

ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE

Analisi di Algoritmi di Rendering basati sull'illuminazione globale

Elaborato in
Computer Graphics

Relatore
Prof.ssa DAMIANA LAZZARO

Presentata da
FRANCESCA PRETE

III sessione
Anno Accademico 2017-2018

Indice

Introduzione.....	1
Capitolo 1: Dal modello 3D all'immagine sullo schermo.....	3
1.1 Creazione di un'immagine a schermo a partire da un modello 3D.....	3
1.2 Eliminazione delle superfici nascoste.....	4
1.2.1 Coerenza.....	5
1.2.2 Extents e Bounding Volume.....	5
1.2.3 Back face culling.....	6
1.2.4 Partizionamento spaziale.....	6
1.3 Algoritmi per la determinazione delle superfici visibili.....	7
1.3.1 Z-buffer.....	7
1.3.2 Depth Sort.....	8
1.3.3 Ray Tracing.....	9
1.3.3.1 Bounding volume.....	9
1.3.3.2 Slabs.....	10
1.3.3.3 Gerarchie e spatial partitioning.....	10
1.4 Modelli di illuminazione.....	10
1.4.1 Tipi di superfici e di sorgenti luminose.....	11
1.4.2 Modello di illuminazione locale di Phong.....	12
1.4.2.1 Scene senza fonti luminose.....	12
1.4.2.2 Illuminazione ambientale.....	13
1.4.2.3 Illuminazione diffusiva.....	13
1.4.2.4 Illuminazione speculare.....	14
1.4.2.5 Modello completo di Phong.....	15
1.5 Tecniche di shading.....	15
1.5.1 Constant shading.....	16
1.5.2 Interpolated shading.....	16
1.5.3 Mach Banding.....	17
1.5.4 Gouraud shading.....	17
1.5.5 Phong shading.....	18
Capitolo 2: Ray Tracing.....	19
2.1 Generalità sul Ray tracing.....	19

2.2	Tipi di raggi luminosi e funzionamento.....	20
2.2.1	Albero del ray tracing.....	23
2.3	Calcolo delle intersezioni.....	24
2.3.1	Intersezione algebrica raggio-sfera.....	25
2.3.2	Intersezione geometrica raggio-sfera.....	26
2.3.3	Confronto tra i metodi.....	28
2.3.4	Problemi di precisione.....	29
2.3.5	Intersezione raggio-piano.....	30
2.3.6	Intersezione con un poligono.....	31
2.4	Fisica delle superfici.....	32
2.4.1	Frequenza e lunghezza d'onda.....	32
2.4.2	Illuminazione delle superfici e colore.....	33
2.4.3	Riflessione.....	34
2.4.4	Tipi di trasmissione della luce.....	35
2.4.5	Precisazioni sulla riflessione e sulla propagazione del colore.....	38
2.4.5.1	Riflessione diffusiva $I_{dr}(\lambda)$	38
2.4.5.2	Riflessione speculare $I_{sr}(\lambda)$	39
2.4.5.3	Riepilogo sul modello di shading.....	41
2.4.5.4	Semplificazioni del modello.....	41
2.4.5.5	Hall Shading.....	41
2.5	Aliasing.....	43
2.5.1	Anti-aliasing.....	43
2.5.2	Uniform point sampling.....	45
2.5.3	Campionamento di tipo Poisson disk.....	46
2.5.4	Jittering di una griglia regolare.....	47
2.5.5	Distributed ray tracing.....	49
2.5.6	Shading.....	49
2.5.7	Motion Blur.....	50
2.5.8	Implementazione del distributed ray tracing.....	51
2.5.9	Riepilogo del distributed ray tracing.....	51
2.6	Velocizzazione delle tecniche di ray tracing.....	52
2.6.1	Bounding volume, gerarchie e slabs.....	53
2.6.2	Suddivisione spaziale.....	56
2.7	Immagini realizzate in Ray Tracing.....	58

Capitolo 3: Radiosity.....	61
3.1 Concetto base dal radiosity.....	61
3.2 Nozioni base.....	62
3.2.1 BRDF.....	65
3.2.2 L'equazione del rendering.....	67
3.2.3 Dall'equazione di rendering a quella del radiosity.....	68
3.3 Accenni al funzionamento dell'algoritmo.....	70
3.4 Calcolo del Form Factor.....	70
3.4.1 Approccio al form factor col metodo dell'emisfera.....	71
3.4.2 Approccio al form factor col metodo dell'emicubo.....	73
3.5 Immagini realizzate con il metodo del Radiosity.....	73
Bibliografia.....	79

Introduzione

Con il termine Computer Graphics, spesso abbreviato con CG o CGI (*Computer Generated Imagery*), si intende la disciplina informatica che consiste nella creazione e manipolazione delle immagini attraverso l'utilizzo del computer.

Inizialmente utilizzata per scopi militari, la CG è entrata a far parte della vita quotidiana e il suo obiettivo, col passare del tempo, è diventato quello di aggiungere sempre più realismo alle immagini generate. Sicuramente, i campi di utilizzo più comuni e conosciuti sono quelli che riguardano il mondo cinematografico, sia per quanto riguarda la realizzazione di interi film d'animazione che per la sola aggiunta di effetti speciali. Trova applicazione negli spot pubblicitari, in cui vengono inseriti elementi grafici in immagini reali, nei videogiochi, realizzati interamente in CG, e, come si può facilmente intuire, anche nel campo riguardante la meteorologia. Oltre a questi campi, il suo utilizzo viene impiegato in altri, un po' meno evidenti, ma non meno importanti. Molti degli aspetti che riguardano la visualizzazione di un insieme di dati, anche in tre dimensioni, può essere facilmente riprodotto in un computer e visualizzato a schermo. Quindi possiamo includere nella lista anche la generazione di funzioni e modelli matematici in due o tre dimensioni, la produzione di mappe geografiche, la visualizzazione e l'analisi di dati generati in esami di medicina diagnostica e in sistemi di progettazione assistita per la realizzazione di parti meccaniche, planimetrie di edifici e la stampa di circuiti.

Il processo di creazione di immagini in ogni campo è piuttosto complesso. Lo scopo che si è posto alla CG nel tempo è stato quello di aggiungere realismo alle immagini generate, perciò sono state introdotte diverse tecniche al fine di migliorare il risultato finale.

La fase iniziale consiste nella realizzazione e modellazione di una scena 3D attraverso l'utilizzo

di adeguati software. Generata la scena è il momento di ricavarne un'immagine, visibile da un qualsiasi dispositivo in due dimensioni, ponendo una vista da cui guardarla. Il processo di creazione delle immagini, utilizzando tecniche che creino un effetto realistico, sarà l'argomento principale dei prossimi capitoli.

La tesi è così organizzata:

- il capitolo 1 tratta in generale il processo che porta dal modello all'immagine sullo schermo. Fissato un punto di vista sulla scena, ciò comprende: la determinazione di quali superfici sicuramente non saranno visibili dall'osservatore; quali superfici saranno visibili in base alla disposizione degli oggetti nella scena; a come l'illuminazione possa influire sulle superfici e i tipi di illuminazione esistenti; gli algoritmi di shading per la determinazione del colore di un singolo punto di un poligono.
- Il capitolo 2 è incentrato sull'analisi e l'approfondimento del modello di illuminazione globale Ray Tracing: si discute del suo funzionamento, dell'albero del ray tracing, delle ottimizzazioni del calcolo delle intersezioni, delle tecniche per velocizzarlo, di come viene risolto il problema dell'aliasing.
- Il capitolo 3 tratta di un altro metodo di illuminazione globale: il radiosity, passando dall'equazione del radiosity alla sua discretizzazione, descrivendo nel dettaglio il calcolo del form factor ed evidenziando le sue principali differenze con il ray tracing.

Capitolo 1: Dal modello 3D all'immagine sullo schermo

La Computer Graphics si interessa di costruire immagini sintetiche, create e manipolate con l'utilizzo di specifici hardware e software grafici.

1.1 Creazione di un'immagine a schermo a partire da un modello 3D

Con il termine *rendering* intendiamo, in Computer Graphics, il processo che permette il passaggio dalla rappresentazione digitale di una scena tridimensionale, definita in fase di modellazione, ad una *vista* della scena, intesa come una proiezione bidimensionale della scena visibile da un qualsiasi dispositivo di output^[1]. Ciò che si cerca di simulare è quindi il processo di creazione delle immagini passando da una scena 3D complessa a una in 2D.

Il processo di rendering può essere schematizzato nei seguenti passaggi:

- definizione degli oggetti, o parti di essi, che compongono la scena che intendiamo visualizzare attraverso primitive geometriche elementari;
- applicazione di trasformazioni geometriche di modellazione per ricondurre gli oggetti della scena a un unico spazio di riferimento cartesiano;
- definizione della vista che intendiamo visualizzare;
- back-face culling e clipping sul volume di vista, ovvero rimozione di parti della scena al di fuori del volume di vista e di parti di oggetti non visibili all'osservatore;
- applicazione delle trasformazioni e proiezione sul piano di vista;
- eliminazione delle superfici nascoste da altri elementi della scena;
- rasterizzazione delle primitive geometriche e rappresentazione come insieme di pixel;
- determinazione del colore di ogni pixel, in funzione della luce nella scena e delle

caratteristiche del materiale associato alla primitiva a cui il pixel appartiene.

L'ordine con cui vengono eseguite le operazioni non è sempre definito, infatti, è possibile portare avanti contemporaneamente gli ultimi tre punti della lista.

In questo capitolo, si discuterà nel dettaglio della fase relativa all'eliminazione delle superfici nascoste e della fase di colorazione di un pixel.

1.2 Eliminazione delle superfici nascoste

Nella realizzazione di un qualsiasi tipo di scena viene definita una vista, ovvero la posizione e la direzione da cui la scena viene osservata. Nel momento in cui una vista viene creata è necessario determinare quali oggetti, o quali parti di essi, sono visibili in modo da poter escludere superfici non visibili, perchè nascoste da altri oggetti. I metodi per l'eliminazione delle superfici nascoste vengono classificati come segue:

- *image precision*: data un'immagine di mn pixel, si determina quale superficie sarà visibile per ognuno di essi. Si considera un raggio che parte dal centro di proiezione e si interseca con la superficie di ogni oggetto nella scena e viene determinata quale intersezione è più vicina al centro di proiezione. Si colora il pixel in esame usando la gradazione di colore delle superficie nel punto di intersezione.

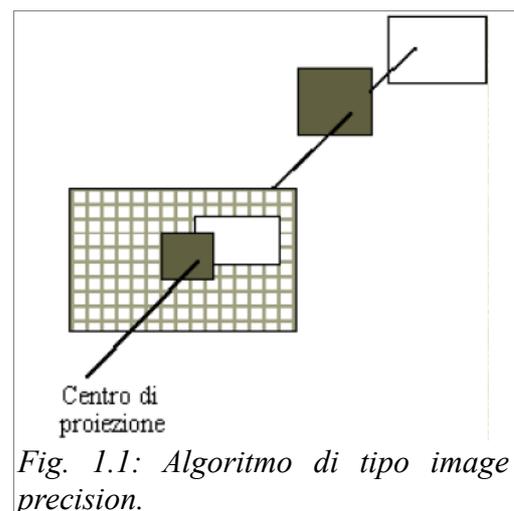


Fig. 1.1: Algoritmo di tipo *image precision*.

- *object precision*: dati k oggetti, questi vengono confrontati determinando quale o quali parti di essi sono visibili. In questo caso vengono presi i k poligoni a coppie e vengono confrontati tra di loro per determinare se uno oscura l'altro, se sono entrambi visibili o se sono entrambi parzialmente visibili.

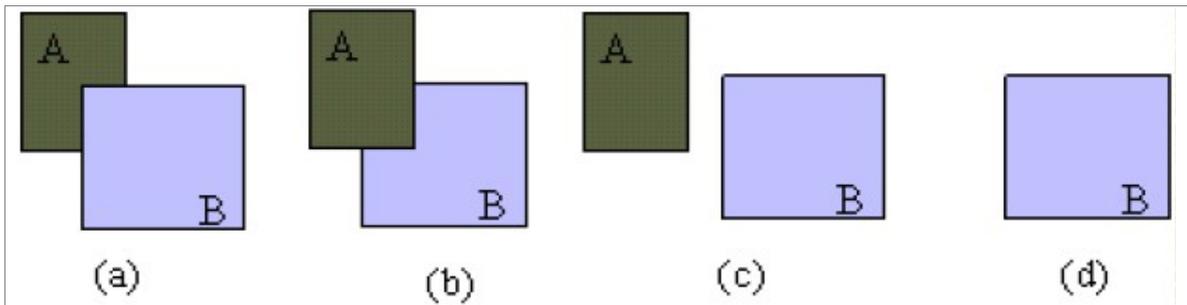


Fig. 1.2: Visualizzazione di tutti i casi possibili quando si fanno confronti con l'object precision. Nel caso (a), B oscura parzialmente A. Nel caso (b), A oscura parzialmente B. Nel caso (c), il due poligoni non si intersecano. Nel caso (d), B oscura completamente A.

Entrambi i metodi hanno un costo computazione molto elevato, perciò sono state create tecniche per evitare il più possibile i calcoli più onerosi.

1.2.1 Coerenza

Questo tipo di tecnica prevede il riutilizzo di alcune parti della scena presumibilmente simili. Ciò può verificarsi sia per quanto riguarda superfici adiacenti in cui vi sarà una differenza minima nella profondità sia nel caso di scene una successiva all'altra, in cui la vista sarà abbastanza simile e vi saranno cambiamenti minimi.

1.2.2 Extents e Bounding Volume

Un altro modo per ridurre i calcoli è quello di usare gli extents. Questi si costruiscono utilizzando la proiezione sul piano 2D di ogni oggetto e realizzando per ogni proiezione un rettangolo allineato agli assi che contenga l'oggetto e che sia il più piccolo possibile. A questo punto verrà fatto un confronto iniziale tra i vari extents in modo tale da escludere quelli che non si sovrappongono. In caso contrario, si verifica se le proiezioni si sovrappongono o se la sovrapposizione è solo degli extents. Quando utilizziamo gli extents andiamo a analizzare una singola dimensione, ad esempio z. Per verificare che due oggetti non si sovrappongano in z è sufficiente controllare se il valore massimo in z di ogni oggetto sia minore del valore minimo in z dell'altro oggetto.

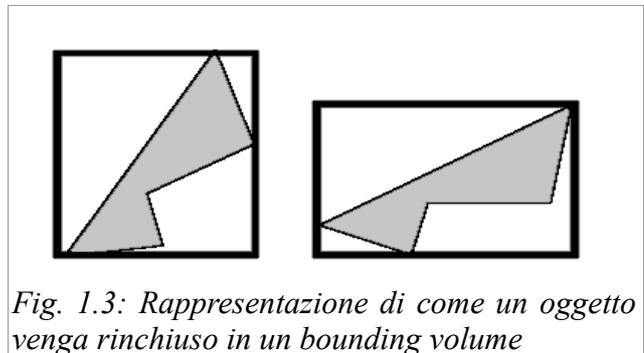


Fig. 1.3: Rappresentazione di come un oggetto venga rinchiuso in un bounding volume

1.2.3 Back face culling

La tecnica del back face culling consiste nel controllo preventivo dei poligoni che sicuramente non saranno visibili dall'osservatore. Per fare ciò, dato un oggetto qualsiasi, vengono definite le normali a ogni superficie e si controlla per ognuna di esse se l'angolo creato dalla direzione di vista e la normale alle superfici sono entro i 90° , in caso contrario la superficie non viene considerata visibile e viene esclusa dai calcoli.

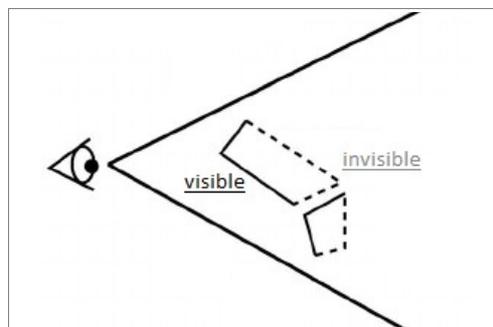


Fig. 1.4: Rappresentazione di una vista in cui i bordi tratteggiati degli oggetti non saranno visibili all'osservatore e potranno essere esclusi dal calcolo delle intersezioni.

1.2.4 Partizionamento spaziale

Questo ultimo metodo consiste nella suddivisione dello spazio in una griglia regolare di volumi più piccoli. Ad ogni cella della griglia verrà poi associata una lista degli elementi in essa contenuti, sia interamente che parzialmente. Nel momento in cui tracciamo il raggio le intersezioni saranno fatte solo con gli oggetti contenuti nelle celle che il raggio attraversa. In questo modo si evita di dover effettuare confronti con oggetti anche molto distanti dal raggio.

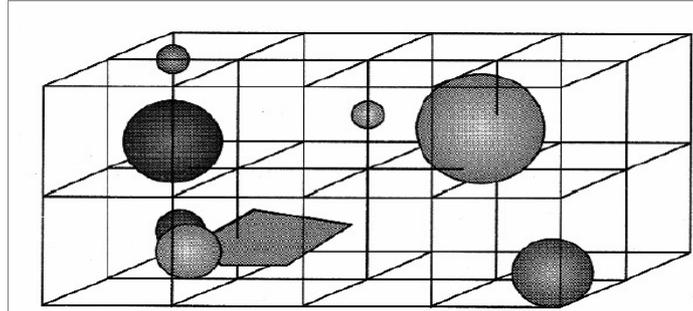


Fig. 1.5: Suddivisione dello spazio in una griglia regolare

1.3 Algoritmi per la determinazione delle superfici visibili

Come abbiamo anticipato gli algoritmi per determinare gli elementi visibili della scena vengono classificati in due grandi categorie: image precision e object precision.

1.3.1 Z-buffer

L'algoritmo dello z-buffer opera in image precision ed è un algoritmo molto semplice. Non richiede che gli oggetti nella scena siano in qualche modo ordinati e i tempi di calcolo tendono ad essere indipendenti dal numero di oggetti nella scena. La tecnica z-buffer fa uso di un buffer di memoria (z-buffer o depth buffer) che ha le stesse dimensioni del frame-buffer. Per ogni posizione della vista che si va generando, rappresentante un pixel, il frame buffer contiene il colore assegnato al pixel; nello z-buffer, vengono salvate le distanze dall'osservatore del punto corrispondente sulla primitiva visibile. Il frame buffer verrà inizializzato con il colore dello sfondo e si comincerà a analizzare ogni poligono al quale avremo precedentemente assegnato un valore di profondità per ogni punto. Per ogni (x,y) dei buffer è necessario un confronto tra l'elemento salvato nello z-buffer e la profondità del nuovo elemento analizzato. Nel caso in cui il nuovo elemento risulti più vicino di quello precedentemente salvato, i valori verranno sostituiti in entrambi i buffer.

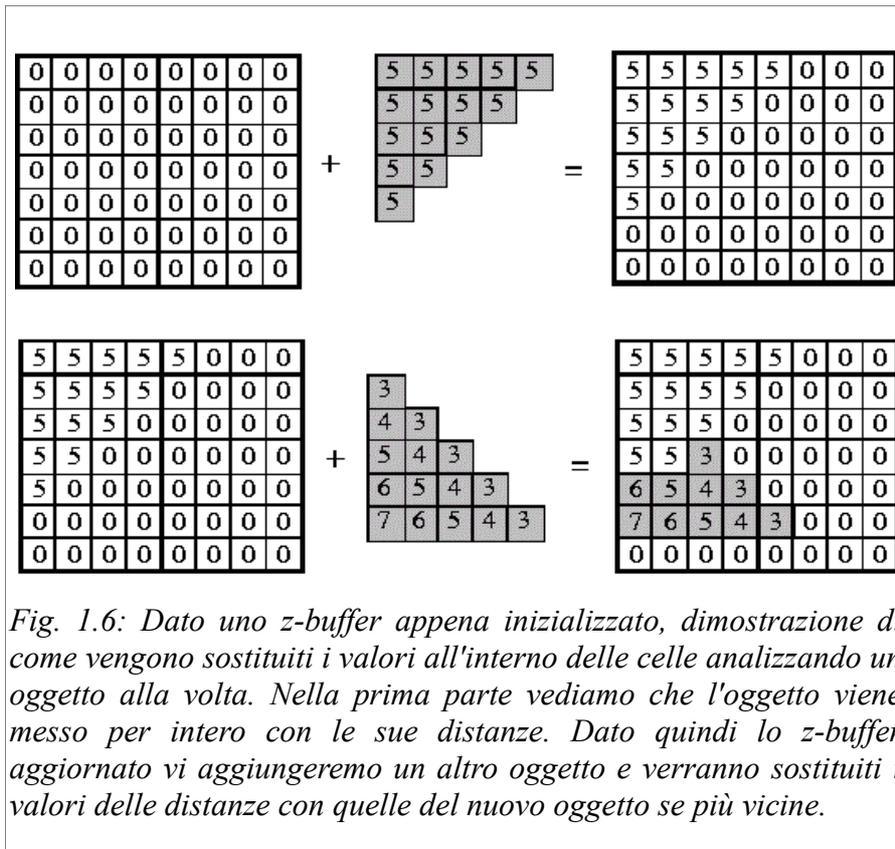


Fig. 1.6: Dato uno z-buffer appena inizializzato, dimostrazione di come vengono sostituiti i valori all'interno delle celle analizzando un oggetto alla volta. Nella prima parte vediamo che l'oggetto viene messo per intero con le sue distanze. Dato quindi lo z-buffer aggiornato vi aggiungeremo un altro oggetto e verranno sostituiti i valori delle distanze con quelle del nuovo oggetto se più vicine.

1.3.2 Depth Sort

Un altro modo è quello di realizzare una lista di priorità che definisce l'ordine in cui dovranno essere visualizzati gli oggetti. Questo algoritmo è di tipo ibrido, lavora sia in object precision che image precision. La lista ben costruita dovrà essere creata dando priorità bassa agli oggetti più lontani e in modo tale che non vi siano sovrapposizioni in z tra gli oggetti. Nel caso in cui due poligoni si sovrappongono in z, al momento in cui si fa la sua scan-conversion, viene fatto un confronto tra i due poligoni effettuando diversi test. Nel momento in cui vengono effettuati tutti i test e nessuno di essi risulta verificato, si procede all'inverso ripetendo il secondo e il terzo test.

Dati due poligoni P e Q si verifica se:

- i poligoni non si sovrappongono in x e y;
- P sia completamente dietro il piano creato dalla superficie di Q;
- Q sia completamente davanti al piano creato da P;
- le proiezioni non si sovrappongono.

Se anche scambiando i poligoni i test non si verificano, si ricade nel caso di sovrapposizione ciclica.

Al fine di risolvere questo problema è possibile dividere il poligono che impedisce la creazione di una lista ordinata in due poligoni separati.

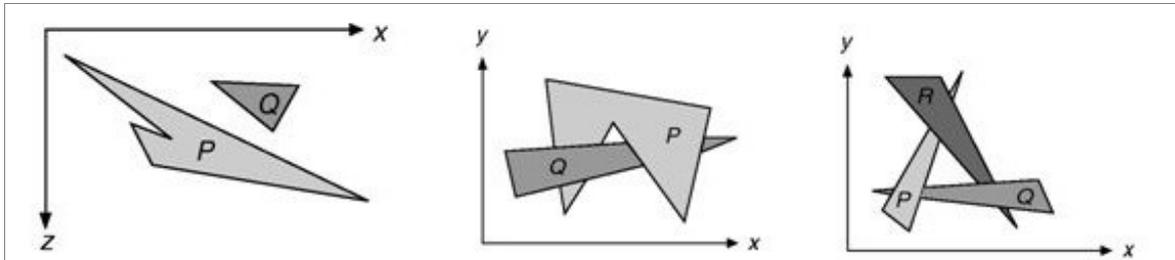


Fig. 1.7: Esempi di due poligoni separati che potranno essere ordinati e due casi in cui avremo una sovrapposizione ciclica tra i poligoni

1.3.3 Ray Tracing

Un altro algoritmo di tipo image precision è il ray tracing. L'algoritmo consiste nel considerare un centro di proiezione dal quale parte un raggio che passa attraverso il centro di ogni pixel del piano di vista e viene intersecato con ogni oggetto della scena. Il colore di ogni pixel è posto uguale a quello dell'oggetto con cui l'intersezione risulta più vicina all'osservatore. Questo tipo di

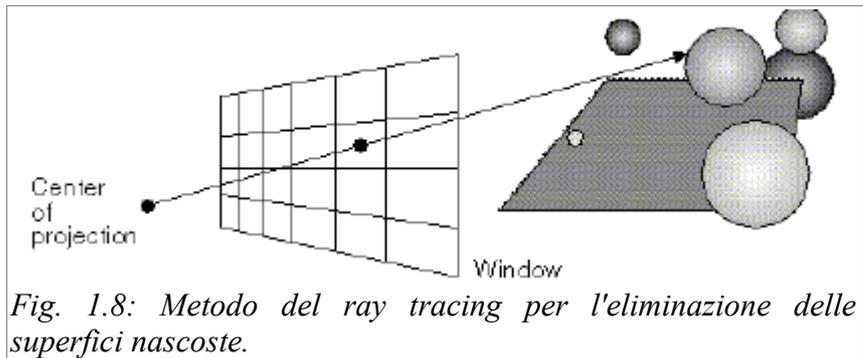


Fig. 1.8: Metodo del ray tracing per l'eliminazione delle superfici nascoste.

algoritmo risulta molto costoso, poiché gli oggetti presenti nella scena possono assumere qualsiasi forma. Esistono delle forme per cui il calcolo delle intersezioni è più semplice: la sfera e il poligono. Queste verranno utilizzate nei modi più disparati al fine di ridurre i calcoli da fare. Più avanti approfondiremo le intersezioni di un raggio con questi oggetti.

1.3.3.1 Bounding volume

Un buon metodo per velocizzare il calcolo delle intersezioni è quello di utilizzare i bounding volume, i quali sono dei contenitori dalla forma semplice e per cui è facile calcolare l'intersezione, al cui interno si racchiudono oggetti più complessi. In questo modo quando si calcolano le intersezioni, queste vengono fatte prima tra raggio e bounding volume in modo tale da evitare di considerare ulteriormente un oggetto complesso, quando già nella fase preliminare non troviamo

intersezioni.

1.3.3.2 Slabs

Un caso particolare di bounding volume è quello delle slabs che consistono nella creazione di bounding volume costituiti da intersezioni di coppie di piani paralleli, il più vicino possibile all'oggetto. Ogni bounding volume di oggetti in tre dimensioni dovrà essere costituito da almeno tre coppie di slabs. Se c'è un'intersezione tra un raggio, espresso in forma parametrica, ed una coppia di piani paralleli, queste intersezioni sono determinate per ogni coppia di valori del parametro. Si avrà un'intersezione più vicina all'osservatore e una più lontana in corrispondenza di un valore del parametro t , chiamate t_{near} e t_{far} rispettivamente. La presenza di intersezioni col bounding volume si calcolerà semplicemente valutando l'intersezione del raggio con le varie slabs. Se il massimo valore intersecato dei t_{near} è maggiore del valore minimo di t_{far} non avremo intersezioni.

1.3.3.3 Gerarchie e spatial partitioning

Al fine di ridurre ulteriormente le intersezioni è possibile creare una gerarchia di bounding volume in modo tale che ogni nodo abbia al suo interno altri bounding volume fino ad arrivare agli oggetti veri e propri. Con una struttura creata in questo modo è possibile, nel caso in cui non vi siano intersezioni tra il bounding volume principale e il raggio, escludere l'intero gruppo di elementi, poiché se non vi sono intersezioni con il bounding volume principale sicuramente non ci saranno neanche con i singoli elementi della gerarchia.

In alternativa alle gerarchie, possiamo suddividere la scena in una griglia regolare dove ogni cella ottiene determinati oggetti. Al momento di determinare le intersezioni verranno analizzati in modo ordinato solamente gli oggetti contenuti nelle celle attraversate dal raggio fermandosi alla prima intersezione verificata.

1.4 Modelli di illuminazione

Come abbiamo detto, ogni elemento della scena ha determinate proprietà come il colore e il materiale. Quando realizziamo una scena e vi inseriamo un singolo oggetto con le sue sole caratteristiche, visivamente creeremo un oggetto uniformemente colorato, piatto e illuminato dalla sola luce ambientale. Al fine di rendere gli oggetti il più realistici possibile e riuscire a mantenere

l'effetto di tridimensionalità sono stati introdotti i modelli di illuminazione e shading. Questi particolari modelli determinano il colore di ogni singolo punto di un oggetto nella scena e di conseguenza del pixel che corrisponde a quel punto, in base alle sue proprietà di materiale e colore, ma tenendo conto anche dell'illuminazione della scena e della sua incidenza in ogni singolo punto degli oggetti colpiti.

Esistono due diversi tipi di illuminazione:

- illuminazione locale: è presente la luce ambientale e nella scena una sorgente luminosa illumina gli oggetti della scena, ma non viene tenuto conto di come il colore di un punto sia influenzato dalle riflessioni degli oggetti presenti nella scena.
- illuminazione globale: questo metodo ha tutte le caratteristiche dell'illuminazione locale, ma tiene conto anche del modo in cui il rimbalzo della luce nella scena, influenza il colore del punto.

1.4.1 Tipi di superfici e di sorgenti luminose

Quando inseriamo nella scena una fonte luminosa, questa può interagire con la superficie che colpisce in base alla lunghezza d'onda della sorgente luminosa, all'angolo di incidenza della luce sulla superficie, alla natura del materiale irradiato e alle proprietà del materiale.

Quando un raggio luminoso colpisce una superficie, una parte della luce viene assorbita, mentre una parte viene riflessa. Ciò dipende da un'altra caratteristica importante, l'opacità, che indica la quantità di luce che penetra attraverso la superficie. L'opacità può assumere un valore tra 0 e 1, 0 corrisponde a una superficie trasparente e 1 una superficie completamente opaca.

- Superficie trasparente: trasmette la luce ed è possibile guardare attraverso l'oggetto e vedere altri elementi della scena.
- Superficie traslucida: trasmette la luce diffondendola, non è trasparente ma si può avere una visuale (non nitida) di altri oggetti guardandoci attraverso.
- Superficie opaca: non trasmette luce

Quando si parla di riflessione della luce, si possono avere tre diverse situazioni. Con le superfici speculari, la maggior parte della luce viene riflessa in un piccolo intervallo di direzioni intorno alla direzione di riflessione facendo risultare la superficie molto luminosa in un intorno di quel punto. Nel caso delle superfici diffuse la luce viene diffusa in tutte le direzioni. Invece in quelle traslucide troviamo un processo chiamato rifrazione per cui la luce che penetra nell'oggetto,

fuoriesce da un altro punto dello stesso. Questi tipi di trasmissione della luce saranno meglio trattati quando parleremo di modelli di illuminazione globale.

Le sorgenti luminose possono essere di diversi tipi:

- point light: ha una posizione nello spazio, non ha volume o area e emette luce ugualmente in ogni direzione;
- directional light: non ha una posizione nello spazio e si presume posizionata all'infinito. Emette luce in modo costante;
- spot light: è una sorgente luminosa puntuale, illumina in una direzione e un angolo emettendo luce in modo differente in ogni direzione;
- area light: tipo di sorgente puntuale caratterizzata da un'area.

Nella scelta di quale modello di illuminazione utilizzare si deve considerare che a una maggiore richiesta di realismo corrisponde un costo computazionale più elevato. Perciò dovremmo sicuramente tenere conto della potenza di calcolo disponibile dalla macchina utilizzata, di quanto è complessa la scena e quante immagini dovranno essere prodotte.

Di seguito sarà trattato il più importante modello di illuminazione locale, dovuto a Phong.

1.4.2 Modello di illuminazione locale di Phong

Generalmente semplice e molto utilizzato, consente di ottenere un buon risultato con un costo computazionale non troppo elevato. Questo modello è costituito da tre diverse componenti luminose: la luce ambientale, la riflessione diffusiva e quella speculare.

1.4.2.1 Scene senza fonti luminose

Prima di raggiungere il modello di Phong, si analizza in questo paragrafo un modello più semplice, ma meno realistico in cui ogni oggetto ha la sua intensità luminosa senza fonti esterne di illuminazione. In questo caso avremo un mondo di sagome monocromatiche. Ogni modello di illuminazione può essere espresso da un'equazione di illuminazione. In questo primo caso l'equazione sarà semplicemente $I=k_i$ dove I è l'intensità e k_i è la luminosità dell'oggetto.

1.4.2.2 Illuminazione ambientale

Nella realtà gli oggetti che non sono illuminati direttamente dalla luce non appaiono neri, sono illuminati dalla luce ambientale. Questo tipo di illuminazione presume che esista una luce che illumina in parte anche gli elementi che non vengono direttamente colpiti da sorgenti luminose, ma che vengano colpite dalla luce riflessa dall'ambiente. In questo modo l'equazione dell'illuminazione si trasforma in $I=I_a k_a$ dove I_a è l'intensità della luce ambientale e k_a è il coefficiente di riflessione ambientale che varia tra 0 e 1 e varia in base al materiale. La sola illuminazione ambientale non è sufficiente a creare una scena realistica poiché anche in questo caso avremo superfici uniformemente illuminate.

1.4.2.3 Illuminazione diffusiva

Supponendo la presenza di una sorgente luminosa puntiforme che emette luce uniformemente in ogni direzione, la luminosità di ogni punto dipende dalla distanza dalla sorgente luminosa e dalla direzione in cui la luce “colpisce” la superficie. Nel caso in cui l'oggetto analizzato sia completamente opaco, esso risulterà uniformemente luminoso da ogni angolazione, in quanto la luce verrà riflessa uniformemente in ogni direzione. In generale la luminosità di un punto dipende dall'angolo θ che si forma tra il raggio luminoso che colpisce la superficie in un punto e la normale in quello stesso punto. L'equazione dell'illuminazione diffusa è $I=I_p k_d \cos(\theta)$ dove l'angolo θ per contribuire all'illuminazione del punto deve essere un valore compreso tra 0° e 90° . I_p rappresenta l'intensità della sorgente luminosa e k_d è il coefficiente di riflessione diffusiva del materiale che varia tra 0 e 1. Se i vettori N , che rappresenta il vettore normale alla superficie nel punto studiato, e L , vettore che rappresenta il raggio luminoso che colpisce la superficie, sono normalizzati la formula si può scrivere come:

$$I = I_p k_d N \cdot L$$

Se la sorgente luminosa è sufficientemente distante dall'oggetto possiamo dire che forma lo stesso angolo con tutte le superfici con la stessa normale, diventando una sorgente luminosa direzionale in cui L è costante. L'equazione di illuminazione contenente sia l'illuminazione ambientale che quella diffusiva diventa:

$$I = I_a k_a + I_p k_p (N \cdot L)$$

Per rendere realistico l'effetto dalla luce sulla superficie e distinguere superfici identiche, ma a distanze diverse dalla fonte luminosa; è necessario introdurre un fattore f_{att} , detto fattore di

attenuazione che è inversamente proporzionale alla distanza della luce, l'equazione dell'illuminazione diventa:

$$I = I_a k_a + f_{att} I_p k_p (N \cdot L)$$

Una possibile scelta per il fattore di attenuazione è $f_{att} = 1/(d_L)^2$, dove d_L rappresenta la distanza dalla sorgente luminosa. In questo caso il valore di f_{att} cresce significativamente nel caso in cui la luce sia abbastanza vicina. In alternativa una soluzione migliore potrebbe essere scegliere f_{att} in questo modo:

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

limitandolo a 1 per evitare che diventi un fattore di amplificazione.

1.4.2.4 Illuminazione speculare

Quando la superficie di un oggetto non è completamente opaca, la luce non viene riflessa allo stesso modo in tutte le direzioni. Nel caso degli specchi, l'angolo che si crea tra N , la normale alla superficie, e R , la direzione di riflessione, è uguale a θ , angolo formato dalla direzione della luce e la normale al punto di incidenza.

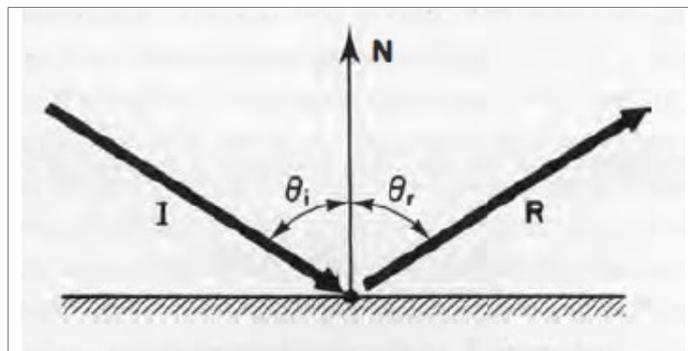


Fig. 1.9: Angolo di riflessione speculare in base all'angolo di incidenza

Consideriamo adesso il vettore V che rappresenta la direzione di vista. L'osservatore posto nella direzione di vista vede la riflessione speculare solo se l'angolo α creatosi tra il vettore R e il vettore V è pari a 0. Quando si tratta la riflessione speculare bisogna tenere conto anche del caso in cui le superfici non siano completamente riflettenti. Per simulare l'illuminazione di materiali che non sono riflettori perfetti, Phong ha usato il modello

$$I = I_p k_s \cos^n \alpha$$

Questo modello presuppone che si abbia riflessione massima per $\alpha=0$ e che decada rapidamente per α che aumenta. n viene detto coefficiente di *shininess*. L'equazione di illuminazione che include anche la luce speculare sarà:

$$I = I_a k_a + f_{att} I_p [k_d \cos \theta + k_s \cos^n \alpha]$$

in cui k_s è il coefficiente di riflessione speculare e varia da 0 a 1. Come nel caso della luce diffusiva, se normalizziamo i vettori R e V la formula diventa:

$$I = I_a k_a + I_p k_d (N \cdot L) + I_p k_s (R \cdot V)^n$$

Nel caso in cui vi siano m sorgenti luminose è sufficiente modificare la formula in questo modo:

$$I = I_a k_a + \sum_{(1 \leq i \leq m)} I_p [k_d (N \cdot L_i) + k_s (R_i \cdot V)^n]$$

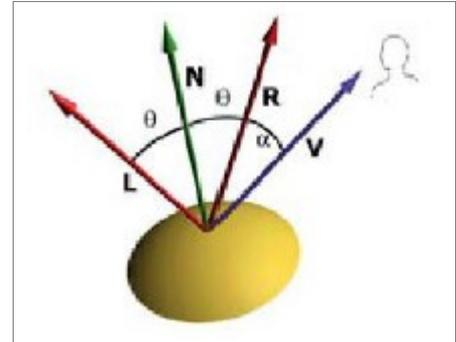


Fig. 1.10: Dato un punto, direzione L da cui proviene la luce, la normale N , direzione di riflessione della luce R e direzione di vista V con i relativi angoli.

1.4.2.5 Modello completo di Phong

Con questa formula il modello di illuminazione di Phong è completo, ma ci sono alcuni problemi per quanto riguarda il realismo, in quanto il modello presuppone che le sorgenti luminose utilizzate siano puntiformi, le riflessioni vengono modellate solo a livello locale e non si considerano tutte le inter-riflessioni tra gli oggetti.

Quando trattiamo oggetti colorati è necessario calcolare separatamente il contributo luminoso per ogni componente rosso, verde e blu.

$$I_r = I_a k_{ar} + I_p [k_{dr} (N \cdot L_i) + k_{sr} (R_i \cdot V)^n]$$

$$I_g = I_a k_{ag} + I_p [k_{dg} (N \cdot L_i) + k_{sg} (R_i \cdot V)^n]$$

$$I_b = I_a k_{ab} + I_p [k_{db} (N \cdot L_i) + k_{sb} (R_i \cdot V)^n]$$

L'intensità totale sarà la sommatoria dei contributi luminosi di tutte le sorgenti.

1.5 Tecniche di shading

Con shading intendiamo il processo con cui viene determinato il colore di ogni pixel di una superficie con l'utilizzo di un modello di illuminazione. Se il modo più semplice è calcolare

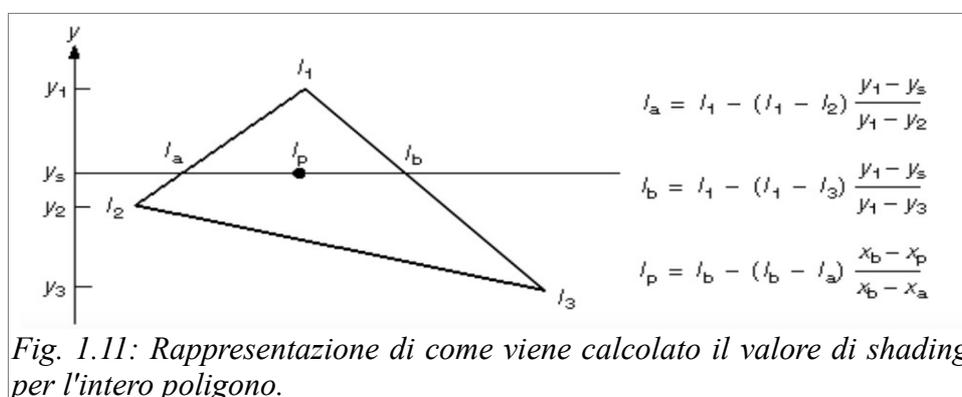
singolarmente il colore di ogni pixel e utilizzare un adeguato modello di illuminazione, questo procedimento risulta molto lungo e costoso in termini computazionali. Perciò è necessario trovare dei metodi per velocizzare e migliorare questo procedimento.

1.5.1 Constant shading

Il constant o flat shading consiste nell'eseguire il calcolo del modello dell'illuminazione solo una volta per poligono. Purtroppo questo metodo dà un risultato accettabile solo se vengono rispettate determinate proprietà: la sorgente luminosa deve essere posta all'infinito in modo tale che l'angolo formato dalla direzione della luce L e la normale nel punto di incidenza, N , sia costante su tutto il poligono; l'osservatore deve essere anch'esso posto all'infinito per rendere l'angolo formato tra la direzione della luce L e la direzione di vista V costante su tutto il poligono; la superficie che si considera è un poligono. Nel caso in cui una delle prime due ipotesi non si verifichi è possibile in alternativa determinare un unico valore per L e V , che possono essere calcolati relativamente al centro del poligono o per il primo vertice. Più i poligoni sono piccoli, migliore sarà l'effetto finale. L'effetto che verrà a crearsi non sarà ottimale poiché saranno ben visibili le suddivisioni dei poligoni.

1.5.2 Interpolated shading

Un'alternativa al constant shading è l'interpolated shading. Con questo metodo piuttosto che scegliere arbitrariamente un punto in cui calcolare il colore per tutto il poligono, verranno calcolati i valori assunti nei suoi vertici e successivamente interpolati per ottenere un valore per ogni punto del poligono.



1.5.3 Mach Banding

Nonostante l'interpolated shading renda meglio l'uniformità degli oggetti, si avrà comunque lo stesso effetto di visibilità dei bordi dei poligoni e ciò crea discontinuità. A causa di ciò otterremo un effetto chiamato Mach Banding per cui un oggetto messo vicino a uno più scuro risulta più chiaro e viceversa, evidenziando ancora di più la differenza di colore tra i poligoni. Questo effetto è dovuto alla tendenza dell'occhio umano di vedere delle bande di rinforzo nel passaggio da un colore più chiaro a uno più scuro che porta a visualizzare un colore più scuro nel punto in cui i due colori si uniscono.

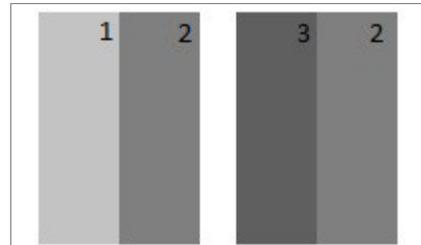
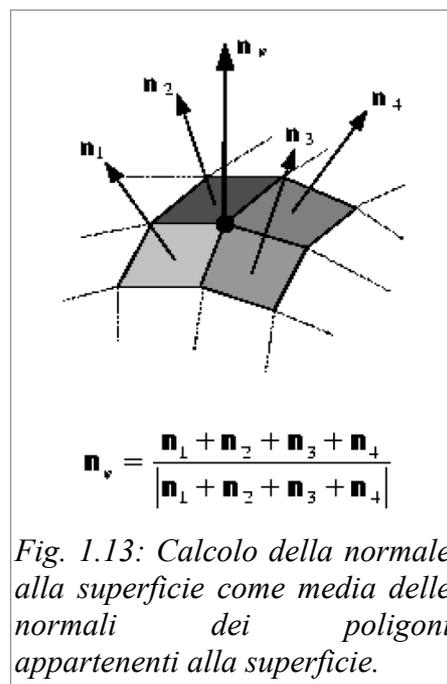


Fig. 1.12: Dimostrazione dell'effetto mach banding. In entrambe le immagini il colore 2 è lo stesso, ma vicino al colore 3 risulta più chiaro di quanto sembri di fianco al colore 1.

1.5.4 Gouraud shading

Una tecnica di shading che supera i limiti delle tecniche appena viste è il Gouraud shading. Si tratta di un metodo di tipo interpolated shading in cui la normale alla superficie in un vertice viene approssimata con la media delle normali dei poligoni che condividono quel vertice. Con questa tecnica si ottiene un effetto liscio e più uniforme nelle varie superfici.



1.5.5 Phong shading

Al fine di ottenere un effetto maggiormente realistico riguardo alla riflessione speculare, viene introdotto il Phong shading. Più costoso, ma anche quello che rende al meglio le superfici, si utilizza in un modo molto simile al Gourand shading. Il calcolo delle normali nei vertici è lo stesso, ma quando si calcola lo shading dei singoli pixel, si utilizzano normali calcolate interpolando linearmente e rinormalizzando all'interno dello spazio delle normali. Il costo computazionale elevato è dovuto al fatto che nel suo calcolo vengono interpolati interi vettori e l'equazione viene calcolata per ogni pixel^[6].

Capitolo 2: Ray Tracing

I modelli di illuminazione globale si distinguono in modelli di tipo view-dependent e modelli di tipo view-independent. I modelli di tipo view-dependent discretizzano il piano di vista per determinare in quali punti valutare l'equazione di illuminazione in base alla vista stabilita. Un rappresentante di questo tipo di modello è il Ray Tracing che sarà analizzato in questo capitolo. Invece, i modelli di tipo view-independent discretizzano l'intero ambiente valutando l'equazione dell'illuminazione da ogni punto e da ogni direzione di vista. Principale rappresentante di questo metodo è il Radiosity. Nel ray tracing l'effetto della luce nella scena verrà calcolato solamente relativamente al punto di vista fissato. D'altra parte nel radiosity gli effetti della luce verranno calcolati per tutta la scena, anche per le parti che non saranno visualizzate a schermo. Oltre ad avere tutte le caratteristiche dei modelli locali, questi modelli gestiscono tutte le inter-riflessioni tra gli oggetti.

2.1 Generalità sul Ray tracing

Nel creare un rendering di una qualsiasi scena in cui sono presenti fonti luminose è necessario tenere conto di come e in che modo i raggi luminosi influiscono sulla scena. Nella Fig. 2.1 il raggio A rappresenta quei raggi luminosi che colpiscono un punto qualsiasi della scena i quali, per le proprietà delle superfici colpite, non verranno riflesse in nessuna direzione e così il percorso del raggio luminoso si interrompe. Il raggio di tipo B interseca lo specchio, quindi una superficie perfettamente riflettente, e viene riflesso sul divano, inquadrato dalla camera, e verrà quindi rappresentato sul piano di vista. Il raggio di tipo C illumina semplicemente un punto del divano inquadrato, mentre D illumina un punto che sarà visualizzabile per riflessione. Un raggio di tipo E,

- *reflection rays*: quelli che si creano con la riflessione della luce su un oggetto;
- *transparency rays*: che trasmettono la luce attraverso gli oggetti.

Il primo passo per scegliere il colore da assegnare a un determinato pixel è quello di tracciare un *eye ray* dalla camera che, passando per il pixel in esame, raggiunge un punto sulla scena appartenente ad un determinato oggetto. Determinato il punto P di intersezione, bisogna prima di tutto determinare da quali fonti luminose viene direttamente colpito il punto. Si tracciano quindi degli *shadow rays* dal punto P verso le fonti luminose. Nel caso in cui nel tragitto non vi siano elementi opachi tra P e la fonte luminosa (come tra il punto P e la fonte luminosa A nell'esempio in Fig. 2.2) il punto sarà illuminato dai raggi provenienti da quella fonte. Se invece, come nel caso in figura tra P e B , i raggi colpiscono un oggetto opaco nella traiettoria, il colore di P non verrà influenzato da quella determinata fonte.

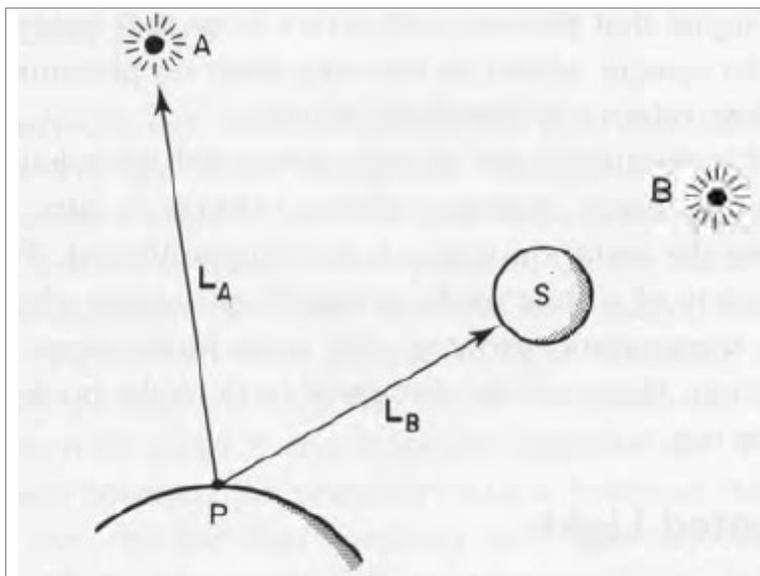


Fig. 2.2: Rappresentazione dei raggi ombra che vanno dal punto preso in esame P nella direzione delle fonti luminose. Il raggio L_A colpirà la sorgente luminosa A portando ad un suo contributo nel punto P . Al contrario il raggio L_B incontrerà sul suo tragitto un oggetto S che impedirà alla luce proveniente da B di contribuire alla luminosità di P .

All'illuminazione di un punto, oltre alla luce diretta, contribuisce un altro tipo di illuminazione che è quella della *propagated light* o luce propagata, che rappresenta tutti quei raggi che arrivano in un punto per trasmissione della luce tra gli oggetti. Per ora ci concentreremo in particolare sulla luce

speculare riflessa e trasmessa. Quando osserviamo un qualsiasi oggetto riflettente da una particolare angolazione, noteremo senz'altro che dato uno specifico angolo di incidenza sulla superficie è possibile vedere sulla superficie di un altro oggetto della scena i contributi luminosi dati dalle fonti luminose. Per comprendere meglio come ciò accade possiamo prendere in esame la Fig. 2.3. Dalla direzione di vista non è possibile vedere direttamente la fonte luminosa e il rimbalzo che si creerebbe specularmente sul punto P non permette di vederla, ma supponendo che tutti gli oggetti visualizzati siano riflettenti, possiamo notare come seguendo specularmente il percorso della luce nella scena, questa rimbalzi tra gli oggetti arrivando a contribuire alla luminosità di P . Il raggio B in figura mostra che il punto P viene direttamente illuminato dalla fonte luminosa, ma dalla direzione di vista non sarà possibile vedere direttamente la fonte.

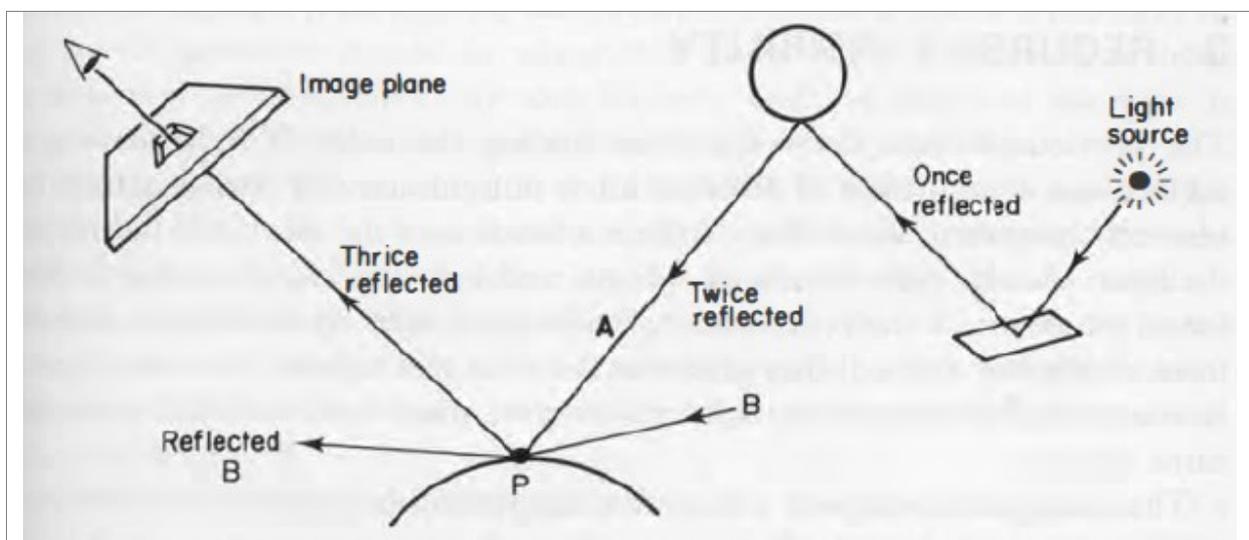


Fig. 2.3: Rappresentazione di come la luce rimbalzi tra le superfici e influenzi il punto P . Il raggio B rappresenta il raggio che parte direttamente dalla fonte luminosa e *reflected B* sarà la direzione da cui sarà visibile la luce riflessa direttamente.

Esattamente come i *reflection rays* anche i *transparency rays* hanno una sola direzione di trasmissione della luce. Quando un raggio luminoso penetra in un oggetto con trasparenza si crea un processo chiamato rifrazione per cui la luce che attraversa l'oggetto, uscirà da un altro suo punto.

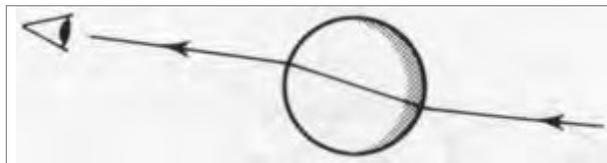


Fig. 2.4: Rappresentazione di come la luce che viene trasmessa attraverso l'oggetto colpisca l'osservatore.

In entrambi i casi è necessario determinare il colore del raggio luminoso riflesso e trasmesso, in base al colore della fonte luminosa e degli oggetti che vengono attraversati o in cui ci si riflette per aggiungerlo come contributo luminoso alla luce diretta.

2.2.1 Albero del ray tracing

Fatte tutte queste considerazioni possiamo riassumere il processo partendo da un generico punto P di una superficie colpito da un *eye ray*:

- dato P, tracciare i raggi per la luce speculare;
- se P appartiene a una superficie riflettente, tracciare il raggio di riflessione;
- se P appartiene a una superficie trasparente, tracciare il raggio di trasmissione;
- ripetere il tutto per ogni punto colpito da raggi di riflessione e di trasmissione fino a che non vi siano più elementi da controllare.

Per comprenderne meglio il funzionamento possiamo prendere in esame l'esempio nella Fig. 2.5. Dato quindi l'*eye ray* E, la prima interazione con l'ambiente è la sua intersezione con la superficie 3. Da questo punto di intersezione saranno tracciati i raggi riguardanti la luce speculare nelle direzioni delle fonti luminose S_1 e S_2 , la riflessione R_1 e di trasmissione T_1 . Questo procedimento viene eseguito ricorsivamente per ogni superficie colpita dai raggi di riflessione e trasmissione.

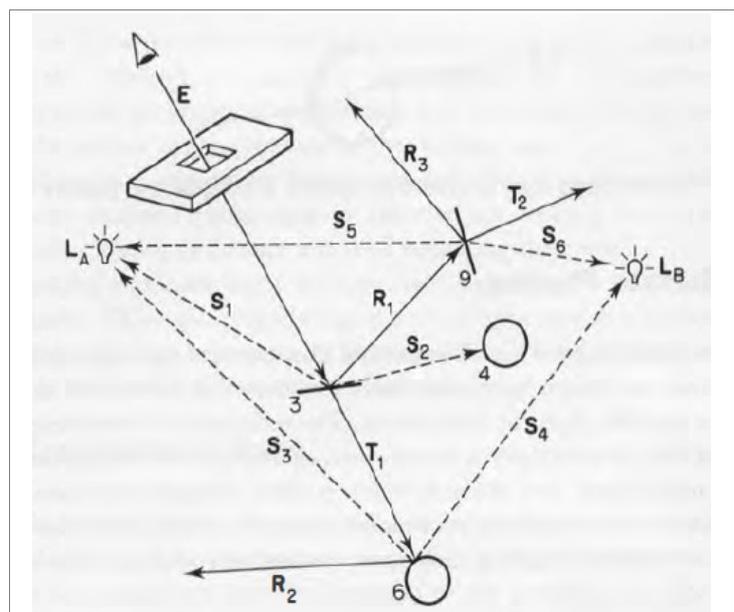
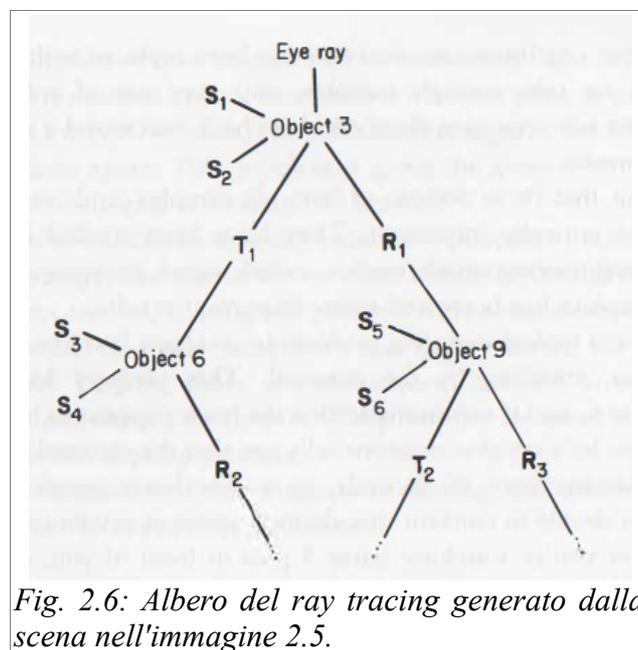


Fig. 2.5: Esempio di costruzione di un albero del ray tracing.

Nel nostro caso vedremo come il raggio di trasmissione T_1 colpisca la superficie 6. Come prima tracciamo i raggi per la luce speculare S_3 e S_4 e il raggio di riflessione R_2 . In questo caso la superficie colpita non è trasparente e non verrà tracciato alcun raggio di trasmissione. Seguendo il raggio R_2 noteremo come non colpisca alcuna superficie facendo terminare l'algoritmo per questo ramo. Seguendo invece il percorso del raggio di riflessione R_1 finiremo per colpire la superficie 9 generando i soliti raggi per la riflessione speculare S_5 e S_6 , il raggio di riflessione R_3 e, siccome la superficie è trasparente, il raggio di trasmissione T_2 . Sia il raggio di riflessione R_3 che quello di trasmissione T_2 non colpiranno alcun oggetto e avendo terminato il percorso di ogni raggio generato, l'algoritmo termina. Questo tipo di approccio consiste nel partire da un singolo raggio che, ogni volta che colpisce una superficie, ne genera altri. Senza considerare i raggi di luce speculare, i raggi riflessi e trasmessi verranno trattati allo stesso modo del raggio principale, procedendo ricorsivamente fino a che gli ultimi raggi per ogni percorso non colpiscano altre superfici. Per comprendere meglio l'algoritmo possiamo immaginarlo come una struttura ad albero dove ogni nodo, corrispondente a un oggetto della scena colpito, avrà al massimo due figli, che saranno gli altri oggetti della scena colpiti dai raggi. La struttura generata dalla Fig. 2.5 possiamo riassumerla come nella Fig. 2.6.



2.3 Calcolo delle intersezioni

Il cuore dell'algoritmo del *Ray Tracing* è sicuramente il calcolo delle intersezioni, sia che si

tratti di intersecare il raggio iniziale con il punto P della scena che della ricerca della presenza di oggetti tra il punto P e la fonte luminosa. Come abbiamo già detto, il calcolo delle intersezioni è un processo molto costoso, per cui è stato necessario trovare degli oggetti per cui i calcoli sono più semplici. Prima fra questi è la sfera.

2.3.1 Intersezione algebrica raggio-sfera

Prima di tutto definiamo un raggio in forma parametrica come:

$$R(t) = R_0 + R_d \cdot t \quad \text{con } t > 0$$

dove:

$$R_{origin} \equiv R_0 \equiv [X_0 \ Y_0 \ Z_0]$$

$$R_{direction} \equiv R_d \equiv [X_d \ Y_d \ Z_d]$$

$$\text{dove } X_d^2 + Y_d^2 + Z_d^2 = 1 \quad \text{normalizzato}$$

Per i valori di $t < 0$ i punti si troveranno dietro l'origine del raggio. È necessario normalizzare il vettore per permettere a t di essere uguale alla distanza dall'origine. In questo modo la formula rappresenterà tutti i punti sul raggio al variare del valore di t .

Dato il

$$\text{Centro della sfera} \equiv S_c \equiv [X_c \ Y_c \ Z_c]$$

$$\text{Raggio della sfera} \equiv S_r$$

la superficie della sfera sarà costituita dai punti $[X_s \ Y_s \ Z_s]$ tali che:

$$(X_s - X_c)^2 + (Y_s - Y_c)^2 + (Z_s - Z_c)^2 = S_r^2$$

Per risolvere il problema dell'intersezione, l'equazione del raggio viene sostituita nell'equazione della sfera al posto di (X_s, Y_s, Z_s)

$$X = X_0 + X_d \cdot t$$

$$Y = Y_0 + Y_d \cdot t$$

$$Z = Z_0 + Z_d \cdot t$$

$$(X_0 + X_d \cdot t - X_c)^2 + (Y_0 + Y_d \cdot t - Y_c)^2 + (Z_0 + Z_d \cdot t - Z_c)^2 = S_r^2$$

questa equazione può essere espressa in termini di t come:

$$A \cdot t^2 + B \cdot t + C = 0$$

dove:

$$A = X_d^2 + Y_d^2 + Z_d^2$$

$$B = 2 \cdot (X_d \cdot (X_0 - X_c) + Y_d \cdot (Y_0 - Y_c) + Z_d \cdot (Z_0 - Z_c))$$

$$C = (X_0 - X_c)^2 + (Y_0 - Y_c)^2 + (Z_0 - Z_c)^2 - S_r^2$$

Poiché il raggio è normalizzato, si ha $A = 1$ e conoscendo S_r^2 , l'equazione ha soluzione per t data da:

$$t_{0,1} = \frac{-B \pm \sqrt{B^2 - 4C}}{2}$$

Se il radicando è negativo non ci sono intersezioni. In alternativa si otterranno due soluzioni corrispondenti ai punti di intersezione. Se entrambi i valori sono positivi, l'intersezione corrisponde al valore di t minore. Se invece un valore di t è negativo, l'intersezione avviene prima dell'inizio del raggio e l'intersezione corrisponde al valore di t positivo. Una volta determinato quale valore di t considerare, bisogna calcolare l'effettivo punto di intersezione.

$$r_{intersect} \equiv r_i = [x_i \ y_i \ z_i] = [X_0 + X_d \cdot t, Y_0 + Y_d \cdot t, Z_0 + Z_d \cdot t]$$

il vettore normale alla superficie sarà perciò

$$r_{normal} \equiv r_n = \left[\frac{x_i - X_c}{S_r}, \frac{y_i - Y_c}{S_r}, \frac{z_i - Z_c}{S_r} \right]$$

Se il raggio ha origine all'interno della sfera r_n dovrà essere negato.

Prima di calcolare l'intersezione con la sfera, conviene:

- testare se l'origine del raggio è fuori dalla sfera;
- trovare il punto del raggio che più si avvicina alla sfera;
- se il raggio è esterno alla sfera, non esiste intersezione;
- altrimenti calcola il valore sotto la radice quadrata per calcolare i valori di t ;
- se il valore ottenuto è negativo, non abbiamo intersezione;
- altrimenti calcolare i valori di t ;
- calcolare il punto di intersezione e poi la normale.

2.3.2 Intersezione geometrica raggio-sfera

Dati i seguenti valori

distanza tra il centro della sfera e l'origine del raggio $\equiv OC = S_c - R_0$

quadrato della lunghezza di $OC \equiv L_{OC}^2 = OC \cdot OC$

se $L_{OC}^2 < S_r^2$ allora l'origine del raggio è all'interno della sfera, altrimenti sarà all'esterno e il raggio potrebbe non colpire la sfera.

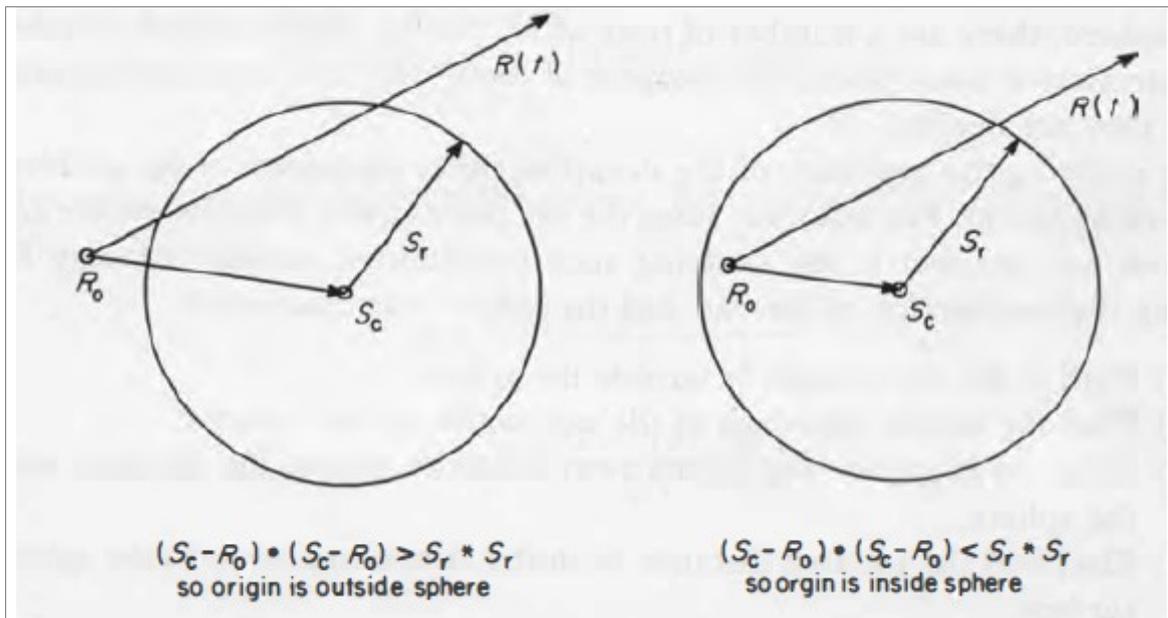


Fig. 2.7: Se se il quadrato della distanza S_r , che in questo caso corrisponde al raggio della sfera, è minore del quadrato della distanza tra l'origine del raggio e il centro della sfera: allora l'origine del raggio si trova all'esterno della sfera. Al contrario si trova all'interno.

In ogni caso il passo successivo è quello di trovare il punto sul raggio più vicino al centro della sfera che può essere tradotto nel calcolo della distanza di un punto da una retta.

$$\text{distanza del punto del raggio più vicino al centro} = t_{ca} = OC \cdot R_d$$

se il valore ottenuto è < 0 , allora il raggio punta lontano dalla sfera e l'algoritmo termina.

Con $t_{ca} > 0$ e data la distanza dal punto alla superficie della sfera

$$t_{hc}^2 = S_r^2 - D^2 \text{ dove } D \text{ è la distanza tra } t_{ca} \text{ e il centro della sfera}$$

$$D^2 = L_{OC}^2 - t_{ca}^2$$

otterremo

$$t_{hc}^2 = S_r^2 - L_{OC}^2 + t_{ca}^2$$

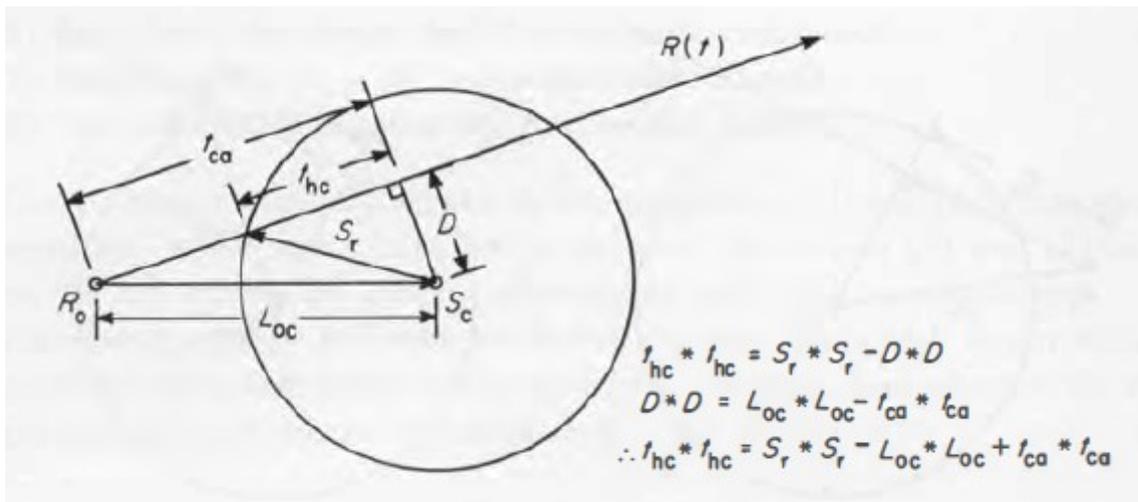


Fig. 2.8: Calcolo del quadrato della distanza t_{hc} . Conoscendo L_{oc} e t_{ca} possiamo ricavare la distanza D che verrà elevato alla seconda e successivamente sottratto al quadrato di S_r per ricavare t_{hc} .

se il valore ottenuto è < 0 , il raggio non colpisce la sfera. A questo punto potremmo calcolare semplicemente la distanza dei punti di intersezione

$$t = t_{ca} \pm \sqrt{t_{hc}^2}$$

2.3.3 Confronto tra i metodi

Poniamo

$$B = -2 \cdot t_{ca}$$

$$C = L_{oc}^2 - S_r^2$$

e confrontiamo l'equazione

$$t = t_{ca} \pm \sqrt{t_{hc}^2}$$

$$t = \frac{-B}{2} \pm \sqrt{\left(\frac{-B}{2}\right)^2 - C}$$

$$t = \frac{-B \pm \sqrt{B^2 - 4C}}{2}$$

A partire da questa soluzione algebrica, più complessa a livello di calcolo rispetto a quella geometrica, possiamo effettuare delle semplificazioni. Sostituendo B con $-2 \cdot NB$, dove

$$NB = -\frac{B}{2} \quad \text{si ha:}$$

$$t = \frac{-(-2 \cdot NB) \pm \sqrt{(-2 \cdot NB)^2 - 4C}}{2}$$

$$t = \frac{-(-2 \cdot NB) \pm 2 \cdot \sqrt{NB^2 - C}}{2}$$

$$t = NB \pm \sqrt{NB^2 - C}$$

sostituendo NB con B della formula algebrica

$$NB = X_d \cdot (X_c - X_0) + Y_d \cdot (Y_c - Y_0) + Z_d \cdot (Z_c - Z_0)$$

dove NB è equivalente a t_{ca} .

Per rimuovere il \pm e sapendo che l'origine del raggio sarà fuori dalla sfera solo per $C > 0$, elimineremo la necessità di calcolare t_0 e t_1 . Se $NB < 0$, il raggio punterà lontano dalla sfera.

Quando dobbiamo risolvere un problema di intersezione è sempre utile analizzare la natura del problema e procedere in modo da velocizzare il più possibile il calcolo. La soluzione algebrica risolve generalmente il problema delle intersezioni, mentre invece il metodo geometrico è incentrato sulle caratteristiche del raggio e sui requisiti del problema.

2.3.4 Problemi di precisione

Lavorando in floating-point è molto probabile che vi sia una perdita di dati e spesso accade che l'origine del raggio sia un punto della sfera stessa. In questi casi $t = 0$ viene ignorato come punto di intersezione e quando vengono calcolati i valori di t , spesso avremo valori non esattamente uguali a 0 e cioè causa dei problemi nel momento in cui tracciamo i raggi luminosi. Infatti potrà erroneamente risultare un punto della superficie ombreggiato da se stessa.

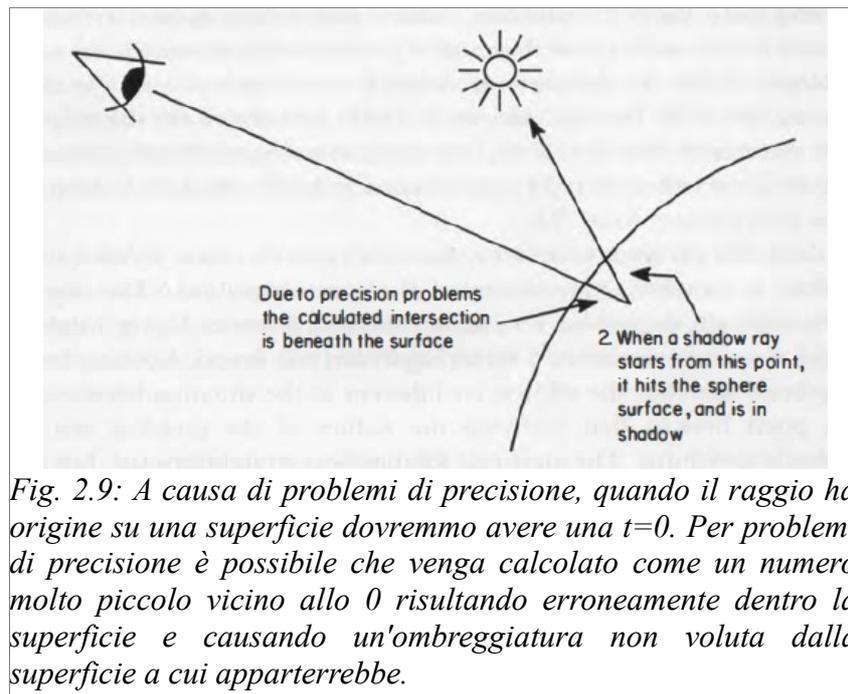


Fig. 2.9: A causa di problemi di precisione, quando il raggio ha origine su una superficie dovremmo avere una $t=0$. Per problemi di precisione è possibile che venga calcolato come un numero molto piccolo vicino allo 0 risultando erroneamente dentro la superficie e causando un'ombreggiatura non voluta dalla superficie a cui apparterebbe.

Un metodo per evitare che ciò accada è quello di segnalare quando l'origine è sulla superficie della sfera. In alternativa si potrebbe controllare se il valore di t è minore di un certo minimo che ci siamo posti per decidere se effettivamente sarà sulla superficie della sfera o no.

2.3.5 Intersezione raggio-piano

Definito il raggio, specificando l'origine e la direzione

$$R_{origin} \equiv R_o \equiv [X_o Y_o Z_o]$$

$$R_{direction} \equiv R_d \equiv [X_d Y_d Z_d]$$

dove $X_d^2 + Y_d^2 + Z_d^2 = 1$ (normalizzato)

i punti sul raggio possono essere rappresentati in forma parametrica come:

$$R(t) = R_o + R_d \cdot t \quad \text{con } t > 0$$

Come nel caso della sfera, t rappresenta la distanza dall'origine del raggio.

Definito il piano in termini di $[A B C D]$ dove il piano è

$$piano \equiv A \cdot x + B \cdot y + C \cdot z + D = 0$$

e $A^2 + B^2 + C^2 = 1$

Il vettore unitario normale del piano viene definito come

$$P_{normal} \equiv P_n = [A B C]$$

e la distanza dall'origine al piano è semplicemente D e il segno del suo valore determina la

posizione del piano. La distanza dall'origine del raggio all'intersezione del piano si calcola sostituendo $R(t)$ nell'equazione del piano

$$A \cdot (X_0 + X_d \cdot t) + B \cdot (Y_0 + Y_d \cdot t) + C \cdot (Z_0 + Z_d \cdot t) + D = 0$$

e risolvendo per t otterremo

$$t = -\frac{A \cdot X_0 + B \cdot Y_0 + C \cdot Z_0 + D}{A \cdot X_d + B \cdot Y_d + C \cdot Z_d}$$

che in notazione vettoriale si può tradurre con

$$t = -\frac{P_n \cdot R_0 + D}{P_n \cdot R_d}$$

Se $v_d = P_n \cdot R_d = 0$, il raggio è parallelo al piano e non ci sono intersezioni. Se $v_d > 0$ la normale al piano punta lontano dal raggio. Altrimenti si calcola il valore di t e se questo è < 0 l'intersezione avviene prima dell'origine, quindi non viene considerata. In alternativa viene calcolata l'intersezione

$$r_i = [x_i \ y_i \ z_i] = [X_0 + X_d \cdot t, Y_0 + Y_d \cdot t, Z_0 + Z_d \cdot t]$$

2.3.6 Intersezione con un poligono

Una volta che abbiamo stabilito l'intersezione raggio-piano è il momento di stabilire se abbiamo intersezione con un preciso poligono. L'algoritmo che utilizzeremo consiste nel inviare un raggio in una direzione arbitraria e contare il numero di intersezioni con i segmenti incontrati. Se il numero di segmenti incontrati è dispari, ci troveremo all'interno del poligono, altrimenti all'esterno.

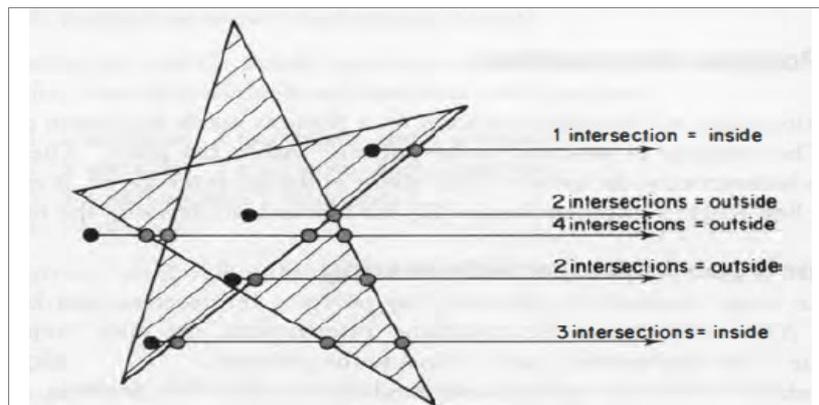


Fig. 2.10: Esempi di test di intersezione con un poligono tracciando un raggio in una direzione arbitraria dal punto di intersezione del raggio intersecato col piano. Se il numero di intersezioni tra il raggio e i bordi del poligono è pari, l'intersezione raggio-piano è all'esterno del poligono. Se è dispari è all'interno.

Definiti

un poligono come una serie di N punti $G_n \equiv [X_n Y_n Z_n]$ dove $n = (0, 1, \dots, N-1)$

il piano $A \cdot X + B \cdot Y + C \cdot Z + D = 0$

la normale al piano $P_{normal} \equiv P_n = [A B C]$

e l'intersezione col piano $R_i \equiv [X_i Y_i Z_i]$ sul piano $[A B C D]$

il primo passo è quello di trasformare il poligono in un piano in due dimensioni. Per farlo ruoteremo gli assi finché uno di essi non sarà parallelo al piano in modo da creare le coordinate (U, V) di ogni punto sulle altre due dimensioni rimaste. Un modo per scegliere quale coordinata togliere è quello eliminare quella che nel vettore della normale ha il valore assoluto più alto.

2.4 Fisica delle superfici

Quando parliamo di fisica delle superfici intendiamo tutte quelle proprietà che influenzano il modo in cui noi la vediamo, come il colore, i raggi luminosi che la colpiscono e di come il colore cambi in base a questi. In particolare si parla delle frequenze e lunghezze d'onda, il modo in cui i colori variano in base ai fotoni che colpiscono la superficie e la riflessione e diffusione della luce.

2.4.1 Frequenza e lunghezza d'onda

Esistono in generale due principali modelli che descrivono i comportamenti della luce: *wave model*, che paragona la luce alle onde dell'acqua e il *particel model* secondo cui la luce è un insieme

di particelle. Molte tecniche di ray tracing si basano sul secondo metodo che vede la luce come un percorso rettilineo di particelle di luce. Entrambi non sono in grado di descrivere esattamente il comportamento della luce poiché in alcuni casi si comporta come un'onda e in altri come particelle. Una particella di luce viene detta fotone e ha un particolare movimento vibratorio. Ogni fotone ha una particolare frequenza di vibrazione strettamente legata alla frequenza d'onda. Assumendo di avere un fotone che vibra secondo uno schema e si muove nello spazio e dato un punto di inizio del movimento vibratorio (punto A), il fotone seguirà un percorso ondulatorio fino a tornare alla stessa posizione da cui eravamo partiti (punto B).



Fig. 2.11: Lunghezza d'onda di un fotone data da un intero ciclo del suo movimento.

Se il fotone si muove a velocità costante, ogni volta che completa il suo movimento da A a B si dice che ha completato il *ciclo di vibrazione*. Questa distanza è la lunghezza d'onda. All'aumentare della frequenza, il fotone impiegherà meno tempo a completare il giro e questo occuperà meno spazio. Sapendo che la velocità della luce è costante, possiamo dire che

$$\lambda = \frac{c}{f}$$

dove λ è la lunghezza d'onda, c la velocità della luce $c \approx 3,00 \times 10^8 \text{ m s}^{-1}$ e f la frequenza. Inoltre è importante sapere che l'energia di un fotone è direttamente collegata alla sua frequenza

$$E = hf$$

dove E è l'energia e h è una costante $h \approx 6,63 \times 10^{-34} \text{ J s}$

da ciò possiamo dedurre che esiste una correlazione tra la frequenza (e l'energia) e il colore percepito.

2.4.2 Illuminazione delle superfici e colore

È importante studiare in che modo la luce interagisce con le superfici. Molti dettagli non vengono correttamente visualizzati, ma esistono molti modi per approssimare le immagini nel miglior modo possibile. Come abbiamo già precedentemente accennato, esistono diversi tipi di interazioni della luce con le superfici: riflessione speculare, riflessione diffusiva, trasmissione

speculare, trasmissione diffusiva. Tutti questi tipi di interazione influiscono insieme alle caratteristiche del materiale sul calcolo del colore finale. Molti colori, come ad esempio il bianco, non vengono generati da un unico fotone ma da un gruppo di fotoni di colori e frequenze diversi che vengono percepite dall'occhio come un unico colore. Ogni tipo di luce viene generato da una diversa combinazione di frequenze di fotoni diversi. A questo punto è possibile impostare uno strumento per contare il numero medio di fotoni a ogni lunghezza d'onda visibile e creare un grafico. Tale grafico rappresenta lo *spettro*.

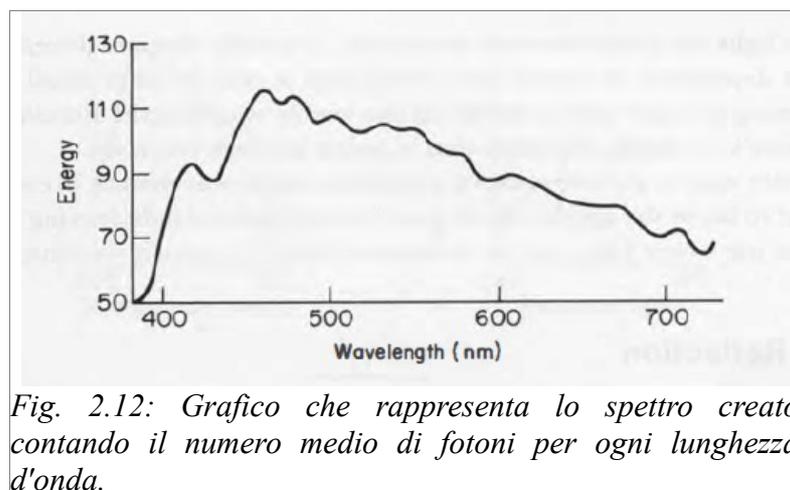


Fig. 2.12: Grafico che rappresenta lo spettro creato contando il numero medio di fotoni per ogni lunghezza d'onda.

Quando si parla di intensità della luce, ci si riferisce alla quantità di fotoni in una determinata lunghezza d'onda. Un buon modo per rappresentare meglio questi dati è quello di assegnargli un raggio anche se in questo caso avremo problemi nella rappresentazione corretta della rifrazione. Per ovviare il problema è possibile generare un raggio per ogni gruppo di fotoni della stessa lunghezza d'onda.

2.4.3 Riflessione

Dato un oggetto d'oro e una fonte luminosa nella scena, possiamo descrivere il fenomeno della riflessione da una particolare angolazione disegnando uno spettro di riflessione, che indica la percentuale della luce che arriva alla superficie riflessa per ogni lunghezza d'onda. Quindi, per trovare la quantità di luce riflessa, che in questo caso è una luce dorata, moltiplicheremo la quantità di luce che arriva alla superficie per la percentuale di riflessione del materiale.

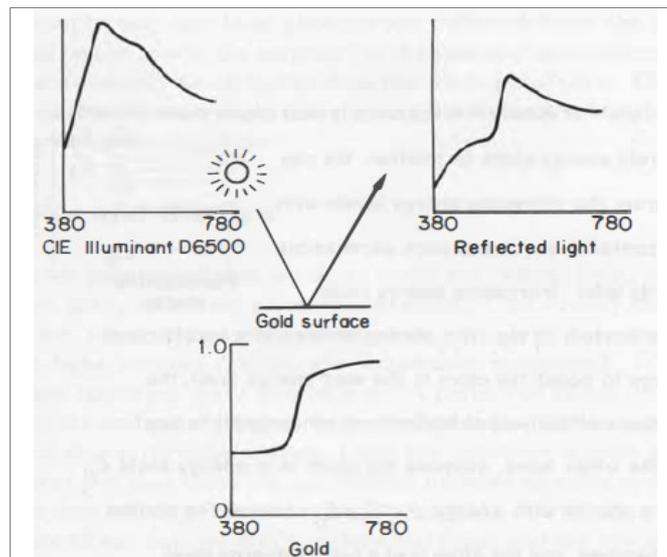


Fig. 2.13: Rappresentazione di come viene riflessa la luce in un oggetto dorato illuminato dallo spettro dell'immagine 2.m. Lo spettro viene moltiplicato con lo spettro della superficie per ogni lunghezza d'onda realizzando lo spettro di riflessione.

Questo accade perché, come abbiamo detto, i fotoni vibrano e per ogni oggetto esiste una frequenza per cui vibrano, chiamata *resonant frequency*. Dati due oggetti che hanno la stessa *resonant frequency*, di cui uno vibra e l'altro no, se messi vicini anche l'oggetto che non vibrava inizierà a farlo. Se si considera un fotone che vibra contro la superficie di un oggetto, gli atomi dell'oggetto che già vibrano, otterranno parte dell'energia dei fotoni. Se il fotone ha abbastanza energia da portare l'atomo a un livello di energia superiore, il fotone verrà assorbito elevando l'atomo. In alternativa l'atomo assorbirà l'energia per poco tempo irradiandola poi sotto forma di calore. Se il fotone è però alla giusta frequenza, l'atomo assorbirà l'energia vibrando al livello successivo e distruggendo il fotone. Successivamente l'energia verrà rilasciata fino a tornare a quella iniziale. A questo punto verrà creato un nuovo fotone portante l'energia rilasciata. Questo fenomeno spiega per quale motivo un oggetto è di un determinato colore e come la luce viene riflessa.

2.4.4 Tipi di trasmissione della luce

Abbiamo già accennato ai modi in cui avviene la trasmissione della luce che tocca una superficie. Quando ne studiamo il comportamento ci interessiamo a una caratteristica specifica di ogni punto dell'oggetto che è quella della *normale alla superficie*. La normale è un vettore che

indica la direzione perpendicolare alla superficie in un punto. Quando trattiamo la riflessione speculare per le superfici luminose, abbiamo detto che l'angolo di riflessione θ_r è uguale all'angolo creato dal raggio di incidenza θ_i della luce e la normale. Algebricamente lo scriveremo come

$$R = \alpha I + \beta N$$

$$\theta_i = \theta_r$$

quindi dati $\cos(\theta_i) = -I \cdot N$ e $\cos(\theta_r) = N \cdot R$ otterremo

$$\begin{aligned} \cos(\theta_i) &= \cos(\theta_r) \\ -I \cdot N &= N \cdot R \\ &= N \cdot (\alpha I + \beta N) \\ &= \alpha(N \cdot I) + \beta(N \cdot N) \end{aligned}$$

e sapendo che $|N| = 1$ e $N \cdot N = 1$ avremo

$$= \alpha(N \cdot I) + \beta$$

impostando $\alpha = 1$ otterremo $\beta = -2(N \cdot I)$

la formula completa della direzione di riflessione speculare sarà

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

in cui I è il raggio di incidenza, N la normale e R il raggio di riflessione.

Quando stiamo invece trattando con delle superfici ruvide, il comportamento della luce è molto diverso. Abbiamo già parlato di come un fotone che colpisce un atomo possa essere assorbito o re-irradiato. Nel secondo caso non abbiamo una vera e propria direzione di riflessione, perciò quando un raggio luminoso (quindi una grande quantità di fotoni) colpisce una superficie di questo tipo, i fotoni vengono re-irradiati in tutte le direzioni con uguale intensità. L'unica parte inerente alla geometria è quella riguardante l'ampiezza della riflessione. Ciò dipende dall'ampiezza dell'angolo creato tra il raggio luminoso incidente e la normale. La quantità di luce sarà proporzionale al coseno di questo angolo.

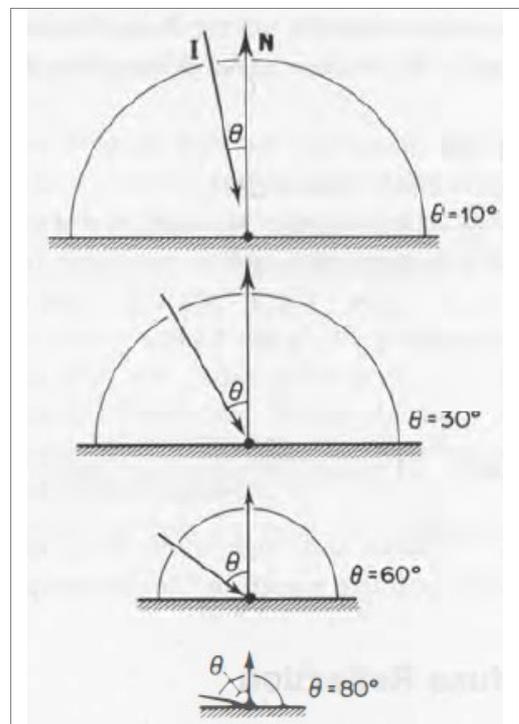


Fig. 2.14: In base all'angolo di incidenza varierà l'ampiezza di riflessione della luce.

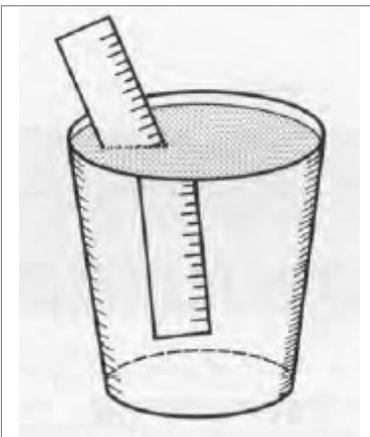


Fig. 2.15: Esempio di trasmissione.

Nel caso di oggetti trasparenti, quando i raggi luminosi colpiscono la superficie questi passeranno attraverso all'oggetto. In Fig. 2.15 si nota l'effetto di cui stiamo parlando nel momento in cui il righello all'interno del bicchiere passa dal "toccare" l'aria, all'acqua. Questo fenomeno è detto trasmissione, o rifrazione. Per comprendere meglio come funziona introdurremo il concetto di indice di rifrazione che descrive la velocità della luce in quel determinato materiale. Nel calcolare come la luce viene rifratta per trasmissione, introduciamo la legge di Snell.

Dati un punto sulla superficie P , la sua normale N e l'angolo di incidenza θ_i , e conoscendo l'indice di rifrazione di entrambi i materiali n_1 ed n_2 , esiste una relazione tra questi elementi e l'angolo creato dalla direzione della rifrazione della luce θ_t .

$$\frac{\sin(\theta_i)}{\sin(\theta_t)} = n_{21} = \frac{n_2}{n_1}$$

$$T = \alpha I + \beta N$$

$$\text{e } n_{21} = \frac{1}{n_{12}}$$

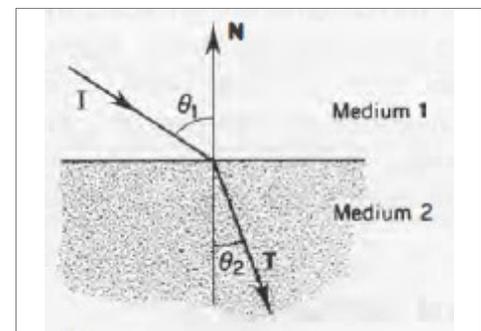


Fig. 2.16: Legge di Snell della rifrazione.

Inoltre l'indice di rifrazione dipende anche della lunghezza d'onda del raggio luminoso, perciò nei prismi la luce incidente viene divisa e le diverse lunghezze d'onda vengono rifratte in quantità diverse.

Oltre a questi fenomeni, ne esiste un altro chiamato *total internal reflection* che avviene nel momento in cui la luce cerca di passare da un oggetto di media densità a uno di densità minore a un angolo troppo stretto. Il risultato è che la luce invece che venire trasmessa avrà anche un effetto speculare.

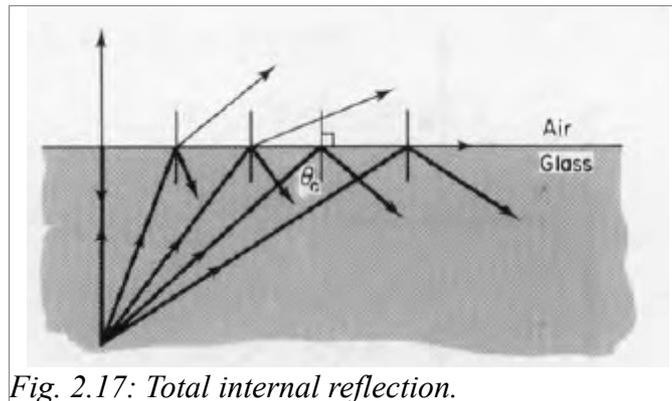


Fig. 2.17: Total internal reflection.

Considerata la precedente premessa sul calcolo della luce trasmessa, è il momento di trovare i valori di α e β in $T = \alpha I + \beta N$.

2.4.5 Precisazioni sulla riflessione e sulla propagazione del colore

Fino ad ora abbiamo fatto delle importanti semplificazioni nella spiegazione di questi fenomeni assumendo che esista perfettamente una singola direzione di riflessione R in cui la luce speculare, scontrandosi contro la superficie di un oggetto, si dirige. Ma nel caso in cui, più realisticamente, la superficie sia anche solo lievemente ruvida questa affermazione non è più vera. La luce che colpisce i punti vicini a quello di perfetta riflessione, verrà comunque riflessa parallelamente alla direzione analizzata. Inoltre abbiamo anche ignorato l'influenza del colore della luce che colpisce la superficie. Data ad esempio una scena in cui sono presenti un contenitore di metallo e una luce bianca posta sopra l'inizio della direzione di vista, otterremo un effetto per cui, nonostante la riflessione, esisteranno zone che saranno più tendenti al colore del metallo stesso che a quello della luce. Se vogliamo creare delle immagini realistiche è necessario tenere conto di queste caratteristiche. Prima di tutto dobbiamo considerare il fatto che la natura dell'interazione tra la luce incidente e la superficie dipende sia della lunghezza d'onda che dell'angolo di incidenza. Poi bisogna considerare che la quantità di colore può variare in base alla posizione della vista.

2.4.5.1 Riflessione diffusiva $I_{dr}(\lambda)$

Come abbiamo già detto parlando della riflessione diffusiva, il colore e la luce emessa dalla superficie dipendono dall'angolo di incidenza del raggio luminoso con la superficie e la normale e siccome la luce viene re-irradiata in tutte le direzioni, non ha importanza da che angolazione guardiamo l'oggetto.

$$I_{dr}(\lambda) = I_{lj}(\lambda) F_{dr}(\lambda) (N \cdot L) \quad \text{in cui}$$

$I_{dr}(\lambda)$ è la luce diffusiva riflessa

$I_{lj}(\lambda)$ è lo spettro della sorgente luminosa j

$F_{dr}(\lambda)$ è la curva di riflessione

N e L rispettivamente la normale e il vettore del raggio luminoso

2.4.5.2 Riflessione speculare $I_{sr}(\lambda)$

Nel caso di riflessione speculare in presenza di una superficie ruvida è necessario trattare la superficie come se fosse un insieme di tanti piccoli riflettori. Nella Fig. 2.18 abbiamo degli esempi del comportamento della luce in diversi casi. Le immagini mostrano una superficie ruvida colpita

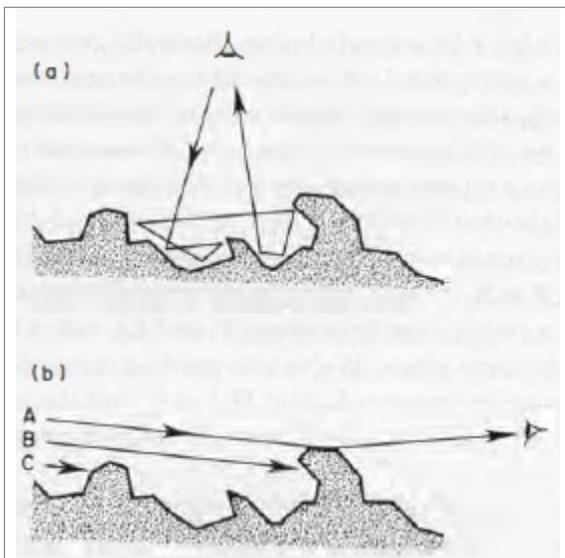


Fig. 2.18: Rimbalzo della luce tra le sfaccettature delle superfici ruvide. Nel caso (a) i troppi rimbalzi influenzeranno il colore della luce portandolo a diventare sempre più simile a quello del materiale. Nel caso (b) il colore della luce sarà ben visibile.

dal raggio luminoso che seguendo la riflessione speculare del raggio stesso rimbalzerà sulle piccole sfaccettature della superficie illuminando leggermente ogni punto colpito. Nel caso in cui i rimbalzi siano pochi, l'effetto potrà essere percettibile. Nel caso della figura 2.18(b) infatti vedremo direttamente il colore della luce. Se al contrario, come nel caso della figura 2.18(a), vi sono molti rimbalzi, la luce verrà lentamente influenzata dal colore del materiale attenuando sempre di più il colore della luce. Riassumendo possiamo dire che il colore e l'intensità della luce speculare che viene riflessa in un determinato punto dipendono dalla direzione e dal colore della luce che colpisce la superficie, dal colore dell'oggetto e da come sono disposte le microfaccettature.

Per effettuare questo tipo di calcolo molto complesso partiamo da una classica situazione di vista. Come sempre, dati un punto P e una fonte luminosa, dobbiamo calcolare la luce proveniente dalla fonte luminosa L_j che viene riflessa nella direzione dell'osservatore V . A questo punto aggiungeremo un vettore H_j tra i vettori V e L_j in modo tale che tra H_j e gli altri due vettori si creino due angoli della stessa ampiezza θ e evidenzieremo l'angolo creato tra H_j e N . Ricordando che

la riflessione speculare perfetta la abbiamo solo quando l'angolo di incidenza è uguale a quello di riflessione, finché H_j crea due angoli θ uguali, otterremo questo tipo di riflessione per tutti i vettori paralleli a H_j . Nel momento in cui la luce tocca una superficie, la sua intensità diminuisce in base alla lunghezza d'onda e all'angolo di incidenza. Per comprendere come e quanto una luce viene colorata dal materiale toccato prima di venire riflessa utilizzeremo la seguente formula $I_{sr}(\lambda)$.

$$I_{sr}(\lambda) = \frac{F(\lambda, \Theta)}{\pi} \frac{D(\alpha) G(N, V, L)}{(N \cdot L)(N \cdot V)}$$

dove

- $G(N, V, L)$ descrive l'orientamento della sfaccettatura e serve a calcolare anche l'eventualità che la luce riflessa da una sfaccettatura rimbalzi su altre sfaccettature prima di lasciare la superficie. H_j , il vettore normalizzato che riflette perfettamente la luce dalla fonte all'osservatore, è fondamentale per questo calcolo

$$H = \frac{(L+V)}{|L+V|}$$

$$\gamma = \frac{2(N \cdot H)}{(V \cdot H)}, G = \min\{1, \gamma(N \cdot V), \gamma(N \cdot L)\}$$

- $D(\alpha)$ descrive quante sfaccettature della superficie sono orientate nella direzione

di H . Il valore di D dipende da α , che corrisponde all'angolo creato tra L e H , e una variabile m che descrive la ruvidità della superficie. Più il valore di m è basso, più la superficie è liscia. Esistono due diverse formule per il calcolo di D :

- la formula più veloce è la seguente

$$D(m, c, \alpha) = ce^{-(\alpha/m)^2}$$

dove c è una costante scelta arbitrariamente e e è la costante di Eulero

- nel caso potessimo permetterci di spendere più tempo potremmo usare questa formula più accurata

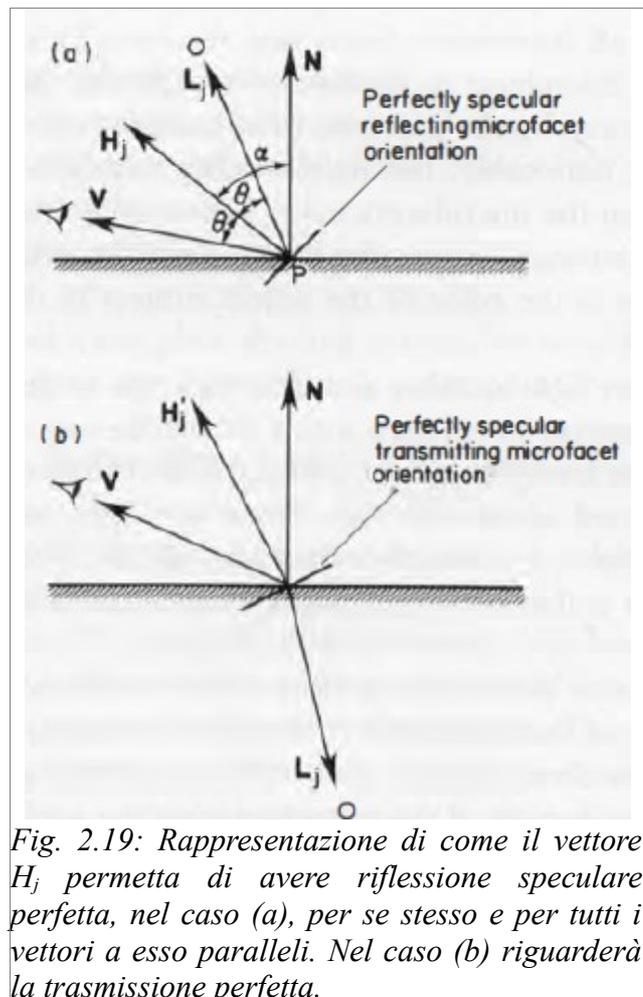


Fig. 2.19: Rappresentazione di come il vettore H_j permetta di avere riflessione speculare perfetta, nel caso (a), per se stesso e per tutti i vettori a esso paralleli. Nel caso (b) riguarderà la trasmissione perfetta.

$$D(m, \alpha) = \frac{1}{m^2 \cos^4(\alpha)} e^{-\left[\frac{\tan(\alpha)}{m}\right]^2}$$

Come spesso accade, la superficie può avere diversi gradi di ruvidità. In quel caso useremo la seguente formula

$$D = \sum_k w_k D(m_k, \alpha)$$

- $F(\lambda, \theta)$ descrive la riflessione a una lunghezza d'onda λ a un angolo θ

$$F(g, c) = \frac{1}{2} \frac{(g-c)^2}{(g+c)^2} \left\{ 1 + \frac{[c(g+c)-1]^2}{[c(g-c)-1]^2} \right\}$$

dove

$$c = \cos(\theta) = V \cdot H$$

$$g^2 = n_\lambda^2 + c^2 - 1$$

n_λ = indice di rifrazione a una determinata lunghezza d'onda

2.4.5.3 Riepilogo sul modello di shading

Abbiamo trovato come calcolare la luce diffusiva e quella speculare. Dalle due formule per entrambi i tipi di contributi luminosi possiamo trovare delle analogie per ricavare la formula della trasparenza diffusiva e speculare finendo per avere sette equazioni: due di riflessione, due di rifrazione, due per calcolare la riflessione e rifrazione da altri oggetti e una per raggrupparle.

2.4.5.4 Semplificazioni del modello

Essendo il modello appena analizzato un processo molto lungo, anche se permette di avere un effetto realistico, dobbiamo utilizzare modelli più semplici anche se con risultati meno soddisfacenti.

2.4.5.5 Hall Shading

Questa tecnica consiste nel suddividere la luce incidente in due classi: quella che arriva direttamente dalla fonte luminosa e quella che arriva riflessa da altri oggetti. Avendo quattro diversi tipi di illuminazione tra speculare e diffusiva, e riflessa e trasmessa avremo ben otto tipi di illuminazione.

	Lightsources	Other bodies
Specular reflection	$k_{sr} \sum_j I_{t_j}(\lambda) F_{sr}(\lambda, \theta_r) (N \cdot H_j)^n$	$k_{sr} I_{sr}(\lambda) F_{sr}(\lambda, \theta_r) T_r^{\Delta sr}$
Specular transmission	$k_{st} \sum_j I_{t_j}(\lambda) F_{st}(\lambda, \theta_t) (N \cdot H_j')^{n'}$	$k_{st} I_{st}(\lambda) F_{st}(\lambda, \theta_t) T_t^{\Delta st}$
Diffuse reflection	$k_{dr} \sum_j I_{t_j}(\lambda) F_{dr}(\lambda) (N \cdot L_j)$	$k_a I_a(\lambda) F_a(\lambda)$
Diffuse transmission	-	

Fig. 2.20: Formule di riflessione e trasmissione diffusiva e speculare diretta e proveniente da altri oggetti.

Qui di seguito abbiamo il calcolo del modello utilizzando le equazioni sopra presentate:

$$\begin{aligned}
 I(\lambda) = & k_{sr} \sum_j I_{t_j}(\lambda) F_{sr}(\lambda, \theta_r, j) (\cos \theta_r, j)^n \\
 & + k_{st} \sum_j I_{t_j}(\lambda) F_{st}(\lambda, \theta_t, j) (\cos \theta_r, j)^{n'} \\
 & + k_{dr} \sum_j I_{t_j}(\lambda) F_{dr}(\lambda) (N \cdot L_j) \\
 & + k_{sr} I_{sr}(\lambda) F_{sr}(\lambda, \theta_R) T_r^{\Delta sr} + k_{st} I_{st}(\lambda) F_{st}(\lambda, \theta_T) T_t^{\Delta st} + k_{dr} I_a(\lambda) F_{dr}(\lambda)
 \end{aligned}$$

dove

k_{dr}, k_{sr}, k_{st} sono i coefficienti di riflessione e trasmissione speculare e diffusiva

$I_{sr}(\lambda), I_{st}(\lambda), I_{t_j}(\lambda)$ sono gli spettri dei raggi riflessi, trasmessi e della sorgente j

$F_{dr}(\lambda), F_{sr}(\lambda), F_{st}(\lambda)$ sono le curve di riflessione diffusiva e speculare, e quella di trasmissione speculare

$\theta_{r,j}$ e $\theta_{t,j}$ rispettivamente $\cos^{-1}(N \cdot H_j)$ e $\cos^{-1}(N \cdot H_j')$

θ_R e θ_T sono gli angoli tra la normale e i raggi di riflessione e trasmissione

T_r e T_t la trasmissività di ogni unità di lunghezza del supporto contenente i raggi riflessi e trasmessi

Δ_{sr} e Δ_{st} distanza che attraversano i raggi riflesso e trasmesso

n e n' coefficienti di evidenziazione della riflessione e trasmissione speculari

H_j e H_j' i vettori per avere una perfetta riflessione e trasmissione della luce j

$$k_{dr} + k_{sr} = 1$$

$$k_{dt} + k_{st} = 1$$

$$0 \leq k_{dr}, k_{sr}, k_{dt}, k_{st}, T_r, T_t \leq 1$$

e tutti i vettori sono normalizzati.

2.5 Aliasing

Quando visualizziamo delle immagini a schermo viene a crearsi un fenomeno chiamato aliasing. Si tratta di un effetto tipo scalettatura nella visualizzazione a schermo degli oggetti causata dal fatto che lo schermo è composto da pixel che hanno una certa dimensione. All'aumentare della risoluzione dello schermo l'effetto di scalettatura viene attenuato e aumenterà la precisione di rappresentazione degli oggetti.

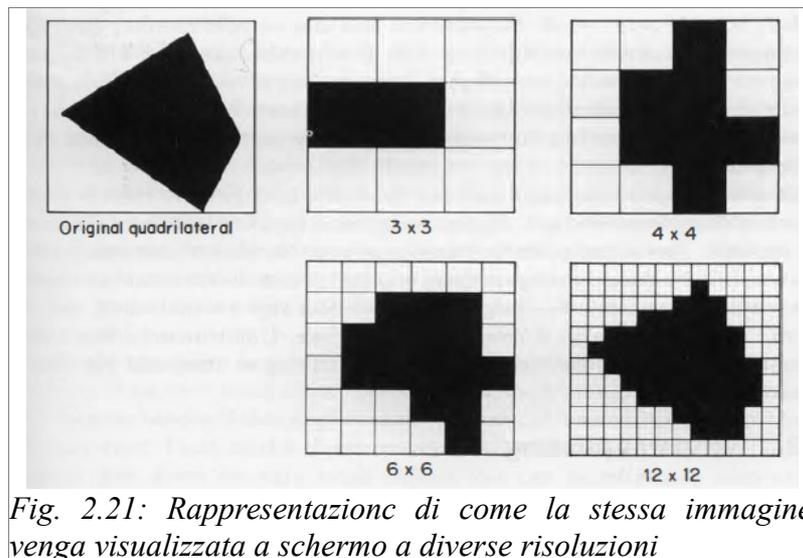


Fig. 2.21: Rappresentazione di come la stessa immagine venga visualizzata a schermo a diverse risoluzioni

L'aumento della risoluzione non è però sufficiente per eliminare l'aliasing. Trattando in un contesto di *spazio* la stessa immagine creata in uno schermo con una buona risoluzione è improbabile che possa essere visualizzata in modo chiaro su un altro dispositivo qualsiasi. Un altro problema causato sempre dalla bassa risoluzione è quello di non colpire con i raggi oggetti molto piccoli e ciò porta alla loro non visualizzazione nell'immagine finale.

2.5.1 Anti-aliasing

Il problema dell'aliasing nasce dal voler rappresentare un fenomeno continuo con degli strumenti che lavorano in modo discreto.

I migliori metodi di attenuazione dell'aliasing in un processo di ray tracing sono:

- *Supersampling*: viene aumentato il numero di raggi per ogni pixel e viene determinato il colore del pixel come media dei valori ottenuti.

- *Adaptive supersampling*: funziona allo stesso modo del supersampling, ma il numero di raggi da far passare per un pixel viene scelto dinamicamente. Si comincia utilizzando cinque raggi fissi di cui uno al centro e gli altri quattro ai vertici. Se ciascuno dei raggi colpisce colori molto simili, si presume che tocchino tutti la stessa superficie. In caso contrario, è necessario suddividere le zone che più si differenziano come colore in una zona più piccola e trattarla come se fosse un pixel, cioè facendovi passare i raggi per analizzare meglio i colori colpiti. Infine il colore verrà calcolato come media pesata.

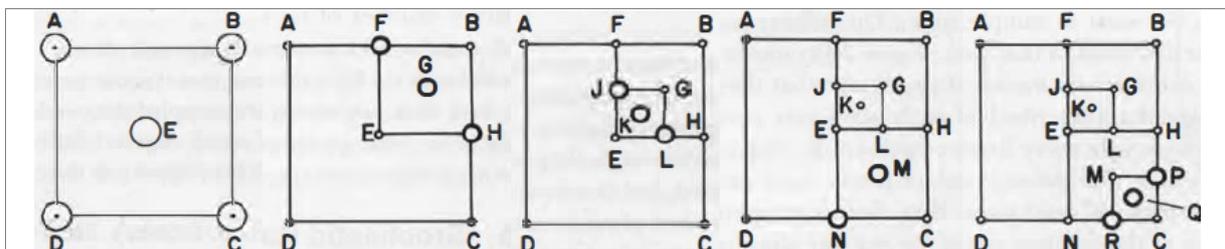


Fig. 2.22: Illustrazione del funzionamento dell'adaptive supersampling. A partire dai cinque raggi iniziali, nelle zone in cui vengono percepiti diversi colori vengono tracciati nuovi raggi per analizzare meglio la zona che copre ogni colore.

- *Stochastic Ray tracing*: questo metodo risolve il problema dei due precedenti, cioè che i raggi vengono inviati secondo una griglia regolare. Questo metodo inizia, come nel caso precedente, utilizzando un numero fisso di raggi. In questo caso però se ne useranno nove e saranno sparpagliati all'interno del pixel. Se distribuiamo i raggi con un determinato criterio possiamo chiamare questo metodo *distributed ray tracing*. Il *stochastic ray tracing* ci aiuta a ottenere il *motion blur*, la profondità di campo, e bordi morbidi sulle ombre. L'unico problema a riguardo è quello del *rumore*, cioè l'inesattezza del colore dei pixel, poiché i raggi vengono inviati in maniera casuale. Può capitare inoltre che per alcuni pixel non sia utile inviare tutti i raggi stabiliti.

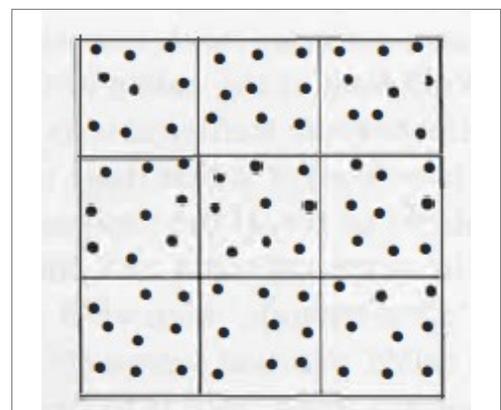


Fig. 2.23: Illustrazione di come nello Stochastic Ray Tracing per ogni pixel venga tracciato un numero fisso di raggi in modo casuale.

- *Statistical supersampling*: questo metodo consiste nel trovare il numero corretto di raggi da utilizzare per ogni pixel. Inizia con l'utilizzo di quattro raggi distribuiti uniformemente. A

questo punto vengono fatti dei test statistici sui colori ottenuti. Se si ritiene che i raggi siano sufficienti a rappresentare ciò che il pixel rappresenta nella scena, cioè che il colore ottenuto dai colori dei raggi inviati rispecchi il punto nella scena, il metodo termina. In caso contrario vengono inviati nuovi raggi finché non si arriva a un colore ritenuto giusto per quel punto.

Abbiamo già parlato di come lo schermo lavori con strumenti discreti: i pixel. La loro dimensione determina un limite massimo alle frequenze rappresentate chiamato *Nyquist limit*. Nel caso volessimo visualizzare frequenze maggiori di questo valore potranno prodursi problemi di aliasing. Quando dobbiamo renderizzare esistono due tipi di approcci. L'approccio analitico permette di filtrare le alte frequenze, ma ha il difetto di essere un calcolo molto complicato e lungo. Al contrario, abbiamo l'approccio discreto, come ad esempio il ray tracing, per cui il calcolo è più semplice e veloce, ma produce problemi di aliasing. Come precedentemente accennato esistono delle tecniche per attenuare l'aliasing: *supersampling* e *adaptive sampling*. Il primo consiste nell'usare più di un campione per pixel riducendo così l'aliasing e aumentando il *Nyquist limit*. Nell'*adaptive sampling* invece, vengono tracciati raggi aggiuntivi vicino ai bordi rendendoli più precisi ma aumentando la complessità dell'algoritmo. Se da un lato abbiamo questi due metodi che prevedono la divisione regolare del pixel, dall'altro abbiamo il *stochastic sampling*. A frequenze al di sopra del *Nyquist limit* non otterremo un campionamento corretto ma con questo metodo possiamo risolvere il problema dell'aliasing con il rumore. Prima di approfondire lo *stochastic sampling* esamineremo l'*uniform point sampling*.

2.5.2 Uniform point sampling

In questo caso per frequenze più alte del Nyquist limit non avremo una campionatura corretta. Se i campioni sono uniformemente distanziati si ha il fenomeno dell'aliasing, che fa apparire come frequenze basse frequenze maggiori del Nyquist limit. Per spiegare come ciò accada possiamo considerare un campionamento monodimensionale nella dimensione del tempo. Dato il segnale $f(t)$ campionato a intervalli regolari nT , dove n è un intero e T è il tempo tra un campione e l'altro, la frequenza di campionatura sarà $1/T$. Questo è equivalente alla funzione *shah* $\text{III}(t/T)$

$$\text{III}(x) = \sum_{n=-\infty}^{\infty} \delta(x-n) \quad \text{dove } \delta \text{ è la funzione di Kronecker}$$

Il teorema della campionatura dice che se $f(t)$ non contiene frequenze maggiori del limite di

Nyquist, la ricostruzione della campionatura produrrà un segnale corretto. Nella figura 2.24 si vede proprio questo effetto nell'esempio (a). Nel caso (b) invece avremo una frequenza maggiore del limite e i valori non creeranno una rappresentazione corretta facendo sembrare i valori provenienti da una frequenza più bassa. Nel caso in cui i punti non siano equidistanti, quello che altrimenti vedremo come aliasing viene definito rumore.

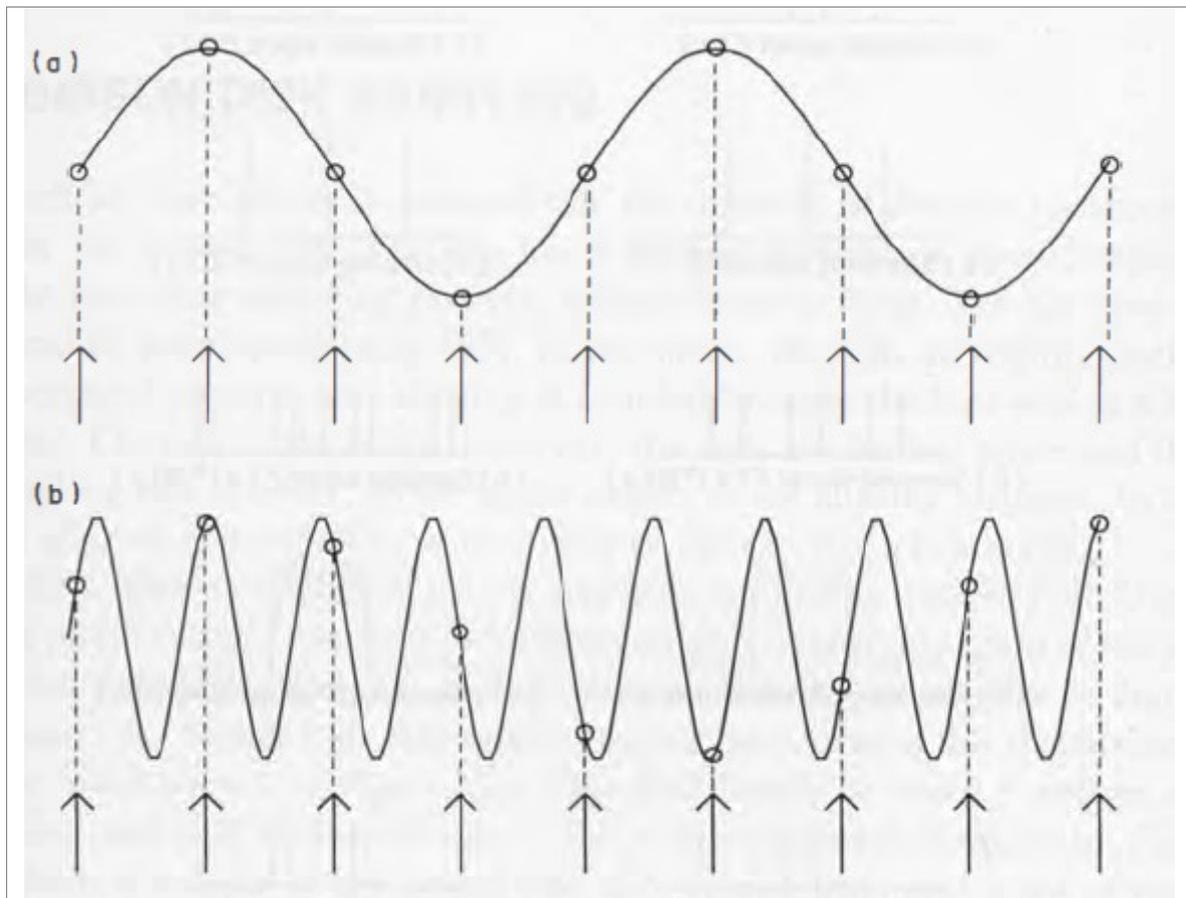


Fig. 2.24: Esempi di campionamenti in cui le frecce segnano i punti di campionamento. Nel caso (a) la frequenza è minore del Nyquist limit e avremo una campionatura corretta. Nel caso (b) la frequenza è maggiore del Nyquist limit e i campioni rappresentano un'onda a frequenza più bassa.

2.5.3 Campionamento di tipo Poisson disk

Un ottimo esempio di distribuzione non uniforme è quella che avviene nell'occhio umano. Al centro del nostro campo visivo le cellule sono organizzate secondo uno schema esagonale in modo che l'effetto dell'aliasing venga evitato. Esternamente invece le cellule sono più distanti e potrebbe verificarsi l'aliasing. Per superare il problema è sufficiente farne una distribuzione non uniforme.

Nell'immagine 2.25(a) sottostante è riportata la distribuzione dei coni al di fuori del centro dell'occhio e in 2.25(b) sua trasformata ottica di Fourier della distribuzione, chiamata *Poisson disk distribution*.

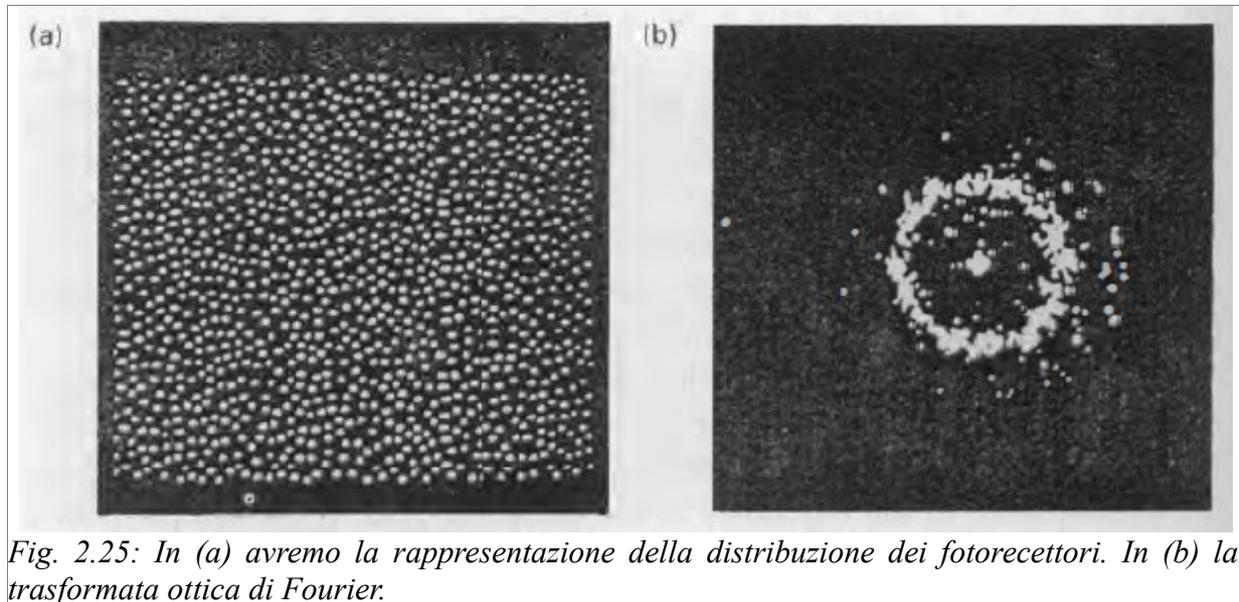


Fig. 2.25: In (a) avremo la rappresentazione della distribuzione dei fotorecettori. In (b) la trasformata ottica di Fourier.

L'unica condizione sulla distribuzione dei punti è quella di avere un minimo di distanza tra loro, in quanto senza questa condizione, tenderebbero ad accumularsi in alcune zone. L'implementazione di questo tipo di campionamento è semplice, anche se costosa. Le posizioni dei punti, sempre con la condizione della distanza, vengono create finché la regione di campionamento non è piena. Successivamente vengono calcolati i valori del filtro che descrive come ciascun campione influisce sui pixel adiacenti che poi saranno normalizzati.

2.5.4 Jittering di una griglia regolare

Questo metodo, come l'aggiunta del rumore, può essere usato per approssimare la Poisson disk distribution. Esistono diversi tipi di jitter, in questa tesi parleremo del regular grid. I risultati ottenuti però non sono buoni come quelli del Poisson disk. Il jitter, analizzato nella sola dimensione del tempo, ha ogni n -esimo elemento moltiplicato per la variabile di tempo T . Il risultato ottenuto verrà distorto da una quantità ζ_n , ottenendo $nT + \zeta_n$. Se i ζ_n non sono correlati avremo attenuazione ad alte frequenze e l'energia persa da questa attenuazione apparirà uniformemente come rumore pari all'attenuazione del segnale. Esiste un tipo di jitter chiamato *uncorrelated jitter* per cui invece che avere un singolo ζ_n avremo anche ζ_m non correlati tra di loro. Possiamo suddividere gli *uncorrelated jitter* in due diversi tipi:

- *Gaussian jitter*: in questo caso i valori di ζ vengono scelti secondo la distribuzione Gaussiana.

$$e^{-(2\pi\nu\sigma)^2}$$

Nell'immagine 2.26 avremo una sua rappresentazione per $\sigma = \frac{T}{6,5}$ e sarà rappresentato da una linea solida.

- *White noise jitter*: qui invece i valori di ζ sono uniformemente distribuiti tra $-\gamma T$ e γT .

$$\left(\frac{\sin(2\pi\gamma\nu T)}{2\pi\gamma\nu T} \right)^2$$

In figura corrisponde alla linea tratteggiata per un valore di $\gamma = \frac{1}{2}$

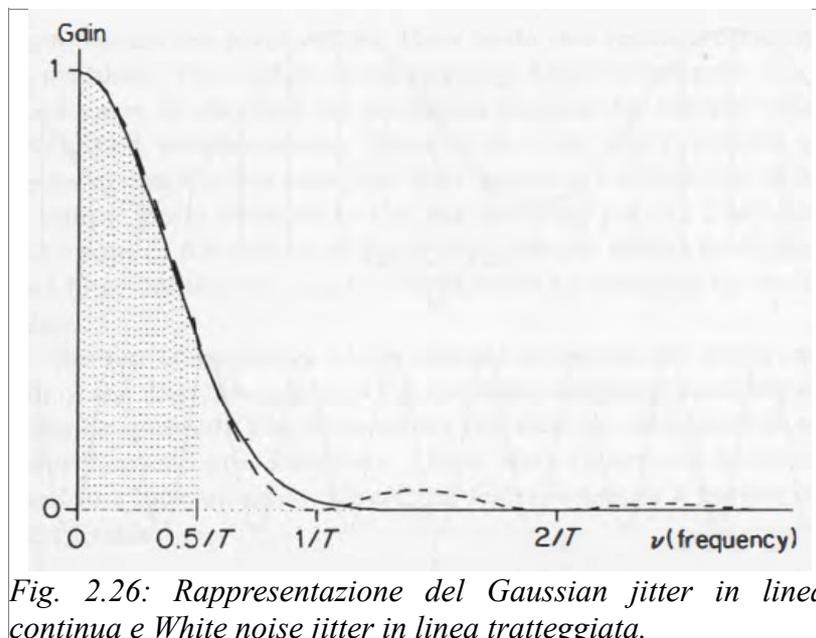


Fig. 2.26: Rappresentazione del Gaussian jitter in linea continua e White noise jitter in linea tratteggiata.

Come intuibile dalla Fig. 2.26, il jittering di una griglia regolare non elimina l'aliasing ma lo diminuisce sostanzialmente. Dato un Nyquist limit di $\frac{0,5}{T}$, la zona ombreggiata sarà all'interno di questo limite. Questo limite può essere ridimensionato variando i valori di γ e σ . Quando dobbiamo estendere il jitter a due dimensioni è sufficiente considerare un pixel come una griglia regolare di uno o più sottopixel, ognuno con il suo punto di campionamento. Il punto di campionamento sarà fatto al centro del pixel e il rumore verrà aggiunto alle coordinate x e y. Successivamente i valori verranno filtrati, ricostruiti e ricampionati su una normale griglia di pixel per ottenere i loro valori. Il modo migliore per fare questa ricostruzione è quello di calcolare un box

filter per cui ogni pixel si ottiene come media dei campioni nel pixel stesso. Il valore finale di ogni pixel è la somma dei valori dei punti di campionamento vicini moltiplicati per il loro valore di filtro, il risultato verrà poi normalizzato.

2.5.5 Distributed ray tracing

L'intensità di ogni pixel sullo schermo può richiedere un complesso calcolo di integrali annidati che permette di creare diversi effetti altrimenti impossibili, come la penombra, effetti sfocati, effetti traslucidi, profondità di campo e motion blur. Essendo però un calcolo molto complesso, gli algoritmi di rendering di immagini fanno delle semplificazioni per evitarne la valutazione e limitandosi a effetti di ombreggiatura, riflessi e rifrazioni nitidi. Questo non è vero però per il ray tracing e il radiosity.

2.5.6 Shading

L'intensità della luce riflessa in un punto su una superficie può essere scritta come la seguente formula

$$I(\phi_r, \Theta_r) = \int_{\phi_i} \int_{\Theta_i} L(\phi_i, \Theta_i) R(\phi_i, \Theta_i, \phi_r, \Theta_r) d\phi_i d\Theta_i$$

dove (ϕ_i, Θ_i) è l'angolo di incidenza e (ϕ_r, Θ_r) l'angolo di riflessione.

Per risolvere più semplicemente questo calcolo vengono fatte delle semplificazioni:

- si assume L come uguale a 0 tranne che per la direzione della sorgente luminosa. Ciò potrebbe causare ombre nette.
- tutte le direzioni che non sono quella della fonte luminosa vengono generalizzate come illuminazione ambientale. Quindi L non dipenderà da ϕ_i e Θ_i e R può essere sostituito con una riflessione ambientale media.
- si assume che la superficie sia uno specchio e che rifletta in una sola direzione dello specchio. Ciò causa riflessi acuti per quanto riguarda la luce speculare e nel caso della luce trasmessa provoca una rifrazione nitida.

Ma i raggi di illuminazione non sono tracciati nella sola direzione della luce, ma vengono distribuiti secondo la funzione di illuminazione L e i raggi riflessi vengono distribuiti in base alla funzione di riflessione R . I vari effetti che possono crearsi in una scena sono:

- *Gloss* o lucentezza: in computer grafica vengono realizzati come se le riflessioni fossero tutte come degli specchi. Le riflessioni dello specchio vengono determinate dai raggi

generati dal metodo del ray tracing e la lucentezza può essere calcolata distribuendo i raggi secondari creati colpendo la superficie.

- *Traslucency* o effetto traslucido: la luce trasmessa attraverso gli oggetti ha un'equazione simile a quella della luce riflessa eccetto per il fatto che avremo la funzione T al posto di quella R . Come prima, la calcoleremo tracciando i raggi secondari nella direzione di trasmissione.
- Penombra: questo fenomeno accade quando la luce viene solo parzialmente oscurata. L'intensità della riflessione è proporzionale all'angolo della porzione della luce visibile. Anche qui, per calcolare la penombra dovremo tracciare diversi raggi secondari dalla sorgente luminosa. Il numero dei raggi tracciati dovrà essere proporzionale all'energia della sorgente luminosa nei suoi diversi punti.

2.5.7 Motion Blur

Il motion blur è un effetto di sfocatura che si vede negli oggetti in movimento. Questa tecnica è molto costosa da realizzare e per realizzarla si può procedere separando il calcolo delle superfici visibili da quello del motion blur. In alcuni casi può essere accettabile procedere in questo modo, ma in generale non è possibile creare un'immagine e successivamente creare l'effetto di sfocatura. Prima di realizzare questo effetto è necessario conoscere lo sfondo, anche se non è sempre facile separarlo dagli altri elementi della scena. Inoltre questo tipo di soluzione non permette modifiche alla visibilità di un singolo oggetto. Considerato un oggetto rotante su un tavolo, se calcolassimo una sola sfumatura per frame avremmo la scena correttamente sfocata, ma allo stesso tempo avremmo erroneamente anche le luci e le ombre sfocate. Gli effetti di sfocatura sulle ombre e sulle luci sono corretti solo nel caso in cui l'oggetto posto sul tavolo si muova o vi sia uno spostamento della fonte luminosa o della fotocamera. Per fare ciò è necessario risolvere il problema del calcolo delle superfici visibili in funzione del tempo insieme a quelle che comprendono le texture e lo shading. Una soluzione a tutti questi problemi è quella di utilizzare il Point sampling che:

- lavora solo nella dimensione dello spazio e non in quella del tempo;
- consiste nell'avere una lista di tutti gli elementi per ogni punto di campionamento in ogni frame;
- impone la restrizione che vertici e bordi si muovano alla stessa velocità, creando buchi negli oggetti che si muovono molto in un solo frame;

- i bordi dei poligoni contenenti questi vertici non rimarranno adiacenti;
- l'algoritmo è limitato ai movimenti lineari, altrimenti si avrebbe un calcolo molto più complesso.

Quindi per concludere, il modo migliore per realizzare questo effetto è quello di distribuire i punti di campionamento nel tempo.

2.5.8 Implementazione del distributed ray tracing

Il calcolo delle superfici visibili è molto semplice. A ogni istante è necessario aggiornare le posizioni degli oggetti e costruire un raggio che parte dall'obiettivo fino a intersecare l'oggetto più vicino al raggio. Un modo per distribuire i raggi nelle altre dimensioni è quello di usare valori casuali non correlati (ad esempio un tempo casuale per ogni raggio) producendo però immagini con molto rumore. Per evitare che ciò accada si può usare la distribuzione di Poisson che assicura che i campioni non si ammuochino e non lascino grossi spazi vuoti tra di loro. Per creare oscillazioni nel tempo, creeremo casualmente un modello nello spazio dello schermo che divideremo in intervalli e assoceremo un punto a ognuno di essi. La correlazione tra la posizione nello spazio e quella in altre dimensioni possono causare aliasing. In alcuni casi è necessario dare dei pesi ai campioni. Un buon approccio sarebbe quello di distribuire uniformemente i campioni e poi pesare ciascun raggio in base al filtro che sarà diviso in modo regolare.

2.5.9 Riepilogo del distributed ray tracing

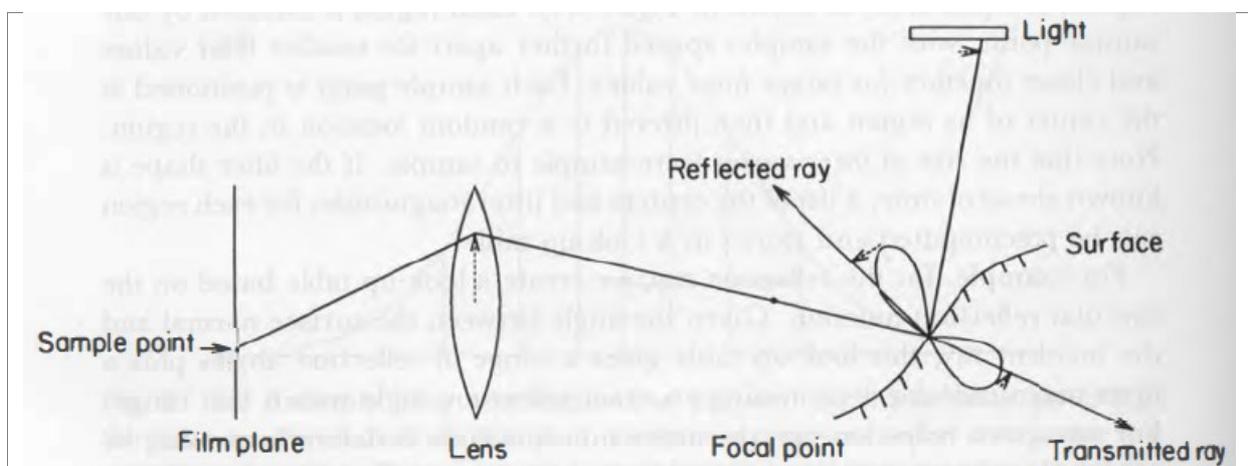


Fig. 2.27: Distributed ray tracing.

Osservando la Fig. 2.27, l'algoritmo del distributed ray tracing può essere così delineato:

- determinare facendo uso del jittering sampling la posizione del raggio nello spazio dello schermo;
- determinare il tempo per ogni raggio;
- aggiornare la posizione della fotocamera e degli oggetti nella scena;
- determinare il punto focale tracciando il raggio dal centro dell'obiettivo alla scena. La distanza tra questo punto e l'occhio viene detta distanza focale;
- determinare la posizione dell'obiettivo;
- determinare il punto visibile per il raggio che passa in questi elementi sopracitati;
- tracciare i raggi di riflessione, trasmissione e ombra.

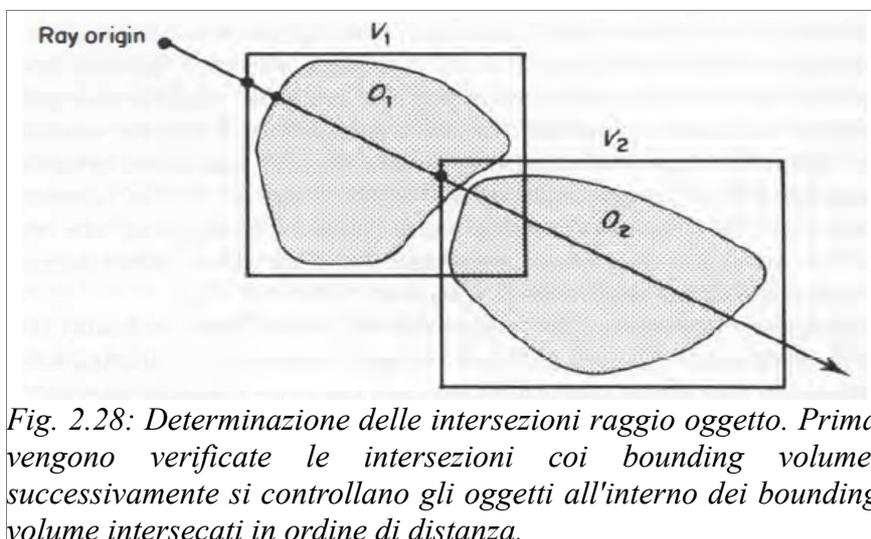
2.6 Velocizzazione delle tecniche di ray tracing

Nonostante il ray tracing abbia molte buone caratteristiche, viene spesso considerato molto costoso a livello computazionale per essere utilizzato. Perciò sono stati introdotti nel tempo diversi metodi riguardanti le strutture dati, numerici, geometrici, ecc. al fine di renderlo più efficiente. Abbiamo già detto come il calcolo delle intersezioni sia un'operazione molto costosa e al contempo la parte principale e più complessa dell'algorithm del ray tracing. Ed è noto come l'algorithm in sé preveda la verifica dell'intersezione con tutti gli oggetti della scena considerando il più vicino. Esistono diverse strategie per accelerare questo processo:

- ridurre il costo medio dell'intersezione di un raggio con l'ambiente. Questa categoria può essere suddivisa in due sotto-categorie: più veloce e meno intersezioni. Per “più veloce” intendiamo algoritmi più efficienti, mentre nell'altro caso utilizzeremo meno intersezioni possibili. A ciò si aggiunge la possibilità di semplificare le intersezioni decomponendo gli oggetti da intersecare in oggetti più semplici;
- ridurre il numero di raggi, in particolare ridurre i raggi da intersecare inclusi quelli secondari. In questo caso, invece che terminare l'albero creato dai raggi secondari nel momento in cui si raggiunge una profondità prefissata o quando si incontra una superficie non riflettente, possiamo porre il criterio di terminazione di Hall che consiste nel prendere in considerazione il massimo contributo al colore di un pixel;
- sostituire il concetto di raggio, tracciando più raggi contemporaneamente.

2.6.1 Bounding volume, gerarchie e slabs

Abbiamo già accennato alle tecniche per la velocizzazione dei calcoli nel ray tracing. I bounding volume consentono di accelerare l'algoritmo racchiudendo un oggetto della scena in un altro oggetto per cui l'intersezione è più semplice da calcolare (come ad esempio le sfere che, come abbiamo già visto hanno un calcolo non molto complesso). In questo modo verrà prima testata l'intersezione con il bounding volume e se ciò accade, verrà poi cercata l'intersezione con l'oggetto. Il numero di intersezioni da verificare non diminuirà, ma sicuramente i calcoli effettuati su oggetti che poi non saranno intersecati si ridurranno enormemente. Per ridurre ulteriormente i calcoli si può introdurre il concetto di gerarchia dei bounding volume per cui un bounding volume ne contiene altri al suo interno e se non vi è intersezione col bounding volume più esterno, sicuramente non ci sarà coi bounding volume più interni e non è necessario verificarne l'intersezione. Una volta che viene trovata un'intersezione con un oggetto e viene posto un limite alla distanza massima da testare, tutti gli oggetti o bounding volume oltre quel limite non necessiteranno di una verifica poiché esisterà già un'intersezione con un oggetto prima di quel limite. Se ad esempio trovassimo un'intersezione, come nel caso in figura 2.28, con i bounding volume V_1 e V_2 verrà testato per primo il caso di V_1 . Trovando quindi un'intersezione con l'oggetto O_1 , non sarà necessario testare l'oggetto presente nei bounding volume successivi, in questo caso O_2 .



Per aumentare l'efficienza dei bounding volume si deve tenere conto di due importanti fattori: la loro aderenza all'oggetto e il costo di intersezione. Utilizzando bounding volume con forme come la sfera o una scatola abbiamo migliorato solo il secondo fattore e ci ha permesso di utilizzare questo tipo di forme anche per le gerarchie di bounding volume. Il costo computazionale associato a un

oggetto e il relativo bounding volume è dato da:

$$\text{Costo} = n \cdot B + m \cdot I$$

dove n è il numero dei raggi testati sul bounding volume, B è il costo di ogni test, m è il numero di raggi che effettivamente colpiscono il bounding volume e I è il costo dell'intersezione del raggio con l'oggetto all'interno del bounding volume. Ponendo che n e I siano fissi, dovremo fare in modo di creare un bounding volume sufficientemente aderente da minimizzare m ma con una forma non troppo complessa per minimizzare B .

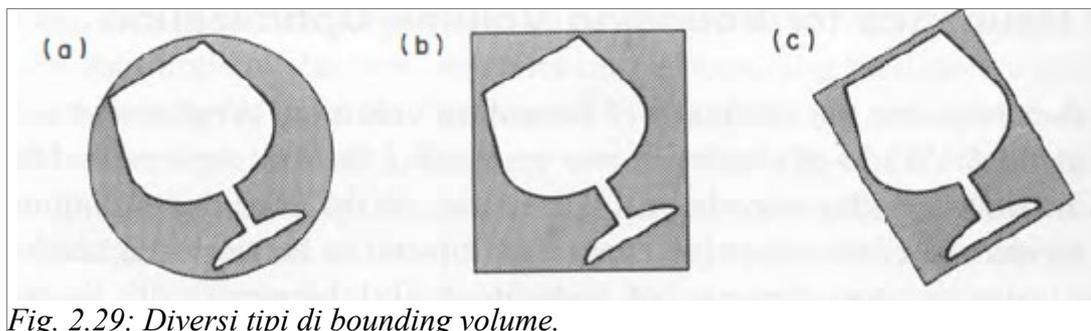


Fig. 2.29: Diversi tipi di bounding volume.

Supponendo che i raggi che tracciamo nell'ambiente abbiano riflessioni e rifrazioni casuali, l'area creata tra il bounding volume e l'oggetto, rappresentata nell'immagine con un colore scuro, diventa la misura corretta per trovare un compromesso alla formula sopra studiata. Nell'immagine 2.29 possiamo vedere come sicuramente il bounding volume (c) sia quello che più si adatta alla forma dell'oggetto contenuto, causando però un costo maggiore nel calcolo di intersezioni rispetto alla figura (b). Nonostante una figura simile (c) costi di più nel calcolo dell'intersezione col bounding volume, nel caso di oggetti molto complessi è conveniente avere bounding volume che più si adattano all'oggetto per diminuire il più possibile il numero di raggi che intersecano l'oggetto e di verificare solo quelli che effettivamente potrebbero intersecarlo. A volte potrebbe essere conveniente usare dei bounding volume particolari come nell'immagine 2.30. Nel caso (a) avremo due diversi bounding volume di forme diverse; per verificare l'intersezione raggio-oggetto dovremo prima controllare che il raggio tocchi entrambi i bounding volume. Nel caso (b) avremo un bounding volume formato dall'unione di due o più bounding volume. Nel caso (c) il bounding volume sarà costituito dall'intersezione di un numero di slabs.

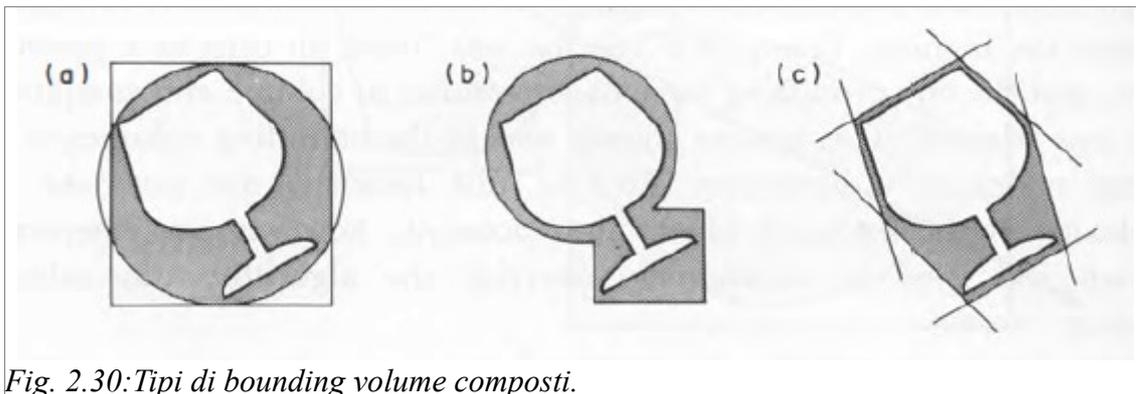


Fig. 2.30: Tipi di bounding volume composti.

Supponendo che i raggi siano uniformemente distribuiti e dato il bounding volume A con all'interno gli oggetti B , la probabilità che un raggio r che colpisce A colpisca anche B si può calcolare come

$$Pr(r \text{ hits } B \mid r \text{ hits } A) = \frac{\langle P(B, d) \rangle}{\langle P(A, d) \rangle} = \frac{S(B)}{S(A)}$$

dove $P(V, d)$ è l'area proiettata di V lungo la direzione d , $\langle \rangle$ indica la media rilevata sulle direzioni d e $S(v)$ è l'area della superficie del volume V . Quando creiamo una gerarchia di bounding volume spesso accade che oggetti molto vicini tra loro vengano raggruppati per ridurre il più possibile l'area vuota. Tuttavia questo tipo di costruzione delle gerarchie tende ad avere grandi fattori di ramificazione riducendo il beneficio di “potatura” dell'albero generato dalla gerarchia di bounding volume. Per evitare questo problema è possibile costruire la gerarchia partendo dalla radice e inserendo in modo incrementale gli oggetti primitivi e scegliendo la via che minimizza gli spazi vuoti. Gli oggetti convessi potranno avere al più due intersezioni con un singolo raggio. Un metodo per utilizzare questa proprietà è quello delle slabs. Questo algoritmo utilizza il concetto di utilizzare una serie di piani paralleli per delimitare gli oggetti e crearvi un bounding volume che contiene l'oggetto. Lo spazio infinito che si crea tra ogni coppia di slabs, min e max, viene chiamato *lastra*. Lavorando in tre dimensioni come minimo avremo bisogno di tre coppie di slabs ma più coppie vengono inserite più il bounding volume generato sarà aderente all'oggetto diminuendo l'area vuota. L'utilizzo delle stesse slabs per tutti gli oggetti della scena può facilitare i calcoli delle intersezioni, poiché si avranno calcoli molto simili.

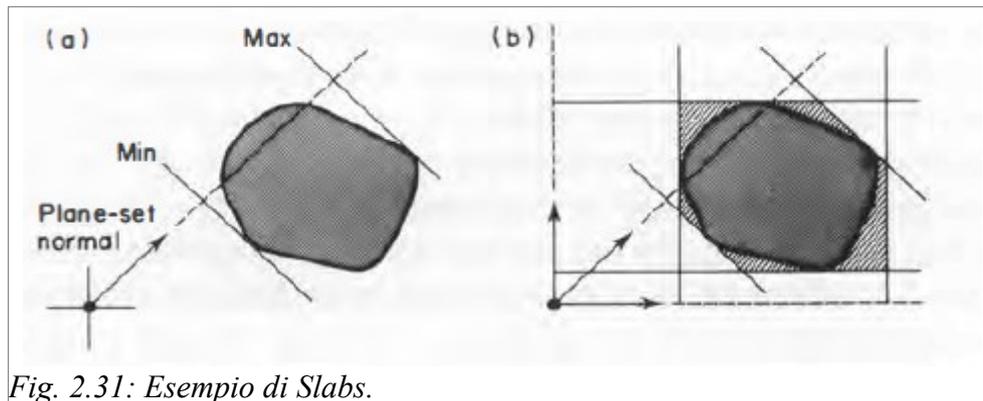


Fig. 2.31: Esempio di Slabs.

2.6.2 Suddivisione spaziale

Abbiamo già visto come sia utile l'utilizzo delle gerarchie di bounding volume per ridurre significativamente il numero delle intersezioni da calcolare. La suddivisione spaziale, a differenza dell'altro metodo, considera una suddivisione dello spazio piuttosto che i singoli oggetti. L'ambiente verrà quindi suddiviso in prismi rettangolari allineati agli assi chiamati *voxel* in modo tale che non si sovrappongano. Le dimensioni dei *voxel* possono essere uniformi o no, ma in ogni caso saranno di aiuto a limitare i controlli delle intersezioni solo a quegli oggetti presenti nei volumi attraversati dal raggio. Ovviamente nel caso in cui nella ricerca delle intersezioni, ordinando i *voxel* per distanza, ne trovassimo una con un determinato oggetto in uno specifico *voxel*, potremmo fermare la ricerca poiché sicuramente l'intersezione trovata sarà quella più vicina.

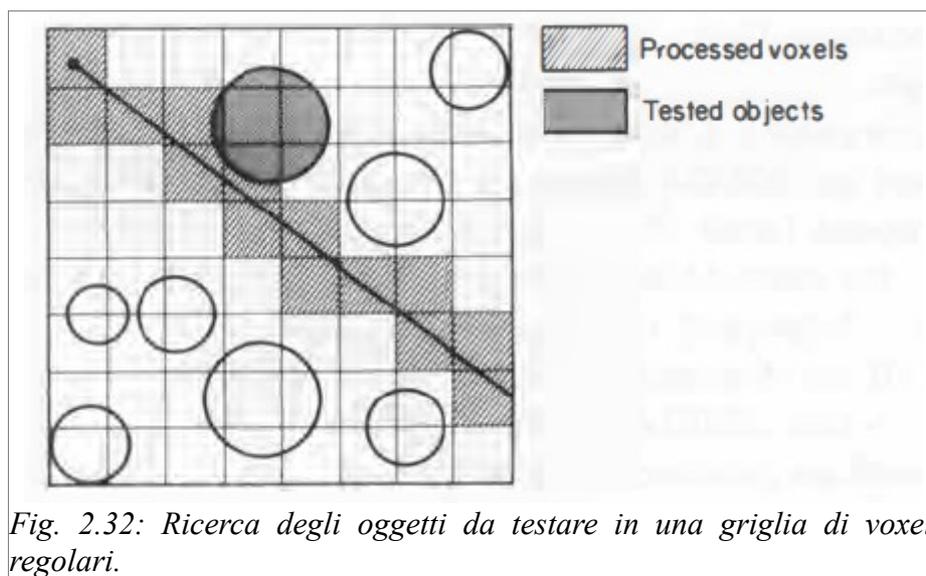


Fig. 2.32: Ricerca degli oggetti da testare in una griglia di voxel regolari.

In alternativa alla suddivisione uniforme dello spazio, c'è la possibilità di suddividerlo non uniformemente in base agli oggetti presenti nella scena. In questo modo zone dell'ambiente molto

popolate verranno suddivise in più *voxel* piccoli e zone poco popolate saranno suddivise in *voxel* di grandi dimensioni. La suddivisione dei *voxel* può avvenire suddividendo l'ambiente parti, le quali saranno ricorsivamente suddivise finché ogni *voxel* non sarà di una dimensione soddisfacente e il calcolo delle intersezioni con gli oggetti in esso contenuti siano semplici. In alternativa potremmo creare prima una suddivisione provvisoria e assegnare a ogni *voxel* dei determinati oggetti che verranno poi testati. Successivamente, dopo un test sugli oggetti che attraversano i bordi di ogni *voxel*, si decide se includerli nel *voxel* o no. Si procede in questo modo finché il numero di oggetti in ogni *voxel* è minore di un numero minimo posto.

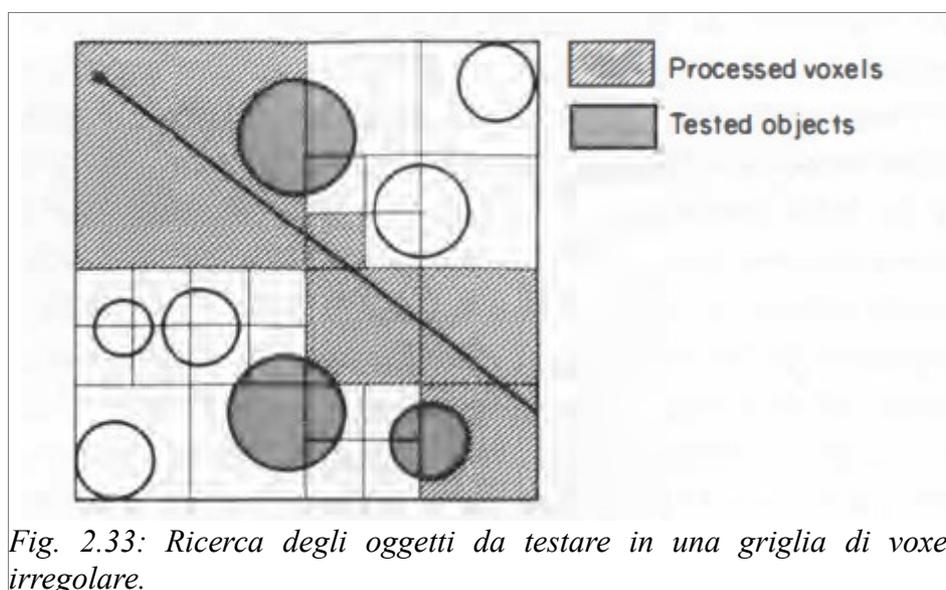


Fig. 2.33: Ricerca degli oggetti da testare in una griglia di voxel irregolare.

In entrambi i casi possono crearsi dei problemi, tra i quali il fatto che un oggetto possa stare in due *voxel* contemporaneamente e che entrambi vengano attraversati dal raggio. Ciò può portare a dover calcolare l'intersezione con uno stesso oggetto più volte (come nel caso dell'oggetto A nell'immagine 2.32). Per risolvere, è sufficiente mantenere uno storico delle intersezioni di ogni singolo raggio con ogni oggetto che viene testato, verificando che un confronto di questo tipo non esista già nello storico. Un altro problema che si può verificare è quello rappresentato dai poligoni

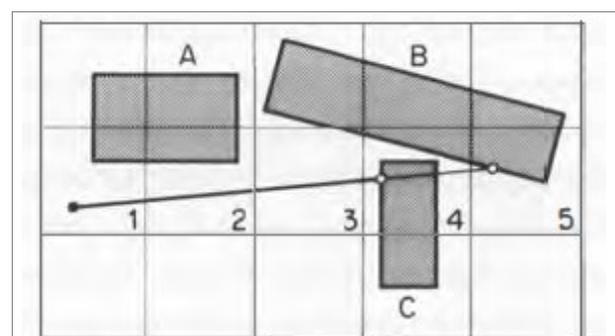


Fig. 2.34: Problema per cui l'oggetto B viene testato prima dell'oggetto C facendo sparire l'oggetto C dalla scena anche se la sua intersezione è prima di quella con l'oggetto B.

B e C. Il poligono B verrà testato prima del poligono C poiché sarà presente già nel terzo riquadro. Effettivamente verrà trovata una intersezione con l'oggetto B, ma sarà nel quinto riquadro.

L'oggetto C dovrebbe essere testato subito dopo B, ma avendo trovato un'intersezione con l'oggetto B, l'intersezione con C non verrà verificata anche se l'intersezione con C è in un riquadro precedente a quella con B. Per evitare che si verifichino sparizioni di oggetti dobbiamo vincolare la ricerca dell'intersezione al *voxel* corrente.

2.7 Immagini realizzate in Ray Tracing

Di seguito vengono mostrate alcune immagini realizzate con l'utilizzo del ray tracing. Si può notare come venga reso con molto realismo l'effetto della luce sui materiali caratterizzati da riflessione speculare. Queste immagini sono tratte da: Fig. 2.35 Oyonale – 3D art and graphic experiments come immagine dimostrativa delle potenzialità del ray tracing, Fig. 2.36 immagine proveniente da una cutscene di Battlefield V, Fig. 2.37 e Fig. 2.38 rispettivamente immagini in game di Star Wars: Battlefront II e Battlefield V, Fig. 2.39 immagine tratta da una clip dimostrativa di una scena in ray tracing.



Fig. 2.35



Fig. 2.36



Fig. 2.37



Fig. 2.38

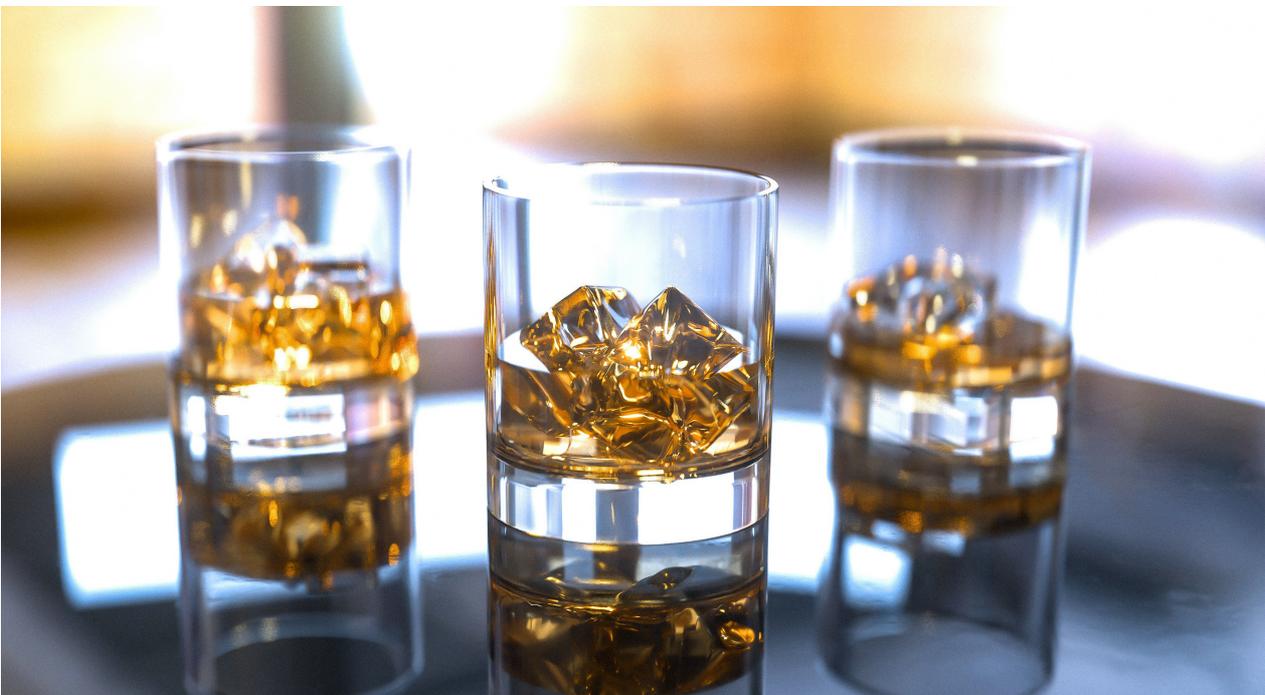


Fig. 2.39

Capitolo 3: Radiosity

3.1 Concetto base dal radiosity

Un altro metodo di illuminazione globale è quello del radiosity. Le origini di questo metodo si possono trovare negli anni ottanta quando venne utilizzato nel campo del trasferimento del calore. Al contrario del ray tracing che valuta l'equazione di illuminazione in base alla vista e ai singoli pixel in cui è discretizzato il piano di vista, il radiosity la risolve in punti distribuiti sulle superfici dell'ambiente, quindi in modo indipendente alla vista. Un buon esempio di differenza tra il ray tracing e il radiosity possiamo trovarlo nell'immagine 3.1.

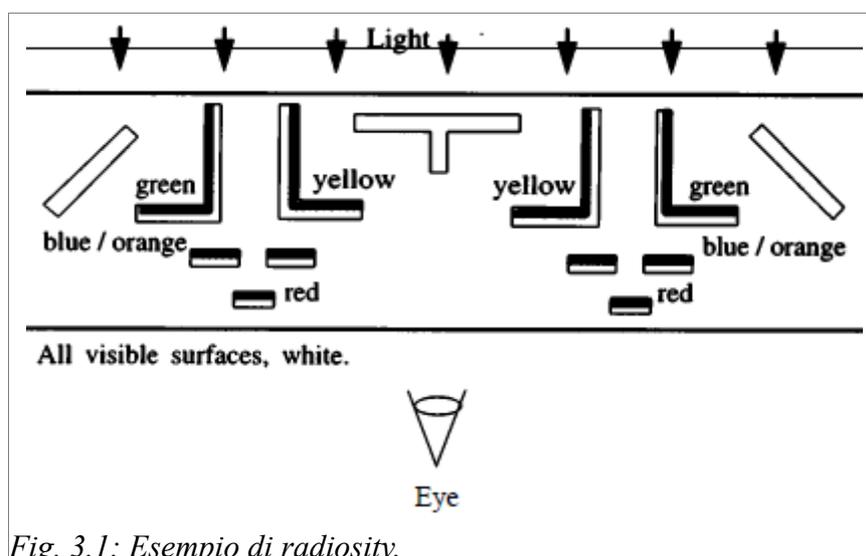


Fig. 3.1: Esempio di radiosity.

Data una scena in cui è presente una serie di pannelli di colore bianco nella direzione dell'osservatore e colorati nella direzione della luce, la luce rifletterà prima le superfici colorate e successivamente quelle bianche prima di raggiungere l'osservatore. Ciò causa una propagazione del

colore dalle superfici colorate a quelle bianche. Quindi potremo dire che il concetto che sta dietro al radiosity è quello per cui nell'ambiente ci sono materiali che diffondono la luce, la riflettono, la rifrangono e la assorbono^[3].

3.2 Nozioni base

Prima di addentrarci nell'algoritmo del radiosity è necessario fare un'introduzione sui concetti base che lo riguardano. In generale la radiazione luminosa è caratterizzata dal colore che viene determinato dalla distribuzione spettrale, e dall'energia che determina l'intensità o la luminosità. Introduciamo ora i concetti di radiometria e fotometria. La radiometria si occupa delle radiazioni appartenenti all'intero intervallo delle lunghezze d'onda e la fotometria estende il concetto analizzando come la luce agisca sull'occhio umano. Un elemento molto importante è il flusso radiante Φ , ovvero la velocità di emissione o assorbimento dell'energia luminosa da una superficie ed è misurabile in Watt (W). Dato quindi il flusso radiante è possibile ricavare l'irradianza $E(x)$ come il rapporto tra il flusso ricevuto da un elemento infinitesimo di superficie in x e la sua area dx :

$$E(x) = \frac{d\Phi}{dx}$$

La stessa formula viene utilizzata per calcolare la luminosità o radiosity $B(x) = \frac{d\Phi}{dx}$. L'unica differenza tra i due concetti è che l'irradianza corrisponde all'energia ricevuta, mentre il radiosity corrisponde a quella emessa. Quando parliamo di intensità radiante, o luminosa, intendiamo il flusso radiante emesso in un angolo solido infinitesimo $d\omega$ in una determinata direzione w :

$$I(w) = \frac{d\Phi}{d\omega}$$

L'angolo solido sotteso da un oggetto rispetto a un generico punto P corrisponde all'area della proiezione dell'oggetto su una sfera unitaria centrata in P . Dato quindi un infinitesimo di superficie dx centrato in x e orientato con la normale rispetto a un punto y posto a distanza r , l'angolo solido sarà:

$$d\omega = \frac{dx \cos \theta}{r^2}$$

in cui

- θ è l'angolo formato tra la normale e la congiungente y a x ;

- $dx \cos \theta$ rappresenta l'area proiettata di dx lungo la congiungente y ed x ;

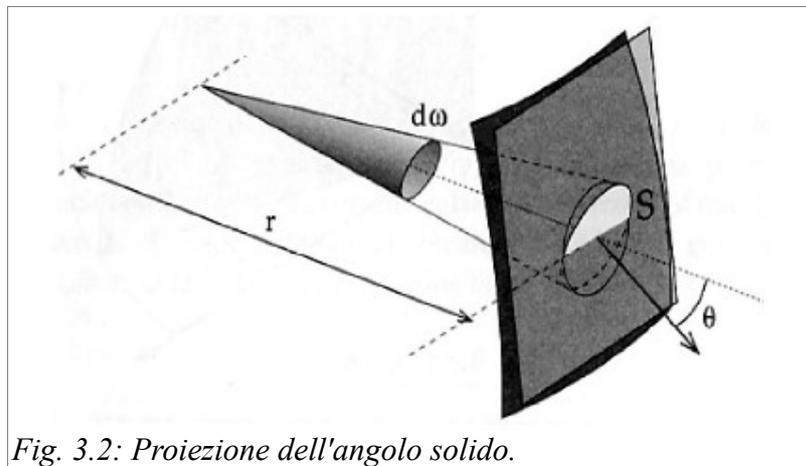


Fig. 3.2: Proiezione dell'angolo solido.

Ponendo a questo punto una sorgente luminosa puntiforme su y con intensità radiante I avremo che l'irradianza nel punto x sarà:

$$E(x) = \frac{d\Phi}{dx} = \frac{I \cos \theta}{r^2}$$

La radianza $L(x, w)$ nel punto x e in una direzione θ è la densità superficiale dell'intensità radiante in x nella direzione θ , o densità del flusso, considerando l'area della superficie proiettata si calcola come:

$$L(x, w) = \frac{dI(w)}{dx(w \cdot n)} = \frac{d^2\Phi}{dw dx (w \cdot n)}$$

perciò avremo che l'area della superficie dx equivale all'area della proiezione di dx con normale n sul piano perpendicolare a w ottenendo così $dx(w \cdot n)$.

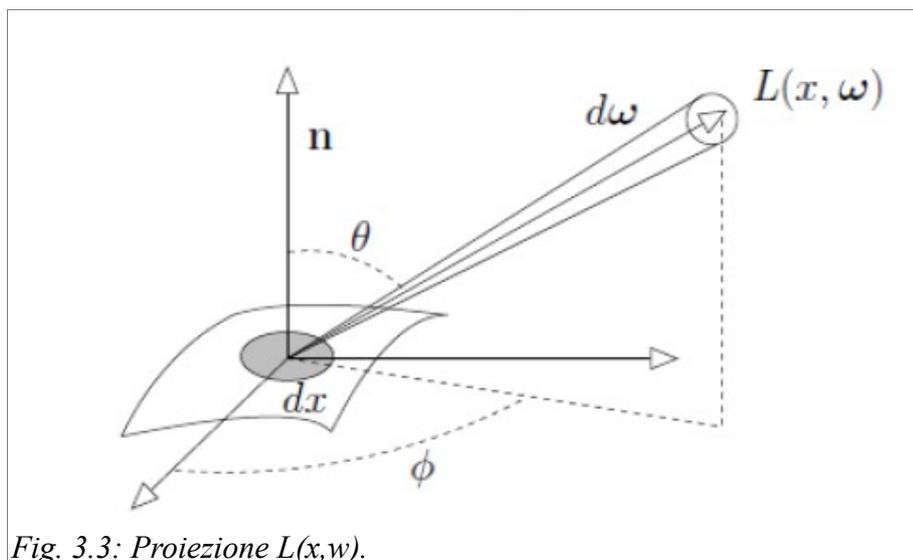


Fig. 3.3: Proiezione $L(x, w)$.

La direzione w è definita da due angoli: quello creato dalla direzione sulla superficie ϕ e quello di “elevazione” θ (rispetto alla normale n alla superficie). Perciò possiamo fare la seguente semplificazione: $(w \cdot n) = \cos \theta$. In generale possiamo dire che dati due punti x e y , la radianza che lascia x verso y è uguale a quella che raggiunge y da x . Quindi possiamo dire che la radianza non è influenzata dalla distanza dei punti analizzati.

A questo punto, data la semisfera delle direzioni attorno alla normale in x , Ω , e a partire dalla semplificazione $(w \cdot n) = \cos \theta$ si ha:

$$L(x, w)(w \cdot n) = \frac{d^2 \Phi}{dw dx}$$

integrando la quale si ottiene:

$$\int_{\Omega} L(x, w)(w \cdot n) dw = \int_{\Omega} \frac{d^2 \Phi}{dw dx} dw = \frac{d \Phi}{dx} = B(x)$$

3.2.1 BRDF

La BRDF, o Bidirectional Reflectance Distribution Function, caratterizza il materiale della superficie e viene definita come $\rho(x, w_i, w_r)$ e consiste nel rapporto tra la radianza riflessa da x nella direzione di w_r e l'irradianza della luce incidente nel punto x da un angolo solido infinitesimo $d\omega_i$ centrato in w_i .

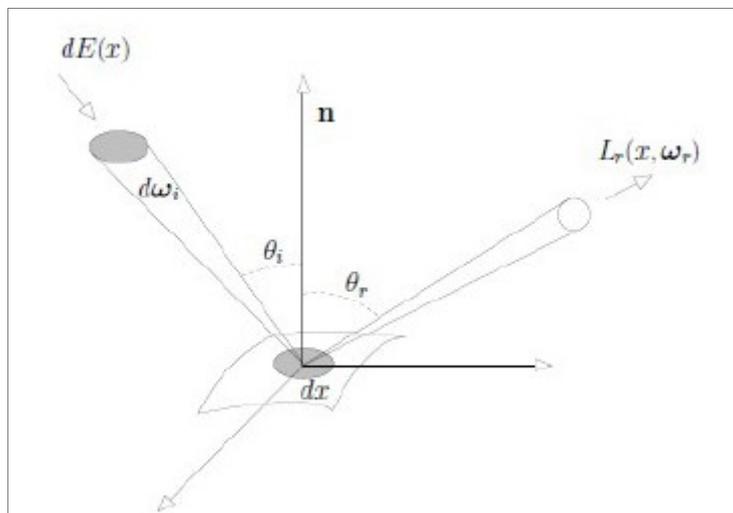


Fig. 3.4: BRDF. Rapporto tra la radianza in uscita verso w_r con l'irradianza in entrata da w_i .

Abbiamo già detto che l'energia che parte da un punto x e va verso y è uguale all'energia che arriva su y da x , perciò l'irradianza proveniente da w_i dipenderà direttamente dalla radianza della luce incidente $L_i(x, w_i)$. Quindi avremo:

$$dE(x) = L_i(x, w_i)(w_i \cdot n) dw_i$$

in cui $(w_i \cdot n) = \cos \theta_i$

La formula della BRDF può essere scritta come:

$$\rho(x, w_i, w_r) = \frac{L_r(x, w_r)}{L_i(x, w_i)(w_i \cdot n) dw_i}$$

Se volessimo considerare i contributi di irradianza da tutte le direzioni di incidenza potremmo calcolare la radianza riflessa in una direzione w_r come:

$$L_r(x, w_r) = \int_{w_i \in \Omega} \rho(x, w_i, w_r) L_i(x, w_i)(w_i \cdot n) dw_i$$

Nel caso di superfici Lambertiane, che prevedono una diffusione perfetta, la BRDF è costante poiché la radianza della superficie non dipende dalla direzione. Quindi avremo

$$\rho(x, w_i, w_r) = \rho(x) \quad \text{e}$$

$$L_r(x, w_r) = \rho(x) \int_{\Omega} L_i(x, w_i)(w_i \cdot n) dw_i = \rho(x) E(x) = L(x)$$

quindi il calcolo del radiosity sarà

$$B(x) = \int_{\Omega} L(x, w)(w \cdot n) dw = L(x) \int_{\Omega} \cos \theta dw = \rho(x) E(x) \pi$$

$\pi \rho(x)$ prende il nome di Albedo.

In generale la BRDF nella realizzazione di superfici realistiche è piuttosto complessa e quando si tenta la sua riproduzione al calcolatore si è costretti a utilizzare delle approssimazioni. Le due approssimazioni più comuni, ma anche più semplici, sono quelle che realizzano la riflessione diffusiva e quella speculare. Sappiamo che la riflessione diffusiva riflette la luce ugualmente in tutte le direzioni e che un riflettore speculare si comporta esattamente come uno specchio, riflettendo la luce in base all'angolo di incidenza generato dal raggio incidente e la normale alla superficie. Ma bisogna anche considerare il caso in cui la superficie colpita dal raggio luminoso sia lucida. In quel caso la riflessione non avverrà in un'unica direzione ma si disperderà in un cono luminoso creato attorno alla direzione di riflessione speculare perfetta. Questo tipo di riflessione viene detta glossy. In fig. 3.5 si può vedere come la BRDF sia la somma di tutti questi tipi di riflessione.

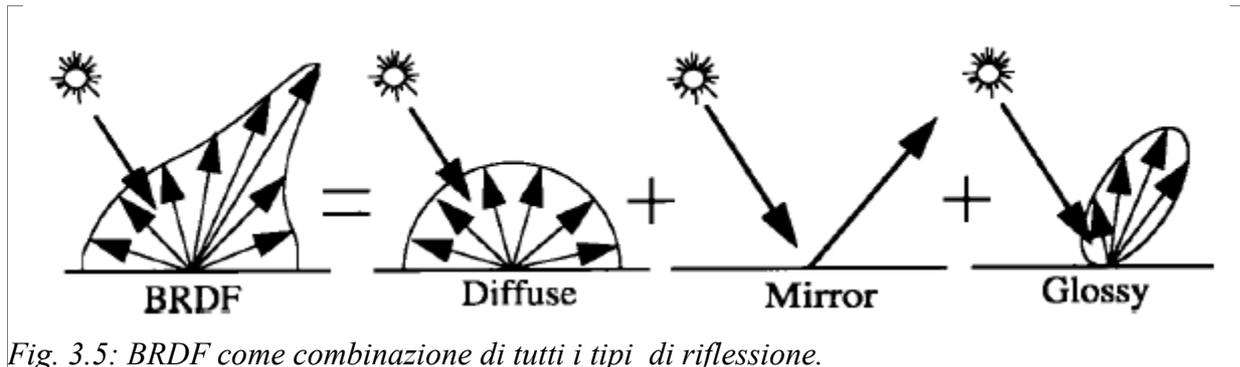


Fig. 3.5: BRDF come combinazione di tutti i tipi di riflessione.

3.2.2 L'equazione del rendering

Abbiamo detto che la radianza riflessa in direzione w_r come conseguenza dell'irradianza lungo la direzione w_i è

$$L_r(x, w_r) = \rho(x, w_i, w_r) L_i(x, w_i) (w_i \cdot n)$$

e che la radianza totale non è altro che la sommatoria dei contributi luminosi dovuti a tutte le direzioni incidenti θ_i e può essere scritta come il seguente integrale.

$$L_r(x, w_r) = \int_{\Omega} \rho(x, w_i, w_r) L_i(x, w_i) (w_i \cdot n) dw_i$$

Se a questa sommatoria aggiungiamo la radianza emessa $L_e(x, w)$ otteniamo l'equazione di rendering, ovvero la radianza totale emessa dal punto x nella direzione w

$$L(x, w) = L_e(x, w) + \int_{\Omega} \rho(x, w_i, w_r) L(x, w_i) (w_i \cdot n) dw_i$$

La radianza incidente nel punto x in direzione w_i , $L_i(x, w_i)$ è uguale alla radianza emessa da un punto y posto in direzione di x e da cui x è visibile. Quindi

$$L_i(x, w_{xy}) = L_e(y, w_{yx}) \text{ con } w_i = w_{xy}$$

e l'angolo solido infinitesimo dw sotto cui dx è visibile da y è dato da

$$dw = \frac{dy \cos \theta_{yx}}{\|x - y\|^2}$$

A questo punto è necessario introdurre il concetto della visibilità $V(x, y)$ che assumerà valore 1 se x è visibile da y . Includendo questo termine nell'equazione di rendering avremo

$$L(x, w) = L_e(x, w) + \int_{y \in S} \rho(x, w_{xy}, w) L(y, w_{yx}) G(x, y) dy$$

in cui il termine $G(x, y)$ sarà

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

e dipende solo dalla geometria della scena.

3.2.3 Dall'equazione di rendering a quella del radiosity

Il radiosity si basa sulla versione semplificata dell'equazione del rendering nel caso di superfici Lambertiane, ovvero composte da materiale completamente diffusivo e che diffonde la luce equamente in tutte le direzioni. Perché ciò avvenga si considera:

- la BRDF costante sul dominio angolare, cioè non dipende dall'angolo di incidenza;
- la radianza non dipende dalla direzione e coincide con la radiosità:

$$B(x) = L(x) \int_{\Omega} \cos \theta \, d\omega = \pi L(x)$$

Possiamo riscrivere l'equazione del rendering (moltiplicata per π) come:

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

da cui ricaveremo l'equazione della radiosità facendo le dovute sostituzioni:

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

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

questo valore prende il nome di fattore di forma e dipende solamente dalla geometria della scena.

$V(x, y)$ prende il nome di visibilità e assume valore 1 quando x è visibile da y . θ_{xy} e θ_{yx}

corrispondono agli angoli formati dal raggio che congiunge x e y con la normale rispettivamente sulla superficie dx e sulla superficie dy .

A questo punto è necessario effettuare una discretizzazione per poter risolvere il problema con elementi finiti. Perciò suddivideremo la scena in N patch P_j ciascuna con la sua area A_j e considereremo costanti B , B_e e ρ_d su ciascuna di esse. Per ogni patch avremo quindi che

$$B_i = \frac{1}{A_i} \int_{x \in P_i} B(x) \, dx$$

$$E_i = \frac{1}{A_i} \int_{x \in P_i} B_e(x) \, dx$$

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

è possibile approssimare l'integrale con una somma ottenendo

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

integrando da entrambi i lati otterremo

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

sostituendo in quest'ultima equazione le definizioni precedentemente date

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

e considerati B_j e ρ_i costanti in ogni patch, possiamo ricavare l'equazione di radiosità in versione discreta

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

questa equazione può essere scritta anche in versione matriciale come

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

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

che crea un sistema lineare $N \times N$ e la cui risoluzione, utilizzando un procedimento iterativo, permette di ottenere la radiosità per ogni patch.

3.3 Accenni al funzionamento dell' algoritmo

Poste queste basi e conosciuta l'equazione del radiosity è il momento di analizzare nel dettaglio l'algoritmo del radiosity. È un algoritmo di tipo view-independent, ciò significa che l'equazione del

radiosity verrà risolta solo una volta e per tutta la scena. La sua applicazione e quindi il suo calcolo è molto complessa e lunga, ma una volta realizzato permette di visualizzare la scena da qualsiasi punto e angolazione senza richiedere ulteriori calcoli. Il radiosity presume che tutte le superfici siano composte da un materiale perfettamente diffusivo, cioè con diffusione Lambertiana, e si specializza sul creare riflessioni diffuse tra gli oggetti della scena. Ogni superficie che riflette diffusivamente la luce porterà con sé sia la luminosità che il colore dell'oggetto da cui proviene che sarà influenzato da tutti i rimbalzi precedentemente fatti nella scena. Ciò che si vuole creare è un effetto per cui una superficie “assorba” una parte del colore proveniente da altre superfici della scena. Il risultato sarà, come abbiamo analizzato in Fig. 3.1, che una qualsiasi superficie bianca sarà influenzata dalla luce diffusiva proveniente da un oggetto colorato qualsiasi, ad esempio rosso, assorbendo una parte del colore in entrata e diventando rosa.

Quando implementiamo l'algoritmo del radiosity in una scena, dobbiamo immaginare una specie di stanza illuminata in cui la luce che parte da un punto x nella direzione di un altro punto y è uguale a quella ricevuta da j dalla direzione di i . Conosciuta l'equazione di rendering precedentemente analizzata sappiamo che l'energia che raggiunge una superficie corrisponde all'energia assorbita sommata a quella che verrà re-irradiata verso altre superfici. Inoltre la quantità di energia irradiata per riflessione su una superficie influirà sulla quantità ricevuta in un altro punto. Perciò ogni volta che l'energia viene riflessa è necessario conoscere l'energia incidente proveniente da altre superfici della scena. Sostanzialmente il ruolo del radiosity è quello di comprendere e calcolare questo valore.

3.4 Calcolo del Form Factor

A questo punto, l'unico elemento mancante della formula del radiosity è il form factor F_{ij} e, come sappiamo, ognuno di essi è dipendente dalla geometria della scena. Dividendo la superficie in patches piatte, sappiamo già che non è possibile che la quantità di energia uscita da una patch i colpisca se stessa, perciò $F_{ii} = 0$. In generale assumiamo che la luce che colpisce una patch sia uniforme e per questo motivo, quando calcoliamo il form factor, possiamo farlo per un generico punto della patch i .

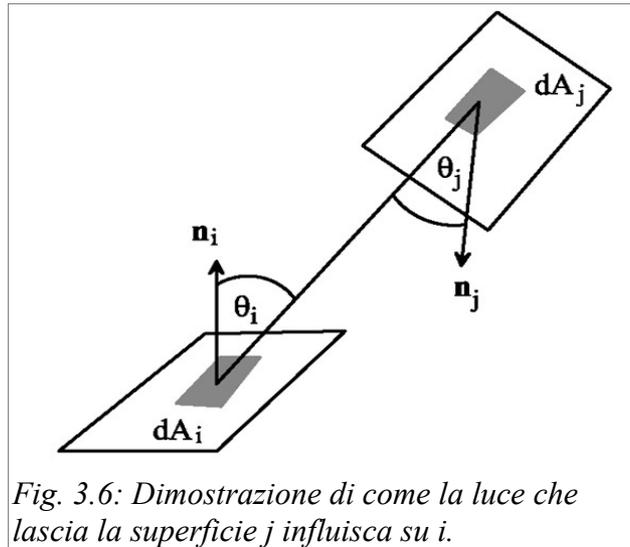


Fig. 3.6: Dimostrazione di come la luce che lascia la superficie j influisce su i .

Sappiamo che la luce che influenza una superficie proviene da tutte le direzioni entro un'emisfera centrata sul punto analizzato. Per ogni direzione, la quantità di energia che lascia un generico punto su una superficie j , in direzione del punto analizzato della patch i , è l'angolo sotteso verso quella direzione dalla patch j e sarà ridotta a $\cos\theta_j$. Di conseguenza la quantità di energia che colpisce la patch i sarà $\cos\theta_i$. Di conseguenza all'assunzione del fatto che la luce all'interno di una singola patch sia uniforme, il form factor si calcola come:

$$F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{\pi r^2} dA_j dA_i$$

I coseni nella formula sono rispettivamente l'angolo di incidenza della patch i e quello della patch j . Entrambi gli angoli sono fondamentali per determinare quanta luce può ricevere o emettere una determinata superficie in una specifica direzione.

3.4.1 Approccio al form factor col metodo dell'emisfera

Esistono diversi metodi per calcolare l'energia in entrata su un generico punto di una patch i , primo fra questi è il metodo dell'emisfera. Da questo momento in poi ci riferiremo a i come al punto sulla patch. Dato quindi i , questo metodo prevede di posizionare un'emisfera di raggio unitario, il cui centro sarà il punto i , con orientamento verso l'esterno della superficie. Tutta l'energia che attraversa questa emisfera si può considerare come energia che arriva in i . Il procedimento sarà quello di proiettare, sull'emisfera di i , la patch j con centro di proiezione i . L'area che viene proiettata sull'emisfera dipende dall'orientamento della patch j . Più il punto in cui la patch j , che

colpisce la patch i , è vicino al centro di proiezione, più il suo contributo sarà alto. Infatti, nel caso in cui la patch j colpisca verticalmente la patch i , il suo contributo sarà maggiore rispetto all'essere colpita da qualsiasi altra angolazione.

Come possiamo vedere dalla figura 3.7, si vengono a creare due proiezioni: la proiezione sull'emisfera corrisponde all'energia proveniente dalla patch j in base all'orientamento rispetto all'emisfera, mentre il punto in cui la proiezione colpisce la superficie della patch determina la quantità di energia proveniente da j influenza i . L'energia ricevuta è proporzionale all'area ottenuta sull'emisfera e inversamente proporzionale al quadrato della distanza tra le patches. Definiamo r come la distanza tra dA_i e A_j e ricordiamo che l'emisfera generata ha raggio unitario,

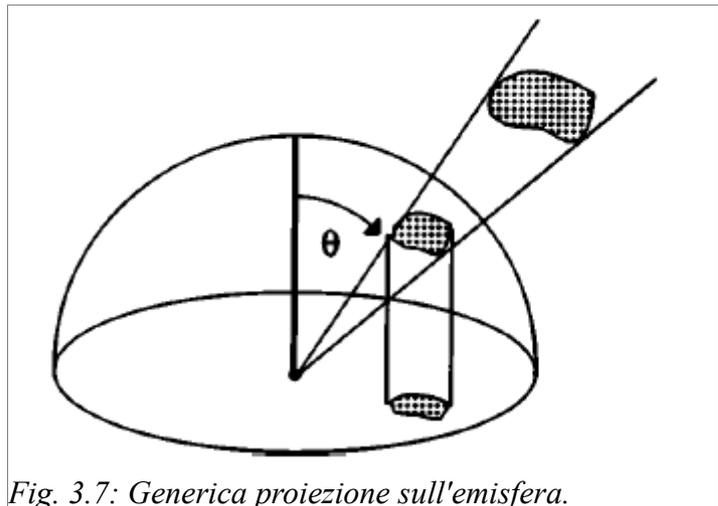


Fig. 3.7: Generica proiezione sull'emisfera.

l'area del cerchio generato sulla patch sarà uguale a π . La proiezione dell'area sulla patch non sarà altro che una frazione di π che sarà proporzionale al form factor. La proiezione di A_j sull'emisfera sarà data da

$$A_j \frac{\cos \theta_j}{r^2}$$

in cui il coseno è semplicemente l'angolo di A_j e r^2 deriva dalla legge dell'inverso del quadrato delle distanza che, nel nostro caso, afferma che l'intensità della luce che parte da una sorgente luminosa è inversamente proporzionale al quadrato della distanza tra la superficie colpita e la sorgente.

Considerata ora l'area proiettata sulla patch in base all'area sull'emisfera, il calcolo della luce proveniente da j che influisce sul punto i si calcola come:

$$A_j \frac{\cos \theta_i \cos \theta_j}{r^2}$$

Per cui, considerando un'area dA_i sulla superficie i e tutte le patch j , è possibile approssimare il calcolo del form factor come:

$$F_{ij} \approx \int_{A_j} \frac{\cos \theta_i \cos \theta_j}{r^2}$$

3.4.2 Approccio al form factor col metodo dell'emicubo

Un altro metodo per calcolare il form factor è quello dell'emicubo. Questo metodo prevede un'approssimazione del metodo dell'emisfera e consiste appunto nell'utilizzo di un emicubo suddiviso in celle. Più piccole saranno le celle, più il calcolo sarà accurato. Come nel metodo precedente, le patches vengono proiettate sull'emicubo con la differenza che la proiezione potrà cadere su una o più celle dell'emicubo. Questo approccio facilita la creazione della proiezione di j , ma si perde la possibilità di calcolare il peso del contributo della patch non potendo lavorare con gli angoli. La struttura dell'emicubo permette però di conoscere, in base alla cella colpita, quale parte dell'emisfera viene influenzata. Sicuramente più una cella è vicina al centro della faccia dell'emicubo, più questa influenzerà la superficie. Nel momento in cui l'emicubo viene generato viene associato il peso che avrà ogni determinata cella. Quindi il calcolo sarà:

$$\Delta F_p = \frac{\cos \theta_i \cos \theta_p}{r^2} \Delta A$$

in cui θ_p è l'angolo tra la normale alla cella p e il vettore che congiunge dA_i e p e A è l'area della cella. Il valore di F_{ij} viene calcolato come somma dei form factor delle celle in cui è inclusa la proiezione di A_j .

Entrambi i metodi prevedono una grande quantità di calcoli e di tempo, ma sappiamo che una volta che vengono effettuati non è necessario ripeterli e una volta cambiata la vista sarà sufficiente applicare lo shading. In ogni caso, il metodo del radiosity è conveniente solo quando la resa dell'immagine è più importante della velocità di calcolo e quando la stessa scena viene inquadrata da diverse angolazioni.

3.5 Immagini realizzate con il metodo del Radiosity

Di seguito alcune immagini realizzate con il metodo del radiosity. Si può notare come venga reso con molto realismo l'effetto della luce sui materiali caratterizzati da riflessione diffusiva. Queste immagini sono tratte da: Fig.3.8 e Fig. 3.9 immagini dimostrative del funzionamento del

radiosity, Fig. 3.10, Fig. 3.11 e Fig. 3.12 immagini realizzate per l'arredamento di ambienti.

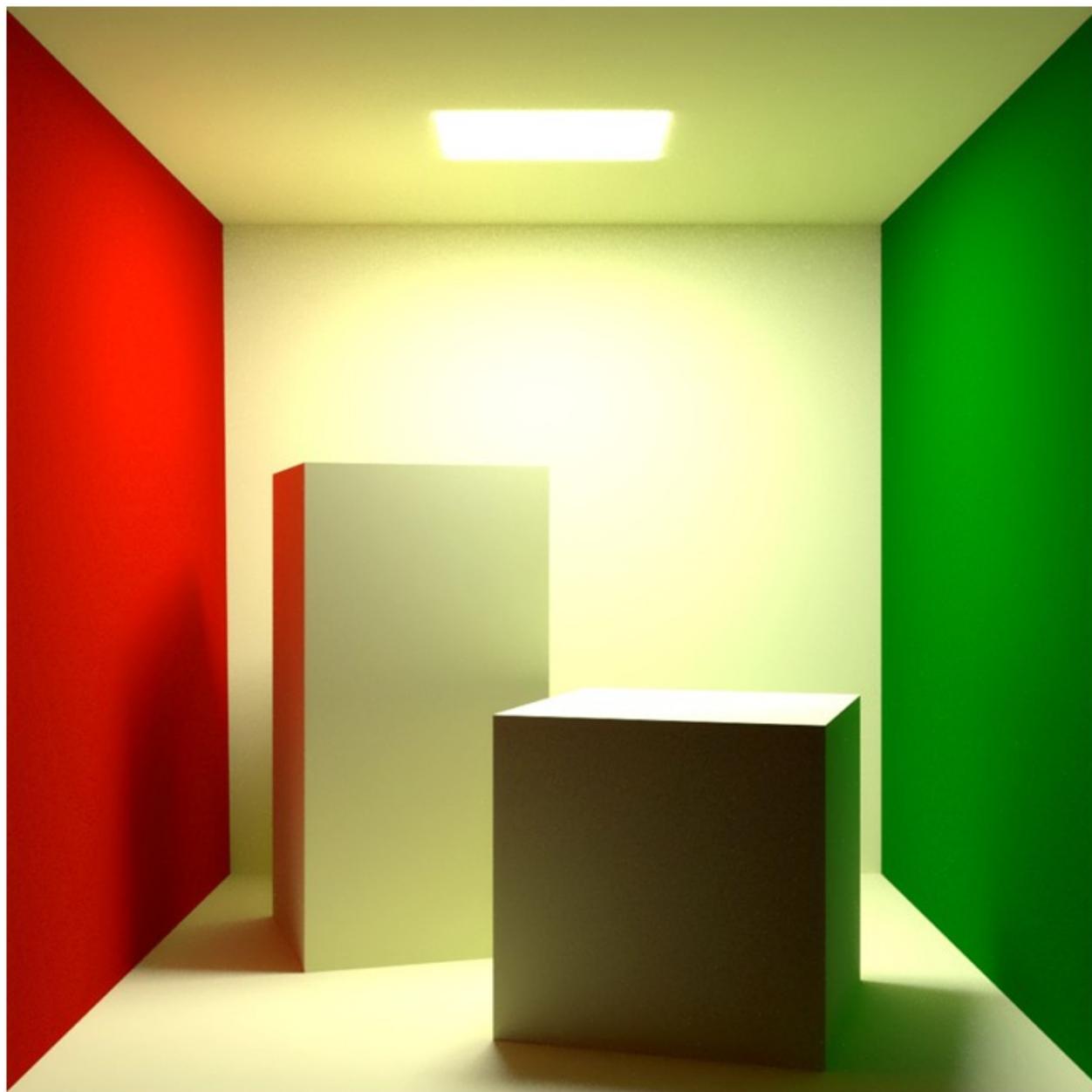


Fig. 3.8

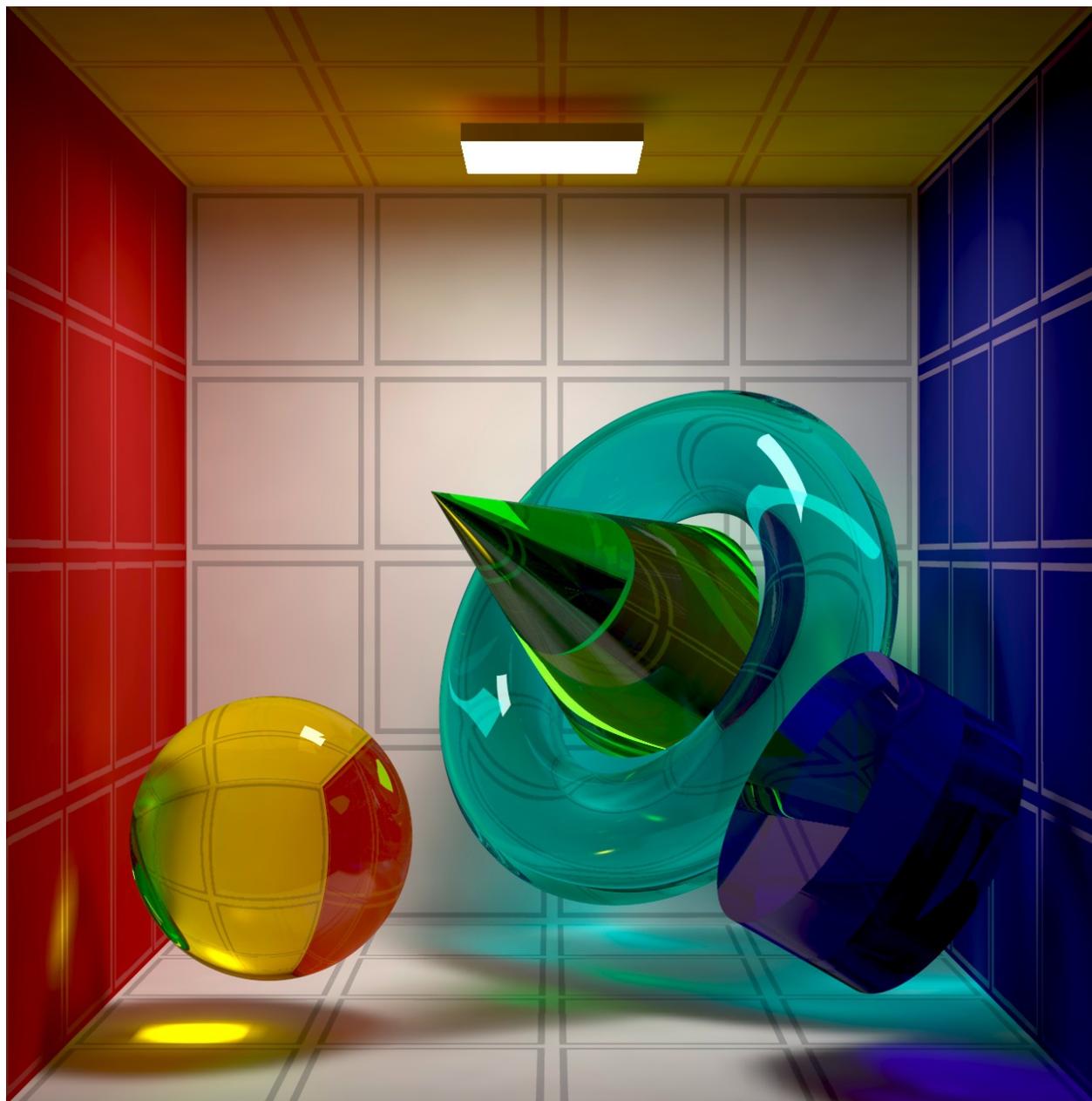


Fig. 3.9

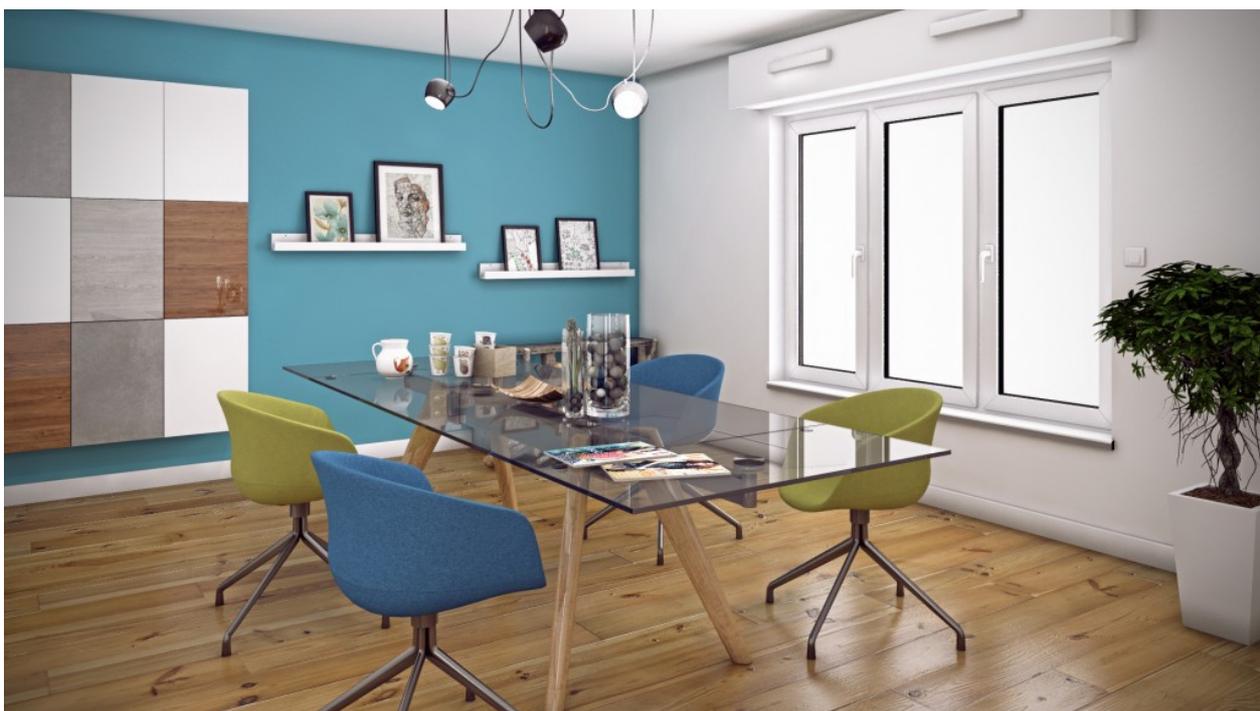


Fig. 3.10



Fig. 3.11



Fig. 3.12

Bibliografia

- [1] Riccardo Scateni, Paolo Cignoni, Claudio Montani, Roberto Scopigno, *Fondamenti di grafica tridimensionale interattiva*, Mc-Graw-Hill, 2005
- [2] Eric Haines, Pat Hanrahan, Robert L.Cook, *An introduction to Ray Tracing*, Academic Press, 1991
- [3] Michael F. Cohen, John R. Wallace, *Radiosity and Realistic Image Synthesis*, Academic Press Professional, 1995