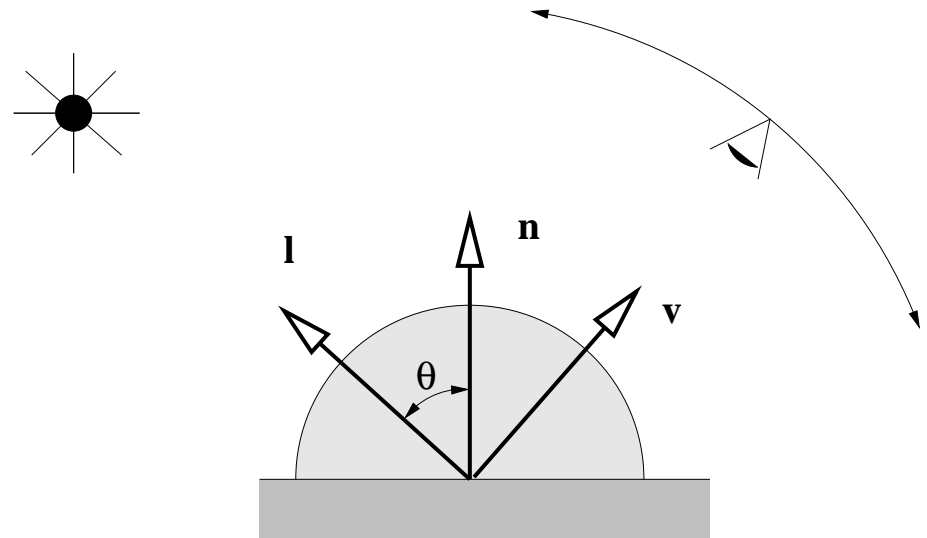


©E.Angel

## Componente diffusa

- Il secondo contributo a  $I^{out}$  è la cosiddetta **componente diffusa** o di **riflessione lambertiana**.
- Non dipende dalla direzione di vista  $\mathbf{v}$ , ma solo da  $\mathbf{n}$  e  $\mathbf{l}$ : una superficie lambertiana appare egualmente luminosa da qualunque direzione la si guardi.
- 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

- L'intensità di questa luce riflessa dipende solo dall'angolo tra la direzione della luce vista da  $P$ , ovvero  $\mathbf{l}$ , e la normale in  $P$ , che indichiamo con  $\mathbf{n}$  e supponiamo di norma unitaria.



- Il modello lambertiano prevede un contributo all'intensità del punto pari a

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

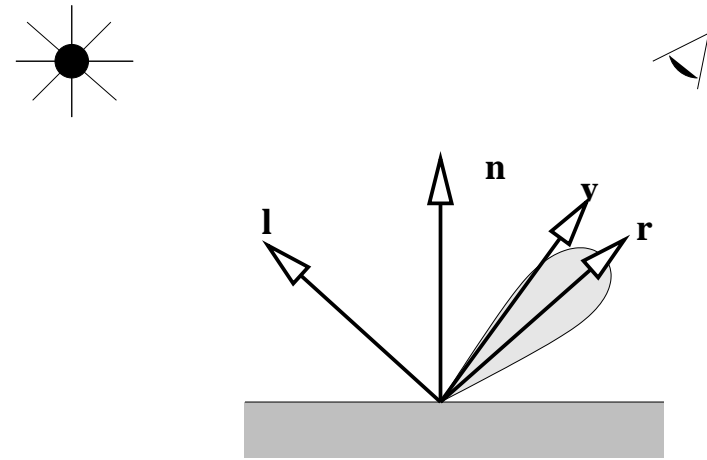
dove  $I$  è l'intensità della luce incidente,  $k_d$  è il **coefficiente di riflessione diffusa** del materiale che compone l'oggetto e  $\theta$  è l'angolo tra  $\mathbf{l}$  e  $\mathbf{n}$

- Tale angolo si presuppone compreso nell'intervallo  $[-\pi/2, \pi/2]$ , altrimenti si pone l'intensità della componente diffusa pari a zero.

## Componente speculare (di Phong)

- La luce però può anche essere riflessa da un corpo **specularmente**, ovvero seguendo le leggi della riflessione dell'**ottica geometrica**

- In tal caso la luce viene riflessa in una direzione  $\mathbf{r}$  che è ottenuta “specchiando”  $\mathbf{l}$  rispetto a  $\mathbf{n}$
- L'osservatore vede la luce riflessa solo se  $\mathbf{v} \cdot \mathbf{r} = 1$ .



- Si può dimostrare che per calcolare  $\mathbf{r}$  vale la seguente formula:

$$\mathbf{r} = 2\mathbf{n}(\mathbf{n} \cdot \mathbf{l}) - \mathbf{l}$$

- La componente speculare in realtà vuole modellare non solo la riflessione speculare perfetta, ma anche la caratteristica di **lucidità** (*glossy*) della superficie.

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

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

dove  $I$  è la luce incidente,  $k_s$  è il **coefficiente di riflessione speculare** ed  $n$  l'**esponente di riflessione speculare** del materiale.

- L'esponente  $n$  modula la lucidità della superficie. Per  $n$  che tende ad infinito si ha una riflessione speculare perfetta.
- La formulazione di Blinn (1977) è leggermente diversa:

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

è lo *halfway vector*, il vettore che biseca l'angolo formato da  $\mathbf{l}$  e  $\mathbf{v}$ .

- il vantaggio della formulazione di Blinn è che gestisce correttamente il caso in cui l'angolo tra  $\mathbf{r}$  e  $\mathbf{v}$  sia  $> \pi/2$
- si osservi che 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.
- entrambe sono modelli empirici, privi di significato fisico, che hanno il merito di produrre superfici lucide **visivamente** credibili.



## Luce ambientale

- I modelli locali non simulano la riflessione multipla della luce (interriflesioni)
- Questo significa che se una superficie punto di un oggetto non è rivolto verso la luce, allora risulterà completamente nero.
- Si introduce allora nel modello locale un termine che prende il nome di **luce ambientale**, caratterizzata con una intensità  $I_a$ , che tiene conto dell'illuminazione indiretta dovuta alle interriflesioni.
- Alla superficie viene quindi assegnato un **coefficiente di riflessione ambientale**  $k_a$  compreso tra 0 ed 1.
- Il contributo all'intensità finale del punto da parte della luce ambientale sarà dato da

$$I_a^{out} = I_a k_a$$

## Sommiamo il tutto

- Sommando quanto visto fino ad ora, otteniamo la seguente espressione per l'intensità di colore di un punto  $P$

$$I^{out} = I_a k_a + I (k_d(\mathbf{n} \cdot \mathbf{l}) + k_s(\mathbf{n} \cdot \mathbf{h})^n) \quad (1)$$

- 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)

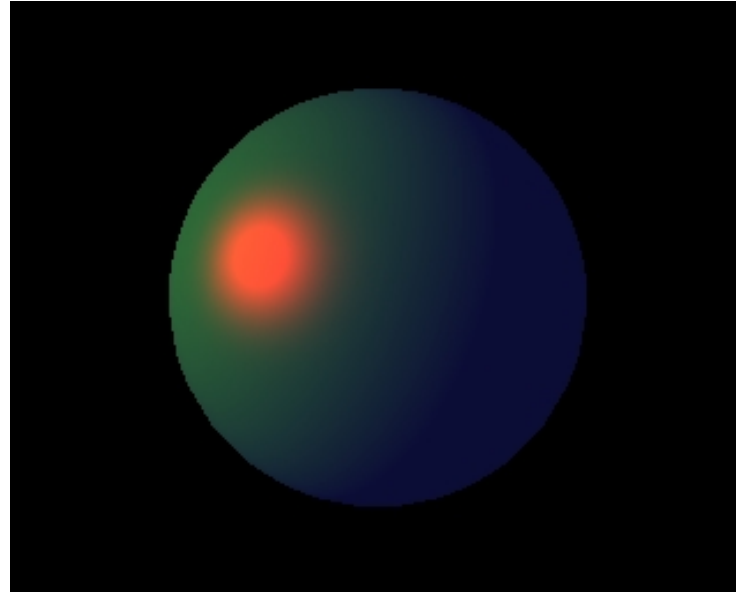
$$I^{r,out} = I_a^r k_a^r + I^r (k_d^r(\mathbf{n} \cdot \mathbf{l}) + k_s^r(\mathbf{n} \cdot \mathbf{h})^n)$$

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

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

- Il colore della superficie si determina fissando i coefficienti di luce ambientale  $(k_a^r, k_a^g, k_a^b)$  e di diffusione  $(k_d^r, k_d^g, k_d^b)$  (di solito sono uguali).
- La superficie apparirà del colore specificato dalla terna di coefficienti quando illuminata da luce bianca.
- Le riflessioni speculari (highlights) invece sono di solito del colore della luce, non della superficie, quindi i coefficienti  $(k_s^r, k_s^g, k_s^b)$  sono sempre (1,1,1), indipendentemente dal colore della superficie.

- La figura sotto mostra il risultato ottenuto assegnando colori diversi ai coefficienti ambientale, di diffusione e speculare di un oggetto.



- Può succedere che una o più componenti cromatiche eccedano il loro limite massimo (256 o 1 a seconda della rappresentazione).
- Questo accade specialmente quando si sommano i contributi di diverse sorgenti luminose.
- In tal caso si possono adottare varie strategie; la più semplice è quella di effettuare un **clamping**, ovvero se un  $I^{out}$  supera il valore massimo lo si pone pari a questo. Oppure si possono rinormalizzare i valori di  $I^{out}$  al valore più alto ottenuto nell'immagine

- La formula (1) prende il nome di **modello di Phong**, anche se, a rigore, il contributo di Phong riguarda il solo termine speculare.
- Il modello, nella forma più generale, prevede tre componenti separate per le sorgenti di luce (come sul Buss): luce speculare  $I_s$  (ovvero che verrà riflessa specularmente dalle superfici), luce diffusa  $I_d$  (ovvero che verrà diffusa dalle superfici, e luce ambientale  $I_a$  (che interagisce il coefficiente ambientale della superficie).
- Nella trattazione precedente 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 (speculare, diffusa e ambientale) pur non avendo un significato fisico, serve a rendere il modello più flessibile (posso simulare effetti globali con un modello locale).
- Per esempio, la luce ambientale  $I_a$  potrà in generale avere un colore diverso dalla  $I$ . Questo serve a rendere 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, perché così quando una luce viene spenta (o accesa) l'illuminazione ambientale diminuisce (o aumenta)



(1) Ambientale

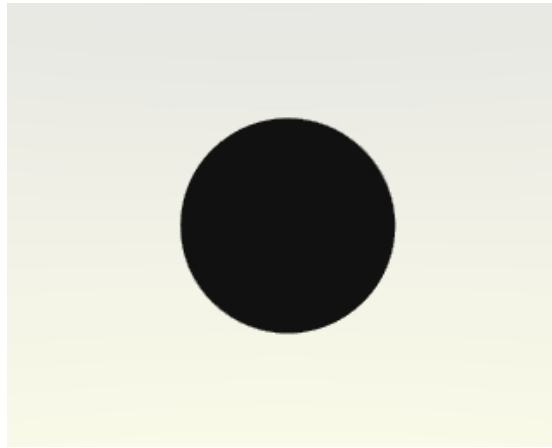


(2) Ambientale + Diffusa

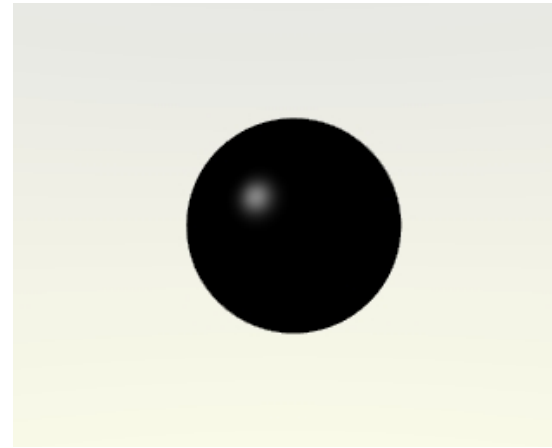


(3) Ambientale + Diffusa + Speculare

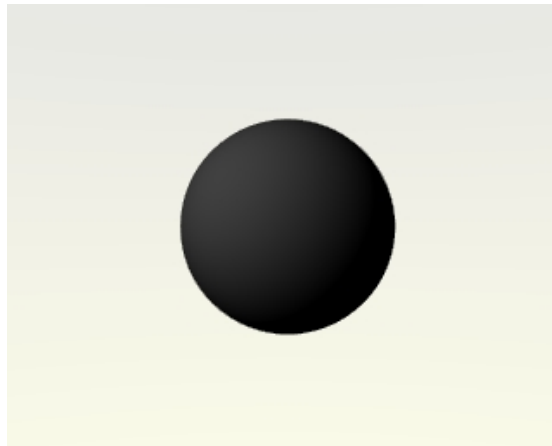
©Alan Watt



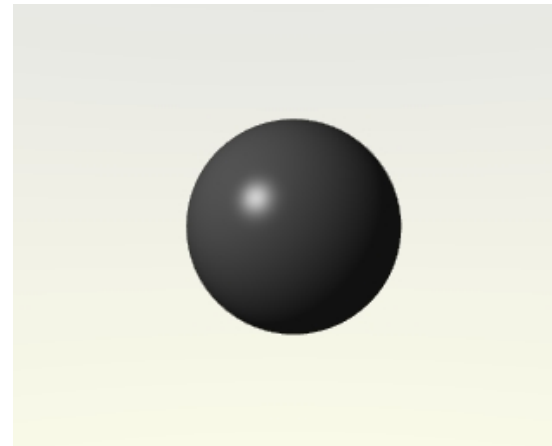
(4) Ambientale



(5) Speculare

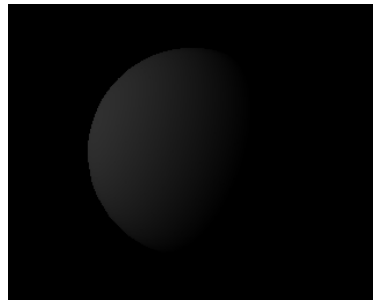


(6) Diffusa

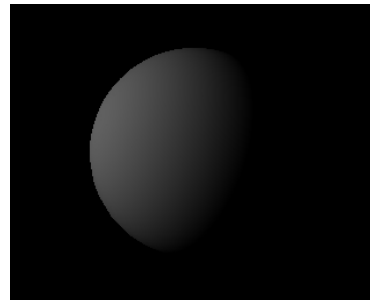


(7) Totale

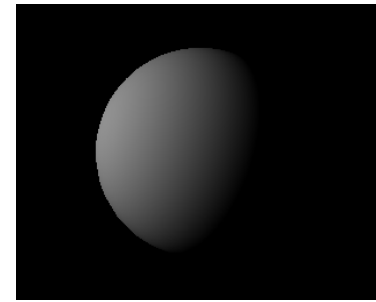
$I_a = 0$



(8)  $k_d = 0.3$

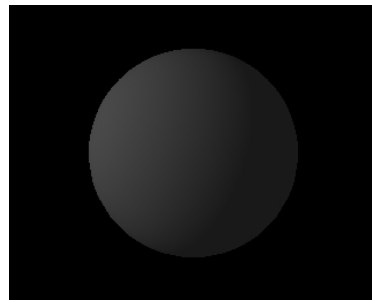


(9)  $k_d = 0.6$

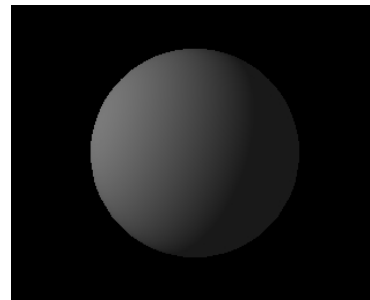


(10)  $k_d = 0.9$

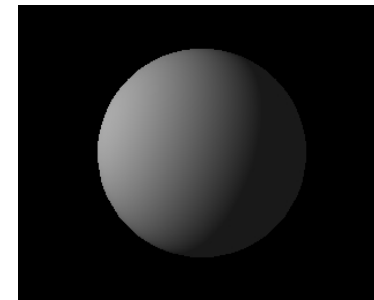
$I_a = 0.1$



(11)  $k_d = 0.3$

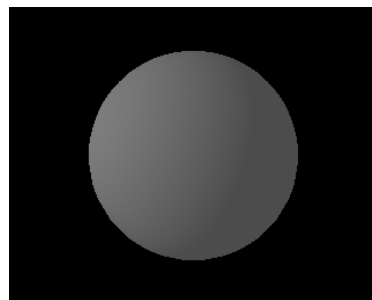


(12)  $k_d = 0.6$

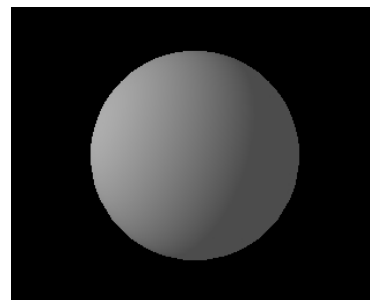


(13)  $k_d = 0.9$

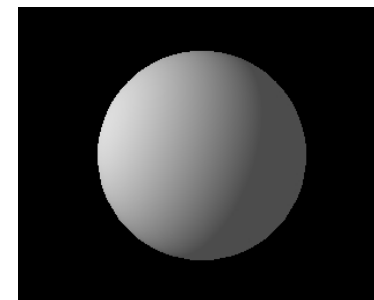
$I_a = 0.3$



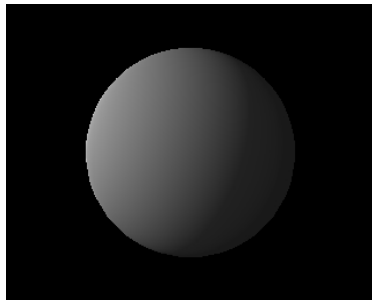
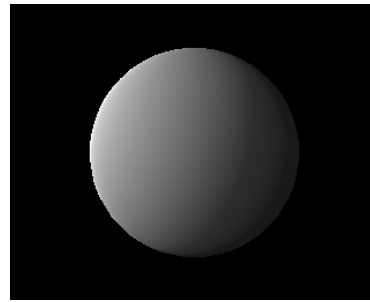
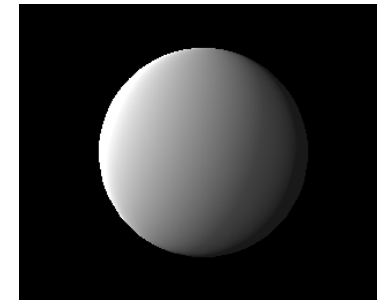
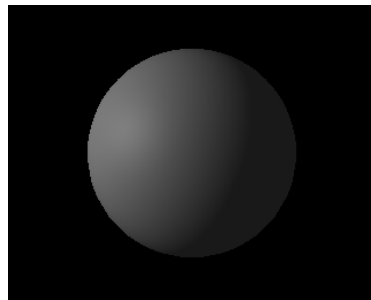
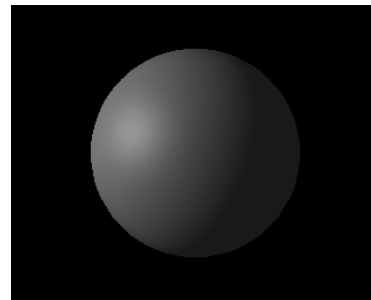
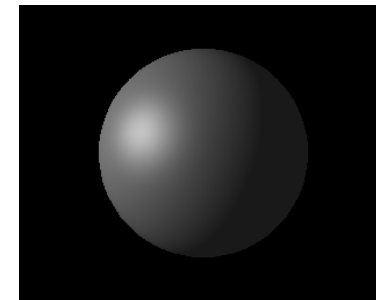
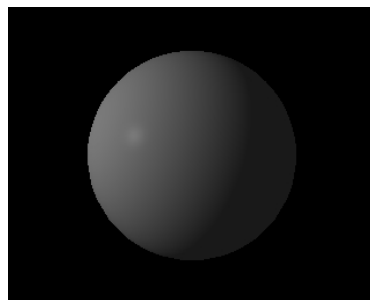
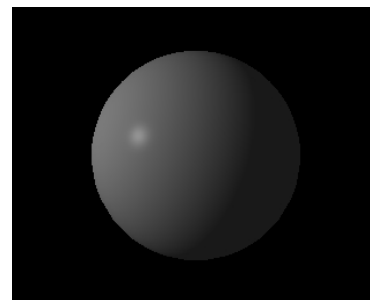
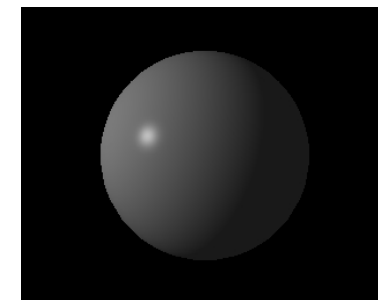
(14)  $k_d = 0.3$



(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  $\omega$  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  $\omega_L$  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 la irradianza

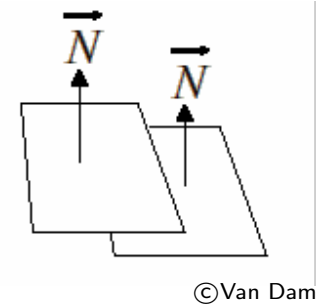
$L(x, \omega_L)d\omega$  con  $I$  si vede che la BRDF di Phong è:

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

- Nota: vi sono alcune imprecisioni nella derivazione, ma il risultato è corretto.

## Attenuazione della luce

- 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 della luce emessa da un punto che raggiunge una superficie decresce con l'inverso del quadrato della distanza:

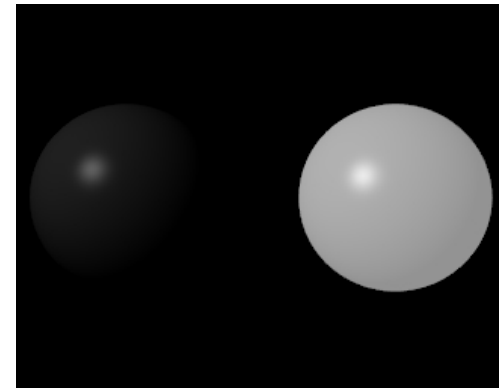
$$f_{att} = \min \left( \frac{1}{c_1 + c_2 d_L + c_3 d_L^2}, 1 \right)$$

dove  $d_L$  è la distanza percorsa dalla luce dalla sorgente alla superficie,  $c_1, c_2, c_3$  sono coefficienti che consentono di modulare l'effetto.

- Nota: In teoria dovrebbe essere  $f_{att} = \frac{1}{d_L^2}$ , ma la formula scritta sopra produce risultati empiricamente migliori.
- Comunque **il modello di Phong non ha l'ambizione di essere un modello fisico.**

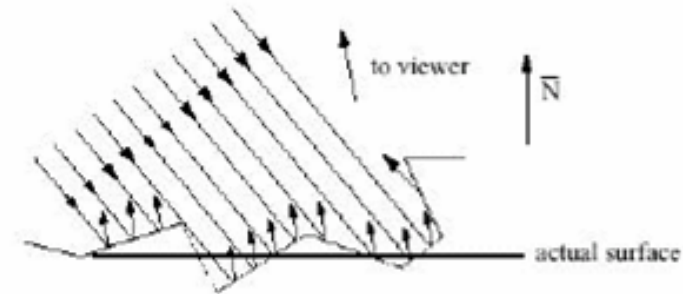
## Emissione

- Si può associare ad un oggetto una **intensità di emissione**  $I_e^{out}$  da aggiungere all'intensità calcolata con la formula (1).
- 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 di Cook-Torrance (cenni)

- Il modello di Cook-Torrance è più sofisticato di quello di Phong, ed è basato sul modello *microfacets* della superficie, che descrive la superficie come formata da piccole facce disposte in modo variabile.



©Van Dam

- Il modello di Cook-Torrance differisce da quello di Phong nella componente speculare.

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

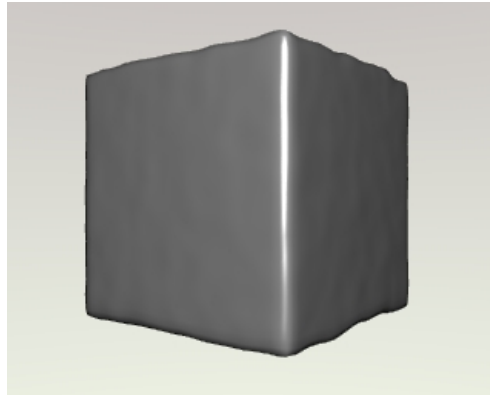
dove  $D(\mathbf{l}, \mathbf{v})$  misura la frazione di microfacette orientate in modo da dare riflessione speculare da  $\mathbf{l}$  a  $\mathbf{v}$ ,  $G(\mathbf{l}, \mathbf{v})$  misura la diminuzione di luce riflessa a causa dell'occlusione da parte di altre microfacette,  $F(\mathbf{l}, \mathbf{v},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

- È un modello che ha un significato fisico.

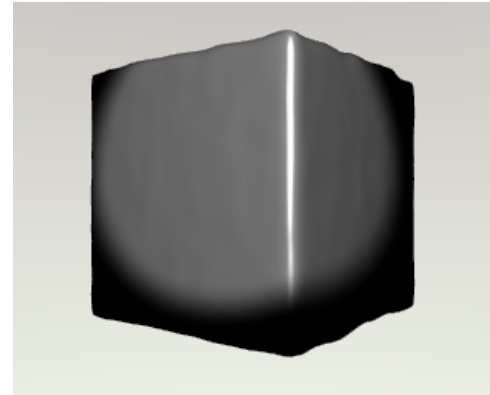
## Tipi di luci

- Fino ad ora abbiamo trattato essenzialmente solo due tipi di luci, la **luce ambientale** ed una **sorgente puntuale**. Sebbene siano i tipi di luci più usate, per raggiungere certi effetti è comodo introdurre almeno tre altri tipi di luce
- **Spotlight**: si tratta di una luce che simula il funzionamento di un faretto direzionale. Anziché emettere raggi di luce di ugual intensità in tutte le direzioni, una spotlight emette con maggiore intensità lungo una direzione specifica e l'intensità dei raggi decade man mano che ci si allontana da questa direzione. Bisogna specificare: a) la direzione  $s$  della spotlight, b) l'angolo di apertura massimo oltre il quale l'intensità è 0 (cutoff angle), una legge di decadimento angolare che controlla quanto velocemente l'intensità decresce deviando dalla direzione dello spotlight. Se  $I$  è l'intensità lungo la direzione  $\omega$ , l'intensità lungo un'altra direzione  $\omega'$ , si prende pari a  $I(\omega \cdot \omega')^k$  dove  $k$  è un parametro che determina l'andamento del decadimento.
- **Distant light**: si tratta di una sorgente posta all'infinito i cui raggi arrivano sulla scena tutti paralleli (luce direzionale). Per specificarla quindi è sufficiente una intensità ed una **direzione** (non una posizione). Viene usata per simulare sorgenti luminose tipo il sole; è quindi adatta per esterni.

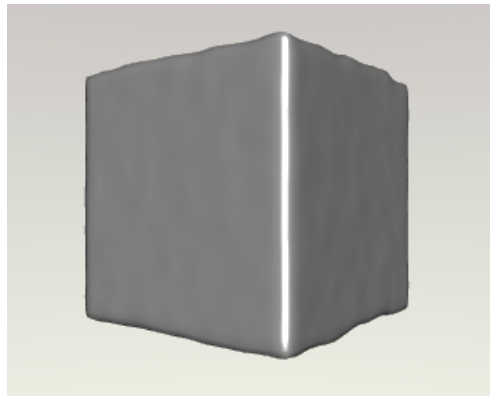
- **Hemispherical light:** in questo caso i raggi luminosi arrivano da una semisfera e può essere utile per simulare la luce diffusa (al posto di una luce ambientale) in esterni.



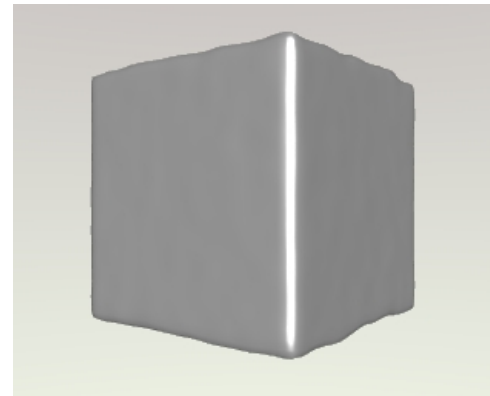
(26) Point light



(27) Spot light



(28) Distant light

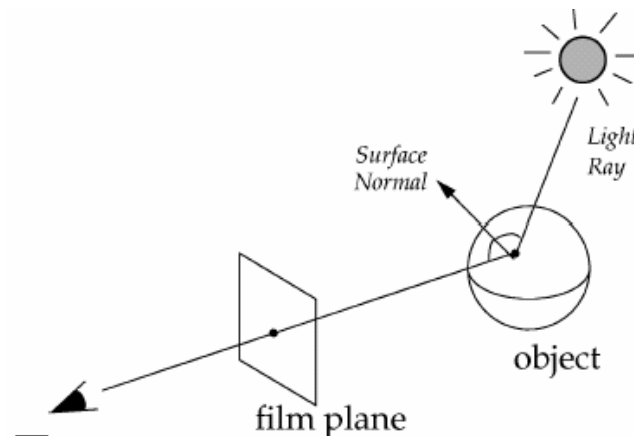


(29) Hemi light

# Ray-tracing

## Rendering con ray casting

- Il rendering avviene secondo il paradigma del **ray-casting**: si “gettano” (cast) i raggi uscenti da ciascun pixel per calcolarne il colore.



©Van Dam

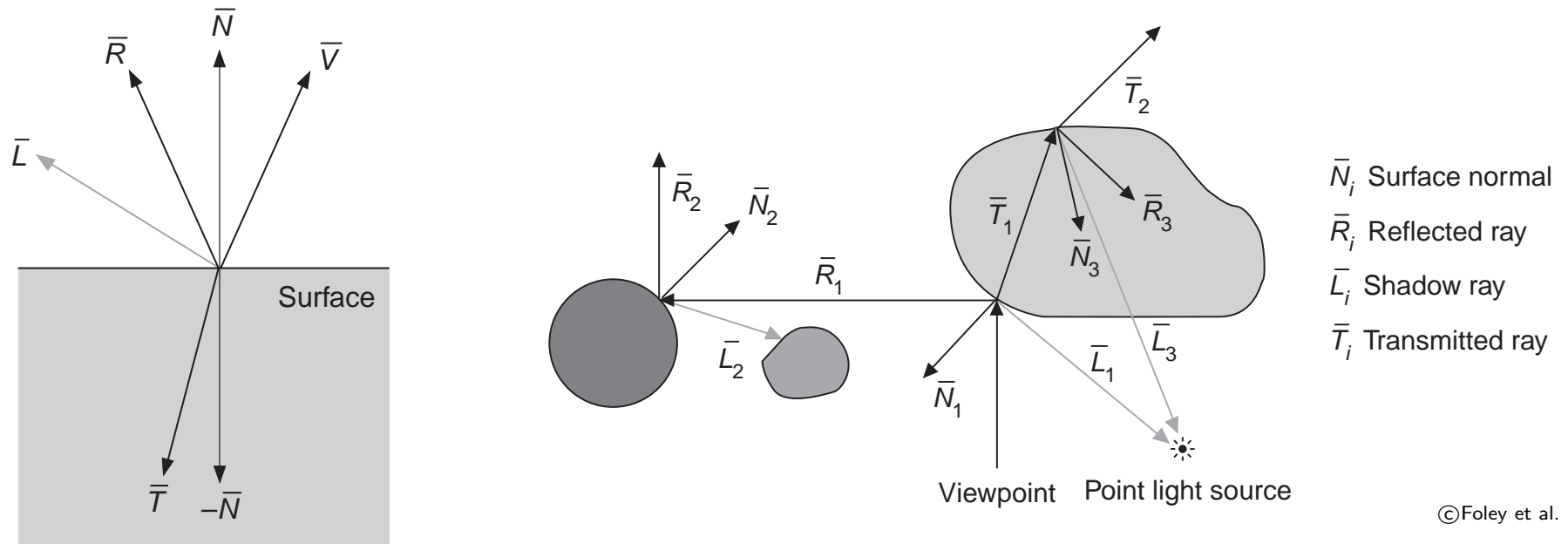
- Dovendo assegnare un colore ad ogni pixel, consideriamo il raggio ottico uscente da ciascun pixel.
- Un raggio ottico è una semiretta uscente dal COP che interseca il piano vista.
- Se il raggio non interseca alcun oggetto della scena allora gli assegno il colore di sfondo.
- Se il raggio interseca 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).
- In generale, c'è un pesante onere computazionale nel calcolo delle 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.
- Il ray-casting è un modo intuitivo e facilmente implementabile per effettuare il rendering, anche se poco efficiente, come vedremo.

# Ray-tracing

- Il **ray-tracing** (Whitted, 1980), applica una tecnica ricorsiva e globale per calcolare l'illuminazione in un punto della superficie.
- Ray-tracing 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 quel punto (si scoprono tracciando gli shadow rays).
- A partire dal punto di intersezione del raggio con la superficie, si tracciano altri due raggi: uno raggio di riflessione ed uno di rifrazione, per raccogliere altri contributi all'illuminazione, dovuti alla riflessione ed alla rifrazione (se l'oggetto è trasparente).
- Si applica il processo ricorsivamente, finché un raggio incontra una sorgente luminosa oppure si perde all'infinito.



- Il **modello di illuminazione di Whitted** si scrive:

$$I^{out} = I_a k_a + I(k_d(\mathbf{n} \cdot \mathbf{l}) + k_s(\mathbf{n} \cdot \mathbf{h})^n) + k_s I_r + k_t I_t \quad (2)$$

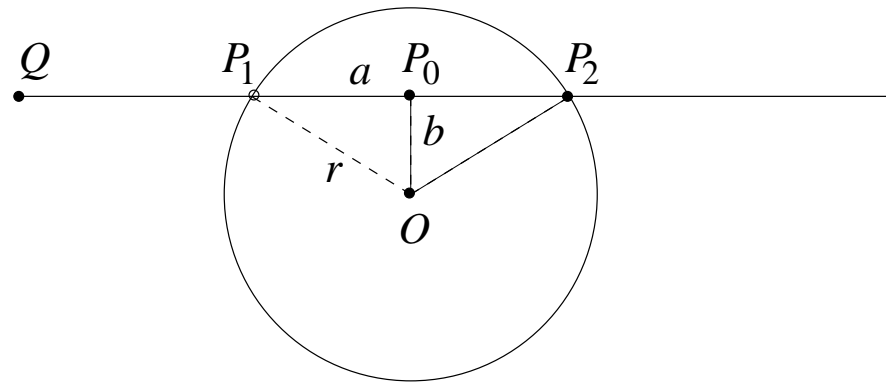
- La prima parte coincide con il modello locale di Phong, ed è il contributo dovuto alla illuminazione diretta da parte della sorgente luminosa.
- Gli ultimi due termini sono i contributi della luce riflessa specularmente che incide lungo la direzione  $\mathbf{r}$ ,  $I_r$ , e della luce trasmessa che incide lungo una direzione  $\mathbf{t}$ ,  $I_t$ , determinata dalla legge di Snell per la rifrazione. Si calcolano ricorsivamente.

## Intersezioni raggio-oggetto

- L'operazione fondamentale nel ray-tracing (come pure il ray-casting) è l'intersezione del raggio tracciato dal piano immagine con gli oggetti (poligoni) della scena.
- Iniziamo con alcune formule per calcolare l'intersezioni tra rette e varie figure geometriche, utili per la grafica 3D; in tutti i casi la retta è descritta dalla sua equazione vettoriale  $P = Q + t\mathbf{v}$  e la soluzione equivale a trovare il valore di  $t$  per cui si ha intersezione.
- Il caso per la semiretta con origine in  $Q$  è uguale solo che alla fine bisogna controllare il segno di  $t$ . Si ha intersezione solo se  $t > 0$ .

## Intersezione Retta-Sfera

- Data una sfera di centro  $O$  e raggio  $r$ , ed una retta  $P = Q + t\mathbf{u}$  si vuole determinare se la retta interseca o meno la sfera.



- Si calcola il punto  $P_0$  sulla retta che è più vicino a  $O$ . Se questo è interno alla sfera allora la retta la interseca, altrimenti no.
- Il segmento  $(P_0, O)$  deve essere ortogonale alla retta, dunque:

$$(P_0 - O) \cdot \mathbf{u} = 0$$

- Sviluppando questa equazione ponendo  $P_0 = Q + t_0\mathbf{u}$  e tenendo conto che  $\mathbf{u} \cdot \mathbf{u} = 1$ , si ottiene:

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

- Per controllare se il punto  $P_0$  è interno alla sfera basta verificare se  $\|P_0 - O\|^2 \leq r^2$ .

- Per calcolare i punti di intersezione serve la lunghezza  $a$ , che si ottiene semplicemente come  $a = \sqrt{r^2 - b^2}$  dove  $b = \|P_0 - O\|$ .
- i due punti in cui la retta interseca la sfera sono dunque:

$$P_1 = Q + (t_0 - a) \cdot \mathbf{u} \quad P_2 = Q + (t_0 + a) \cdot \mathbf{u}.$$

## Intersezione Retta-Piano

- L'eventuale punto di intersezione tra la retta  $P = Q + t\mathbf{v}$  ed il piano  $(O - P) \cdot \mathbf{u} = 0$  deve soddisfare l'equazione:

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

.

- Dunque si ottiene per  $t_0$  la seguente soluzione

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

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 dunque 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 (calcolata per esempio con il prodotto vettore).
- 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  $\mathcal{P}$  sia convesso. In tal caso un punto  $R$  giace all'interno di  $\mathcal{P}$  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  $P_{i+1} \forall i = 1 \dots n$ ).
- Si dimostra che il punto  $R$  giace a sinistra della retta orientata  $\overrightarrow{PQ}$  se solo se

$$\begin{vmatrix} R_x - P_x & Q_x - P_x \\ R_y - P_y & Q_y - P_y \end{vmatrix} < 0$$



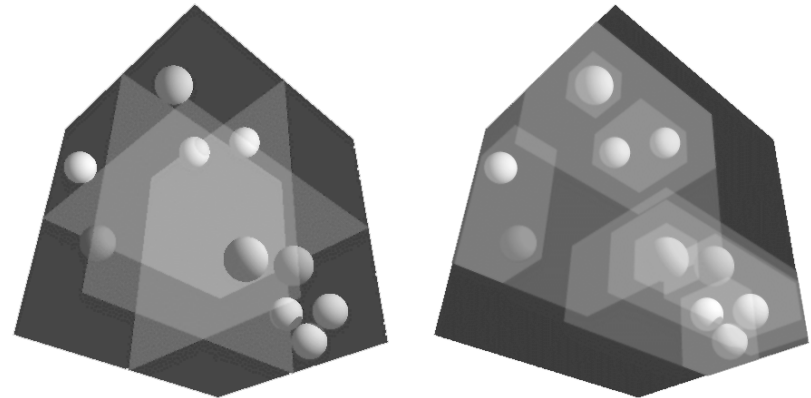
## Intersezione Retta-Poliedro

- Sia data la solita retta  $P = Q + t\mathbf{v}$  e siano dati  $n$  piani identificati dai punti  $O_n$  e dai vettori  $\mathbf{u}_n$ . Ci si chiede se e dove la retta interseca il poliedro convesso definito da tali piani.
- Un poliedro convesso è l'intersezione dei semispazi (negativi) delimitati dai piani contenenti le sue facce. Una retta (orientata) che interseca un piano crea una semiretta orientata che ha origine nel punto di intersezione e si estende illimitatamente nella direzione della retta.
- Ciascuna intersezione è classificata come entrante o uscente a seconda che la retta orientata entri nel semispazio negativo o ne esca, rispettivamente (entra se  $\mathbf{u} \cdot \mathbf{v} < 0$ ).
- È immediato verificare che, se  $t_i$  è il valore di  $t$  per cui il piano  $i$ -esimo interseca la retta, il segmento di retta contenuto all'interno del poliedro è delimitato dal massimo  $t_i$  entrante e dal minimo  $t_i$  uscente. Se i valori sono invertiti l'intersezione è nulla.

- Si procede dunque nel seguente modo:
  1. Si settano due valori  $t_0 = -\infty$  e  $t_1 = \infty$
  2. Per ogni piano  $i$  si cerca l'intersezione della retta con il piano
    - Se la retta non interseca il piano corrente (è parallela) e giace alla destra del piano, allora è esterna al poliedro, e si esce.
    - Se la retta interseca il piano corrente in entrata e  $t_i > t_0$  allora  $t_0 = t_i$
    - Se la retta interseca il piano corrente in uscita e  $t_i < t_1$  allora  $t_1 = t_i$
  3. Dopo aver iterato su tutti i piani l'intersezione è data dal segmento delimitato dai punti  $Q + t_0\mathbf{v}$  e  $Q + t_1\mathbf{v}$ , se  $t_1 > t_0$ . Altrimenti non si ha intersezione.
- Nel caso di intersezione con la semiretta, si considerano solo i valori di  $t_0 > 0$ .
- Nel caso di parallelepipedi le cose si semplificano, possiamo trattare contemporaneamente coppie di facce parallele.

## Tecniche di sfoltimento

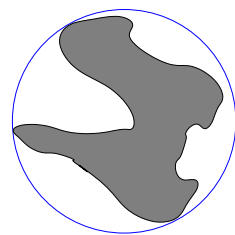
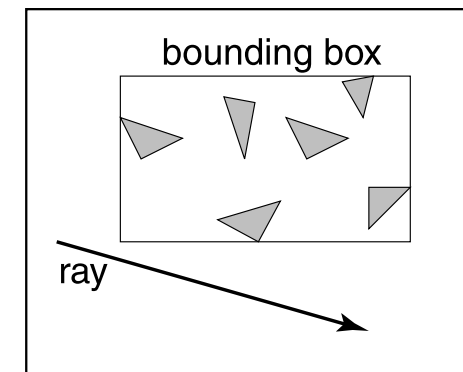
- Implementare un ray-tracer elementare è molto semplice. Basta controllare l'intersezione di ciascun raggio con ogni primitiva (triangolo). Questo ha un costo lineare nel numero delle primitive ed è praticabile solo per modelli semplici.
- Per raggiungere costo **sub-lineare** sono necessarie strutture 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)).
- Le principali tecniche che vedremo sono:
  - Volume di contenimento
  - Partizionamento spaziale
  - Volumi di contenimento gerarchici
  - Partizionamento spaziale gerarchico
- Tipicamente si considerano oggetti convessi. Se non lo sono, si spezzano in parti convesse.



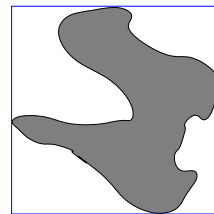
Hierarchical space partitioning e hierarchical bounding volumes

## Volumi di contenimento

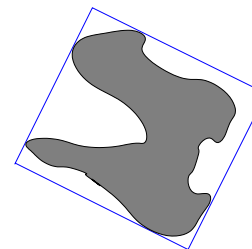
- Una prima tecnica per “sgrezzare” le intersezioni consiste nel racchiudere gli oggetti in solidi (“volumi”) che li contengono (bounding volume), con i quali sia facile testare l’intersezione: se non c’è intersezione con il bounding volume non c’è intersezione con l’oggetto racchiuso.
- Questo non rende 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



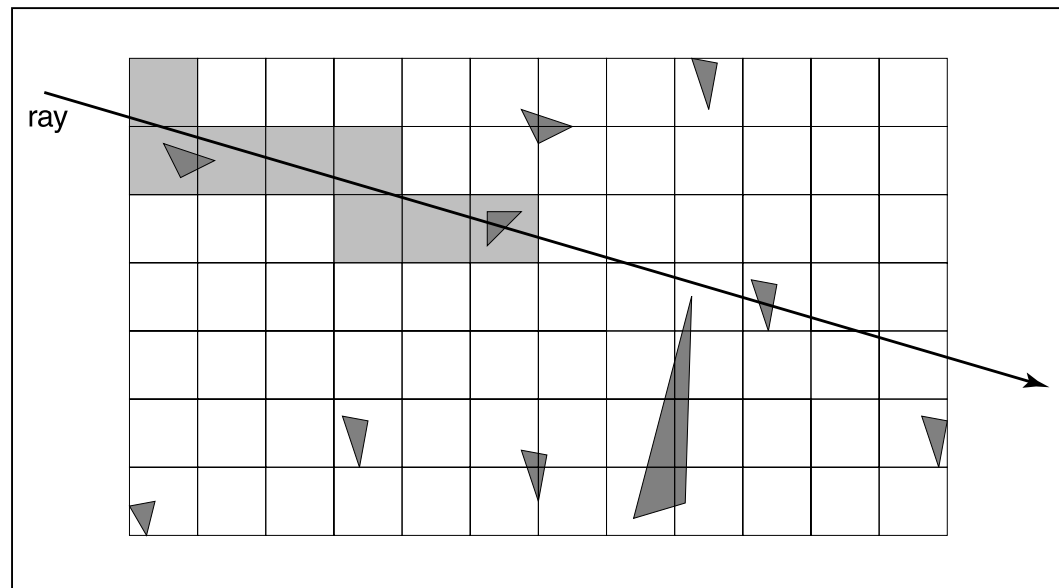
AABB



OBB

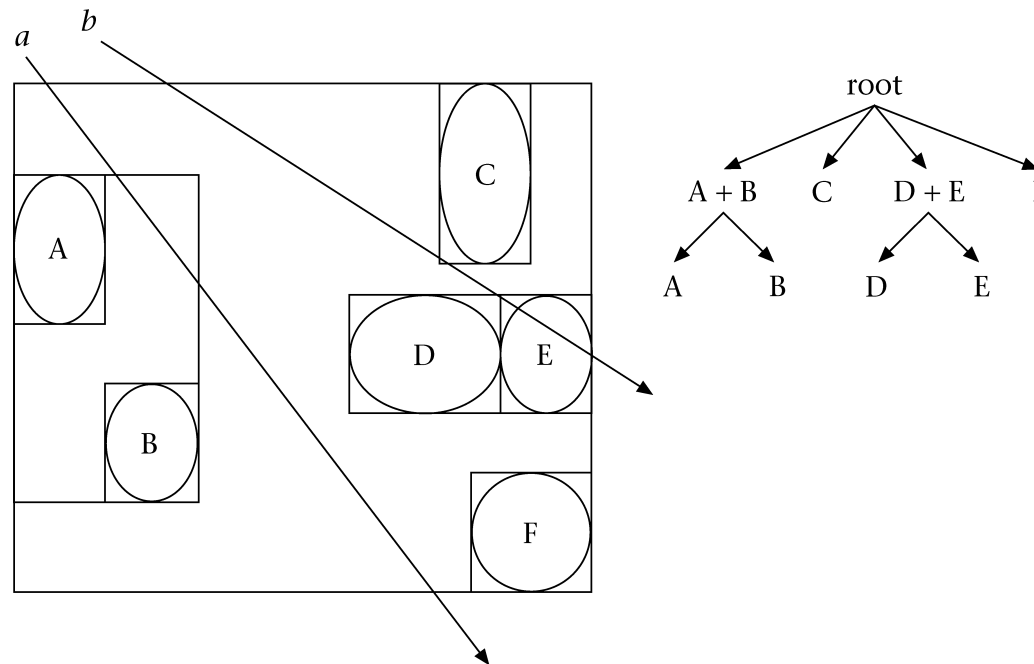
## Partizionamento spaziale

- 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 intersecate con il raggio.



## Volumi di contenimento gerarchici

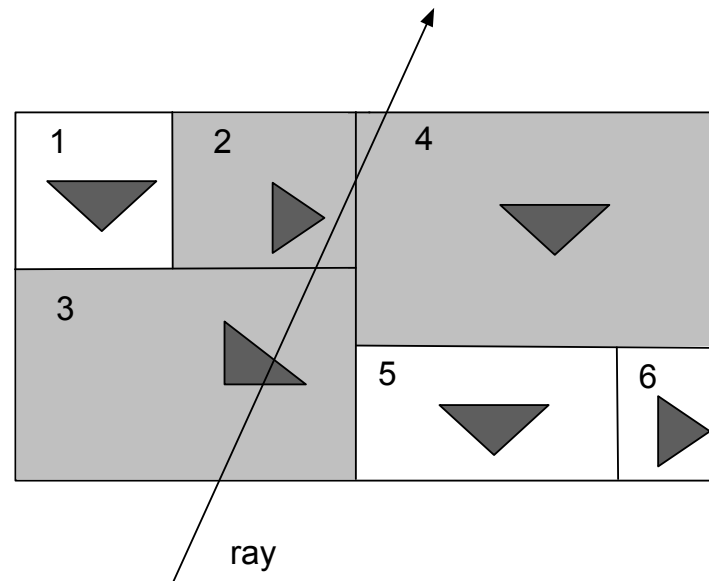
- Si costruisce una gerarchia di volumi di contenimento, dove al livello più alto si ha un volume che racchiude tutta la scene, ed al livello più basso si hanno volumi di contenimento per i singoli oggetti (o per un numero prefissato di primitive).
- È una struttura organizzata bottom-up.
- È stata introdotta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



## Partizionamento spaziale gerarchico

- La tecnica sfrutta una partizione gerarchica dello spazio in regioni, come quelle prodotte da Octree, k-d tree e BSP tree.
- È una struttura organizzata top-down.
- 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.

- Si usano anche **kd-tree** con dimensione di taglio alternata.
- 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 dal BSP. 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.







©Alan Watt

# Radiosity

- **Radiosity** (Goral et al. 1984) è un algoritmo globale che è un po' il complementare di 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.
- È 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.

## Equazione della radiosità.

- Il Radiosity si basa sulla **equazione della radiosità**, che è la versione semplificata dell'equazione del rendering nel caso di superfici lambertiane.
- Grazie all'assunzione di superfici lambertiane:
  - la BRDF è costante sull dominio angolare:  $\rho(x) = \frac{\rho_d(x)}{\pi}$
  - la radianza non dipende dalla direzione, e coincide con la radiosità (a meno di  $\pi$ ):  

$$B(x) = L(x) \int_{\Omega} \cos \theta d\omega = \pi L(x)$$
- Riscriviamo dunque l'equazione del rendering (molt. per  $\pi$ ):

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

Otteniamo così l'**equazione della radiosità**:

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

## Fattore di forma

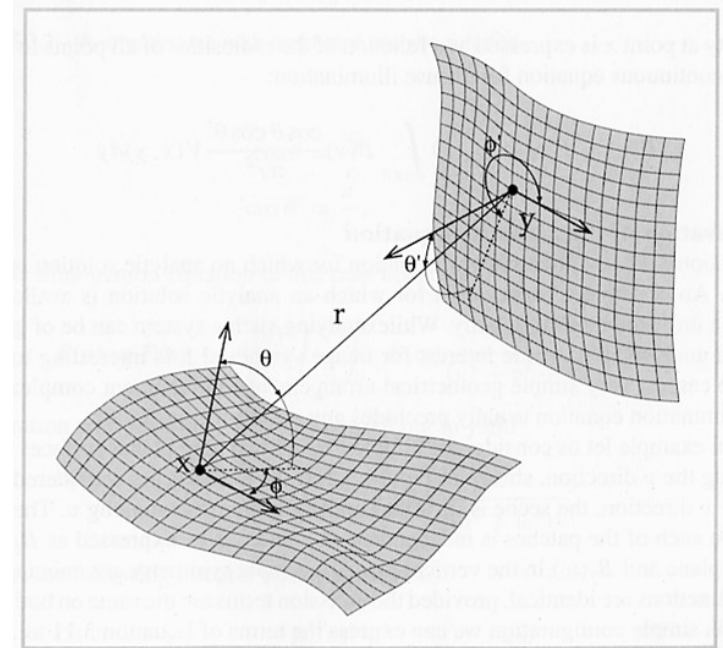
- Il termine

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

prende il nome di **fattore di forma** e dipende solo dalla geometria della scena.

- ricordiamo che:

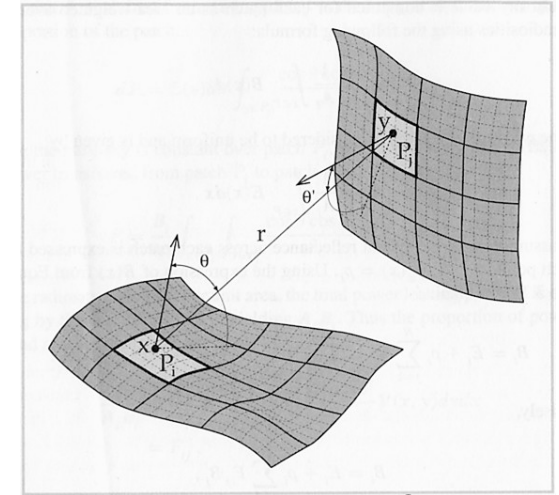
- $V(x, y)$  che vale 1 sse  $x$  è visibile da  $y$ ,
- $\theta_{xy}$  (risp.  $\theta_{yx}$ ) è l'angolo formato dal raggio che congiunge  $x$  ed  $y$  con la normale in  $x$  (risp. in  $y$ ).



©U. Wisconsin

## Equazione della radiosità discretizzata

- Approccio agli 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  $\rho_d$  sono costanti (un valore per patch).



©U. Wisconsin

- 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 sulle patch:

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

- Integriamo entrambe i membri:

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

- Usiamo le definizioni:

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

- Usiamo la costanza di  $B_j$ , e  $\rho_i$  sulla patch, ottenendo la **versione discreta della equazione della radiosità**:

$$B_i = E_i + \rho_i \sum_{j=1}^N B_j F_{ij} \quad (3)$$

- In forma matriciale:

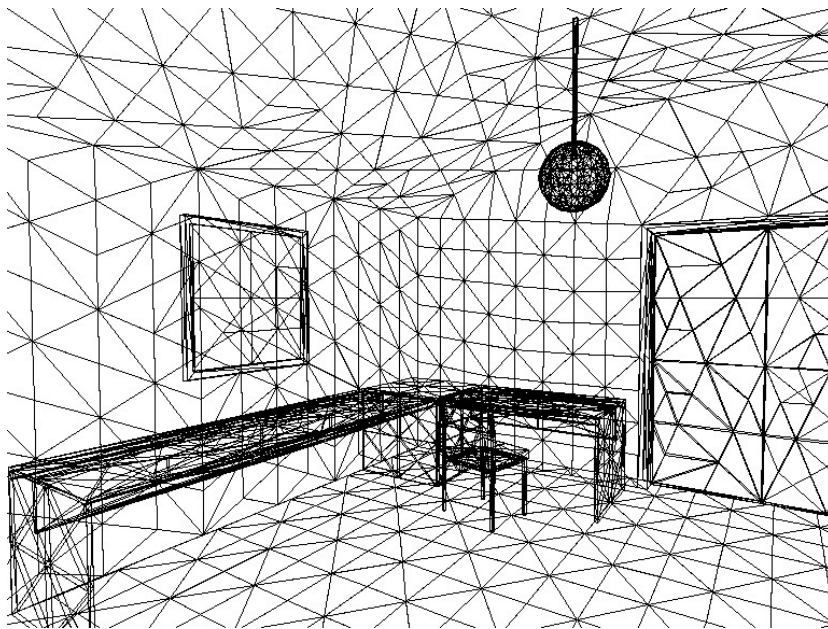
$$\begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_N \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix} + \begin{bmatrix} \rho_1 F_{11} & \rho_1 F_{12} & \cdots & \rho_1 F_{1N} \\ \rho_2 F_{21} & \rho_2 F_{22} & \cdots & \rho_2 F_{2N} \\ \vdots & \ddots & \vdots & \\ \rho_N F_{N1} & \rho_N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

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

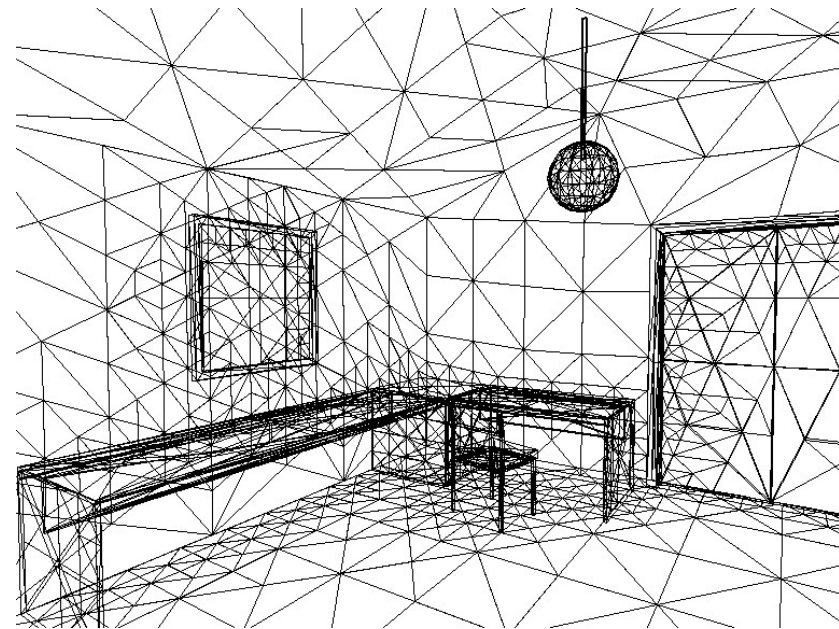
- È un sistema lineare  $N \times N$  denso e di grandi dimensioni (i patch sono solitamente di più dei poligoni).
- La risoluzione fornisce la radiosità per tutti i patches, view-independent.
- Come risolverlo? La soluzione standard è applicare il **rilassamento di Gauss-Seidel**. È un metodo iterativo che ad ogni passo aggiorna la radiosità di un singolo patch  $i$  impiegando la radiosità di tutti gli altri. È come applicare iterativamente la formula (3).

# 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.



(30) Mesh iniziale



(31) Raffiamento adattativo

©Alan Watt

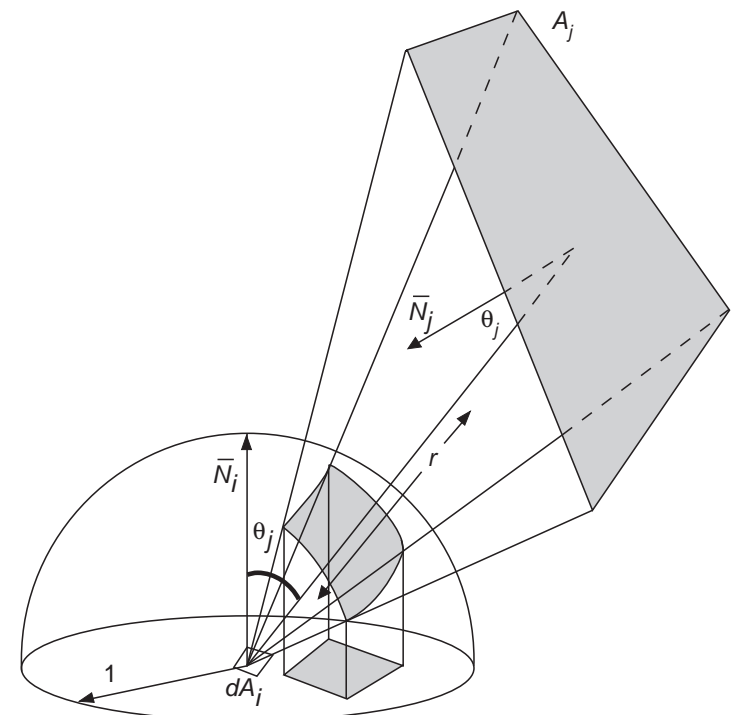


## Calcolo del fattore di forma

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

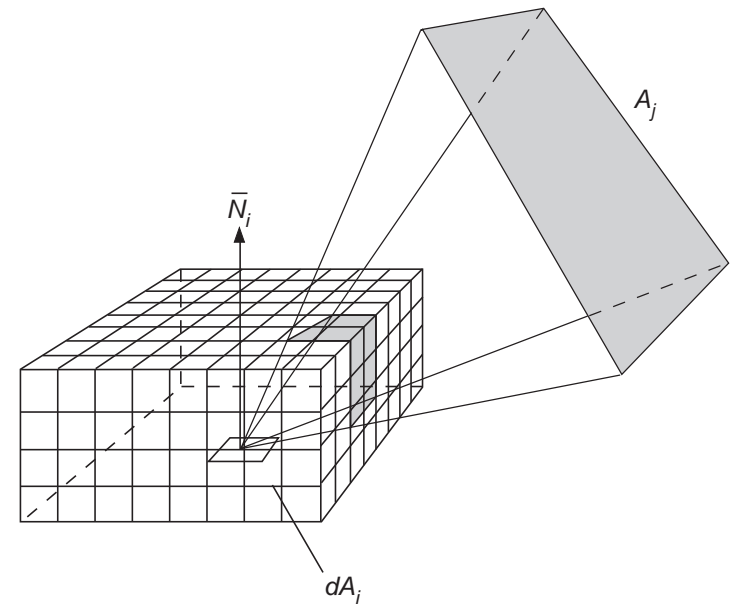
$$F_{ij} = \frac{1}{A_i} \int_{x \in P_i} \underbrace{\left( \int_{y \in P_j} F(x, y) dy \right)}_{F'_{ij}} dx$$

- $F'_{ij}$  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  $x$ , e poi sulla base della semisfera. Il rapporto tra l'area di questa proiezione e l'area della base della semisfera è il fattore di forma  $F'_{ij}$ .
- Se assumiamo  $F'_{ij}$  costante sulla patch, basta calcolarlo in un punto e si ha  $F_{ij} = F'_{ij}$ .



©Foley et .al

- Il **metodo del semicubo** (hemicube) è una 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 una pipeline di rasterizzazione (v. più avanti).
- Il form factor  $F'_{ij}$  viene approssimato come somma dei delta form factor delle celle del semicubo interessate dalla proiezione di  $P_j$ .



©Foley et .al



©Alan Watt



(32) ray-casting con Phong



(33) ray-tracing

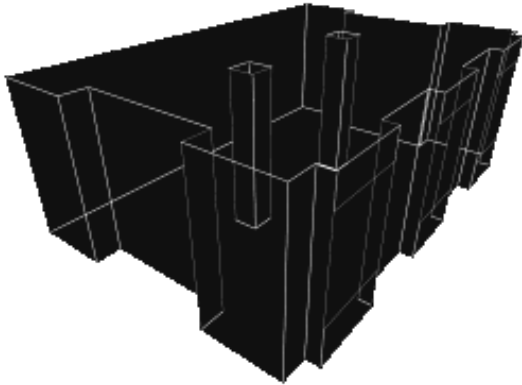


(34) radiosity + ray-tracing

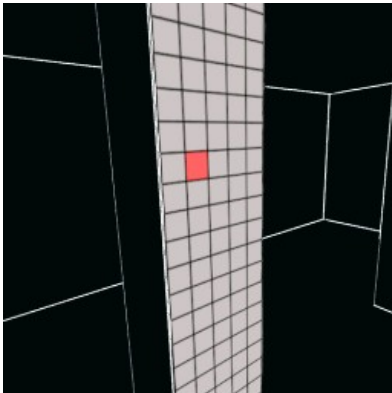


(35) RADIANCE (Monte Carlo)

## Un esempio di Radiosity

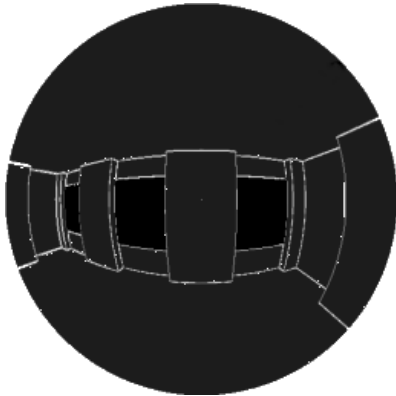


Una scena: stanza con colonne e finestre.

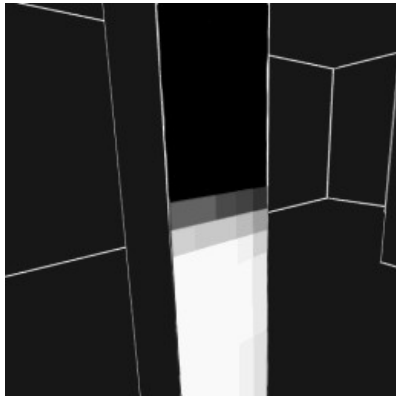


Consideriamo una particolare patch (in rosso) sulla colonna

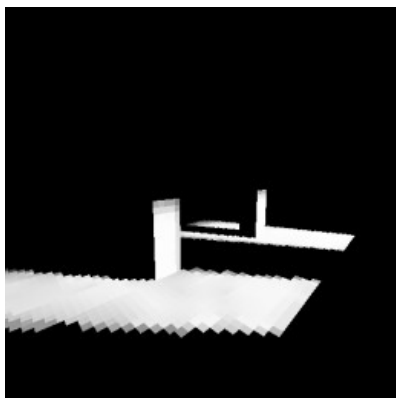
Immagini tratte da <http://freespace.virgin.net/hugo.elias/radiosity/radiosity.htm>



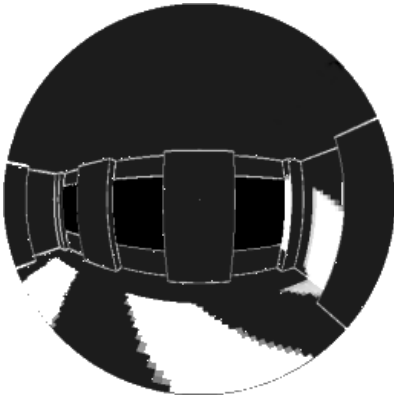
Questo è ciò che vede la patch: nessuna sorgente luminosa. Questa vista emisferica – tra l'altro – è quella che serve a calcolare il fattore di forma con il metodo di Nusselt. È stata creata solo per illustrazione, comunque, visto che il fattore di forma viene invece calcolato con il metodo del semicubo, più pratico.



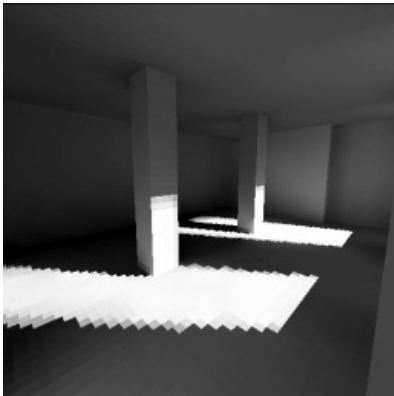
Dopo la prima passata la colonna si presenta così: le patch più in basso vedono la luce (fuori dalla finestra) e vengono illuminate, mentre la nostra è al buio (nera).



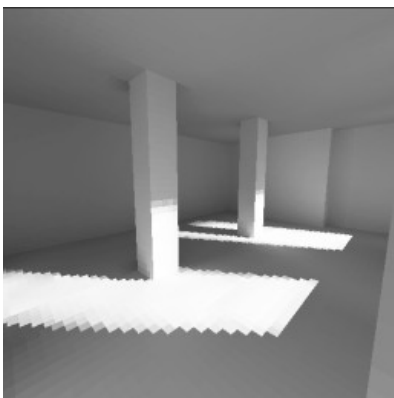
Così è la scena dopo la prima passata: solo le patch che vedono direttamente la luce sono illuminate.



Questo è quello che vede la nostra patch adesso: ci sono altre patch illuminate in vista, quindi adesso è raggiunta dalla luce diffusa dalle altre patch ...



... infatti dopo la seconda passata non è piu' completamente nera, come pure il resto della colonna.



Questa è la scena dopo la 16<sup>a</sup> passata.