

Costruzione di Interfacce Lezione 14 Rendering Locale e Globale

cignoni@isti.cnr.it
<http://vcg.isti.cnr.it/~cignoni>

Rendering Locale

- ❖ Il modello di rendering che abbiamo finora visto è detto locale
- ❖ Ogni primitiva è trattata in maniera indipendente da tutte le altre
- ❖ Il concetto di scena, la presenza di altri oggetti etc, interessano solo l'utente/programmatore e non il renderer
- ❖ Il modo in cui è disegnato un triangolo dipende solo da:
 - ❖ Caratteristiche del triangolo stesso,
 - ❖ Lo stato dello zbuffer
 - ❖ Le luci.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

2

Rendering locale

- ❖ Questo tipo di modello ha numerosi vantaggi:
 - ❖ Semplicità
 - ❖ Parallelismo a livello di primitiva
 - ❖ Costo costante per primitiva

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

3

Rendering locale

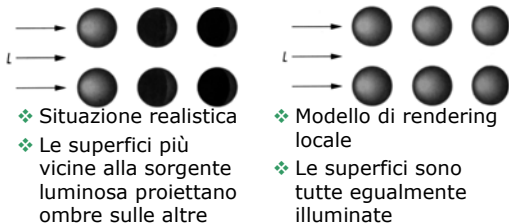
- ❖ Limitazioni
- ❖ Nel mondo reale:
 - ❖ Se un oggetto blocca la luce proveniente da una sorgente, gli oggetti al di là di esso restano in ombra
 - ❖ Se un oggetto è riflettente, la luce che si riflette da esso illumina gli altri oggetti
- ❖ Un modello di illuminazione locale non può produrre nessuno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

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

4

Rendering globale



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

5

Rendering globale

- ❖ Un'ulteriore limitazione del modello nasce dal fatto che, tenendo conto solo di fenomeni locali, non permette di modellare:
- ❖ La gestione del **calcolo delle ombre**, ossia dell'individuazione delle parti della superficie di oggetti della scena che non ricevono radiazione luminosa da una o più sorgenti, e quindi del calcolo corretto dell'illuminazione

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

6

Rendering globale

- ❖ Un'ulteriore limitazione del modello nasce dal fatto che, tenendo conto solo di fenomeni locali, non permette di modellare:
- ❖ 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 pressoché nulla)

Rendering globale

- ❖ Un'ulteriore limitazione del modello nasce dal fatto che, tenendo conto solo di fenomeni locali, non permette di modellare:
- ❖ La **riflessione diffusiva** tra gli oggetti nella scena (inter-riflessione)

Rendering globale

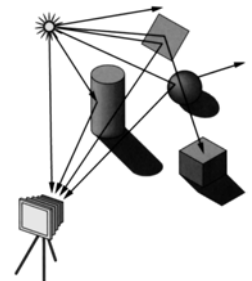
- ❖ Per risolvere i problemi di
 - ❖ punto di vista e luci nella scena
 - ❖ trasparenza
 - ❖ riflessioni speculari da altri oggetti
 - ❖ ombre
- ❖ si applica il metodo **Ray Tracing**

Rendering globale

- ❖ Il metodo **Radiosity**, invece, consente di risolvere i problemi relativi a
 - ❖ sorgenti luminose di forma qualsiasi
 - ❖ modello di illuminazione globale (che tenga conto delle inter-riflessioni diffuse tra gli oggetti)

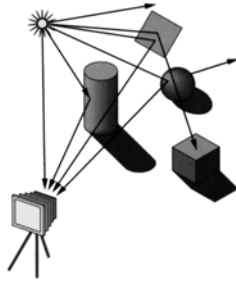
Ray Tracing

- ❖ Il metodo **ray tracing** è basato sull'osservazione che, di tutti i raggi luminosi che lasciano una sorgente, i soli che contribuiscono all'immagine sono quelli che raggiungono l'osservatore



Ray Tracing

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



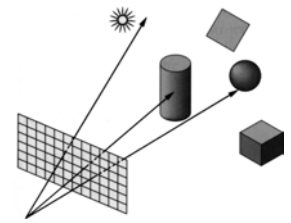
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

13

Ray Tracing

- ❖ Mentre non è possibile seguire la **traiettoria** di ciascun raggio possiamo determinare i raggi che contribuiscono all'immagine se invertiamo la traiettoria dei raggi, e consideriamo solo quelli che partono dalla posizione dell'osservatore



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

14

Ray Tracing

- ❖ Questa è esattamente l'idea alla base del metodo ray tracing, che simula **all'indietro** il cammino compiuto dalla radiazione luminosa per giungere all'osservatore
- ❖ Poiché si deve assegnare un colore a ciascun pixel, si deve considerare almeno un raggio luminoso per ogni pixel
- ❖ Questo raggio è detto **raggio primario**

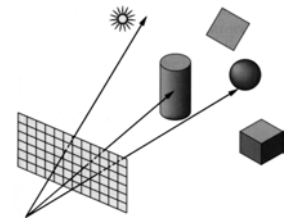
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

15

Ray Tracing

- ❖ Ciascun raggio primario può intersecare una superficie, o una sorgente luminosa, oppure può andare all'infinito senza intersezioni
- ❖ In questo caso verrà assegnato un colore di sfondo



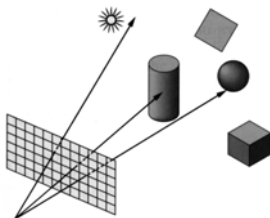
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

16

Ray Tracing

- ❖ I raggi che colpiscono le superfici (che supponiamo per ora opache) richiedono il calcolo di una gradazione di colore, shading, per il punto di intersezione.
- ❖ Se si usa un modello di shading locale (come ad esempio phong) il risultato sarebbe uguale al rendering in opengl



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

17

Ray Tracing

- ❖ Avendo a disposizione tutta la scena, per calcolare lo shading di un punto, ci possiamo *guardare intorno*
- ❖ In ray tracing ciò significa che possiamo calcolare
 - ❖ Ombre portate
 - ❖ Riflessione diretta
 - ❖ Trasparenza e rifrazione

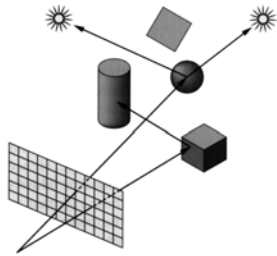
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

18

Ray Tracing e ombre portate

- ❖ Una volta che abbiamo determinato quale sia il punto della superficie visibile per un certo pixel, si generano e si tracciano **raggi ombra**, diretti dal punto sulla superficie verso ogni sorgente luminosa, per calcolare se il punto di intersezione è in ombra o no rispetto ad una sorgente luminosa



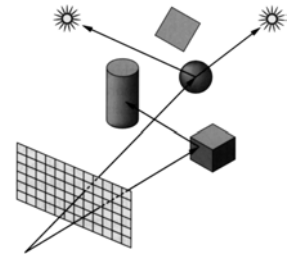
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

19

Ray Tracing e ombre portate

- ❖ Se un raggio ombra interseca una qualunque superficie prima di incontrare la sorgente, la luce è bloccata e non può raggiungere il punto considerato, che rimane dunque in ombra, almeno rispetto a questa sorgente



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

20

Ray Tracing e ombre portate

- ❖ Se le superfici sono tutte opache, e se non si considerano gli effetti della luce diffusa da superficie a superficie, abbiamo un'immagine che ha, oltre all'illuminazione, 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.

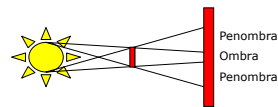
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

21

Ray Tracing e Ombre portate

- ❖ Limiti
 - ❖ Ombre nette: il meccanismo del raggio ombra (come del resto anche l'eq di shading di phong) assume che la sorgente di luce sia puntiforme, cosa inesistente in natura: assenza di penombra



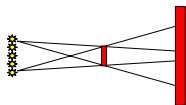
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

22

Ray Tracing e Ombre portate

- ❖ Su può modellare sorgenti di luci non puntiformi utilizzando molte luci
- ❖ Il numero di raggi ombra aumenta e quindi aumenta anche il tempo di rendering



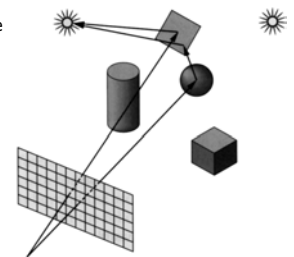
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

23

Ray Tracing e superfici riflesse

- ❖ Se alcune superfici sono invece altamente riflettive, possiamo, oltre a calcolare il raggio ombra, seguire il raggio primario riflesso come se fosse un nuovo raggio primario
- ❖ Il raggio riflesso determina il contributo dovuto alla riflessione di altri oggetti della scena sulla superficie stessa.



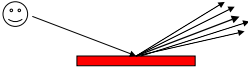
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

24

Ray Tracing e superfici riflesse

- ❖ Il calcolo degli effetti di riflessione ha un costo (si spara un altro raggio) e un limite:
- ❖ Modella accuratamente solo 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 sfuocato)
- ❖ Si può modellare sparando molti raggi... (molto costoso!)



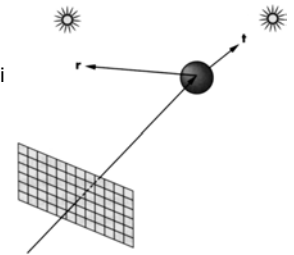
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

25

Ray Tracing

- ❖ Il metodo ray tracing consente inoltre di trattare in modo adeguato anche i casi in cui le superfici siano a trasparenza non nulla



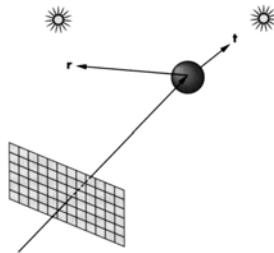
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

26

Ray Tracing

- ❖ Quando un raggio luminoso colpisce un punto di una superficie a trasparenza non nulla, la luce è parzialmente assorbita, e parte di questa luce contribuisce al termine di riflessione diffusa



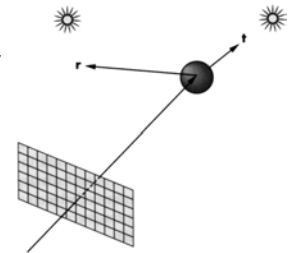
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

27

Ray Tracing

- ❖ Il resto della luce incidente si divide in un raggio trasmesso ed un raggio riflesso, il raggio riflesso prosegue il suo cammino secondo le regole della rifrazione.



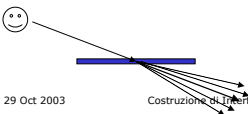
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

28

Ray Tracing e superfici trasparenti

- ❖ Limiti
 - ❖ Al solito solo superfici perfettamente lisce.
 - ❖ Vetri smerigliati, appannati ecc sono simulabili solo a gran costo sparando numerosi raggi
 - ❖ Per modellare correttamente la rifrazione occorre costruire i nostri oggetti in maniera solida (non solo superfici...):



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

29

Ray Tracing

- ❖ Il nucleo di un algoritmo raytracing è il test intersezione segmento3d (una porzione di un raggio) con una primitiva di modellazione (triangolo, sfera, ecc.).
- ❖ Il costo del rendering di una scena in raytracing è soprattutto il tempo di esecuzione di un enorme numero di questi test.
- ❖ È facilmente estendibile al trattamento di primitive di modellazione qualsiasi (ad esempio primitive solide, sfere, coni, blocchi, etc.): basta il codice per il calcolo delle intersezioni tra i raggi e la nuova primitiva

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

30

Ray Tracing in pratica

- ❖ Un'implementazione banale di un raytracer che gestisca modelli fatti solo di sfere perfette è molto semplice
- ❖ Un'implementazione molto efficiente di un raytracer è abbastanza complessa.
 - ❖ La cosa più importante è l'implementazione efficiente del test intersezione raggio primitiva

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

31

Speedup Intersezione

- ❖ Numerose tecniche: Obiettivo abbassare la complessità del test di intersezione a qualcosa che sia sublineare nella grandezza della scena

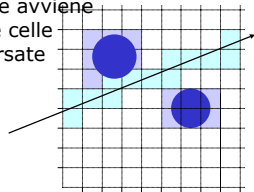
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

32

Ray Tracing ottimizzazioni

- ❖ Indicizzazione dello spazio della scena con una griglia (gerarchica o meno) che dica quali primitive sono contenute in una certa porzione dello spazio
- ❖ Il test d'intersezione avviene controllando tutte le celle del grigliato attraversate da un dato raggio (nell'ordine del raggio)



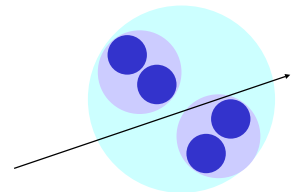
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

33

Ray Tracing Ottimizzazioni

- ❖ Indicizzazione delle primitive della scena mediante gerarchie (alberi) di bounding volumes (sfere o bounding box)
- ❖ Il test di intersezione viene fatto visitando l'albero dei bounding volumes partendo dalla radice e testando i figli solo se il padre interseca il raggio



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

34

Ray Tracing

- ❖ Gestisce correttamente
 - ❖ Hidden surface removal
 - ❖ Ombre portate luci puntiformi
 - ❖ Superfici riflettenti perfette
 - ❖ Effetti rifrazione perfetta
 - ❖ Estendibile
 - ❖ Parallelizabile

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

35

Ray Tracing conclusioni

- ❖ Gestisce correttamente ma con un costo spesso molto alto
 - ❖ Sorgenti luminose non puntiformi
 - ❖ Superfici riflettenti o trasparenti non perfettamente lisce
- ❖ Non gestisce illuminazione non diretta: Cioè quella quantità di luce che non arriva direttamente dalla sorgente luminosa ma raggiunge la superficie dopo aver rimbalzato su altre superfici

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

36

Radiosity

Radiosity

- ❖ Il metodo **radiosity** è invece concepito per la visualizzazione, la più realistica possibile, delle superfici perfettamente diffuse
- ❖ Senza entrare nei dettagli, vediamo solo le idee di base di questa tecnica

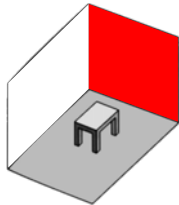
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

38

Radiosity

- ❖ Consideriamo una scena costituita semplicemente da due pareti perfettamente diffuse, una bianca ed una rossa
- ❖ Se visualizziamo la scena supponendo di avere una sorgente luminosa distante, ogni parete assumerà un colore costante



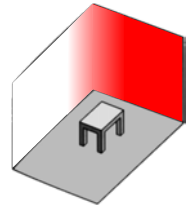
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

39

Radiosity

- ❖ Nella realtà, invece, la riflessione diffusa della parete rossa, colpisce la parete bianca, col risultato che della luce di colore rosso andrà ad aggiungersi alla luce bianca riflessa dalle parti di parete più vicine alla parete rossa



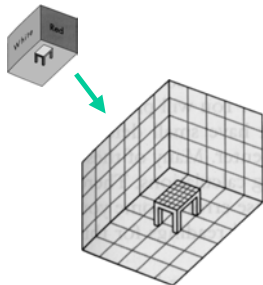
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

40

Radiosity base

- ❖ Nel metodo radiosity la scena viene suddivisa in *pezze* (*patches*), ovvero in molti poligoni piatti e di dimensioni limitate, ciascuno dei quali è considerato perfettamente diffusivo



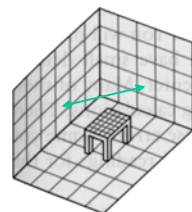
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

41

Radiosity

- ❖ Il metodo prevede due passi per determinare le gradazioni di colore da assegnare alle varie pezze
- ❖ Il primo passo consiste nel determinare, per ogni coppia di pezze, i **fattori di forma (form factor)**, che descrivono come la luce che lascia una pezza influenza l'altra



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

42

Form Factor

- ❖ In pratica I form factor definiscono quanta parte dell'energia che esce da una patch arriva su un'altra patch, tenendo in considerazione di occlusioni, orientamento delle patch, distanza ecc.
- ❖ Il calcolo dei form factor delle patch di una scena sarebbe inerentemente quadratico, ma la maggior parte dei sono praticamente nulli.
 - ❖ Patch lontane non si influenzano.

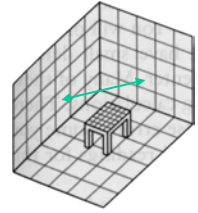
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

43

Radiosity

- ❖ Una volta determinati i fattori di forma, sapendo quali patch emettono luce, capire come si distribuisce la luce all'interno di una scena corrisponde ad un sistema di equazioni lineari:
- ❖ La somma delle quantità di luce (radianza) che arrivano su una patch deve essere uguale alla radianza che esce piu' la luce assorbita dalla patch stessa



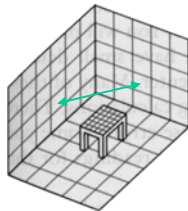
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

44

Radiosity

- ❖ In pratica dopo aver calcolato I form factor si risolve un sistema di eq. Lineari enorme (ma sparso!) e alla fine si ha per ogni patch la quantità di luce che la raggiunge (shading).



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

45

Radiosity

- ❖ Il risultato del calcolo della radiosity è relativo alla componente diffusa dello shading di una superficie, quindi è indipendente dalla posizione dell'osservatore.
- ❖ Si può mostrare il risultato del calcolo interattivamente, in opengl disabilitando il calcolo dell'illuminazione e colorando le sup in proporzione alla radiosity calcolata.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

46

Radiosity

- ❖ In sintesi
 - ❖ Modella in maniera accurata la distribuzione dell'illuminazione in una scena composta solo da superfici perfettamente diffusive.
 - ❖ L'accuratezza della soluzione trovata dipende da:
 - ❖ Accuratezza della suddivisione in patch della scena (probabilmente è la cosa piu' complessa)
 - ❖ Accuratezza nel calcolo dei form factor
 - ❖ Accuratezza nella soluzione del sistema

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

47

Radiosity

- ❖ In pratica funziona molto bene, per calcolare l'illuminazione di scene architettoniche (le pareti sono diffusori quasi perfetti) e per mostrarle poi interattivamente
- ❖ Usato in molti giochi per calcolare l'illuminazione della scena una sola volta...

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

48

Radiosity

- ❖ La tecnica presentata è estendibile (con una differente e più complessa definizione di form factor) anche alla gestione superfici non perfettamente diffuse, non perfettamente opache ecc.
- ❖ Con questo approccio, avendo abbastanza pazienza, si può simulare in maniera praticamente perfetta tutti gli effetti di illuminazione di una scena.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

49

Comparazione

Solo ray-tracing



29 Oct 2003

Ray-tracing & radiosity



Costruzione di Interfacce - Paolo Cignoni

50



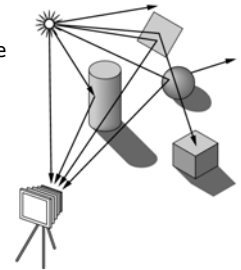
29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

51

Photon Mapping

- ❖ L'affermazione fatta durante l'introduzione del ray tracing e che è alla base del ray tracing stesso:
non è possibile seguire il percorso di tutti i raggi di luce partendo dalla sorgente luminosa
- ❖ È in realtà stata recentemente smentita dalla tecnica detta photon mapping.



29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

52

Photon mapping

- ❖ L'idea è quella di sparare un enorme numero di particelle (fotoni) dalla sorgente di luce e registrare tutti i punti delle superfici che colpiscono prima di essere definitivamente assorbiti.
- ❖ La parte principale di questo approccio è la scelta delle direzioni in cui sparare le particelle che deve adattarsi alla scena in maniera tale da raccogliere informazioni accurate sull'illuminazione che raggiunge le varie superfici.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

53

Photon Mapping

- ❖ Alla fine del processo su ogni superficie sono presenti un gran numero di fotoni (ognuno con una sua energia e direzione di provenienza) che sono utilizzati per calcolare lo shading (e non solo diffusivo) della superficie.
- ❖ La qualità del risultato finale dipende da quanti fotoni, e soprattutto come sono distribuiti nella scena.

29 Oct 2003

Costruzione di Interfacce - Paolo Cignoni

54

OpenGL?

- ❖ E' possibile sfruttare la pipeling di rendering di opengl, che è progettata per fare rendering locale, per ottenere effetti di rendering globale?
- ❖ Si, ma non è facile.
- ❖ Soprattutto occorre considerare tutta la nostra scena assieme e farla passare più volte nella pipeline di rendering.

OpenGL!

- ❖ Con un po' di fatica, si possono gestire in maniera ragionevolmente corretta in opengl:
 - ❖ Ombre portate
 - ❖ Riflessioni speculari piate
 - ❖ Rifrazione
 - ❖ Riflessioni approssimate su superfici curve