



Rendering globale



-
- Il modello di rendering che abbiamo finora studiato è detto locale.
 - Ogni primitiva è trattata indipendentemente dalle altre.
 - Il concetto di scena, la presenza di altri oggetti nella scena, interessano solo il programmatore e non il renderer.
-



Vantaggi del rendering locale

- Il modo con cui viene visualizzato un poligono dipende solo dalle caratteristiche del poligono e del raggio di luce che lo colpisce direttamente.
 - Questo tipo di modello ha molti vantaggi:
 - Semplicità
 - Parallelismo a livello di primitiva
 - Costo costante per primitiva.
-



Limiti del rendering locale.

- Nel mondo reale:
 - Se un oggetto blocca la luce proveniente da una sorgente, gli oggetti al di là di esso restano **in ombra**.
 - **Se un oggetto è riflettente, la luce che si riflette da esso illumina gli altri oggetti.**
 - Un modello di illuminazione locale non può produrre nessuno di questi effetti.
-



-
- Il modello di illuminazione di Phong, anche se accoppiato con la tecnica di shading di Phong, non permette la rappresentazione visiva corretta di alcuni fenomeni:
 - Luci ed osservatori posizionati all'interno della scena.
 - Valutazione delle componenti emissive e trasmissive della radiazione luminosa.
 - Nel modello di Illuminazione di Phong, l'illuminazione globale è stata modellata con un termine di illuminazione ambientale, costante per tutti i punti su ogni oggetto.
 - Non dipende dalla posizione dell'oggetto o dell'osservatore, o dalla presenza od assenza di oggetti vicini che potrebbero bloccare la luce ambientale.
-



- Non permette di modellare:
 - La **riflessione diffusiva** tra gli oggetti nella scena (inter-riflessione)
 - La **riflessione speculare** di radiazioni tra gli oggetti (ad esempio la simulazione della riflessione su uno specchio, cioè su una superficie la cui componente diffusiva è pressochè nulla)
 - La **gestione delle ombre**, ossia dell'individuazione delle parti di superficie di oggetti della scena che non ricevono radiazione luminosa da una o più sorgenti, e quindi del calcolo corretto dell'illuminazione.



- Molta della luce in ambienti del mondo reale non proviene da sorgenti luminose dirette.
- Esistono due classi differenti di algoritmi per generare immagini che enfatizzano il contributo della illuminazione globale.



Ray tracing

- Che mette assieme eliminazione delle superfici nascoste e shading per gestire le ombre, la riflessione e la rifrazione.
 - Così la riflessione e la trasmissione speculare globale completano l'illuminazione locale di tipo ambientale, speculare, diffusa.
-



Radiosity

- Questo metodo separa la fase di shading dalla fase di eliminazione delle superfici nascoste.
 - **Modella tutte le interazioni dell'ambiente con le sorgenti luminose prima in una fase indipendente dalla vista**, poi calcola una o più immagini dal punto di vista desiderato usando uno tra i classici algoritmi di eliminazione delle superfici visibili e di shading interpolativo.
-



Distinzione tra Ray-tracing e Radiosity

- Ray tracing è un algoritmo **view-dependent**.
- Radiosity è un algoritmo **view-independent**.



-
- Gli algoritmi **view-dependent** discretizzano il piano di vista per determinare i punti in cui valutare l'equazione di illuminazione, fissata la direzione di vista.
 - Gli algoritmi **view-independent** discretizzano l'ambiente e lo elaborano allo scopo di fornire abbastanza informazione per valutare l'equazione dell'illuminazione in ogni punto e da ogni direzione di vista.
-



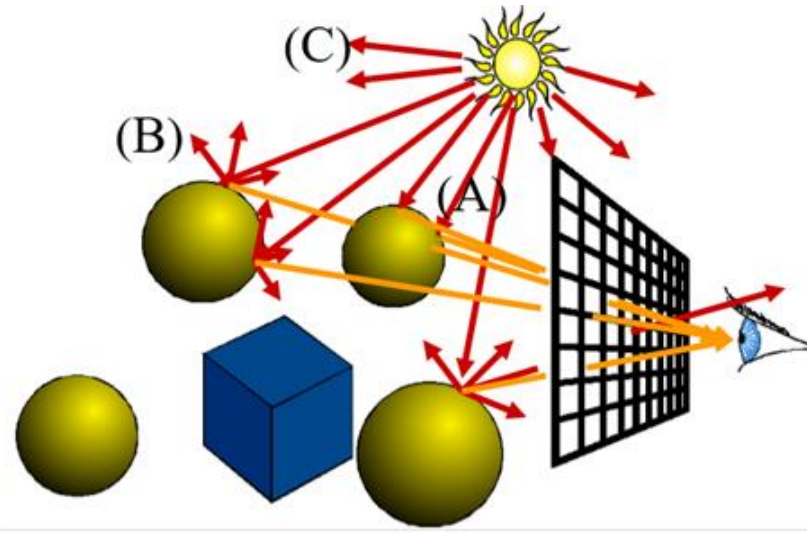
-
- Gli algoritmi **view-dependent** sono adatti per trattare **fenomeni speculari** che sono altamente dipendenti dalla posizione dell'osservatore, ma realizzano lavoro extra quando modellano fenomeni diffusivi, che cambiano poco su grandi aree dell'immagine.
 - Al contrario, gli algoritmi **view-dependent** modellano efficientemente **il fenomeno diffusivo**, ma richiedono eccessive quantità di memoria per catturare abbastanza informazione riguardo ai fenomeni speculari.
-



Ray Tracing

- L'algoritmo del Ray casting per l'eliminazione delle superfici nascoste è stato esteso per trattare le ombre, le riflessioni, le rifrazioni.
- Il metodo del Ray tracing è basato sull'osservazione che,

di tutti i raggi luminosi che lasciano una sorgente, i soli che contribuiscono l'immagine sono quelli che raggiungono l'osservatore.



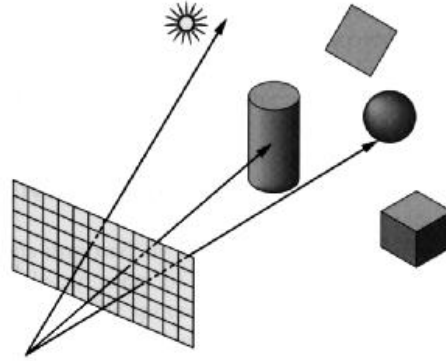
- (A) Raggio che raggiunge l'osservatore
- (B) Raggio assorbito dal materiale
- (C) Raggio che non raggiunge la scena



-
- I raggi luminosi possono raggiungere l'osservatore sia direttamente, sia per effetto delle interazioni con le altre superfici.
 - La maggior parte dei raggi che lascerà la sorgente non raggiungerà l'osservatore e dunque non contribuirà all'immagine.
-

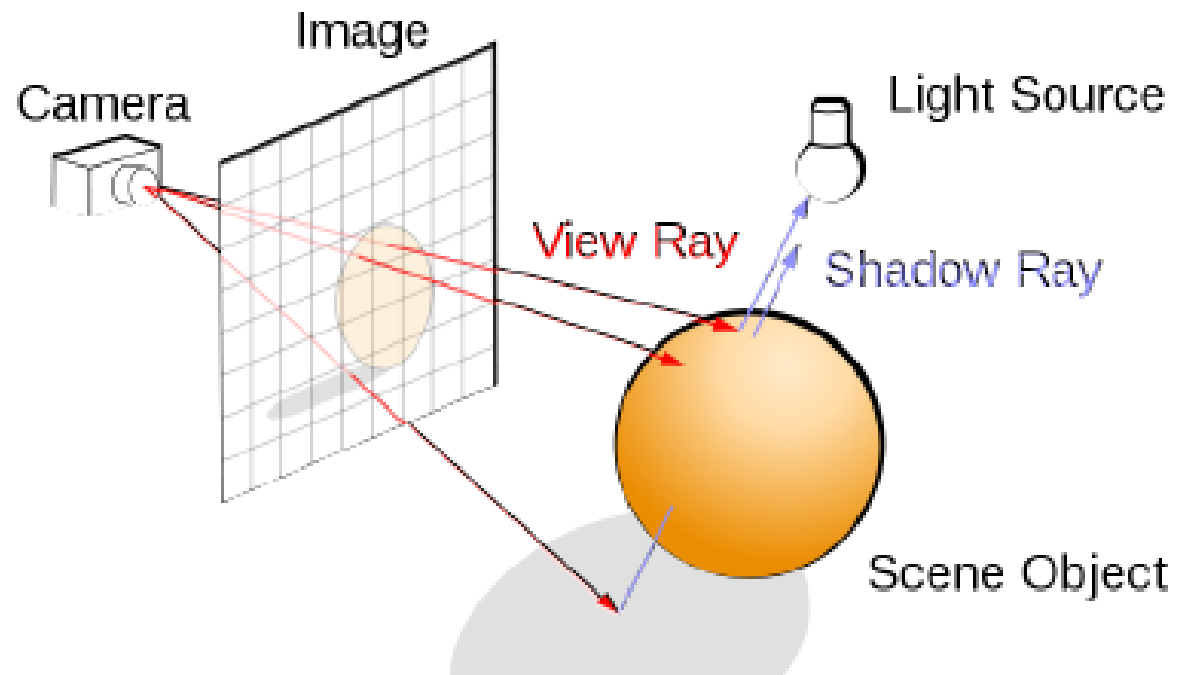


- Non è possibile seguire la traiettoria di ogni raggio.
- E' possibile determinare i raggi che contribuiscono all'immagine, se invertiamo la traiettoria dei raggi e consideriamo solo quelli che escono dall'osservatore.
- Questa è l'idea alla base del metodo Ray-tracing, che **simula all'indietro il cammino compiuto dalla radiazione luminosa per giungere all'osservatore.**
- Il processo di lanciare i raggi dall'occhio verso la sorgente di luce, per disegnare un'immagine, viene chiamato a volte **backwards ray tracing** (o "ray tracing inverso"), dal momento che i fotoni viaggiano in senso opposto a quello usuale
- Poiché si deve assegnare un colore ad ogni pixel, si deve considerare almeno un raggio luminoso per ogni pixel.
- Questo raggio è detto **raggio primario.**



- Ciascun raggio primario può intersecare o una superficie, o una sorgente luminosa, o può andare all'infinito senza intersezioni.
 - In questo caso verrà assegnato il colore dello sfondo.
 - Per determinare la gradazione di colore (*shading*) del punto di intersezione del raggio con una superficie è necessario applicare un modello di illuminazione e di shading scelto tra quelli descritti precedentemente.
-

Data la posizione dell'osservatore, a ogni pixel del piano immagine corrisponderà un raggio che possiamo seguire a ritroso dall'osservatore alla scena, detto **raggio primario** o **view ray**





-
- Avendo a disposizione tutta la scena, per calcolare lo shading di un punto, ci possiamo guardare intorno.
 - Ciò significa che con il ray tracing possiamo calcolare **ombre portate, riflessione diretta, trasparenza e rifrazione.**
-



RAY CASTING

versione semplificata del ray tracing

non considera il contributo luminoso riflesso dovuto alla presenza di altri oggetti in scena

tuttavia gestisce sia l'illuminazione diretta che la visibilità delle superfici e le ombre.



RAY CASTING

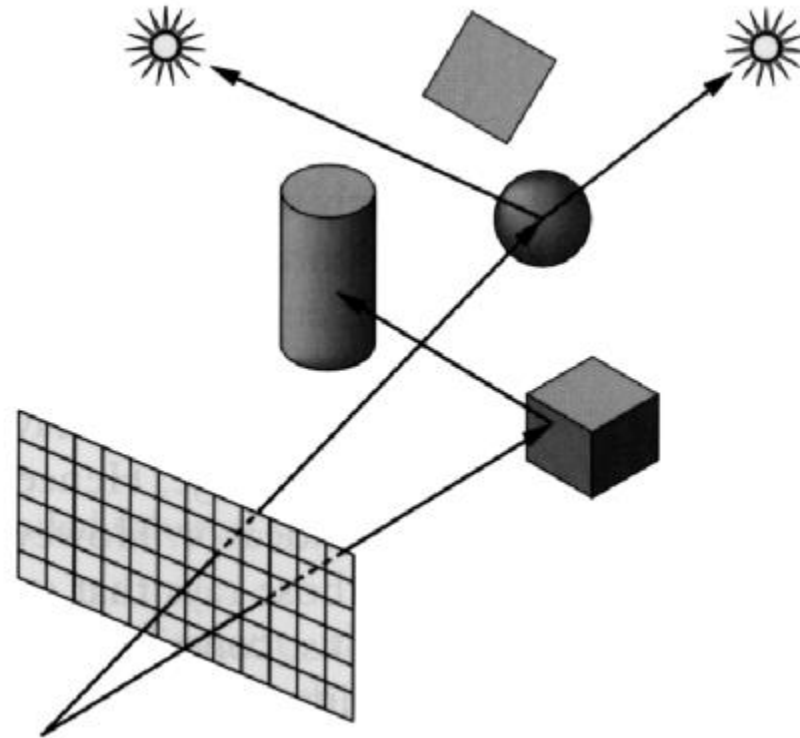
L'algoritmo utilizza il modello di riflessione locale di Phong per il calcolo di illuminazione diretta dei punti visibili dall'osservatore (primi punti di intersezione).

Per ogni pixel (x,y) della viewport, l'algoritmo costruisce un raggio dal punto di vista in scena (ray) e assegna al pixel il colore del primo oggetto intersecato

color[x,y] = castRay(ray),



- Sia **p** il primo punto di intersezione (oggetto) che il raggio incontra.
- L'intensità e il colore di questo punto **p** è calcolata come contributo di tutte le sorgenti luminose che incidono su **p**.
- Tracciando da **p** un raggio (detto raggio **shadow**) per ogni sorgente luminosa:
 - se un raggio **shadow interseca una superficie prima di incontrare la risorsa luminosa**, la luce non raggiunge il punto **p** considerato che quindi non avrà il contributo di quella risorsa luminosa. Di conseguenza, se nessuna risorsa luminosa è raggiungibile dal punto **p** allora **il punto è in ombra**;
 - se almeno **un raggio shadow non incontra occlusioni**: il punto **p** risulterà **illuminato**.
- L'intensità (colore) del pixel viene calcolata dall'applicazione del modello di **illuminazione di Phong nel punto intersezione p sommando il contributo di ogni sorgente luminosa visibile da p**.
- Il raycasting opera un'illuminazione diretta e in più rimuove superfici non visibili e aggiunge ombre, al costo dei calcoli di intersezione oggetto/risorse luminosa.





$$I_r = I_a k_a + \sum_{i=1}^m S_i f_{att_i} I_{p_i} \left(k_d (N \cdot L_i) + k_s (R_i \cdot V)^n \right)$$

dove S_i è 1 se il punto non è in ombra rispetto alla sorgente luminosa i -esima, è uguale a 0 se il punto è in ombra rispetto alla sorgente luminosa i -esima



col castRay(ray) {

hitObject(ray, hitpoint, n, object); // **calcola il punto di impatto del raggio che parte dall'osservatore con l'oggetto.**

col = object color;

For every light L // **per ogni luce**

Construct rayS(hitPoint, L->getDir()); // **costruisce il raggio rayS che parte dal punto di impatto nella direzione della luce**

For every object ob

Per ogni oggetto nell'ambiente:

hitObject(rayS, hit2, ob);

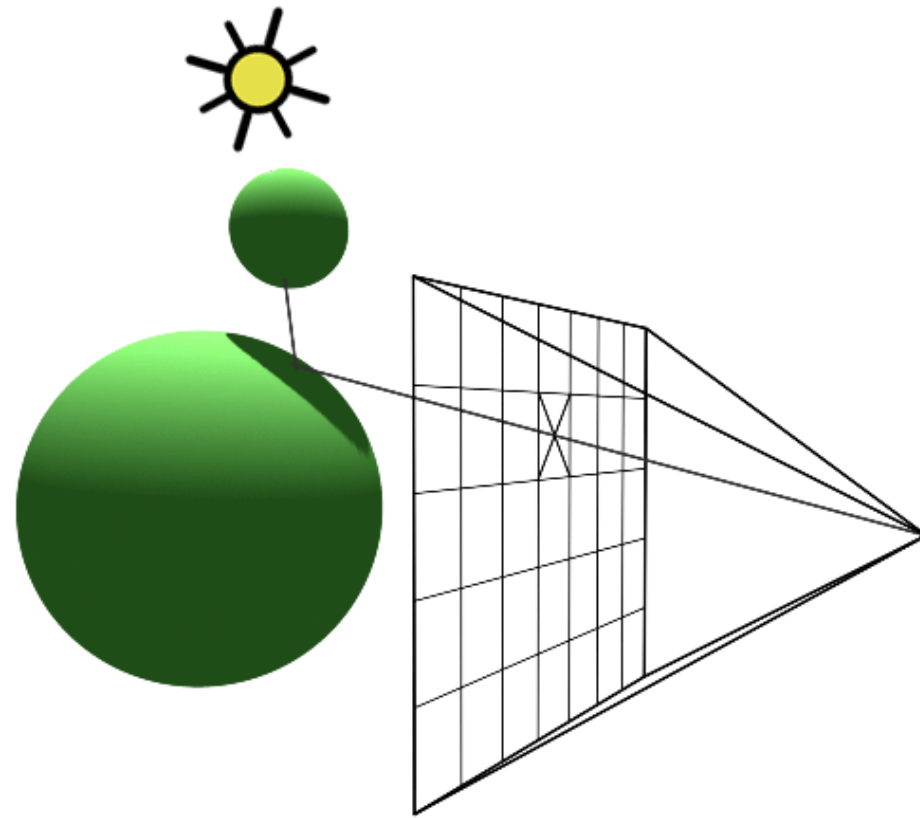
Viene verificato se il raggio rayS interseca l'oggetto chiamando la funzione hitObject(rayS, hit2, ob).

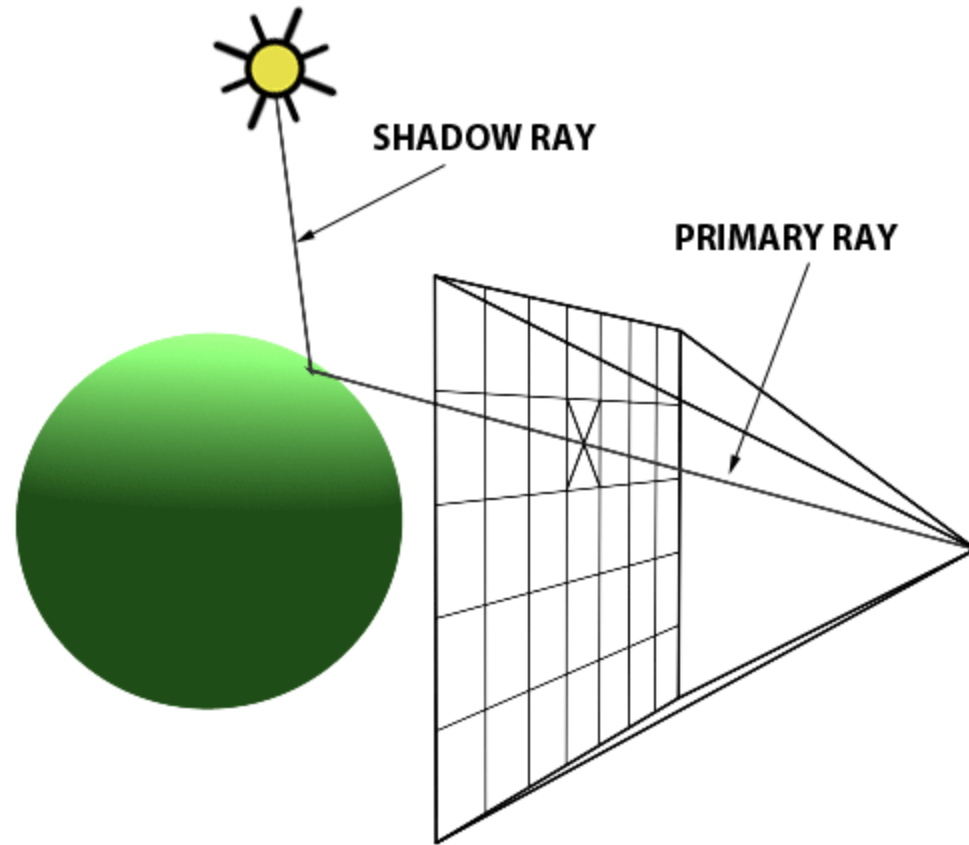
If (rayS does NOT intersect any ob)

Se il raggio rayS non interseca alcun oggetto, significa che il punto è illuminato dalla sorgente di luce, e quindi il colore (col) dell'oggetto viene modificato aggiungendo il contributo della luce. Questo contributo è calcolato chiamando la funzione lighting(L, hitpoint, n).

col=col+lighting(L, hitpoint, n)

Return col; }





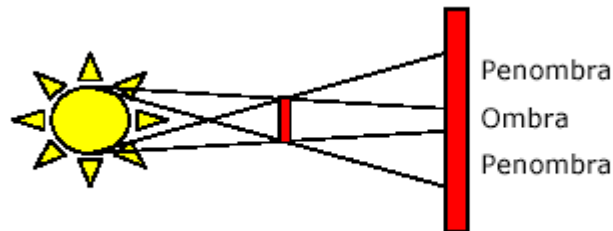


-
- Se **le superfici sono tutte opache** e non si considerano gli effetti della luce diffusa da superficie a superficie, **avremo un'immagine che, oltre all'illuminazione, presenta anche delle ombre.**
 - Il prezzo che si deve pagare per l'introduzione delle ombre è quello di dover sparare un raggio ombra per ogni punto della superficie che abbiamo determinato essere visibile e per ogni luce.
-

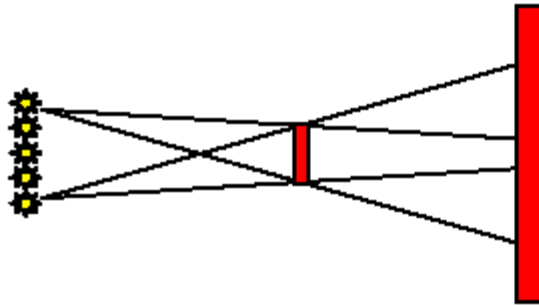


Limiti

- **Ombre nette**: il meccanismo di raggio ombra assume che la sorgente di luce sia puntiforme, cosa inesistente in natura: assenza di penombra.



- Si possono modellare sorgenti di luce non puntiformi utilizzando molte luci.
- Il numero di raggi ombra aumenta e quindi aumenta anche il tempo di rendering.





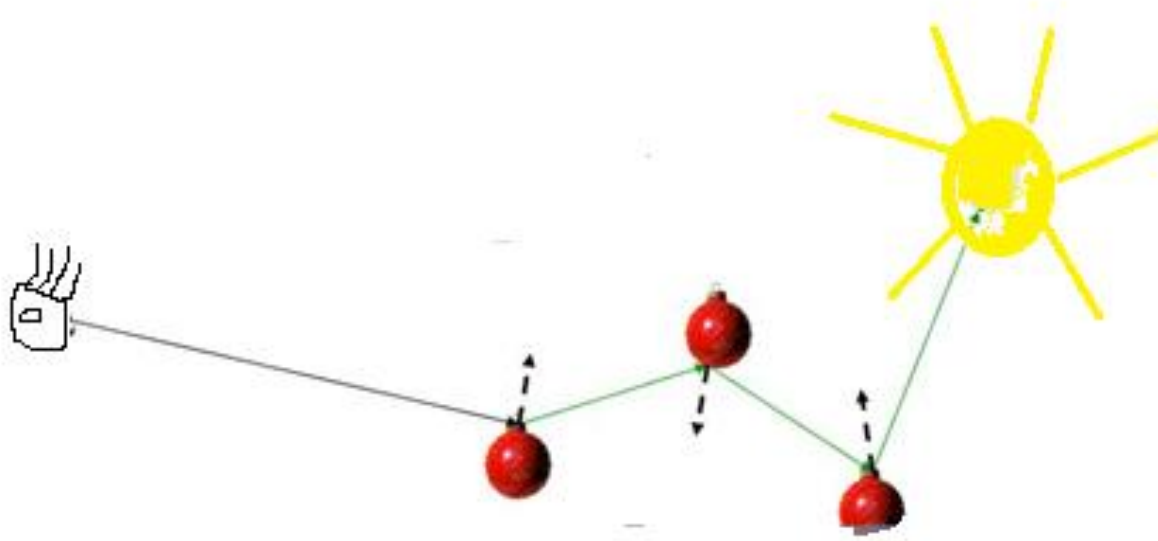
Gestione raggi riflessi

- Per ora si sono considerati raggi luminosi che hanno origine dalla risorsa luminosa colpiscono una superficie e si riflettono verso l'osservatore.

In realtà la luce compie percorsi molto più articolati, rimbalzando in scena molte volte prima di raggiungere l'osservatore grazie ad oggetti **riflettenti** presenti, e **apportando così un contributo luminoso su tutti gli oggetti che vengono incontrati nel percorso dei raggi** (inter-riflessioni).

Per tener conto almeno parzialmente di questo effetto, **ad ogni intersezione raggio-oggetto viene generato un nuovo raggio (raggio riflesso)** nella direzione della riflessione speculare perfetta.

I raggi **riflessi 'rimbalzano' da superficie a superficie contribuendo all'intensità luminosa di ciascuna superficie che intersecano**, finchè intersecano una risorsa luminosa o raggiungono un massimo numero di 'rimbalzi'.



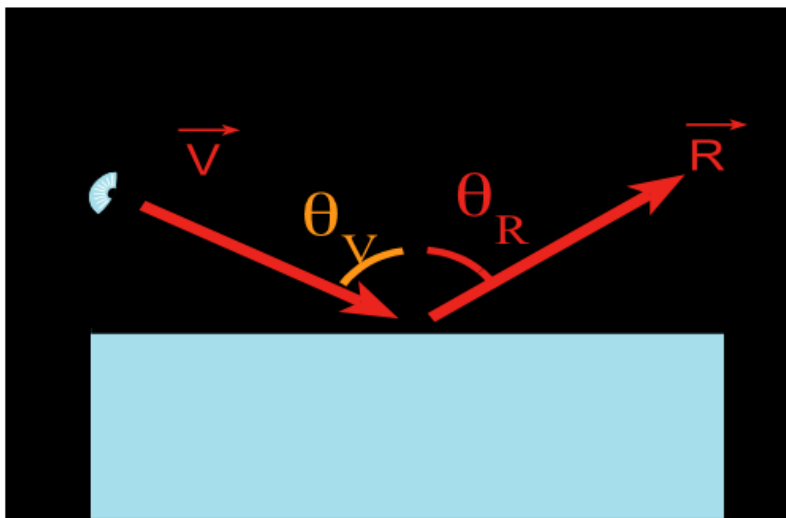


Questi calcoli sono fatti ricorsivamente e devono tener conto dell' attenuazione dovuta alla distanza, cioè dell'assorbimento della luce da parte delle superfici.

La direzione di riflessione è gestita esattamente come se fosse il raggio dall'osservatore (raggio primario).

Stiamo seguendo un raggio la cui direzione **r** è determinata a partire dal vettore **v** (che proviene o direttamente dall'osservatore o in generale da un qualunque altro punto di intersezione con altri oggetti) e dalla normale **n** alla superficie nel punto.

Calcoliamo il vettore di vista riflesso rispetto alla normale con lo stesso procedimento con cui avevamo calcolato la direzione della luce riflessa rispetto alla normale nel modello di Phong



$$R = V - 2 (V \cdot N) N$$

Notiamo che V punta verso l'oggetto



Ray tracing per gestire la riflessione speculare

1. Lancio di un raggio primario:

Inizia con il lancio di un raggio primario dalla telecamera attraverso ogni pixel dell'immagine verso la scena.

2. Intersezione con gli oggetti:

Il raggio primario interseca la scena con gli oggetti. Se c'è un'intersezione, si determina il punto di intersezione e la normale all'oggetto nel punto di impatto.

3. Calcolo della riflessione speculare:

- Se l'oggetto colpito ha proprietà di riflessione speculare (come un materiale lucido o uno specchio), viene **generato un raggio riflesso**. Questo raggio è calcolato utilizzando la legge della riflessione, che prevede che l'angolo di incidenza sia uguale all'angolo di riflessione rispetto alla normale.
- Il raggio riflesso parte dal punto di intersezione e viene lanciato in direzione riflessa rispetto alla normale.

4. Raggio riflesso e nuove intersezioni:

- Il raggio riflesso diventa un **nuovo raggio primario**, e il processo di intersezione viene ripetuto. Questo significa che il raggio può colpire nuovi oggetti nella scena.
- Ogni volta che il raggio colpisce un oggetto, se quest'ultimo è speculare, viene generato un nuovo raggio riflesso e il processo continua fino a quando si raggiunge una profondità massima di riflessione o il raggio non colpisce più alcun oggetto.

5. Calcolo del colore finale:

Il colore del pixel viene calcolato combinando i contributi dai raggi primari e i raggi riflessi. Questo può includere la combinazione di colori degli oggetti colpiti e il calcolo dei contributi di illuminazione.



Ray tracing e superfici riflesse:limiti

- Il calcolo degli effetti di riflessione ha un costo (si spara un altro raggio) e un limite:
 - Modella accuratamente solamente superfici perfettamente lisce parzialmente riflettenti.
 - Nella realtà la maggior parte delle superfici non sono perfettamente lisce, ma riflettono in un insieme di direzioni (il riflesso è sfocato).
 - Si può modellare sparando molti raggi (è molto costoso).
-



Gestione Rifrazione

Il ray tracing è in grado di gestire sia superfici riflettenti che superfici che lasciano passare parzialmente o totalmente la luce.

La luce che colpisce un oggetto trasparente/ semitrasparente viene in parte trasmessa (i raggi sono rifratti).

Quando i raggi luminosi passano da un mezzo "trasparente" a un altro, cambiano direzione, subiscono una deviazione della loro traiettoria



Quando la luce viaggia nel vuoto, viaggia alla **velocità della luce**, indicata con la costante **c** (circa 3×10^8 m/s).

Ma quando viaggia attraverso un altro mezzo, la sua velocità diminuisce.

Se denotiamo la velocità della luce in questo mezzo con **v** , **l'indice di rifrazione (IOR) è il rapporto di c e v**

$$\eta = \frac{c}{v}$$

La luce viaggia più velocemente nell'acqua che nel vetro, ma più lentamente che nell'aria (l'aria ha un indice di rifrazione molto vicino a 1 e in CG trattiamo quasi sempre l'aria come se fosse un vuoto).

$$\text{Se } v_{\text{acqua}} > v_{\text{vetro}} \quad \frac{1}{v_{\text{acqua}}} < \frac{1}{v_{\text{vetro}}} \quad \frac{c}{v_{\text{acqua}}} < \frac{c}{v_{\text{vetro}}} \quad \eta_{\text{acqua}} < \eta_{\text{vetro}}$$

L'indice di rifrazione dell'acqua è 1.3, l'indice di rifrazione del vetro è 1.5

Mezzo o sostanza	Indice di rifrazione	Velocità di propagazione
vuoto	1	$3 \cdot 10^8$ m/s
aria	1,00029	$2,999 \cdot 10^8$ m/s
acqua	1,33	$2,26 \cdot 10^8$ m/s
ghiaccio	1,31	$2,29 \cdot 10^8$ m/s
sale	1,54	$1,95 \cdot 10^8$ m/s
alcool	1,36	$2,2 \cdot 10^8$ m/s
vetro (Crown)	1,5	$2 \cdot 10^8$ m/s
vetro (Flint)	1,65	$1,82 \cdot 10^8$ m/s
solfuro di carbonio	1,63	$1,84 \cdot 10^8$ m/s
sodio liquido	4,22	$0,7 \cdot 10^8$ m/s
arseniuro di gallio	3,6	$0,83 \cdot 10^8$ m/s
silicio	3,4	$0,88 \cdot 10^8$ m/s
diamante	2,417	$1,24 \cdot 10^8$ m/s
quarzo	1,51	$1,98 \cdot 10^8$ m/s

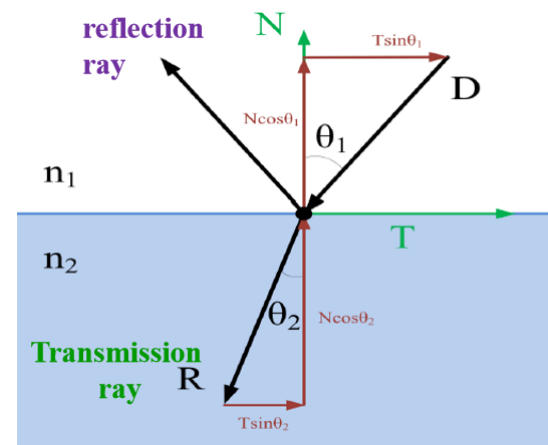


La rifrazione è gestita dalla legge di Snell

D – direzione del raggio incidente
N – normale unitaria alla superficie
R – direzione raggio rifratto
T – tangente unitaria alla superficie
(nel piano N e D)

Due mezzi trasmissivi con indice di rifrazione η_1 (in alto) e η_2 (in basso) in contatto tra loro attraverso una superficie, che viene chiamata interfaccia (linea orizzontale in figura)

Il raggio incidente, il raggio rifratto e la normale alla superficie di separazione dei due mezzi, nel punto di incidenza, giacciono sullo stesso piano.



La **legge di Snell** descrive le modalità di [rifrazione](#) di un raggio luminoso nella transizione tra due mezzi con [indice di rifrazione](#) diverso.

$$\eta_1 \sin(\theta_1) = \eta_2 \sin(\theta_2)$$

$$\sin(\theta_1) = \frac{\eta_2}{\eta_1} \sin(\theta_2)$$

$$\frac{\sin(\theta_1)}{\sin(\theta_2)} = \frac{\eta_2}{\eta_1}$$



Se **il primo mezzo è meno rifrangente del secondo**, allora $\eta_1 < \eta_2$ e, di conseguenza

$$\frac{\eta_2}{\eta_1} > 1 \rightarrow \frac{\sin(\theta_1)}{\sin(\theta_2)} > 1 \rightarrow \sin(\theta_1) > \sin(\theta_2) \rightarrow \theta_1 > \theta_2 \text{ (ricordiamo che } \theta_2 \text{ è l'angolo formato tra il raggio rifratto e la normale)}$$

Quindi il raggio rifratto **si avvicina alla normale alla superficie**.

Se **il primo mezzo è più rifrangente del secondo**, allora $\eta_1 > \eta_2$ e, di conseguenza

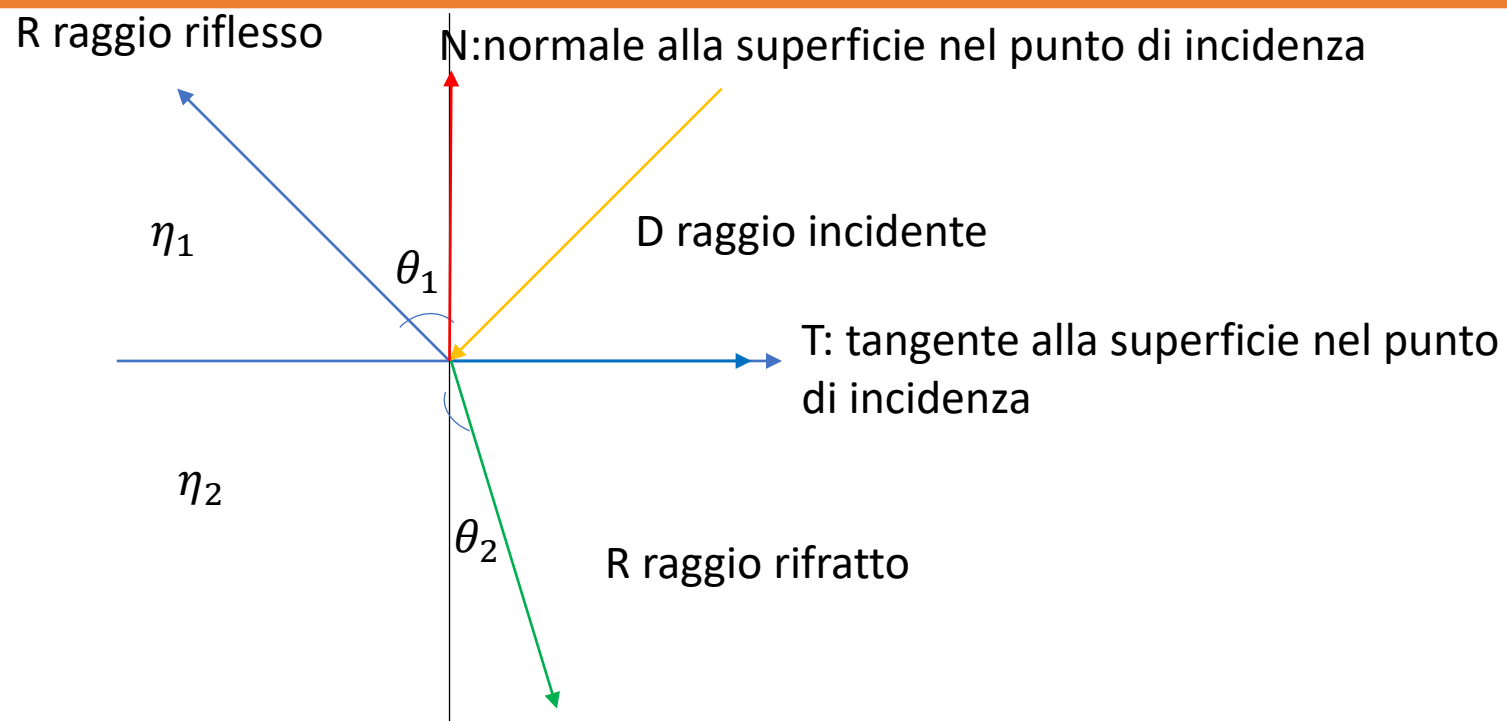
$$\frac{\eta_2}{\eta_1} < 1 \rightarrow \frac{\sin(\theta_1)}{\sin(\theta_2)} < 1 \rightarrow \sin(\theta_1) < \sin(\theta_2) \rightarrow \theta_1 < \theta_2$$

Quindi il raggio rifratto **si allontana dalla normale alla superficie**.



$$\eta_1 < \eta_2$$

il primo mezzo è meno rifrangente del secondo $\rightarrow \theta_1 > \theta_2$



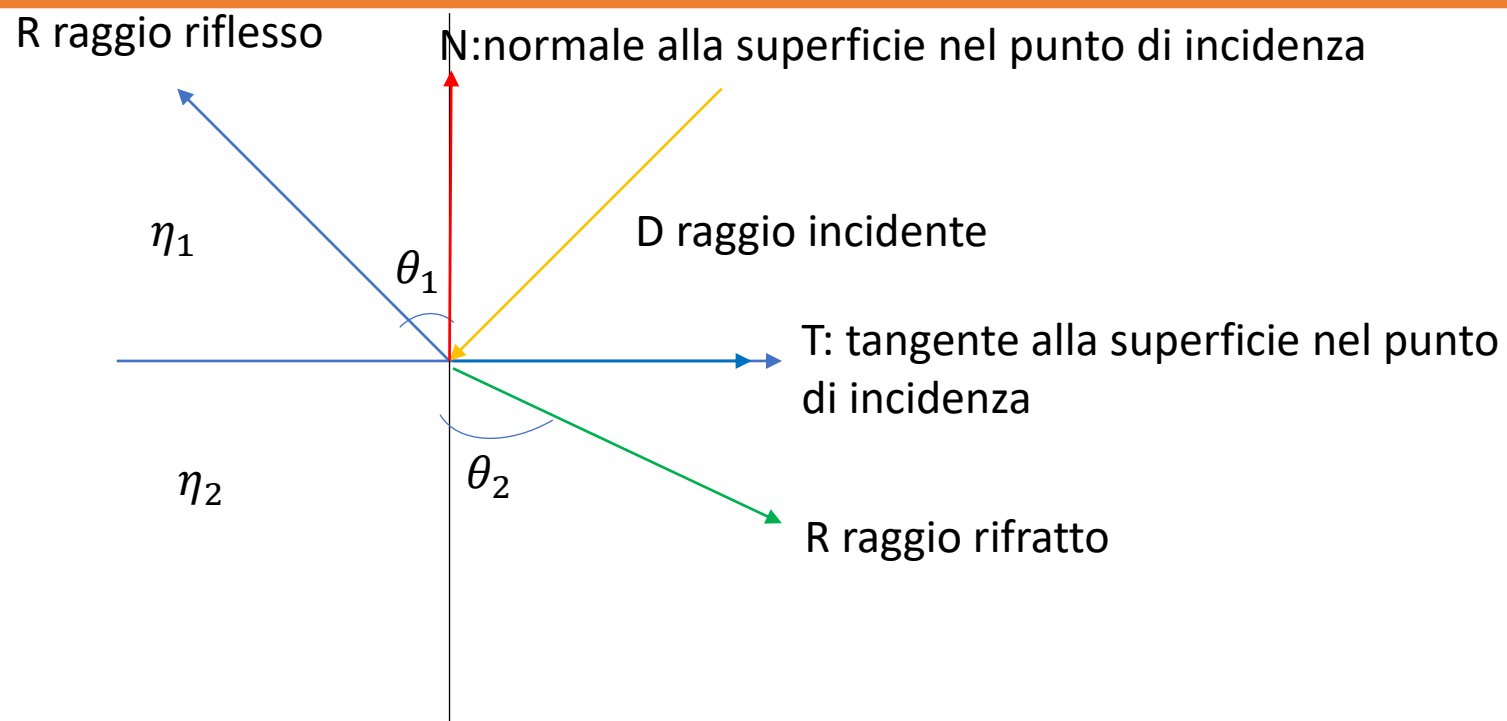
Quindi **il raggio rifratto si avvicina alla normale alla superficie**



$$\eta_1 > \eta_2$$

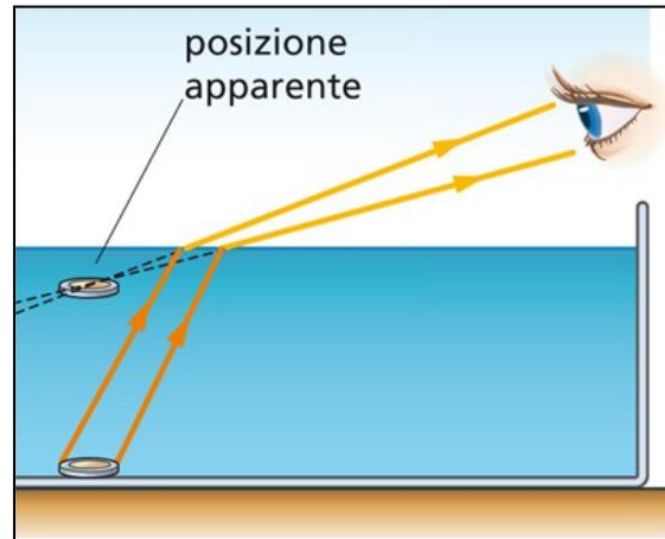
il primo mezzo è più rifrangente del secondo

$$\rightarrow \theta_2 > \theta_1$$



Quindi il raggio rifratto **si allontana dalla normale alla superficie**

Se un osservatore guarda in direzione di un oggetto in acqua, lo vedrà a una profondità inferiore rispetto a quella reale, **a causa del cammino dei raggi luminosi che, provenendo dall'acqua, che ha un indice di rifrazione maggiore di quello dell'aria, si rifrangono allontanandosi dalla normale alla superficie di separazione dei due mezzi.**





Lo stesso effetto si verifica con una cannuccia parzialmente immersa in acqua, cioè ogni punto della cannuccia immersa nell'acqua appare più vicina di quanto realmente sia alla superficie attraverso la quale viene osservato.

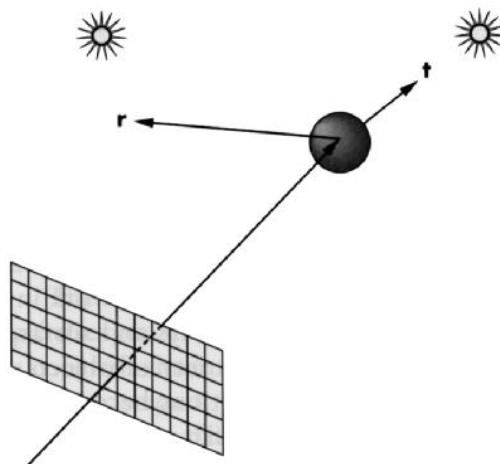




Gestione Rifrazione

Per gestire la trasparenza degli oggetti con ray tracing ad ogni intersezione raggio-oggetto semitrasparente viene generato un nuovo raggio nella direzione della trasparenza (secondo la legge di rifrazione) e calcolato il contributo trasmesso.

Ogni volta che un raggio colpisce una superficie produce, in generale, un raggio riflesso ed uno rifratto.





Pseudocodice

Come nel ray tracing standard, il processo inizia con il lancio di un raggio primario dalla telecamera attraverso ogni pixel dell'immagine verso la scena.

Intersezione con gli oggetti:

Il raggio primario interseca la scena con gli oggetti. Se c'è un'intersezione, viene determinato il punto di intersezione e la normale all'oggetto nel punto di impatto.

Verifica delle proprietà del materiale:

Se l'oggetto colpito ha proprietà di rifrazione, come un materiale trasparente, il ray tracing gestisce la rifrazione. La rifrazione segue le leggi della rifrazione di Snell, che descrivono come la direzione del raggio di luce cambia quando attraversa un'interfaccia tra materiali di diverso indice di rifrazione.

Calcolo del raggio rifratto e del raggio riflesso:

Viene calcolato il raggio rifratto ed il raggio riflesso in base alla direzione del raggio incidente, alla normale all'oggetto e all'indice di rifrazione relativo tra i due materiali. .

Raggio rifratto e riflesso e nuove intersezioni:

Il raggio rifratto ed il raggio riflesso diventano due nuovi raggi primari, e il processo di intersezione viene ripetuto. Questo significa che i raggi possono colpire nuovi oggetti nella scena.

Ogni volta che il raggio colpisce un oggetto, se quest'ultimo è rifrattivo, viene generato un nuovo raggio rifratto ed uno riflesso ed il processo continua fino a quando si raggiunge una profondità massima di rifrazione o il raggio non colpisce più alcun oggetto trasparente.



Costruzione geometrica di R (facoltativo)

Proiettiamo R su $-T$ (sia **a** la proiezione) e su $-N$ (sia **b** la proiezione)

$$a = -T \sin(\theta_2)$$

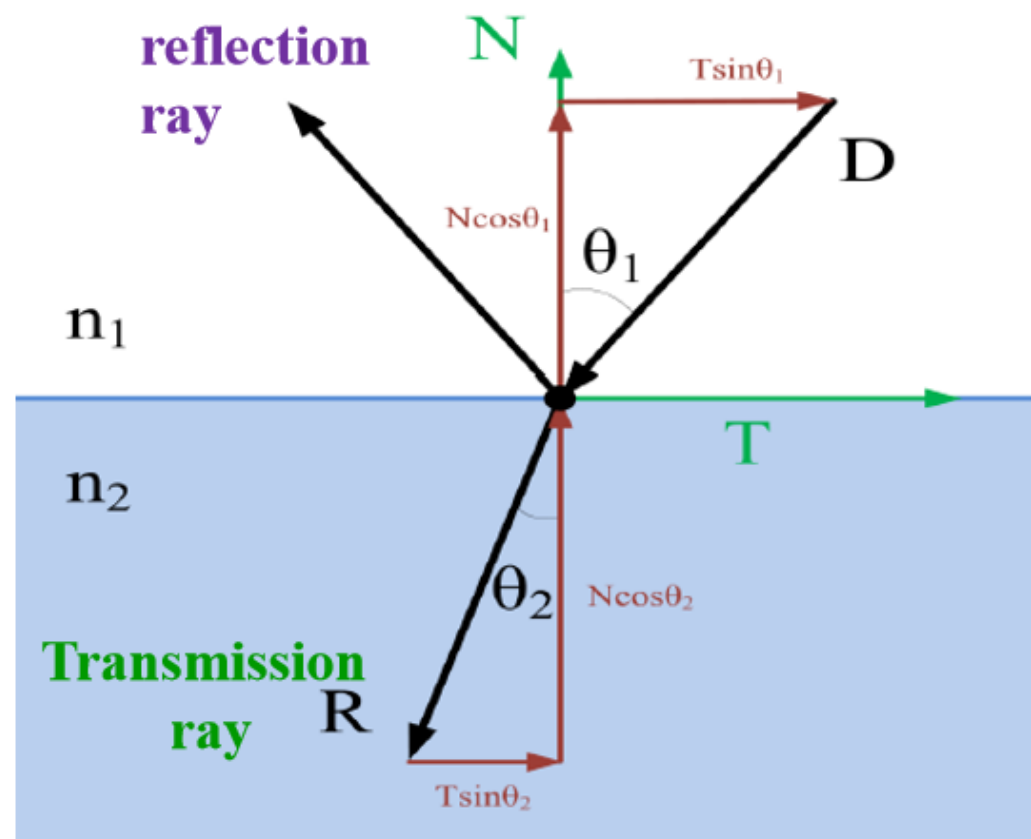
$$R = a + b = -T \sin(\theta_2) - N \cos(\theta_2)$$

$$b = -N \cos(\theta_2)$$

$$-D = N \cos(\theta_1) + T \sin(\theta_1)$$

$$T = \frac{-D - N \cos(\theta_1)}{\sin(\theta_1)}$$

$$R = (D + N \cos(\theta_1)) \frac{\sin(\theta_2)}{\sin(\theta_1)} - N \sqrt{1 - \sin^2(\theta_2)}$$





$$R = (D + N \cos(\theta_1)) \frac{\sin(\theta_2)}{\sin(\theta_1)} - N \sqrt{1 - \sin^2(\theta_2)}$$

Per le leggi di Snell

$$\eta_1 \sin(\theta_1) = \eta_2 \sin(\theta_2) \quad \sin(\theta_2) = \frac{\eta_1}{\eta_2} \sin(\theta_1) \quad \frac{\sin(\theta_2)}{\sin(\theta_1)} = \frac{\eta_1}{\eta_2}$$

$$R = (D + N \cos(\theta_1)) \frac{\eta_1}{\eta_2} - N \sqrt{1 - \sin^2(\theta_2)}$$

$$R = (D + N \cos(\theta_1)) \frac{\eta_1}{\eta_2} - N \sqrt{1 - \left(\frac{\eta_1}{\eta_2} \sin(\theta_1) \right)^2}$$

$$R = (D + N \cos(\theta_1)) \frac{\eta_1}{\eta_2} - N \sqrt{1 - \left(\frac{\eta_1}{\eta_2} \right)^2 \sin^2(\theta_1)}$$



$$R = (D + N \cos(\theta_1)) \frac{\eta_1}{\eta_2} - N \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 \sin^2(\theta_1)}$$

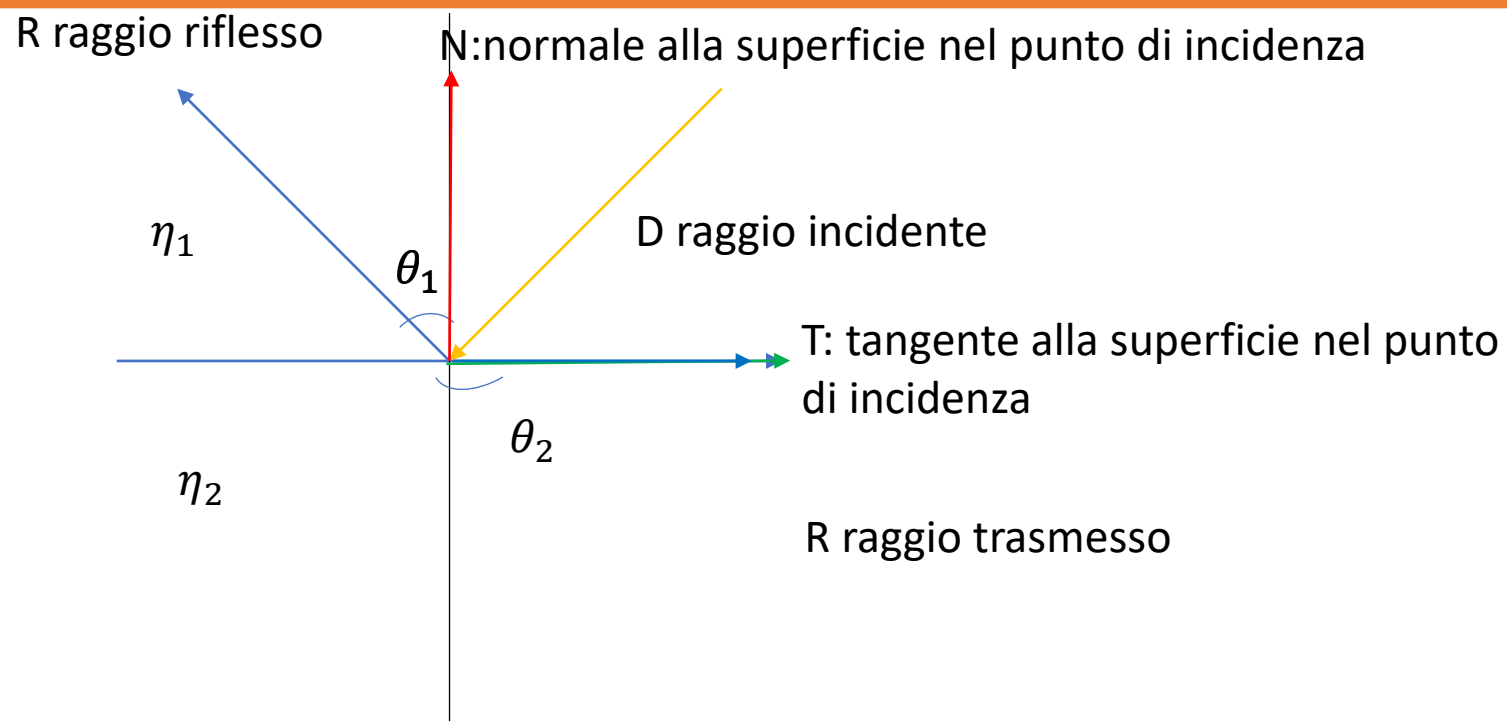
$$R = (D + N \cos(\theta_1)) \frac{\eta_1}{\eta_2} - N \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - \cos^2(\theta_1))}$$

Poichè $\cos(\theta_1) = N \cdot D$

$$R = D \frac{\eta_1}{\eta_2} + N \frac{\eta_1}{\eta_2} (N \cdot D) - N \sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - (N \cdot D)^2)}$$

Se il radicando è negativo, allora non c'è rifrazione. Si parla di total internal reflection.

Questo accade quando la luce tenta di passare da un mezzo più rifrangente (vetro) a un mezzo meno rifrangente (aria) con un angolo basso

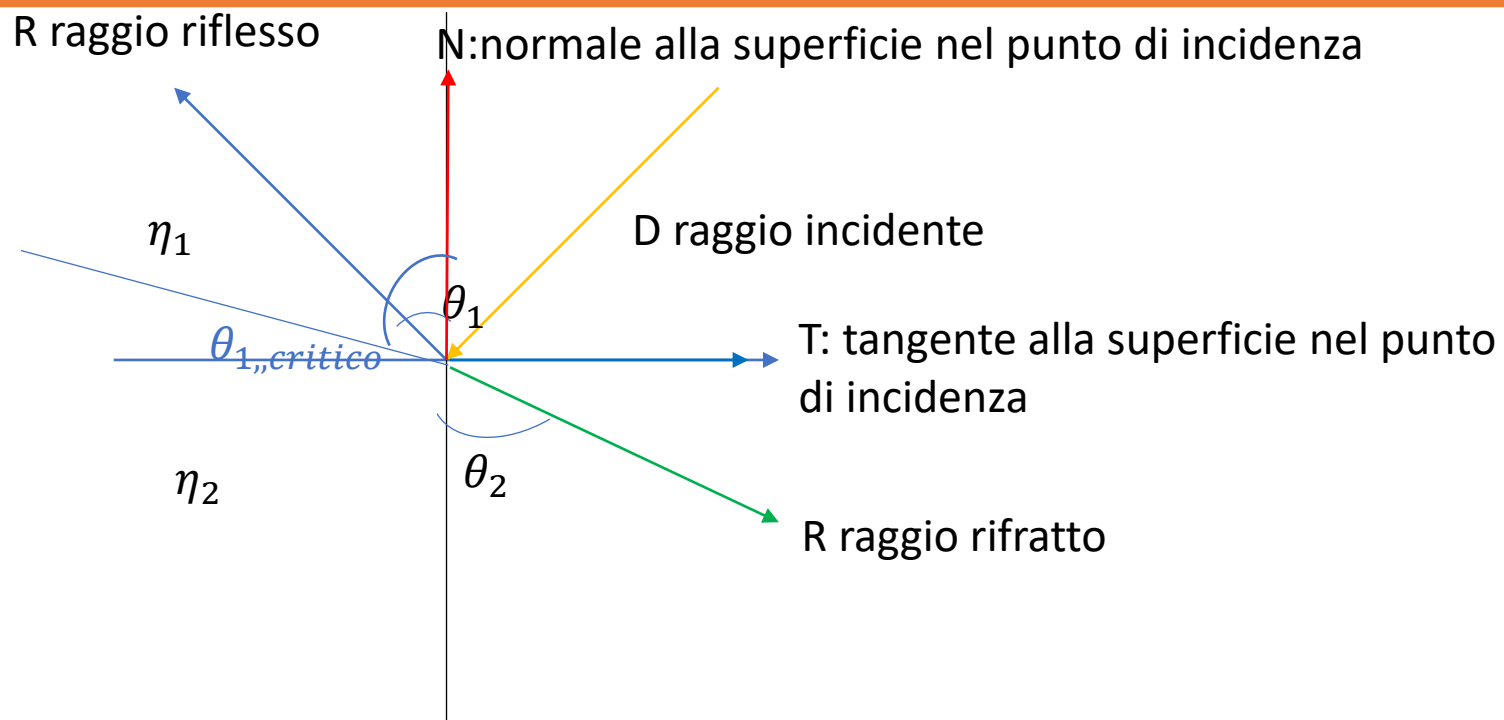


Si definisce angolo critico, $\theta_{1,critico}$ l'angolo di incidenza per cui si assiste al fenomeno della **riflessione totale**, ovvero quel valore per cui il corrispondente angolo di rifrazione è pari a 90° .



$$\eta_1 > \eta_2$$

il primo mezzo è più rifrangente del secondo



$$\theta_1 < \theta_{1,critico}$$

C'è riflessione e rifrazione, se l'angolo di incidenza è minore dell'angolo critico



$$\eta_1 > \eta_2$$

il primo mezzo è più rifrangente del secondo

R raggio riflesso

N: normale alla superficie nel punto di incidenza

η_1
 θ_1 $\theta_{1,critico}$

D raggio incidente

T: tangente alla superficie nel punto di incidenza

η_2

θ_2

$$\theta_1 > \theta_{1,critico}$$

C'è solo rifrazione, se l'angolo di incidenza è maggiore dell'angolo critico



Ray tracing ricorsivo

```
Image raytraceImage(vector eye)
begin
  Image image;
  for (each scanline Y)
    for (each pixel X) image[X][Y]=

      traceray(eye,dirFromEyeToXY,RecursionStep)

  end
End

return image;
```

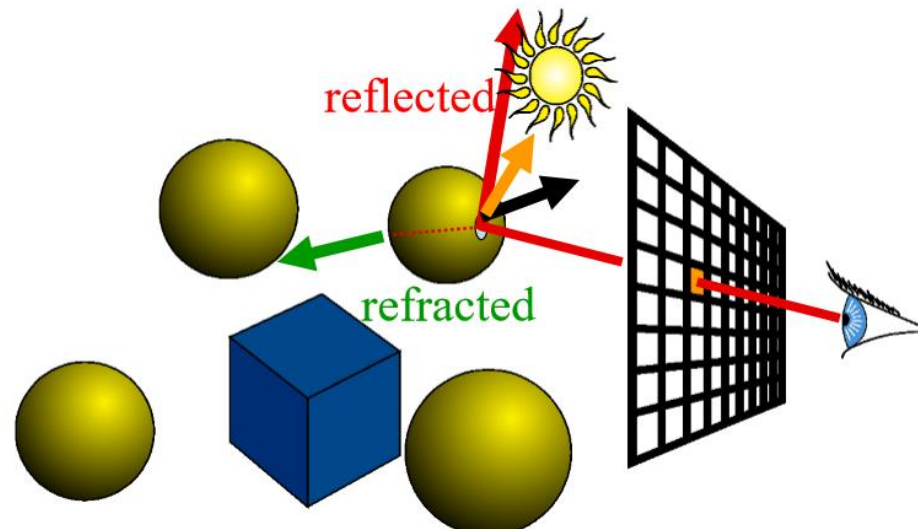
traceray () procedura ricorsiva che traccia un raggio, definito da un punto e una direzione, e calcola il colore della prima superficie intersecata (Inizializza RecursionStep = numero massimo di rimbalzi - passi ricorsivi)



traceray()

- Trova la prima intersezione del raggio con la scena; se il raggio non colpisce alcun oggetto, viene restituito il "colore di sfondo"; se il raggio interseca la luce, viene restituito il colore della luce;
 - Genera raggi d'ombra alle fonti luminose
 - Calcola l'illuminazione di quel punto in base al modello di illuminazione locale
 - Genera un raggio di riflessione r e un raggio di trasmissione t appropriatamente
 - Richiama se stesso ricorsivamente con r e t
 - Combina i contributi dell' illuminazione risultanti ed esce
-

Tutti i raggi riflessi, rifratti e ombre sono chiamati raggi secondari





color c = traceray (punto p, vettore d, int step)

begin

color locale, riflesso, trasmesso;

point q; normal n;

if (step > max) restituisce (colore di sfondo);

[obj, q] = intersect (p, d, status);

if (status = sorgente luminosa) restituisce (colore della sorgente luminosa colore);

if (status = no-intersection) restituisce (colore dello sfondo);

*/ * calcolo della normale al punto di intersezione, della direzione del raggio riflesso e del raggio trasmesso */*

n = normale (q);

r = reflect (q, n); t = trasmit (q, n);

local = phong (q, n, r);

If (obj.mirror) riflesso = traceray (q, r, step + 1);

If (obj.transparent) trasmesso = traceray (q, t, step + 1);

return (local + trasmesso + riflesso);

end



Equazione dell'illuminazione

$$I = I_{local} + K_r I_{reflect} + K_t I_{transmit}$$

K_r in $[0,1]$: fattore che specifica quale frazione della luce dalla direzione della riflessione viene riflessa;

K_t in $[0,1]$ proprietà materiale, specifica la frazione della luce trasmessa attraverso la superficie

$I_{reflect}$: intensità della luce riflessa in entrata, calcolata in modo ricorsivo nella direzione del raggio di riflessione;

$I_{transmit}$ intensità della luce riflessa in entrata, calcolata in modo ricorsivo nella direzione di trasmissione;

Illuminazione I_{local} calcolata mediante modello di illuminazione locale (Phong)

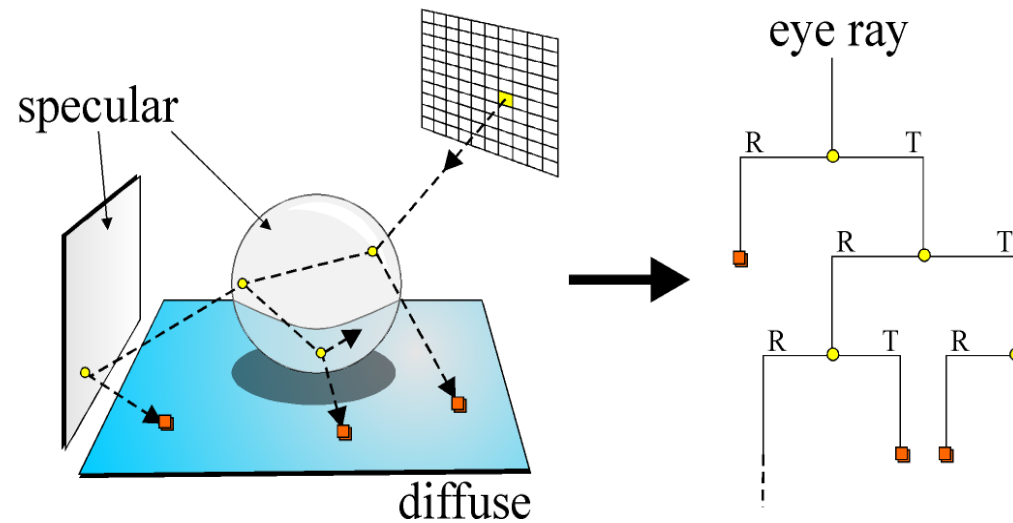


Albero del ray-tracing

- Nell' albero binario ogni nodo corrisponde all'intersezione tra un raggio e una superficie.
- Ogni nodo può avere un ramo per il raggio di riflessione ed un ramo per il raggio di rifrazione.



Recursive Ray Tracing





Criterio di stop

- Stop dopo un numero di rimbalzi
- Stop se i contributi della riflessione e rifrazione diventano troppo piccoli.



-
- Il Ray Tracing gestisce correttamente la riflessione speculare tra gli oggetti poiché è basato su una modellazione globale della radiazione riflessa in modo speculare.
 - La gestione della componente diffusa dell'illuminazione, realizzata usando un modello solo locale, non riesce invece a rappresentare i fenomeni di inter-riflessione della luce tra gli oggetti della scena.
-



```
v = calculate viewpoint
for (x = x_min; x < x_max; x++) {
  for (y = y_min; y < y_max; y++) {
    r = calculate ray passing from viewpoint v to pixels x, y
    c = calculate color using cast_ray(r, 1)
    draw color c and pixel x, y
  }
}

procedure cast_ray(r: ray, d: depth): color
  determine closest intersection i with closest object c
  if (intersection = true)
    compute normal n at intersection
    return color_ray(r, c, i, n, d)
  else
    return color_background
end

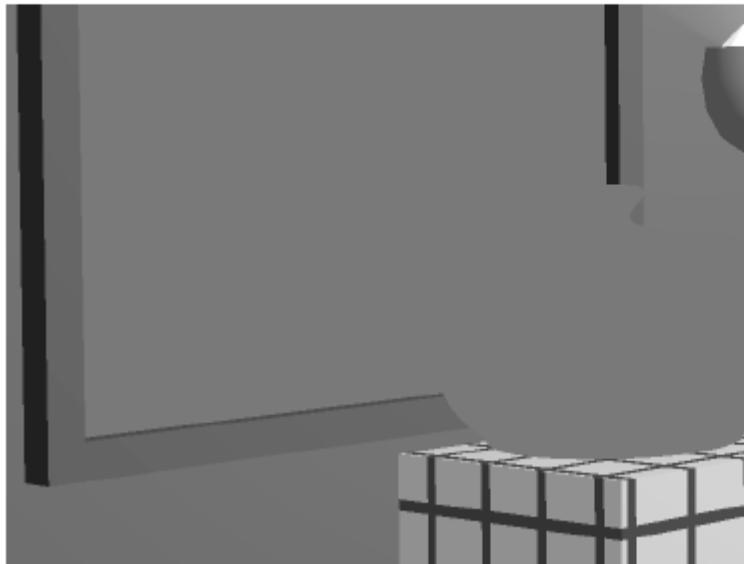
procedure color_ray(r: ray, c: object, i: intersection,
  n: normal, d: depth): color
  set color to ambient color
  for each light do
    calculate ray from intersection to light
    if dot product of normal and direction to light is positive then
      compute how much light is blocked by opaque and transparent
      surfaces and use to scale diffusion and specular terms
      before adding them to color
    end
  end

  if (d < maxDepth) then
    if (c is reflective) then
      rray = calculate ray in reflection direction
      rcol = cast_ray(rray, depth+1)
      scale rcol by specular coefficient and add to color
    end
    if (c is transparent) then
      tray = calculate ray in refraction direction
      tcol = cast_ray(tray, depth+1)
      scale tcol by transmission coefficient and add to color
    end
  end
  return color
end
```

Figure 14: Basic algorithm for recursive ray tracing.



Test Scene.



Ray tree depth 1.

Note only ambient shade
on mirror and teapot



Test Scene.

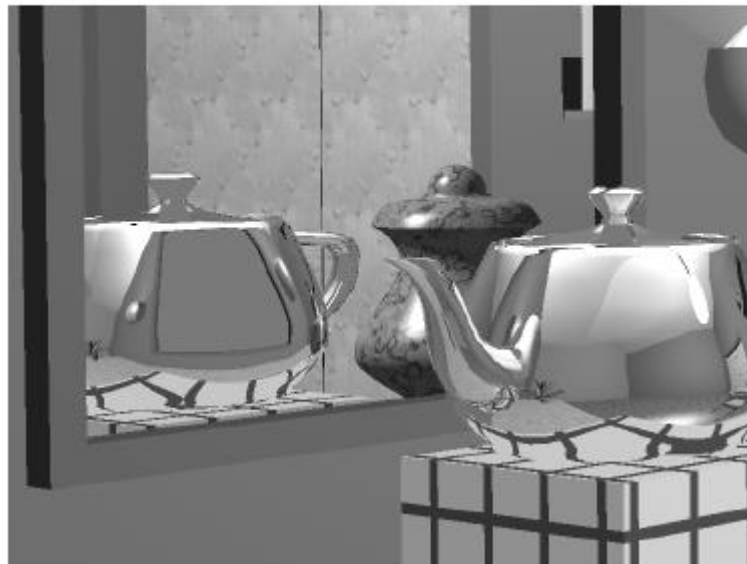


Ray tree depth 2.

Note only ambient shade
on reflection of mirror
and teapot.



Test Scene.



Ray tree depth 3.

Note only ambient shade
on reflection of mirror in
teapot.

Test Scene.



Ray tree depth 4.

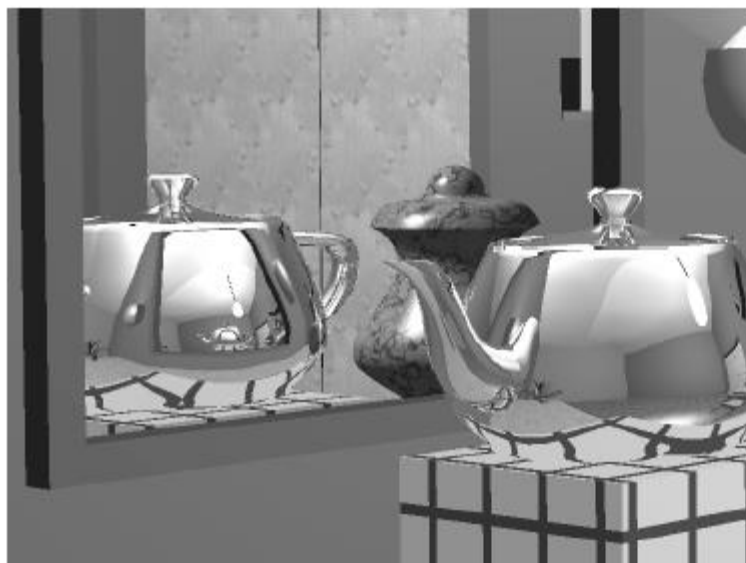
Note ambient shade on
reflection of teapot in
reflection of mirror in
teapot.

Test Scene.



Ray tree depth 5.

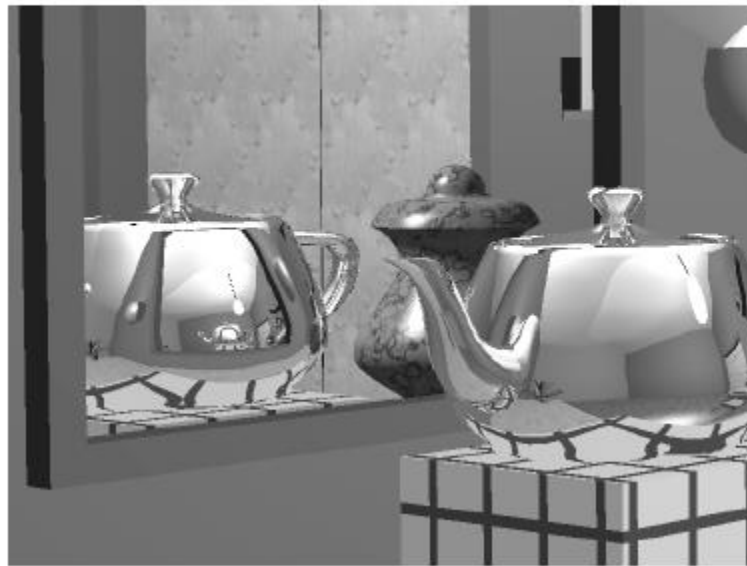
Test Scene.



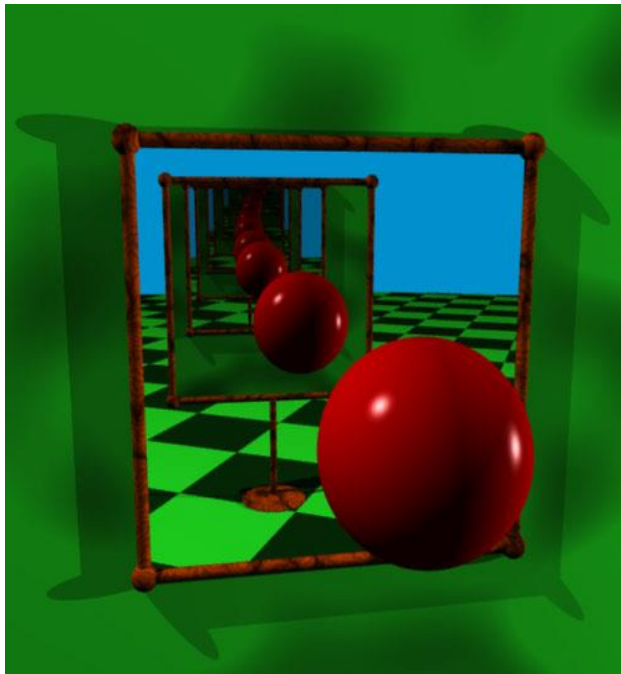
Ray tree depth 6.



Test Scene.



Ray tree depth 7.



Very high maximum recursion level





- Il metodo è tuttavia caratterizzato da un elevato costo computazionale, dovuto principalmente al calcolo delle intersezioni. Il calcolo delle intersezioni è problematico per molti tipi di superficie, perché la maggior parte dei ray tracers trattano solo superfici piane o quadriche.



-
- Il ray tracer usa il modello di Phong per includere un termine diffusivo al punto di intersezione tra un raggio ed una superficie. Tuttavia esso ignora la luce che viene distribuita per diffusione su quel punto, altrimenti dovrebbe considerare un tal numero di punti da diventare impossibile.
 - Perciò la tecnica del ray tracing è più adatta per ambienti altamente riflessivi.
-



Esempi di ray-tracing

