

ALMA MATER STUDIORUM · UNIVERSITÀ DI
BOLOGNA

FACOLTÀ DI SCIENZE MATEMATICHE, FISICHE E NATURALI

CORSO DI LAUREA TRIENNALE IN INFORMATICA

Sintesi di immagini realistiche
mediante l'uso del Photon differentials
come estensione del Photon mapping

Relatore:
Chiar.mo Prof. Giulio Casciola

Tesi di Laurea di:
Fabio Di Lorenzo

Sessione I
Anno Accademico 2011-2012

Indice

Introduzione	3
0.1 Organizzazione della tesi	4
0.2 Ringraziamenti	6
1 Concetti di base e presupposti	7
1.1 Il trasporto della luce	7
1.1.1 Radiometria	8
1.1.2 Equazione di rendering	11
1.1.3 Riflessione e rifrazione della luce	12
1.2 Campionamento di sorgenti luminose	14
1.2.1 Campionamento di una sorgente isotropica	15
1.2.2 Campionamento di una sorgente sferica	17
1.2.3 Campionamento di una sorgente semisferica	19
1.3 Riepilogo	20
2 Metodi di illuminazione globale	23
2.1 Ray tracing	24
2.1.1 Ricorsione	25
2.1.2 Beam tracing	26
2.1.3 Ray differentials	26
2.2 Il metodo Monte Carlo	27
2.2.1 Simulazione Monte Carlo	28
2.2.2 Path tracing	29
2.3 Particle tracing	32

2.3.1	Photon mapping	34
2.4	Photon tracing	35
2.4.1	Emissione da una sorgente luminosa	35
2.4.2	Propagazione dei fotoni	37
2.4.3	Roulette russa	38
2.4.4	Struttura della photon map	40
2.5	Riepilogo	41
3	La stima della densità in computer grafica	43
3.1	Varianza e bias in computer grafica	44
3.2	Stima della densità nel photon mapping	44
3.2.1	Stima della radianza	45
3.2.2	Stima della densità k -nearest-neighbor	45
3.2.3	Stima della densità nella stima della radianza	46
3.3	Stima della densità e kernel locale	47
3.4	Riepilogo	48
4	Stima della densità anisotropica	51
4.1	Funzioni kernel	51
4.2	Raggi differenziali	52
4.2.1	Tecnica generale	53
4.2.2	Ray differentials tracing	55
4.2.3	Footprint per la riduzione della varianza	57
4.3	Photon differentials	60
4.4	Riepilogo	61
5	Implementazione in XCRayt	63
5.1	Implementazione	64
5.1.1	XCRayt	65
5.1.2	La classe Photon_differentials	65
5.1.3	La struttura Photon estesa	72
5.1.4	La classe Photon_map estesa	73

5.2	Integrazione in XCRayt	80
5.2.1	Parametri di configurazione	82
5.2.2	Statistiche	83
5.3	Riepilogo	84
6	Risultati sperimentali	85
6.1	Test della scena	86
	Conclusioni	97

Elenco delle figure

1.1	Campionamento di una sorgente isotropica con simmetria sferica.	16
1.2	Geometria di una sorgente sferica.	18
1.3	Geometria di una sorgente emisferica.	19
2.1	Geometria in funzione di due punti.	30
2.2	Integrale sulla emisfera superiore alla superficie.	31
2.3	Un path dal punto x_1 al punto x_5	33
4.1	Un raggio in object space ed in image space.	54
4.2	Un raggio adiacente al raggio base.	55
4.3	Regione associata al raggio base.	56
4.4	Generazione di nuove variabili nel percorso del raggio di luce. .	58
5.1	Un fotone attraversa la scena in direzione \vec{d} e colpisce una superficie nel punto \vec{p} . I vettori differenziali posizionali $\delta_x\vec{p}$ e $\delta_y\vec{p}$ moltiplicati con le corrispondenti distanze Δ_x e Δ_y definiscono un footprint ellittico asimmetrico. Insieme ad un terzo vettore normale alla superficie nel punto d'intersezione (della stessa lunghezza del semiasse minore) è possibile determinare un footprint ellissoideale asimmetrico.	67
5.2	Un fotone attraversa la scena in direzione \vec{d} e colpisce una superficie diffusiva in \vec{p} . In fase di riemissione in direzione \vec{d}' , l'origine del fotone è spostata simmetricamente indietro verso \vec{p}' , in modo che le aree A ed A' dei footprint corrispondano. .	71

5.3	(a) Sezione di regione kernel approssimata dall'area del footprint del fotone (in giallo). (b) Regione di supporto dopo la trasformazione in una sfera unitaria.	77
6.1	Scena test resa con 5000 fotoni.	88
6.2	Scena test resa con 5000 fotoni, 80 photon differentials per stima e fattore di smooth 1.	89
6.3	Scena test resa con 10000 fotoni.	90
6.4	Scena test resa con 10000 fotoni, 100 photon differentials per stima e fattore di smooth 1.	91
6.5	Scena test resa con 100000 fotoni.	93
6.6	Scena test resa con 800000 fotoni e 400 photon differentials per stima.	94
6.7	Scena test resa con 800000 fotoni, 400 photon differentials per stima e fattore di smooth 3.	95

Elenco delle tabelle

4.1	Funzioni kernel	52
4.2	Stima del footprint per la riduzione del rumore.	59

Sommario

Questa tesi intende presentare una tecnica per la sintesi di immagini realistiche al computer basata sul concetto di particle tracing. Il metodo proposto opera una stima sulla densità locale dei fotoni estendendo il concetto del photon differentials anche alla gestione delle riflessioni diffuse. Si è scelto di implementare il nuovo algoritmo di illuminazione globale all'interno di XCModel come estensione del photon mapping.

Introduzione

La simulazione realistica di fenomeni luminosi è un aspetto fondamentale della computer grafica. Gli algoritmi di illuminazione globale creano immagini foto-realistiche tramite complesse elaborazioni atte a simulare la propagazione della luce. In questo modo la scena virtuale appare indistinguibile da una fotografia del mondo reale.

I recenti progressi nel campo degli algoritmi e la crescente potenza di calcolo degli elaboratori odierni hanno favorito la diffusione di tecniche di resa realistica sempre più avanzate; le applicazioni nell'industria del desing, dell'architettura e dell'intrattenimento (produzione di film, effetti speciali e videogiochi) richiedono un'elevata qualità, e si è ormai diffusa la tendenza a sviluppare soluzioni sempre più sofisticate.

Il *particle tracing* è un concetto molto importante nel contesto dell'illuminazione globale. Gli algoritmi particle tracing hanno rappresentato negli ultimi anni un cambiamento significativo nel processo di rendering.

La maggior parte degli algoritmi particle tracing si sviluppano in due passi. Un primo passo in cui le particelle rappresentanti la luce sono emesse dalla sorgente di luce e riflesse (o rifratte) all'interno della scena, e un secondo passo in cui viene generata l'immagine della scena usando le informazioni sul trasporto della luce generate nel primo passo.

I vantaggi degli algoritmi particle tracing risiedono nella loro capacità di simulare fenomeni luminosi come le caustiche e l'illuminazione indiretta. Tali algoritmi non sono però esenti da problemi.

Ciò che caratterizza gli algoritmi particle tracing è la natura stocastica del processo di emissione e di tracciamento delle particelle attraverso i possibili percorsi della luce. Questa procedura genera del rumore (*noise*) che deve essere gestito per ricostruire in modo efficace l'illuminazione della scena.

La tecnica usata per la riduzione del rumore produce un effetto di sfocamento (*blurring*) nell'illuminazione finale. Tale tecnica diminuisce il rumore regolando l'intensità della luce (*smoothing*) ma genera effetti negativi quando l'intensità luminosa muta velocemente come nelle caustiche e nelle ombre.

Il lavoro svolto in questa tesi è finalizzato al miglioramento del trade-off tra varianza (*noise*) e bias (*blurring*). Il metodo presentato raggiunge questo scopo effettuando una stima della densità locale in combinazione con un filtro anisotropico.

Il nuovo algoritmo è stato implementato come estensione del *photon mapping*. Il photon mapping è un popolare algoritmo particle tracing sviluppato da Henrik Wann Jensen [1]. Questo algoritmo è stato d'ispirazione per la tecnica del *photon differentials* [2]. Estendendo il concetto del photon differentials anche alla gestione delle riflessioni diffuse ho ricostruito l'illuminazione globale di scene tridimensionali, senza continue interrogazioni e/o ricerche all'interno di complesse strutture dati.

La soluzione proposta migliora in modo considerevole il trade-off tra varianza e bias se paragonata con i metodi convenzionali, ed è in grado di ridurre il rumore preservando la forte illuminazione come quella che caratterizza le caustiche.

0.1 Organizzazione della tesi

- Capitolo 1: analizza i modelli fisico-matematici alla base del trasporto della luce e descrive i metodi di campionamento di varie distribuzioni utilizzate nelle differenti fasi del processo di simulazione.
- Capitolo 2: introduce le tecniche del ray tracing e del ray differentials; vengono presentati vantaggi e svantaggi dei metodi di illuminazione glo-

bale basati sull'integrazione di tipo Monte Carlo, come ad esempio il path tracing; prevede un'ampia descrizione di un importante gruppo di algoritmi multi-pass strettamente correlati al concetto di particle tracing, con particolare riferimento al photon mapping.

- Capitolo 3: illustra la stima della densità nel contesto della computer grafica e chiarifica il problema del trade-off tra varianza e bias; spiega come i problemi legati alla stima della densità influenzino in modo prevalente gli algoritmi particle tracing fornendo una panoramica sulle soluzioni adottate per diminuire tali problematiche, con particolare riferimento al photon mapping.
- Capitolo 4: si concentra sull'implementazione del ray differentials alla base della tecnica del photon differentials; esamina la ricostruzione dell'illuminazione di scene tridimensionali, in termini di stima della densità senza l'ausilio della ricerca k nearest neighbor;
- Capitolo 5: presenta il nuovo metodo basato sul photon differentials esteso all'interazione con superfici diffuse; illustra l'implementazione descrivendo in dettaglio le modifiche apportate ad XCMoel per incorporare le funzionalità del nuovo algoritmo (come quella del filtro anisotropico, in grado di adattare la stima della densità locale, in base ad una regione di supporto variabile costruita tramite il footprint).
- Capitolo 6: valuta i risultati raggiunti con il nuovo metodo comparando il photon mapping standard con il nuovo algoritmo basato sul photon differentials; mostra come il metodo presentato sia più efficace nella simulazione delle caustiche rispetto al photon mapping standard; riassume il lavoro svolto in questa tesi e suggerisce la direzione verso un futuro lavoro di ottimizzazione.

0.2 Ringraziamenti

Ringrazio il professore Giulio Casciola per avermi seguito in tutte le fasi della realizzazione di questa tesi ed avermi spinto ad impegnarmi al massimo. Ringrazio tutti i miei parenti che mi hanno sostenuto, in particolar modo, i miei genitori, che con i loro sacrifici mi hanno permesso di completare questo percorso. Ringrazio inoltre i miei amici, per l'affetto che mi hanno sempre dimostrato.

Capitolo 1

Concetti di base e presupposti

Il mondo fisico così come lo conosciamo è visibile ai nostri occhi grazie all'interazione della luce con oggetti ed atmosfera; ciò che vediamo dipende dal modo in cui la radiazione luminosa raggiunge l'osservatore.

Il calcolatore combina le informazioni sulla propagazione della luce per ricostruire l'illuminazione all'interno di una scena tridimensionale ripresa da una telecamera virtuale.

I principi fisici alla base del trasporto della luce saranno esaminati in questo capitolo.

1.1 Il trasporto della luce

La capacità di simulare il trasporto della luce che avviene nel mondo reale è la prerogativa di tutti i metodi di illuminazione globale basati su modelli fisico-matematici. L'ottica fornisce diversi modelli per descrivere il comportamento della luce [3].

Al centro degli algoritmi di illuminazione globale vi è un modello dominante, denominato *ottica geometrica* [4]. Il modello geometrico si basa sul concetto che la luce viaggia in linea retta propagandosi istantaneamente senza essere influenzata da forze esterne.

Ulteriori caratterizzazioni della luce descritte dal modello corpuscolare saranno talora considerate, come il flusso di particelle elementari associate a quantità discretizzabili di luce.

1.1.1 Radiometria

Prima di simulare il trasporto della luce in un ambiente virtuale bisogna definire le unità fisiche fondamentali che permettono di quantificare la luce presente.

Le discipline rilevanti ai fini della quantificazione della luce sono la *radiometria* e la *fotometria* [5]. La fotometria studia l'effetto della luce in base alle proprietà spettrali dell'occhio umano, mentre la radiometria è indipendente da un potenziale osservatore umano.

Per la sintesi di immagini realistiche catturate da una telecamera virtuale viene usata la teoria della radiometria.

Angolo solido

Uno dei concetti matematici che più frequentemente si incontra nella radiometria è quello di angolo solido Ω . Dato un punto \vec{x} ed una superficie S , l'angolo solido sotteso da S è l'area occupata dalla sua proiezione sulla sfera unitaria intorno ad \vec{x} . L'angolo solido sotteso dall'intera superficie sferica misura 4π mentre quello dell'emisfero superiore od inferiore alla superficie, 2π .

Radianza

Le superfici da cui viene originata la luce sono dette *sorgenti luminose*. Una sorgente di luce emette una *potenza radiante* o un *flusso radiante*, Φ , entro un angolo solido Ω sotteso da un'area A . La radianza esprime il flusso radiante per unità di angolo solido per unità di area proiettata su un piano normale ad una direzione,

$$L(\vec{x}, \vec{\omega}) = \frac{d^2\Phi}{dA^\perp d\Omega}. \quad (1.1)$$

Secondo il *teorema di conservazione della radianza*, la radianza è una grandezza invariante. Dato un mezzo senza perdite, la radianza L_i incidente in un punto \vec{x} dalla direzione di un punto \vec{y} , equivale alla radianza L_o emessa da \vec{y} verso \vec{x} ,

$$L_i(\vec{x}, \widehat{\vec{y} - \vec{x}}) = L_o(\vec{y}, \widehat{\vec{x} - \vec{y}}). \quad (1.2)$$

Grandezze cumulative

Dal calcolo integrale della radianza si possono ricavare altre quantità radiometriche fondamentali. Il flusso radiante per unità di angolo solido si ottiene integrando la radianza L sull'area A della superficie. Tale grandezza è detta *intensità radiante*,

$$I(\vec{\omega}) = \int_A L(\vec{x}, \vec{\omega}) |\vec{n} \cdot \vec{\omega}| dA = \frac{d\Phi}{d\Omega}. \quad (1.3)$$

L'*emittenza radiante* M (anche detta *radiosity* B nel contesto dell'illuminazione globale) misura il flusso radiante emesso da una sorgente per unità di area superficiale; l'*irradianza* E misura il flusso radiante incidente su una superficie per unità di area superficiale. Tali grandezze si ottengono rispettivamente integrando la radianza uscente L_o e la radianza incidente L_i su tutte le possibili direzioni della semisfera Ω ,

$$M(\vec{x}) = \int_\Omega L_o(\vec{x}, \vec{\omega}) |\vec{n} \cdot \vec{\omega}| d\Omega = \frac{d\Phi_o}{dA}, \quad (1.4)$$

$$E(\vec{x}) = \int_\Omega L_i(\vec{x}, \vec{\omega}) |\vec{n} \cdot \vec{\omega}| d\Omega = \frac{d\Phi_i}{dA}. \quad (1.5)$$

La potenza radiante totale Φ è data dall'integrale della radianza su tutte le direzioni della semisfera Ω e sull'area considerata A ,

$$\Phi = \int_A \int_{\Omega} L(\vec{x}, \vec{\omega}) |\vec{n} \cdot \vec{\omega}| d\Omega dA. \quad (1.6)$$

Sorgenti di luce puntiforme

Riducendo l'area della sorgente luminosa a zero si ottiene una determinata tipologia di luce che ha origine da un singolo punto \vec{y} , nota come *sorgente di luce puntiforme*. Dal momento che ad una sorgente di luce puntiforme non è associata alcuna superficie, essa irradia luce in tutte le possibili direzioni con simmetria sferica.

A causa dell'assenza di un'area A , l'irradianza non può essere calcolata tramite integrazione. Applicando la *legge dell'inverso del quadrato della distanza*, l'irradianza decresce in proporzione al quadrato della distanza dal punto \vec{x} sulla superficie. Se $\Phi_i = \Phi_o$ ed inserendo, $I(\vec{\omega}) = \frac{d\Phi_o}{d\Omega}$, dall'Equazione 1.3 all'interno dell'Equazione 1.5, allora

$$E(\vec{x}) = \frac{d\Phi_i}{dA} = \frac{d\Phi_o}{dA} = \frac{d\Phi_o}{d\Omega} \frac{|\vec{n} \cdot (\widehat{\vec{x} - \vec{y}})|}{\|\vec{x} - \vec{y}\|^2} = I(\widehat{\vec{x} - \vec{y}}) \frac{|\vec{n} \cdot (\widehat{\vec{x} - \vec{y}})|}{\|\vec{x} - \vec{y}\|^2}. \quad (1.7)$$

La più semplice sorgente di luce puntiforme è detta *isotropica*. Una sorgente di luce puntiforme isotropica emette una potenza radiante Φ , uniformemente in tutte le direzioni con simmetria sferica, ed intensità radiante $I(\vec{\omega}) = \frac{\Phi}{4\pi}$. Per questo tipo di sorgente puntiforme l'irradianza è:

$$E(\vec{x}) = \frac{\Phi}{4\pi} \frac{|\vec{n} \cdot (\widehat{\vec{x} - \vec{y}})|}{\|\vec{x} - \vec{y}\|^2}. \quad (1.8)$$

Fotoni

Secondo la teoria del modello corpuscolare, un flusso di particelle elementari chiamate *fotoni* trasporta una carica di energia $E = hc/\lambda$, dove h è la costante di Planck, c è la velocità della luce e λ è la lunghezza d'onda.

Il fotone ha ispirato la tecnica del photon mapping [1]. Nel photon mapping, ogni fotone trasporta una frazione della potenza radiante totale emessa dalla sorgente di luce.

Data una sorgente di luce puntiforme che emette una potenza radiante Φ_l , uniformemente in tutte le direzioni, l'emettanza radiante totale M rilasciata dalla superficie di una sfera unitaria centrata intorno alla sorgente di luce puntiforme è:

$$M = \frac{\Phi_l}{4\pi}. \quad (1.9)$$

Emettendo n_p fotoni, ognuno dei quali trasporta una frazione della potenza radiante totale della sorgente luminosa, ogni fotone rappresenta una frazione dell'area della sfera unitaria, $A_p = \frac{4\pi}{n_p}$. La quantità di potenza radiante trasportata da ogni fotone è:

$$\Phi_p = \frac{\Phi_l}{n_p}. \quad (1.10)$$

Quando il fotone viene propagato all'interno della scena e proiettato su una superficie, la sua irradianza può essere calcolata come:

$$E_p = \frac{\Phi_p}{A_p}. \quad (1.11)$$

Le informazioni memorizzate a seguito dell'interazione dei fotoni con la scena vengono utilizzate per generare l'immagine finale.

1.1.2 Equazione di rendering

Lo scopo dei metodi di illuminazione globale è quello di simulare tutti i possibili fenomeni luminosi. Questo problema è stato formalizzato e semplificato da Kajiya [6] che ha proposto un'equazione in grado di considerare la maggior parte dei fenomeni provocati dalla luce. Questa equazione è chiamata *equazione di rendering*.

Gli algoritmi di illuminazione globale possono essere classificati in numerosi modi. Uno di questi è l'approccio che essi usano per risolvere l'equazione di rendering.

L'equazione di rendering può essere espressa tramite la seguente formula:

$$L_o(\vec{x}, \vec{\omega}) = L_e(\vec{x}, \vec{\omega}) + \int_{\Omega_i} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) L_i(\vec{x}, \vec{\omega}_i) |\vec{n} \cdot \vec{\omega}_i| d\Omega_i \quad (1.12)$$

dove $L_o(\vec{x}, \vec{\omega})$ è la radianza uscente dal punto \vec{x} sulla superficie, in direzione $\vec{\omega}$, e $L_e(\vec{x}, \vec{\omega})$ è la radianza emessa. Il secondo termine nell'equazione è la radianza riflessa, L_r . Quest'ultima è espressa come l'integrale della radianza incidente, L_i e f_r , su tutte le direzioni lungo la semisfera Ω_i , dove f_r è la *BRDF* (abbreviazione di *bidirectional reflectance distribution function*) [7] ed \vec{n} è la normale alla superficie.

La radianza uscente da un dato punto \vec{x} in direzione $\vec{\omega}$ può essere approssimata dalla sommatoria di tutte le direzioni sulla semisfera incidenti sul punto \vec{x} .

1.1.3 Riflessione e rifrazione della luce

La riflessione della luce su una data superficie è modellata dalla sua BRDF. Questa funzione descrive un modello di illuminazione locale relativamente alla radianza uscente L_o in una direzione $\vec{\omega}$ per la radianza incidente da una direzione $\vec{\omega}_i$,

$$f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) = \frac{dL_o(\vec{x}, \vec{\omega})}{dE(\vec{x}, \vec{\omega}_i)}. \quad (1.13)$$

Integrando su tutte le possibili direzioni di emissione della semisfera Ω è possibile ricavare il coefficiente adimensionale della *riflettanza* ρ ,

$$\rho(\vec{x}, \vec{\omega}_i) = \int_{\Omega} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) d\Omega. \quad (1.14)$$

Per il principio fisico della *conservazione dell'energia* una superficie non può riflettere più luce di quanta ne riceve. Ovvero, $\forall \vec{\omega}_i \in \Omega$:

$$\rho(\vec{x}, \vec{\omega}_i) \leq 1. \quad (1.15)$$

Una seconda importante proprietà della BRDF è la *legge della reciprocità di Helmholtz* [8] che stabilisce come tale BRDF sia indipendente dalla direzione del raggio di luce considerato,

$$f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) = f_r(\vec{x}, \vec{\omega}, \vec{\omega}_i). \quad (1.16)$$

Quest'ultima proprietà risulta fondamentale per la realizzazione di algoritmi di illuminazione globale poichè consente di seguire il percorso della luce sia dalla sorgente all'osservatore che viceversa.

Generalizzazioni

Il modello più semplice di riflessione della luce riguarda le superfici *Lambertiane* (o diffuse ideali), le quali ridirezionano uniformemente l'irradiazione in modo perfettamente casuale nella semisfera Ω . Questo tipo di superfici hanno riflettanza $\rho_d = k_d$ e una BRDF costante:

$$f_d(\vec{x}, \vec{\omega}_i, \vec{\omega}) = k_d \frac{1}{\pi}. \quad (1.17)$$

Esiste poi un caso speciale di superfici che riflettono la radianza incidente basandosi sulla sola normale alla superficie \vec{n} . Per questo tipo di superfici, dette *perfettamente speculari*, la direzione di riflessione è ben determinabile tramite la relazione:

$$\vec{\omega}_s = 2(\vec{\omega}_i \cdot \vec{n})\vec{n} - \vec{\omega}_i. \quad (1.18)$$

Come per le superfici diffuse caratterizzate dallo loro BRDF, una superficie traslucida è caratterizzata dalla propria *BTDF* (abbreviazione di *bidirectional transmittance distribution function*). Il principio della conservazione dell'energia e la reciprocità di Helmholtz restano valide in forma analoga anche per le BTDFs [9].

Tramite la BTDF viene modellata la *rifrazione*. Il ridirezionamento della radianza cambia al variare di un *indice di rifrazione* η . Tale legame è espresso dalla *legge di Snell*:

$$\eta_1 \sin \theta_1 = \eta_2 \sin \theta_2, \quad (1.19)$$

dove η_1 ed η_2 sono gli indici di rifrazione dei due mezzi attraversati, mentre θ_1 e θ_2 sono, rispettivamente, l'angolo con cui un raggio luminoso incide sulla superficie di separazione dei due mezzi e l'angolo con cui esso viene rifratto.

Utilizzando tale relazione, la direzione $\vec{\omega}_r$ del raggio rifratto risulta essere:

$$\vec{\omega}_r = -\frac{\eta_1}{\eta_2}(\vec{\omega} - (\vec{\omega} \cdot \vec{n})\vec{n}) - \left(\sqrt{1 - \left(\frac{\eta_1}{\eta_2}\right)^2 (1 - (\vec{\omega} \cdot \vec{n})^2)} \right) \vec{n}. \quad (1.20)$$

1.2 Campionamento di sorgenti luminose

Le tecniche di campionamento svolgono un ruolo fondamentale quando la resa di immagini dipende da tecniche basate sul campionamento stocastico; come in soluzioni di tipo Monte Carlo, dove l'equazione di rendering è risolta numericamente.

In base alla semplificazione numerica usata per approssimare l'integrale dell'equazione di rendering, l'equidistribuzione dell'insieme di punti su una semisfera o su una sfera potrebbe rappresentare un aiuto importante nella riduzione dell'errore di integrazione.

Le tecniche di campionamento classiche utilizzate dal metodo Monte Carlo per risolvere l'equazione di rendering impiegano variabili casuali, $u, v \in [0, 1]$, per generare modelli di campionamento diversi per dominio di integrazione.

Di seguito verrà esaminata l'equidistribuzione di un insieme di punti su una semisfera ed una sfera unitaria mediante una tecnica di campionamento uniforme denominata *Cosine weighted*.

Il campionamento casuale di tipo cosine weighted utilizza $\varphi = 2\pi u_i$, $\theta = \arcsin \sqrt{v_i}$, $u_i, v_i \in [0, 1]$, $i = 1, \dots, N$ per generare un modello di campionamento.

1.2.1 Campionamento di una sorgente isotropica

Nel caso di una sorgente isotropica con simmetria sferica che emette particelle o fotoni, ogni elemento di angolo solido riceve lo stesso contributo dalla sorgente, come mostrato in Figura 1.1.

Occorre risolvere la funzione di densità di probabilità, *pdf*:

$$\begin{aligned} p(\theta, \phi)d\theta d\phi &= \frac{d\Omega}{4\pi} = \frac{dS}{4\pi r^2} = \frac{r^2 \sin \theta d\theta d\phi}{4\pi r^2} \\ &= p(\theta)d\theta \cdot p(\phi)d\phi = \frac{\sin \theta d\theta}{2} \cdot \frac{d\phi}{2\pi}, \end{aligned} \quad (1.21)$$

dove l'elemento di angolo solido $d\Omega = \frac{dS}{r^2}$.

Qui la funzione di densità di probabilità è separabile in due pdf:

$$p(\theta)d\theta = \frac{\sin \theta d\theta}{2}, \int_0^\pi \frac{\sin \theta}{2} d\theta = 1, \quad (1.22)$$

$$p(\phi)d\phi = \frac{d\phi}{2\pi}, \frac{1}{2\pi} \int_0^{2\pi} d\phi = 1. \quad (1.23)$$

Queste due funzioni di densità di probabilità rappresentano due variabili casuali indipendenti e possono essere campionate separatamente. Settando,

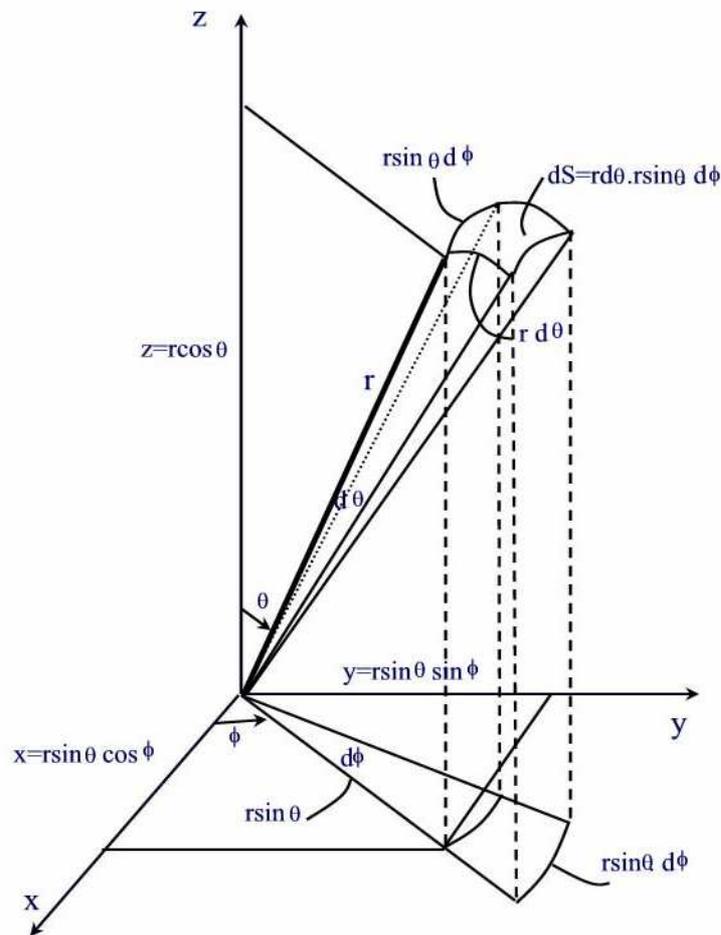


Figura 1.1: Campionamento di una sorgente isotropica con simmetria sferica.

$$C(\phi) = \int_0^\phi \frac{d\phi}{2\pi} = \rho_1,$$

il campionamento dell'angolo azimutale può essere effettuato tramite:

$$\phi(\rho) = 2\pi\rho_1. \quad (1.24)$$

Settando,

$$\begin{aligned}
C(\theta) &= \int_0^\theta \frac{\sin \theta}{2} d\theta = \rho_2, \\
1 - \cos \theta &= 2\rho_2, \\
\cos \theta &= 1 - 2\rho_2.
\end{aligned} \tag{1.25}$$

Oppure, l'angolo polare può essere campionato tramite:

$$\theta = \cos^{-1}(1 - 2\rho_2). \tag{1.26}$$

Ora che gli angoli polare ed azimutale sono stati campionati si possono facilmente determinare i coseni direttori u , v e w per le particelle emanate dalla sorgente isotropica sfruttando queste semplici relazioni:

$$\begin{aligned}
u &= \frac{x}{r} = \frac{r \sin \theta \cos \phi}{r} = \sin \theta \cos \phi, \\
v &= \frac{y}{r} = \frac{r \sin \theta \sin \phi}{r} = \sin \theta \sin \phi, \\
w &= \frac{z}{r} = \frac{r \cos \theta}{r} = \cos \theta.
\end{aligned}$$

1.2.2 Campionamento di una sorgente sferica

Data una sfera di raggio R , definiamo la pdf e la cdf in direzione radiale:

$$pdf : f(r)dr = \frac{dV}{\int dV} = \frac{4\pi r^2 dr}{\int_0^R 4\pi r^2 dr} = \frac{4\pi r^2 dr}{\frac{4}{3}\pi R^3}, \tag{1.27}$$

$$cdf : C(r) = \frac{\int_0^r 4\pi r^2 dr}{\frac{4}{3}\pi R^3} = \frac{\frac{4}{3}\pi r^3}{\frac{4}{3}\pi R^3} = \rho_3. \tag{1.28}$$

Il campionamento del raggio si effettua invertendo la cdf,

$$r = R\rho_3^{1/3}. \tag{1.29}$$

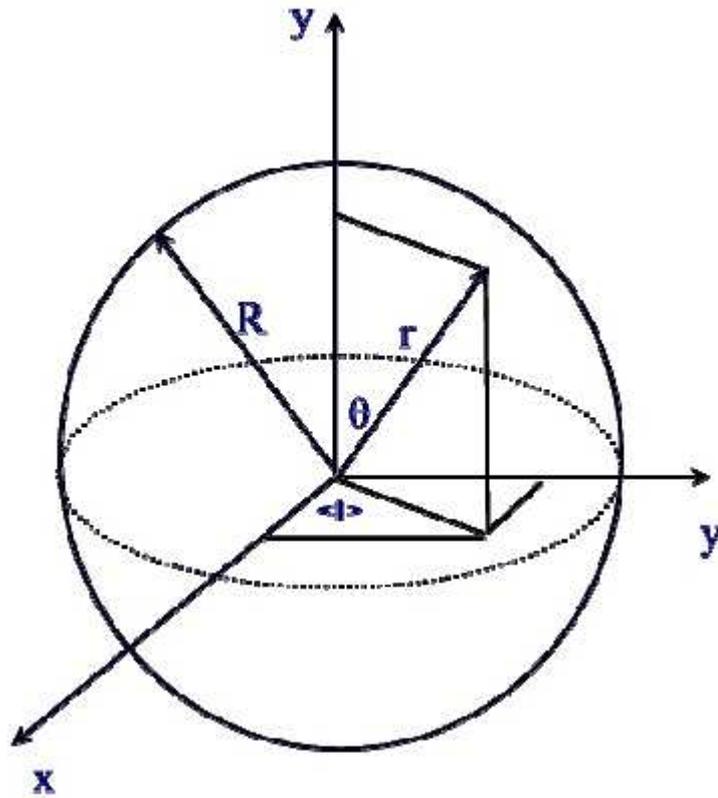


Figura 1.2: Geometria di una sorgente sferica.

Per l'angolo polare θ e l'angolo azimutale ϕ si adotta lo stesso approccio usato per il campionamento di una sorgente isotropica con simmetria sferica. In questo caso ogni elemento di angolo solido riceve lo stesso contributo dalla sorgente. Occorre risolvere la pdf come nell'Equazione 1.21.

Azimut ed inclinazione si ricavano, rispettivamente, come in Equazione 1.24 ed in Equazione 1.26.

Le coordinate campionate per tale sorgente sferica uniforme sono:

$$\begin{aligned}
 x &= r \sin \theta \cos \phi = R \rho_3^{1/3} (1 - \mu^2)^{1/2} \cos(2\pi \rho_1), \\
 y &= r \sin \theta \sin \phi = R \rho_3^{1/3} (1 - \mu^2)^{1/2} \sin(2\pi \rho_1), \\
 z &= r \cos \theta = R \rho_3^{1/3}, \\
 \text{dove: } \mu &= (1 - 2\rho_2).
 \end{aligned}$$

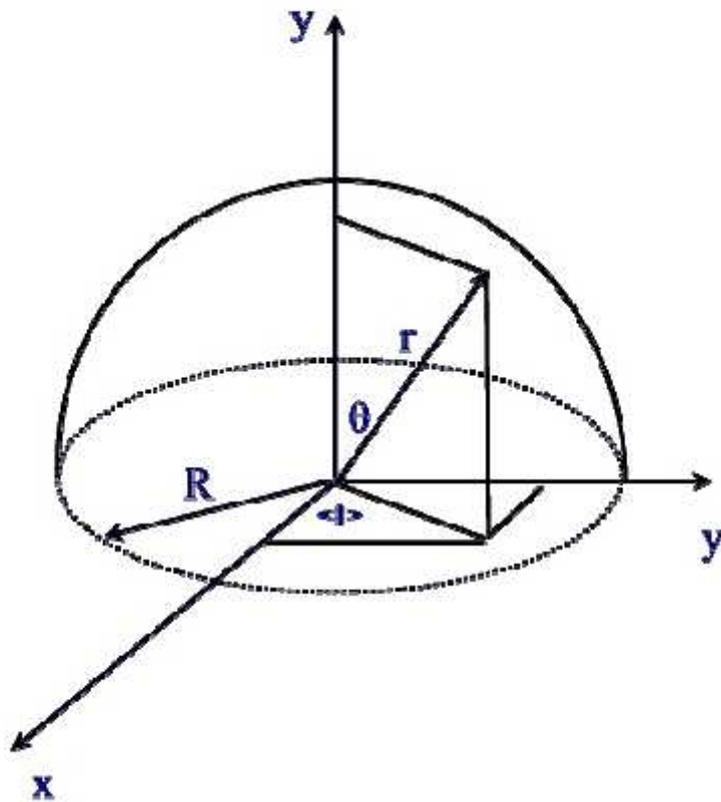


Figura 1.3: Geometria di una sorgente semisferica.

1.2.3 Campionamento di una sorgente semisferica

Questo tipo di sorgente è un caso speciale di sorgente sferica dove l'angolo polare è mostrato in Figura 1.3,

$$\theta \in [0, \frac{\pi}{2}].$$

Quindi,

$$pdf : f(\theta)d\theta = \sin \theta, \int_0^{\pi/2} \sin \theta d\theta = 1. \quad (1.30)$$

Dato che,

$$\begin{aligned} p(\theta, \phi)d\theta d\phi &= \frac{d\Omega}{2\pi} = \frac{dS}{2\pi r^2} = \frac{r^2 \sin \theta d\theta d\phi}{2\pi r^2} \\ &= p(\theta)d\theta \cdot p(\phi)d\phi = \sin \theta d\theta \cdot \frac{d\phi}{2\pi}, \end{aligned} \quad (1.31)$$

La cdf è:

$$\begin{aligned} cdf : C(\theta) &= \int_0^\theta \sin \theta d\theta \\ &= 1 - \cos \theta = \rho_2, \\ \mu &= \cos \theta = (1 - \rho_2) \approx \rho_2, \\ \theta &= \cos^{-1}(1 - \rho_2) \approx \cos^{-1} \rho_2. \end{aligned} \quad (1.32)$$

dato che $(1 - \rho_2)$ è distribuito nella stessa maniera di ρ_2 .

Il campionamento per l'angolo azimutale si effettua come per la sorgente sferica.

1.3 Riepilogo

L'obiettivo di questo capitolo è quello di fornire un valido supporto su alcuni dei concetti fondamentali nel campo della sintesi di immagini realistiche al calcolatore. Per questa ragione vengono forniti i presupposti per familiarizzare con i principali concetti della radiometria e dell'ottica.

Gli algoritmi particle tracing sono un gruppo di algoritmi di illuminazione globale basati sul metodo Monte Carlo. Il più popolare di questi algoritmi è il photon mapping di Henrik Wann Jensen [1].

L'equazione di rendering descrive matematicamente la propagazione della luce in una scena. La radianza L_o , uscente da un punto \vec{x} sulla superficie in una direzione $\vec{\omega} \in \Omega_x$ (dove Ω_x è la semisfera al punto \vec{x}) è la somma della radianza auto-emessa dalla sorgente di luce L_e e tutta la radianza riflessa.

La radianza L_e ha un valore iniziale non nullo se il punto \vec{x} considerato appartiene ad una sorgente solida. La radianza riflessa in direzione $\vec{\omega}$ è un integrale della radianza entrante da tutti i punti provenienti dalla semisfera Ω_x al punto \vec{x} , attenuata dalla BRDF della superficie $fr(-\vec{\omega}_i, \vec{x}, \vec{\omega})$ e la proiezione $\cos \theta_i$.

Quando il punto \vec{x} è su un oggetto trasparente la componente della luce trasmessa viene aggiunta all'equazione di rendering. Questa componente stima la luce totale trasmessa attraverso l'oggetto e la luce entrante al punto \vec{x} da tutte le direzioni opposte rispetto alla semisfera Ω_x .

La luce trasmessa in direzione $\vec{\omega}$ è un integrale simile all'integrale della radianza riflessa, dove il dominio di integrazione è la semisfera $\overline{\Omega}_x$ al punto \vec{x} e la BRDF è sostituita dalla BTDF della superficie. In questo caso il dominio di integrazione per risolvere l'equazione di rendering è una sfera $\Omega^{(x)}$ al punto \vec{x} , dove $\Omega^{(x)} = \Omega_x \cup \overline{\Omega}_x$.

Data la natura dell'equazione di rendering non esistono schemi universali o tecniche di campionamento più adatte per la resa di scene arbitrarie. Molte strategie di campionamento sono pensate per diversi scopi e applicazioni grafiche.

Gli algoritmi basati sul metodo Monte Carlo richiedono il campionamento di varie distribuzioni nelle diverse fasi del processo di simulazione. Il campionamento di queste distribuzioni si ottiene identificando la funzione di densità di probabilità, deducendo la funzione di distribuzione cumulativa corrispondente, ed invertendo.

Per dimostrare questo approccio sono state considerate diverse geome-

trie di sorgenti luminose campionate con la tecnica cosine weighted. Tale metodo utilizza variabili casuali per definire un modello di campionamento uniforme di un insieme di punti su una sfera unitaria, una sfera ed una semi-sfera. La qualità della distribuzione ottenuta ha un effetto sulla generazione dell'immagine.

Capitolo 2

Metodi di illuminazione globale

Gli algoritmi di illuminazione globale si possono classificare in base al loro grado di indipendenza dal punto di vista (*view-independent*). L'esempio classico di algoritmo view-independent è il *radiosity*, mentre quello di algoritmo dipendente dal punto di vista (*view-dependent*) è il *ray tracing*.

La tecnica del radiosity è stata introdotta da Goral *et al.* [10] e Kay *et al.* [11], mentre l'introduzione della tecnica del ray tracing è comunemente attribuita a Whitted *et al.* [12]. Tra queste due classi di algoritmi si collocano gli algoritmi multi-pass, i quali comprendono differenti combinazioni di ray tracing e radiosity.

I metodi view-independent sono indipendenti da una camera virtuale. Essi elaborano e memorizzano l'illuminazione globale di una scena: calcolata la distribuzione luminosa nella scena, questa può essere esaminata in tempo reale da qualsiasi punto di osservazione.

Il vantaggio dei metodi view-independent consiste nel poter generare differenti viste ad un costo computazionale minimo. D'altra parte, tali algoritmi presentano un importante handicap: dato che le informazioni sull'illuminazione globale sono precalcolate, essi richiedono molta memoria. Per limitare il consumo di memoria, gli algoritmi view-independent attuano ipotesi semplificative nei modelli fisico-matematici.

Gli algoritmi view-dependent calcolano l'illuminazione in base ad un sin-

golo punto di vista effettuando una stima dell'illuminazione percepita dall'osservatore, senza dover precalcolare e memorizzare le informazioni sull'illuminazione globale. Tra gli algoritmi di illuminazione globale che appartengono alla categoria view-dependent, si collocano il *path tracing* introdotto da Kajiya [6], ed il *path tracing bidirezionale* introdotto da Lafortune *et al.* [13].

I vantaggi di questi metodi risiedono nella loro capacità di gestire geometrie complesse e le BRDFs. Essi hanno un basso consumo di memoria e sono in grado di simulare la maggior parte dei fenomeni luminosi. Lo svantaggio principale delle soluzioni view-dependent è l'elevato costo computazionale.

Lo scopo degli algoritmi multi-pass è quello di combinare i vantaggi dei metodi view-independent e view-dependent. Il primo passo consiste in una stima dell'illuminazione indiretta tramite un algoritmo view-independent; il secondo passo utilizza un algoritmo view-dependent per elaborare le informazioni ricavate dal primo passo.

In questo capitolo si considerano gli algoritmi più rilevanti per il lavoro svolto in questa tesi.

2.1 Ray tracing

Il *ray tracing* [12] è una tecnica in grado di modellare accuratamente la maggior parte dei fenomeni luminosi. Come qualsiasi altro algoritmo di sintesi di immagini realistiche finalizzato alla risoluzione dell'equazione di rendering, il ray tracing deve localizzare la superficie più vicina visibile da un punto \vec{x} in una direzione $\vec{\omega}$. Questo vincolo è noto come *problema di visibilità*. La *rasterizzazione* fornisce una possibile soluzione a tale problema.

Si ricorre alla tecnica denominata *ray casting*. Questa tecnica effettua una ricerca delle superfici visibili ad ogni pixel del piano di vista di una telecamera virtuale.

Dato un raggio con parametro $t \geq 0$,

$$\vec{z}(t) = \vec{x} + t\vec{\omega}. \quad (2.1)$$

Questo raggio viene sottoposto ad un test d'intersezione con tutte le superfici nella scena. Ogni intersezione rilevata è un *hit point* $\vec{z}_{hit} = \vec{z}(t_{hit})$. La superficie più vicina al punto di vista è quella con il valore t_{hit} minore.

2.1.1 Ricorsione

La tecnica del ray casting è stata inizialmente proposta per risolvere il problema di visibilità da una telecamera virtuale [14]; il ray casting può essere esteso alla simulazione del trasporto della luce dalla sorgente al punto di vista. L'estensione del ray casting richiede che i raggi siano tracciati ricorsivamente ad ogni interazione con le superfici.

I raggi emessi da una telecamera virtuale o da una sorgente luminosa sono detti *raggi primari*. Questi raggi hanno generalmente origine comune e direzioni simili. I *raggi secondari* vengono generati a seguito dell'interazione con una superficie.

Per quanto riguarda i raggi secondari, il ray tracing prevede che i materiali caratterizzati da proprietà riflettenti ritrasmettano all'esterno parte della luce ricevuta tramite un nuovo raggio, detto *raggio riflesso*, avente una direzione ben determinabile secondo i criteri della riflettanza. Altresì, prendendo in considerazione quei materiali che permettono la propagazione al loro interno della luce secondo fenomeni di trasparenza e rifrazione, il lavoro aggiuntivo consiste nel generare nel punto illuminato un nuovo raggio, chiamato questa volta *raggio trasmesso*, con direzione dettata dalla legge di Snell e dipendente dai due materiali divisi dalla superficie in questione.

Data una sorgente luminosa solida o puntiforme, l'illuminazione diretta è determinata tracciando dei *raggi ombra*. Un raggio ombra stabilisce se la sorgente luminosa è visibile o occlusa rispetto ad una superficie.

A causa della limitata precisione dell'aritmetica floating point, il calcolo delle intersezioni non è perfettamente accurato. Per questa ragione viene utilizzato un offset ϵ ; nel caso dei raggi ombra, essi sono in grado di rilevare se una sorgente luminosa risulta occlusa quando $t_{hit} + \epsilon < t_{light}$.

2.1.2 Beam tracing

La ricorsione produce un successore dopo ogni interazione di un raggio isolato con una superficie. Il concetto di raggio isolato può essere ulteriormente esteso all'inclusione di raggi limitrofi.

Nel *beam tracing* [15] viene introdotto il concetto di *beam* (fascio): una piramide di raggi con vertice in un punto; le coordinate (x, y, z) del punto definiscono l'origine da cui si propagano i raggi limitrofi. Questo approccio usa la *coerenza dei raggi vicini*: un raggio e tutti i suoi vicini seguono uno stesso cammino ed intersecano una stessa superficie.

In modo analogo al ray tracing, anche nel beam tracing il percorso di un beam è tracciato ricorsivamente.

2.1.3 Ray differentials

La tecnica del *ray differentials* [16] è un metodo elegante che unisce i benefici del beam tracing alla semplicità del ray tracing.

Questo metodo associa un *footprint* ad ogni raggio considerando la convergenza o la divergenza dei raggi limitrofi. Il footprint può essere calcolato collegando i punti d'intersezione del raggio originario e dei raggi limitrofi.

Il ray differentials introduce un nuovo approccio per approssimare il footprint: dato un raggio inizialmente parametrizzato in termini di coordinate del piano di vista, (u, v) , si calcolano i valori delle derivate rispetto al piano di vista per definire posizione e direzione iniziali dei suoi vicini.

Usando le derivate appena calcolate, la distanza tra un raggio e i suoi vicini può essere stimata con la formula di Taylor del primo ordine. Questa distanza rappresenta il footprint del raggio.

Nel caso dei raggi tracciati da un sorgente luminosa, il footprint permette di associare il flusso radiante ad un'area di dimensione variabile.

La tecnica differenziale differisce dai precedenti approcci in cui si considerano percorsi infinitamente sottili. Le informazioni sul percorso di un vicino sono derivate dal percorso stesso; non vi è alcun riferimento alla geometria

differenziale o ad alcun fondamento matematico che abbia una comprensione del mondo tridimensionale.

2.2 Il metodo Monte Carlo

Il ray tracing [12] scompone la radianza in diretta ed indiretta. L'unica sorgente di luce utilizzata è di tipo puntiforme. In questo modo è possibile calcolare l'illuminazione diretta semplificando l'integrale ad una somma di contributi. Un raggio ombra viene generato per ogni sorgente luminosa presente nella scena per valutare se si verificano occlusioni.

La componente indiretta della radianza è calcolata per la sola riflessione speculare, seguendo al più una singola riflessione ed una singola rifrazione per interazione.

Gli algoritmi di tipo Monte Carlo operano una stima dell'illuminazione globale estendendo il ray tracing tradizionale alla simulazione di tutti i possibili percorsi della luce.

Un ampio gruppo di metodi di illuminazione globale utilizzano la cosiddetta *integrazione di tipo Monte Carlo* al fine di approssimare l'equazione di rendering.

Le immagini di sintesi basate sull'integrazione di tipo Monte Carlo, presentano differenze anche se generate sotto le stesse identiche condizioni e circostanze. Questa differenza (varianza) è causata dalla natura stocastica dell'integrazione di tipo Monte Carlo e si manifesta nell'immagine sottoforma di rumore.

In contrasto con i metodi deterministici che ricorrono alla semplificazione numerica degli integrali coinvolti nell'equazione di rendering, gli algoritmi Monte Carlo sono in grado di approssimare gli integrali complessi per stimare l'illuminazione globale.

Lo svantaggio degli algoritmi Monte Carlo è che essi devono gestire la varianza causata dall'integrazione di tipo Monte Carlo. Questa è un'operazione difficoltosa e computazionalmente costosa.

Gli algoritmi basati sul metodo Monte Carlo sono stati utilizzati in soluzioni view-dependent, view-independent e multi-pass. *Path tracing* [6] e *path tracing bidirezionale* [13] sono esempi di algoritmi view-dependent basati sul metodo Monte Carlo, mentre *Monte Carlo radiosity* è un esempio di algoritmo Monte Carlo view-independent [17].

L'integrazione di tipo Monte Carlo ha avuto un vasto impiego negli algoritmi particle tracing, di cui esistono soluzioni view-dependent, view-independent e multi-pass.

2.2.1 Simulazione Monte Carlo

Alla base della simulazione di alcuni processi fisici vi è una matrice comune derivante da modelli statistici, come ad esempio nell'emissione di un numero casuale di fotoni (pseudo-fotoni in quanto obbediscono a leggi geometriche anzichè fisiche) con una certa funzione di densità di probabilità che ne regola la lunghezza d'onda e la direzione.

Tramite la tecnica della simulazione Monte Carlo, il comportamento della particolare sorgente di luce può essere calcolato numericamente senza ricorrere a generalizzazioni derivanti da modelli statistici.

A fronte di una minore complessità del processo di simulazione, le simulazioni Monte Carlo raramente modellano processi fisici completi; si ricorre a processi analoghi che escludono dal modello alcuni importanti comportamenti dei processi fisici originali.

Il *Path tracing* [6] genera stocasticamente un singolo raggio di riflessione o di rifrazione, ed un singolo raggio ombra ad ogni interazione, eliminando le diramazioni durante la ricorsione; questo perchè ripetendo la stima stocastica anche per i raggi riflessi (o trasmessi) si avrebbe un'esplosione esponenziale del numero di raggi da considerare.

Il *ray tracing distribuito* [18] estende l'algoritmo del ray tracing originale. Questo metodo si basa sul *supersampling*: il sovracampionamento di ogni pixel con una distribuzione casuale e non regolare di raggi.

L'utilizzo di un campionamento distribuito in modo casuale permette di gestire le interazioni diffuse e di simulare effetti come la profondità di campo, il rendering di sorgenti luminose estese e la sfocatura di movimento (motion blur).

Nel *backward ray tracing* [19], ogni raggio trasporta una frazione della potenza radiante della sorgente di luce. Questa tecnica memorizza le informazioni all'interno di una texture map - chiamata *illumination map* - associata alla prima superficie diffusiva considerata a seguito di un'interazione di tipo speculare.

Il *path tracing bidirezionale* [13] calcola i percorsi partendo sia dal punto di vista che dalla sorgente luminosa. I punti estremi dei due percorsi vengono poi collegati attraverso raggi ombra per determinarne la reciproca visibilità e quindi l'eventuale contributo che giunge dalla sorgente.

2.2.2 Path tracing

Una delle leggi fondamentali di geometria ottica enuncia che la radianza che si propaga in linea retta in uno spazio libero non cambia lungo il suo percorso.

Assumendo un punto \vec{x}_i su una superficie non occlusa all'azione di una sorgente di luce \vec{x} , la radianza incidente è uguale alla radianza uscente:

$$L_i(\vec{x}_i, \vec{\omega}_i(\vec{x}, \vec{x}_i)) = L_o(\vec{x}, \vec{\omega}_o(\vec{x}, \vec{x}_i)). \quad (2.2)$$

Dove la direzione è una funzione di due punti:

$$\vec{\omega}(\vec{x}_1, \vec{x}_2) = \vec{\omega}(\vec{x}_1 \rightarrow \vec{x}_2) = \frac{\vec{x}_2 - \vec{x}_1}{|\vec{x}_2 - \vec{x}_1|}. \quad (2.3)$$

Nella convenzione standard, la radianza incidente L_i è espressa in direzione del punto \vec{x}_i verso la sorgente \vec{x} . Usando questa convenzione, la radianza incidente è definita in direzione opposta alla direzione di propagazione della luce,

$$L(\vec{x}_i, \vec{x}) = L(\vec{x}_i \rightarrow \vec{x}) = L_o(\vec{x}_i, \vec{\omega}(\vec{x}_i, \vec{x})). \quad (2.4)$$

Se \vec{x} è visibile da \vec{x}_i , allora \vec{x}_i è visibile da \vec{x} . Questa reciproca visibilità è rappresentata dalla funzione di visibilità, $V(\vec{x}_i, \vec{x})$, la quale ha valore 1 se il segmento che unisce \vec{x}_i ad \vec{x} non interseca nessun oggetto opaco, e valore 0 altrimenti.

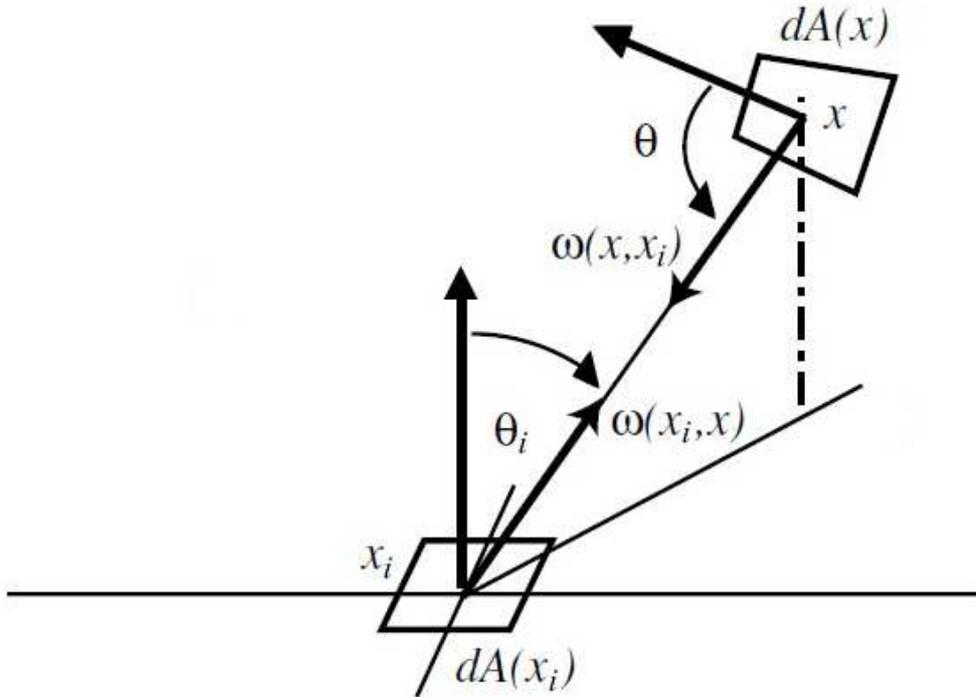


Figura 2.1: Geometria in funzione di due punti.

La radianza riflessa L_r è calcolata con l'equazione di riflessione integrando la radianza incidente su una semisfera centrata rispetto ad un punto sulla superficie ed orientata rispetto alla normale della superficie stessa,

$$L_r(\vec{x}_i, \vec{\omega}_r) = \int_{\Omega_i} f_r(\vec{x}_i, \vec{\omega}_i \rightarrow \vec{\omega}_r) L_i(\vec{x}_i, \vec{\omega}_i) \cos \theta_i d\omega_i. \quad (2.5)$$

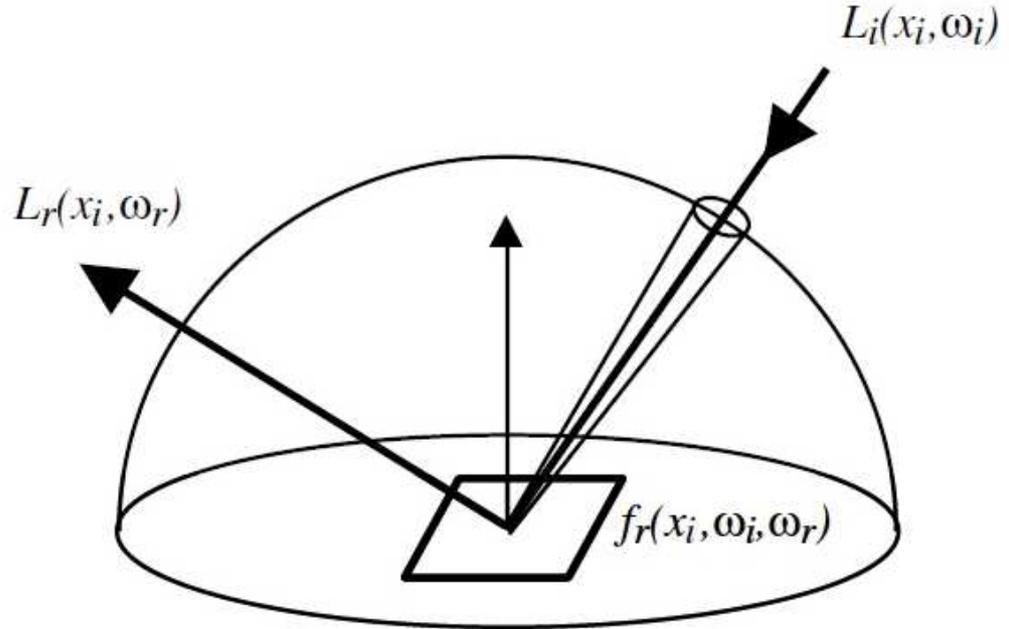


Figura 2.2: Integrale sulla semisfera superiore alla superficie.

La BRDF f_r descrive la probabilità che un raggio di luce incidente si propaghi in una direzione casuale. Per convenzione, entrambi i vettori nella BRDF hanno direzione uscente.

Per risolvere l'integrale dell'equazione di riflessione si effettua un cambio di variabile da angolo solido ad area superficiale. Il cambio di variabile consiste nell'associare l'angolo solido sotteso dalla sorgente all'area superficiale della sorgente stessa,

$$d\omega_i = \frac{\cos \theta_o}{|\vec{x}_i - \vec{x}|^2} dA(\vec{x}). \quad (2.6)$$

L'angolo solido proiettato diventa:

$$\cos \theta_i d\omega_i = G(\vec{x}_i, \vec{x}) dA(\vec{x}), \quad (2.7)$$

dove,

$$G(\vec{x}_i, \vec{x}) = G(\vec{x}, \vec{x}_i) = \frac{\cos \theta_i \cos \theta_o}{|\vec{x}_i - \vec{x}|^2} V(\vec{x}_i, \vec{x}). \quad (2.8)$$

Effettuando il cambio di variabile, l'integrale nell'equazione di riflessione diventa:

$$L_r(\vec{x}_i, \vec{\omega}) = \int_A f_r(\vec{x}_i, \vec{\omega}(\vec{x}_i, \vec{x}), \vec{\omega}) L_o(\vec{x}, \vec{\omega}(\vec{x}, \vec{x}_i)) G(\vec{x}_i, \vec{x}) V(\vec{x}_i, \vec{x}) dA(\vec{x}). \quad (2.9)$$

Dato che la radianza uscente L_o è la somma della radianza emessa L_e e della radianza riflessa L_r , inserendo l'equazione di riflessione si ricava la seguente equazione di rendering:

$$L(\vec{x}_i, \vec{\omega}) = L_e(\vec{x}_i, \vec{\omega}) + \int_A f_r(\vec{x}_i, \vec{\omega}(\vec{x}_i, \vec{x}), \vec{\omega}) L(\vec{x}, \vec{\omega}(\vec{x}, \vec{x}_i)) G(\vec{x}_i, \vec{x}) V(\vec{x}_i, \vec{x}) dA(\vec{x}). \quad (2.10)$$

Diversamente dalla funzione di emissione L_e e dalla BRDF f_r , le quali dipendono dalla geometria della scena, dalle caratteristiche dei materiali, e dalle sorgenti luminose, la radianza L non è conosciuta e deve essere calcolata per tutte le superfici considerate.

L'Equazione 2.10 è un esempio di *equazione integrale*, in quanto la radianza L rappresenta l'incognita sotto il segno dell'integrale. Lo scopo principale del path tracing di tipo Monte Carlo è risolvere questa equazione.

2.3 Particle tracing

Il *particle tracing* è un'importante tecnica adottata in molti algoritmi di illuminazione globale. Nel particle tracing, le particelle sono tracciate dalla sorgente di luce e attraverso la scena usando il ray tracing classico o il ray tracing basato sul metodo Monte Carlo.

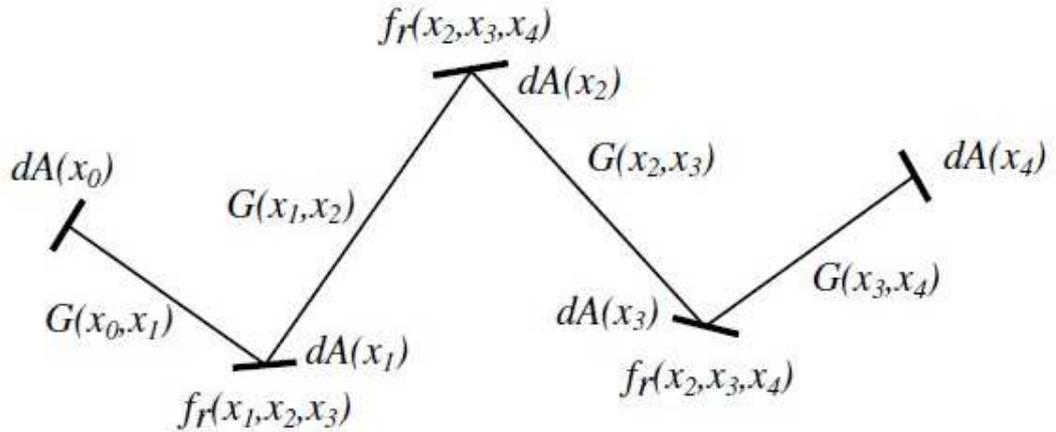


Figura 2.3: Un path dal punto x_1 al punto x_5 .

A seguito delle interazioni con le superfici della scena, le informazioni associate alle particelle sono memorizzate in una struttura dati intermedia di qualche tipo; al termine del processo di tracciamento delle particelle, l'illuminazione nella scena viene ricostruita usando le informazioni assegnate ad ogni particella.

Nel caso di algoritmi view-independent, l'illuminazione generata può essere visualizzata direttamente. Nel caso di algoritmi multi-pass, l'illuminazione deve essere ricostruita mediante tecniche quali il ray tracing o il ray tracing Monte Carlo, questa volta tramite raggi tracciati dalla telecamera virtuale.

Un importante vantaggio del particle tracing è la capacità di eliminare il rumore presente durante la ricostruzione dell'illuminazione basata sul ray tracing di tipo Monte Carlo.

Lo svantaggio consiste nella gerazione di un errore sistematico (bias) introdotto durante l'eliminazione del rumore. Il bias si manifesta sottoforma di offuscamento dell'illuminazione (blur) imposto sull'immagine di sintesi.

Un numero consistente di algoritmi di illuminazione globale adottano il concetto di particle tracing; essi si distinguono l'uno dall'altro per le tecniche usate nelle differenti parti dell'algoritmo. Le principali differenze sono:

- la tecnica usata per tracciare le particelle;
- il tipo di struttura dati usata per memorizzare le particelle;
- il numero ed il tipo di informazioni memorizzate per ogni particella;
- la tecnica usata per ricostruire l'illuminazione;
- il metodo usato per visualizzare la scena.

La combinazione delle tecniche impiegate e le modalità di utilizzo delle stesse determinano l'efficacia e le limitazioni degli algoritmi di illuminazione globale basati sul particle tracing.

2.3.1 Photon mapping

L'algoritmo del photon mapping è un popolare algoritmo particle tracing introdotto da Henrik Wann Jensen [1] come un'estensione della tecnica del ray-tracing classico allo scopo di sostituire la tecnica del ray-tracing Monte Carlo.

Nel photon mapping l'illuminazione indiretta è ricostruita mediante una serie di richieste ad una struttura dati intermedia, chiamata *photon map*, al fine di stimare la densità locale per un dato insieme di punti su una data superficie e, conseguentemente, calcolare la radianza riflessa rispetto ad un dato punto di vista. Jensen chiama questa stima *radiance estimate*

L'algoritmo base del photon mapping si sviluppa in due passi:

- Photon tracing: gestisce il popolamento della photon map usando la tecnica del ray-tracing Monte Carlo per tracciare i raggi dalla sorgente luminosa.
- Radiance estimate: traccia i raggi dal punto di vista tramite la tecnica del ray-tracing per generare l'immagine finale con l'ausilio della photon map.

La sezione seguente descrive dettagliatamente la fase del photon tracing; la stima della radianza è invece approfondita sotto molteplici aspetti nel capitolo successivo.

2.4 Photon tracing

Nella fase del photon tracing, ogni sorgente luminosa presente nella scena emette quantità di fotoni; ogni fotone può essere definito come una particella elementare che possiede un flusso radiante Φ_p che attraversa una superficie in una specifica direzione.

La distribuzione della luce attraverso la scena è poi simulata secondo il modello geometrico. Il percorso di ogni fotone è descritto da un raggio rettilineo che si propaga all'interno della scena in base alle proprietà del materiale delle varie superfici. Le intersezioni fotone-superficie rilevanti, lungo ciascun percorso, sono memorizzate all'interno della photon map.

Nella terminologia usata da Jensen [1], il photon tracing si suddivide in tre distinti stadi: *emission*, *scattering* e *storing*. Il primo stadio indica l'emissione dei fotoni dalla sorgente di luce; lo scattering determina come i fotoni vengono dispersi all'interno della scena; lo storing effettua il salvataggio delle informazioni a seguito delle interazioni dei fotoni con superfici diffuse.

2.4.1 Emissione da una sorgente luminosa

La posizione e la direzione iniziale di ogni fotone dipendono dal tipo di sorgente luminosa da cui questi vengono originati. Ogni sorgente luminosa presente nella scena può essere di vario tipo. Nel caso di una sorgente luminosa di tipo puntiforme, l'origine del fascio luminoso è posta al centro di una sfera di raggio unitario e la direzione è scelta casualmente all'interno di tale sfera mediante una tecnica di *rejection sampling*.

Il rejection sampling consiste nel generare valori casuali e nel considerare validi solo quelli che soddisfano le caratteristiche volute. Nel caso di luci puntiformi, si generano le coordinate che esprimono la direzione del fotone

emesso all'interno di un cubo unitario centrato sulla sorgente e si considerano validi solo i valori che appartengono anche alla sfera unitaria che circonda la sorgente. Nel caso di sorgenti luminose più complesse questa metodologia deve essere estesa in modo da prendere in considerazione la topologia della sorgente.

Per luci complesse, come ad esempio luci aventi una superficie, l'origine è un punto scelto casualmente sulla superficie (usando, come suggerito da Jensen [1], una tecnica di rejection sampling simile a quella utilizzata per scegliere la direzione nel caso di luci puntiformi) e la direzione del raggio è scelta in maniera casuale attorno all'emisfero di raggio unitario esterno alla superficie e avente direzione corrispondente alla normale della superficie stessa.

Per evitare l'aumento o la diminuzione di energia nel sistema, la somma della potenza radiante dei fotoni emessi da una data sorgente luminosa deve sempre essere uguale alla potenza radiante della sorgente luminosa stessa. Data una sorgente luminosa di tipo puntiforme, la potenza radiante di ogni fotone è inizializzata per essere una frazione della corrispondente potenza radiante Φ_l emessa uniformemente in tutte le direzioni dalla sorgente luminosa.

Considerando n_p fotoni emessi dalla sorgente di luce, ognuno dei quali rappresenta una frazione della potenza radiante totale, la soluzione più semplice consiste nel dividere la potenza radiante della sorgente luminosa in modo uniforme tra i fotoni da essa emessi. Ogni fotone trasporta una quantità di potenza radiante Φ_p pari a:

$$\Phi_p = \frac{\Phi}{n_p}. \quad (2.11)$$

Il numero di fotoni emessi non è un parametro diretto della sorgente luminosa. Questo valore è dedotto dallo specifico numero di fotoni dell'intera scena, così come la potenza radiante Φ_l di ogni sorgente di luce è ricavata in base alla potenza radiante totale di tutte le sorgenti luminose della scena Φ_{scene} . Il numero di fotoni emessi per una singola sorgente di luce è definito come:

$$n_p = \frac{\Phi_l}{\Phi_{scene}} \Phi_{scene}. \quad (2.12)$$

Jensen [1] osserva che per ottenere una buona qualità della stima della radianza è importante che tutti i fotoni abbiano approssimativamente potenza radiante uguale. Per i fotoni emessi usando il modello sopra descritto, questo è facilmente verificabile combinando l'Equazione 2.12 all'interno dell'Equazione 2.11, e poi riducendo:

$$\begin{aligned} \Phi_p &= \frac{\Phi_l}{\left(\frac{\Phi_l}{\Phi_{scene}} n_{scene}\right)} \\ &= \frac{\Phi_{scene}}{n_{scene}}. \end{aligned} \quad (2.13)$$

2.4.2 Propagazione dei fotoni

Nell'algoritmo del photon mapping, i fotoni emessi dalla sorgente luminosa sono propagati attraverso la scena seguendo grossomodo le stesse regole del ray-tracing Monte Carlo. A seguito dell'emissione di un fotone, questo viene tracciato attraverso la scena per simulare la dispersione della luce. Ogni fotone è trattato come un raggio che viene sottoposto a test per la verifica delle effettive intersezioni con le superfici della scena, dove, per ogni intersezione fotone-superficie, le proprietà del materiale della superficie sono utilizzate per decidere il corso della successiva azione.

La direzione del flusso è calcolata in base alle caratteristiche del materiale. Quando un fotone interseca una superficie, esso può essere riflesso, rifratto o assorbito. Le funzionalità del ray tracer permettono di determinare se il raggio di luce incidente nel punto di intersezione deve essere propagato attraverso la scena sotto forma di raggio riflesso o di raggio trasmesso.

Durante il photon tracing tutte le superfici possono riflettere e trasmettere fotoni ma per ricostruire l'illuminazione globale della scena è sufficiente memorizzare tutti quei fotoni che colpiscono le superfici diffuse, e riflettere, trasmettere o assorbire gli altri. A seguito di un'interazione, la posizione,

la potenza radiante Φ_p e la direzione incidente ω_p associate al fotone sono memorizzate nella *global photon map*.

I fotoni che popolano la global photon map rappresentano l'illuminazione diretta, l'illuminazione indiretta, e le caustiche. Per ragioni di efficienza può essere conveniente usufruire di una *caustics photon map* per rappresentare le sole caustiche. Diversamente dalla global photon map - spiega Jensen [1] - la caustics photon map memorizza i fotoni che sono stati riflessi o trasmessi tramite una superficie speculare prima di colpire una superficie diffusiva, dopodiché il fotone viene terminato. Nella global photon map, a differenza di quanto avviene per la caustics photon map, le superfici diffuse riflettono i fotoni e non li terminano una volta che questi colpiscono un materiale diffusivo.

Un dato fotone può essere memorizzato più volte lungo il suo percorso. Il processo continua finché il fotone viene assorbito o fino al raggiungimento di un numero massimo di intersezioni o finché non vengono trovate ulteriori intersezioni.

2.4.3 Roulette russa

La maggior parte delle superfici non sono perfettamente riflettenti né completamente trasparenti; in genere, esse assorbono soltanto una parte della luce che le colpisce. Le superfici diffuse così come quelle speculari, riflettono la luce in direzioni multiple. Per queste ragioni, l'implementazione esplicita dello scattering normalmente richiederebbe che la potenza radiante di ciascun fotone sia regolata quando quest'ultimo viene riflesso o rifratto e che più fotoni vengano creati ad ogni intersezione fotone-superficie, per valutare tutti i possibili percorsi.

La valutazione di tutti i percorsi possibili per ogni fotone emesso e la conseguente variazione del contributo di potenza radiante per determinare il flusso di energia trasportato da ogni singola unità sono operazioni costose ed impraticabili. Per questo motivo, il photon tracing non implementa esplicitamente

lo scattering, ma utilizza un sistema di selezione probabilistica denominato *Russian roulette*.

Per comprendere la strategia adottata dalla Russian roulette bisogna innanzitutto analizzare il modello matematico che descrive la distribuzione della luce secondo il campionamento uniforme per il calcolo dell'illuminazione indiretta. In base a questo modello, quando un fotone colpisce una superficie che riflette il 75% di energia e ne assorbe il 25%, esso rimbalza in una direzione casuale uniforme mantenendo il 75% della sua energia iniziale. La quantità finale di energia trasportata è quella che resta dopo vari rimbalzi, ognuno dei quali diminuisce l'energia disponibile.

Nel sistema della Russian roulette la probabilità p che un altro rimbalzo sia valutato è una frazione dell'energia che avrebbe dovuto essere mantenuta secondo il modello matematico sopra descritto. L'idea di base consiste nell'esaminare ognuna delle riflessioni e rifrazioni a cui un singolo fotone è sottoposto nel suo tragitto e, a seconda delle proprietà delle superfici con cui avviene l'interazione scegliere, probabilisticamente, se accettare o scartare l'intero percorso. La riflessione o la trasmissione di un fotone determinano la valutazione del suo percorso completo, viceversa, l'assorbimento provoca la cancellazione del fotone e del suo intero tracciato.

Il sistema della Russian roulette adatta le possibili scelte sottoforma di intervalli numerici proporzionali alle caratteristiche del materiale di cui è composta la superficie intersecata; ogni intervallo numerico viene selezionato mediante l'attribuzione di un numero casuale, ϵ :

$$\epsilon \in [0, \rho_d] \Rightarrow \text{riflessione diffusiva}$$

$$\epsilon \in [\rho_d, \rho_d + \rho_s] \Rightarrow \text{riflessione speculare}$$

$$\epsilon \in [\rho_d + \rho_s, 1] \Rightarrow \text{assorbimento}$$

dove ρ_d e ρ_s (con $\rho_d + \rho_s \leq 1$) sono rispettivamente i coefficienti di riflessione diffusiva e riflessione speculare, ed ϵ è un valore casuale per determinare il

comportamento del fotone.

Simulare la propagazione dei fotoni secondo lo schema della Russian roulette permette di tracciare un unico percorso per ogni fotone. Il risultato è corretto se, indipendentemente dal numero di riflessioni e rifrazioni, ciascun fotone possiede la stessa quantità di energia; il fotone conserva la potenza radiante per tutta la lunghezza del suo percorso, ma si propaga con differenti probabilità.

2.4.4 Struttura della photon map

La struttura che meglio si addice a realizzare una photon map risulta essere il *kd-tree*. Il *kd-tree* può essere schematizzato come un albero multidimensionale che supporta la ricerca binaria; ogni suo nodo infatti viene utilizzato per partizionare lo spazio di ricerca secondo una delle sue dimensioni (un *kd-tree* monodimensionale è un semplice albero binario).

Nel caso della photon map la struttura risulta essere tridimensionale; ogni nodo contiene un fotone ed un riferimento al sottoalbero destro e sinistro, esso inoltre risulta associato ad uno dei piani ortogonali che contengono il fotone e che taglia in due sottopiani una delle tre dimensioni. In tal modo ognuno dei fotoni appartenenti al sottoalbero destro risultano al di sopra di tale piano, mentre quelli del sottoalbero sinistro risultano sotto tale piano.

Ciò rende possibile localizzare un fotone all'interno del *kd-tree* di n fotoni in un tempo medio $O(\log n)$, mentre nel caso peggiore di un albero fortemente sparso si arriva a $O(n)$. Nel caso di un albero bilanciato $O(\log n)$ risulta essere il tempo di ricerca nel caso peggiore.

Non volendo fare alcuna assunzione sulla disposizione dei fotoni memorizzati, si presuppone che il *kd-tree* generato durante la prima fase risulti non bilanciato; poichè esso verrà utilizzato più volte durante la successiva fase di rendering risulta conveniente operare tra le due fasi un'operazione di bilanciamento dello stesso, in modo da avere poi le migliori prestazioni in fase di ricerca.

Si osserva infine come nel caso di ricerche k-nearest-neighbors, il tempo medio richiesto sia dell'ordine di $O(k + \log n)$; questo, assieme al fatto che il kd-tree possa essere realizzato in maniera molto efficiente rende questa struttura la più adatta per la realizzazione della photon map.

2.5 Riepilogo

Questo capitolo intende fornire una panoramica sui principali metodi di illuminazione globale che hanno lo scopo di simulare tutti i possibili fenomeni luminosi.

Gli algoritmi view-dependent producono immagini di sintesi di alta qualità, ma richiedono un elevato costo computazionale; gli algoritmi view-independent sono in grado di simulare solo un insieme limitato di fenomeni luminosi e sono computazionalmente più efficienti.

La tecnica del ray tracing considera una griglia di punti sul piano di vista e traccia raggi infinitamente sottili attraverso la scena. Questo permette di simulare facilmente gli effetti di riflessione e rifrazione.

La tecnica del beam tracing estende il concetto di raggio a quello di fascio avente larghezza finita.

Di particolare interesse è l'approccio presentato nella tecnica del ray differentials, in cui il footprint del raggio è approssimato tramite vettori differenziali: le derivate parziali moltiplicate rispetto ad una distanza finita sul piano di vista.

L'integrazione di tipo Monte Carlo è applicata per stimare l'illuminazione diretta di un punto con arbitrarie proprietà riflessive. Questo metodo utilizza numeri casuali per approssimare gli integrali coinvolti nell'equazione di rendering.

Estensioni del ray tracing classico, come il ray tracing basato sul metodo Monte Carlo, sono in grado di gestire le BRDFs arbitrariamente al fine di realizzare immagini foto-realistiche.

Gli algoritmi basati sul ray tracing Monte Carlo sono largamente usati per simulare l'illuminazione globale. Un ampio gruppo di algoritmi impiegati nella sintesi di immagini realistiche ricorre invece alla tecnica del particle tracing per ricostruire l'illuminazione di scene tridimensionali.

Nel photon mapping, l'illuminazione indiretta è ricostruita attraverso una serie di richieste ad una struttura dati intermedia chiamata photon map. La photon map contiene la collezione di fotoni creata durante la fase del particle tracing in cui i fotoni sono propagati nella scena usando la tecnica del ray tracing Monte Carlo.

Ogni richiesta alla photon map è usata per stimare la radianza riflessa nel punto d'intersezione sulla superficie. In questo modo è possibile effettuare una stima locale della densità dei fotoni vicini al punto d'intersezione. Questa stima è chiamata stima della radianza.

Capitolo 3

La stima della densità in computer grafica

Con l'introduzione del particle tracing, il concetto di *stima della densità* diventa rilevante nell'ambito della computer grafica.

Durante la ricostruzione dell'illuminazione l'intensità varia in proporzione al numero di particelle per una data area. Ciò contribuisce a spiegare il fatto che la ricostruzione dell'illuminazione è in realtà un problema di stima della densità.

Un importante vantaggio del particle tracing risiede nel poter eliminare il rumore tipico che caratterizza l'illuminazione ricostruita basandosi sul ray tracing Monte Carlo puro. D'altro canto l'eliminazione del rumore presenta un compromesso che consiste nell'introduzione di un errore sistematico, detto *bias*.

Il bias si manifesta sottoforma di sfocamento dell'illuminazione. Questo costituisce un punto cruciale di questa tesi: il cosiddetto '*problema del trade-off tra varianza e bias*'.

In questo capitolo viene spiegata la stima della densità nel contesto della computer grafica, con particolare riferimento al suo impiego nel photon mapping; inoltre verrà esaminato il problema del trade-off tra varianza e bias appena introdotto.

3.1 Varianza e bias in computer grafica

I metodi di illuminazione globale che rappresentano un'alternativa alla tecnica del particle tracing, sono basati sul ray tracing Monte Carlo puro.

I vantaggi di questi metodi consistono nel poter simulare tutti i possibili percorsi della luce, nel ridurre al minimo il consumo di memoria, e nell'assenza di vincoli sulle geometrie della scena. Per eliminare la varianza introdotta da questi metodi si ricorre ad operazioni molto costose.

L'alternativa rispetto ai metodi basati sul ray tracing Monte Carlo puro è rappresentata dagli algoritmi particle tracing; il più popolare di questi è il photon mapping di Henrik Wann Jensen [1].

Il vantaggio delle tecniche basate sul particle tracing è che l'illuminazione può essere regolata per ridurre la varianza in modo molto più efficiente; diversamente dagli algoritmi basati sul metodo Monte Carlo puro, dove aumentando il numero di campioni (tipicamente attorno ai 1000, 10000 percorsi per pixel) si riduce in modo significativo la varianza della stima, nel particle tracing si ricorre allo *smoothing*.

Lo smoothing può essere una soluzione molto efficace, in particolare quando l'illuminazione cambia lentamente nello spazio, ma può generare effetti negativi quando l'illuminazione muta velocemente come nelle caustiche e nelle ombre.

Le tecniche basate sul particle tracing riducono la varianza introducendo il bias. Il bias è uno dei difetti più comuni e visibili che caratterizzano gli algoritmi particle tracing.

3.2 Stima della densità nel photon mapping

Nel photon mapping la mappa dei fotoni è usata per ricostruire l'illuminazione durante la fase della stima della radianza. La stima della radianza ha lo scopo di stimare la radianza riflessa in un dato punto nella scena collezionando i k fotoni più vicini dalla photon map. Questi fotoni approssimano la radianza incidente su quel dato punto.

L'accuratezza della stima della radianza dipende da due importanti fattori: la risoluzione della photon map ed il numero di fotoni usati per ogni stima. Il rumore appare in modo visibile nell'illuminazione se si utilizza un basso numero di fotoni, mentre l'illuminazione risulta sbiadita se si utilizza un elevato numero di fotoni.

Questi effetti negativi si manifestano in modo mutuamente esclusivo e non possono essere evitati a meno di memorizzare un numero eccessivo di fotoni nella photon map. Questo comportamento rispecchia appieno il problema del trade-off tra varianza e bias nell'algoritmo del photon mapping.

3.2.1 Stima della radianza

Nel suo libro Jensen [1] deriva un'equazione per approssimare l'integrale continuo nell'equazione di rendering ad una sommatoria dei k fotoni più vicini nella photon map.

Riformulando l'equazione di rendering, il termine della radianza riflessa in un dato punto \vec{x} viene sostituito dall'integrale della potenza radiante incidente sull'unità di area superficiale.

Dato che il flusso incidente rappresenta la radianza incidente L_i nel punto considerato e l'integrale può essere approssimato usando i k fotoni più vicini intorno al punto \vec{x} , l'equazione per la radianza riflessa diventa:

$$L_r(\vec{x}, \vec{\omega}) \approx \hat{L}_r(\vec{x}, \vec{\omega}) = \frac{1}{\pi r(\vec{x})^2} \sum_{i=1}^k f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) \Phi_i, \quad (3.1)$$

dove Φ_i è il flusso radiante rappresentato dall' i -esimo fotone nella photon map, f_r è la BRDF, e $r(\vec{x})$ è il raggio della sfera che comprende i k fotoni più vicini, dove $\pi r(\vec{x})^2$ è la sezione di area trasversale passante per il centro della sfera.

3.2.2 Stima della densità k -nearest-neighbor

Il photon mapping recupera un numero fisso k di fotoni per ottenere un *kernel bandwidth* che si adatti alle condizioni di illuminazione locali.

Il bandwidth viene calcolato in base al raggio $r(\vec{x})$ della minima sfera di contenimento che include i k fotoni più vicini intorno ad \vec{x} .

Questo introduce un collo di bottiglia: dato che il bandwidth deve essere inizializzato tramite un valore globale (r_{max}) si richiede che un insieme temporaneo di k candidati debba essere mantenuto ed aggiornato.

Una possibile ottimizzazione riguarda l'uso di una struttura ordinata del tipo max-heap (ovvero una coda con priorità dipendente dalla distanza dal punto) per mantenere i valori temporanei dei k fotoni durante la ricerca. In tal modo risulta immediato confrontare il nuovo vicino in esame con il più lontano dei fotoni già individuati.

Il contributo dei fotoni può essere calcolato solo dopo aver adattato il bandwidth $r(\vec{x})$ al termine della ricerca. La dimensione della bandwidth è importante in quanto controlla il trade-off tra varianza e bias.

Una bandwidth piccola limita il supporto dei fotoni nel calcolo della stima della radianza; questo riduce il bias ma accresce la varianza nella stima. D'altra parte, una bandwidth grande corrisponde ad una diminuzione della varianza ma, allo stesso tempo, anche ad un aumento del bias.

3.2.3 Stima della densità nella stima della radianza

Uno dei problemi derivanti dalla stima della radianza classica è che i k fotoni più vicini hanno tipicamente distanze diverse dal punto \vec{x} considerato.

Dato che l'accuratezza del contributo calcolato dalla stima della radianza aumenta al diminuire della distanza dal punto \vec{x} , la valenza dei fotoni più vicini ad \vec{x} è maggiore rispetto a quella dei fotoni più lontani.

La stima della radianza classica (Equazione 3.1) valuta il contributo di ogni fotone in modo equivalente. Jensen [1] suggerisce di pesare il contributo di ogni fotone in funzione della sua distanza da \vec{x} .

La stima della radianza è raffinata mediante una qualche funzione filtro, K , e può essere riformulata in forma più generale per essere associata a differenti tecniche di fitraggio.

Una versione generale ispirata alla stima della radianza classica è espressa dalla seguente formula:

$$\hat{L}_r(\vec{x}, \vec{\omega}) = \frac{1}{\pi r(\vec{x})^2} \sum_{i=1}^k K\left(\frac{\|\vec{x} - \vec{x}_i\|^2}{r(\vec{x})^2}\right) f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) \Phi_i, \quad (3.2)$$

dove \vec{x}_i è la posizione dell' i -esimo fotone e $K(y)$ è una funzione che valuta i fotoni in base alla loro distanza da \vec{x} .

Nella stima della densità, $K(y)$ è nota come la *funzione kernel*. La funzione kernel decresce monotonamente attribuendo un maggior peso al contributo dei fotoni vicini al punto \vec{x} rispetto a quelli più lontani.

Come si evince dall'Equazione 3.2, la valutazione di una funzione K allineata con la superficie in un dato punto \vec{x} viene effettuata per ogni i -esimo fotone recuperato dalla photon map. Per ogni fotone, il flusso radiante Φ_i è pesato in base al valore assunto dalla funzione kernel in \vec{x} .

Si noti come la funzione K si comporti come un *filtro isotropico*, dato che essa si affida alla distanza euclidea tra \vec{x}_i ed \vec{x} in rapporto alla bandwidth stimata dalla ricerca k nearest neighbor.

Dal momento che la stima della radianza è sensibile al rumore ed al bias, tipicamente l'illuminazione diretta è calcolata separatamente e il photon mapping viene impiegato per la sola illuminazione indiretta.

Nel photon mapping, la funzione kernel assume valore nullo per $-1 \geq y \geq 1$. Ciò è necessario per limitare l'insieme di dati da esaminare ad ogni stima della radianza in modo da mantenere un basso costo computazionale.

3.3 Stima della densità e kernel locale

Nel photon mapping si ricorre all'adattamento del kernel bandwidth basato sulla stima della densità k nearest neighbor per migliorare il trade-off tra varianza e bias.

Un approccio alternativo è il kernel bandwidth della stima della densità locale dei fotoni: per ogni fotone, un kernel bandwidth r_i è scelto prima della generazione dell'immagine.

Dato che ad ogni fotone è associato una predefinita regione kernel, il loro contributo è calcolato autonomamente. Il contributo della radianza uscente è calcolato indipendentemente per tutti i fotoni il cui footprint soddisfa il criterio di sovrapposizione con il dato punto \vec{x} .

Lo svantaggio di questo metodo consiste nella perdita dell'adattamento del kernel bandwidth basato sulla densità dei fotoni vicino al punto. Per questa ragione tale adattamento deve essere calcolato con metodi alternativi.

Una possibile soluzione a questo problema utilizza il concetto di *photon splatting*. I metodi basati sul photon splatting impiegano una forma di kernel bandwidth variabile; ad ogni fotone è associato una bandwidth che viene usata per stimare il contributo apportato dal fotone nell'illuminazione.

Schjøth *et al.* [2] propongono un metodo alternativo chiamato *photon differentials*. Diversamente dai metodi basati sul photon splatting, che utilizzano filtri kernel isotropici durante la ricostruzione dell'illuminazione, i kernel usati nel metodo del photon differentials sono anisotropici.

3.4 Riepilogo

Questo capitolo ha lo scopo di spiegare il ruolo della stima della densità nel contesto della computer grafica.

Molti metodi di illuminazione globale basati sulla stima della densità hanno in comune il problema del bias. Questo introduce il problema del trade-off tra varianza e bias.

L'equazione generale per la stima della radianza di Jensen utilizza, nella sua forma più semplice, la stima della densità k nearest neighbor ed un kernel uniforme.

La ricerca k nearest neighbor permette di selezionare la bandwidth in base alla concentrazione di fotoni rispetto ad un dato punto limitando ulteriormente

i difetti legati a bias e varianza nella stima della radianza; lo smoothing risulta maggiore a fronte di una scarsa densità di fotoni, minore altrimenti.

La funzione kernel è tipicamente una funzione unimodale simmetrica che decresce monotonicamente; i fotoni vicini al punto \vec{x} hanno un peso maggiore rispetto a quelli più lontani. In questo modo la funzione kernel riduce il bias, dove la variazione di densità è significativa.

Nel photon mapping, la funzione kernel $K(y)$ è una funzione che pesa il contributo del fotone (in termini di flusso radiante trasportato) in base alla distanza del fotone dal punto \vec{x} considerato nella stima della radianza.

Il photon differentials, introdotto nel prossimo capitolo è una tecnica elegante che impiega regioni kernel individuali nella stima della densità locale dei fotoni. Ciò significa che ogni stima locale si basa sulla correlazione di un numero prefissato di kernel.

Capitolo 4

Stima della densità anisotropica

In questo capitolo viene introdotto un nuovo approccio nella riduzione del problema del trade-off tra varianza e bias. Il metodo presentato è denominato *photon differentials* [2]. In termini di stima della densità, in confronto al photon mapping di Jensen [1] che utilizza un kernel isotropico che adatta le sue dimensioni in funzione della massima distanza tra i k fotoni più vicini, il photon differentials utilizza un kernel anisotropico basato sulla tecnica del *ray differentials*.

4.1 Funzioni kernel

L'uso delle funzioni kernel nella stima della radianza è finalizzato alla riduzione del problema del trade-off tra varianza e bias. Funzioni kernel differenti hanno un'efficienza differente.

Dato che la differenza riscontrabile nel miglioramento di questo trade-off è minima a fronte del particolare tipo di funzione kernel considerata, la scelta del kernel più adatto si basa su altre osservazioni, quali l'efficienza computazionale e la differenziabilità.

Jensen [1] presenta una funzione kernel denominata *cone filter*. Questo kernel è stato introdotto per ridurre il bias nell'illuminazione calcolata mediante la stima della radianza generale.

La funzione kernel che si è distinta più delle altre nel raggiungimento del miglior trade-off tra varianza e bias è detta *Epanechnikov kernel*. Questa funzione è nota in ambito statistico per le sua acclarata efficienza nella riduzione dell'errore quadratico medio della stima della densità basata su kernel multivariati.

La funzione kernel di *Silverman* del secondo ordine [20], a dispetto di una minore efficienza in termini di miglioramento del rapporto varianza/bias e di un costo computazionale superiore ha il vantaggio di essere derivabile fino al primo ordine.

Quest'ultima proprietà è molto importante per alcune tecniche di stima della densità, come ad esempio nel *diffusion based photon mapping* [21], dove la funzione kernel di Silverman è preferibile rispetto alla funzione kernel di Epanechnikov per la sua derivabilità.

Kernel	$K(y)$	
Cone	$(1 - \frac{\sqrt{ y }}{g}) / (1 - \frac{2}{3g})$	if $ y < 1, g \geq 1,$
	0	altrimenti
Epanechnikov	$2(1 - y)$	if $ y < 1,$
	0	altrimenti
Silverman	$3(1 - y)^2$	if $ y < 1,$
	0	altrimenti
	$4(1 - y)^3$	$ y < 1,$
	0	altrimenti

Tabella 4.1: Funzioni kernel

4.2 Raggi differenziali

I raggi differenziali sono noti in geometria ottica. Questa tecnica si basa sulle derivate parziali e sull'approssimazione di Taylor del primo ordine. Le derivate parziali descrivono un fascio tracciato attraverso la scena: le derivate

della direzione regolano l'estensione ed il ridirezionamento del fascio, mentre le derivate della posizione quantificano la grandezza del fascio in un dato punto generando il footprint.

Il footprint è utilizzato in molti algoritmi di illuminazione globale. Per il calcolo del footprint sono stati impiegati differenti approcci. Quando le tecniche impiegate si basano sul ray tracing classico, come nel caso del ray differentials [16], in cui ogni raggio è definito tramite due variabili, il footprint è dato dal parallelogramma costruito a partire da una coppia di vettori differenziali.

Una proprietà fondamentale dei raggi differenziali è che essi possono essere propagati attraverso la scena semplicemente calcolando i differenziali delle operazioni usate per la propagazione del raggio originale.

4.2.1 Tecnica generale

Dato un sistema ottico generico, si considerino le coordinate cartesiane locali ad un oggetto (object space) e quelle del piano dell'immagine (image space), e si preveda un raggio che attraversi tale sistema, come quello mostrato in Figura 4.1.

In questa figura, r_0 , rappresenta la posizione iniziale del raggio, mentre u_0 rappresenta la direzione iniziale del raggio; la configurazione del raggio in image space può essere individuata in funzione della configurazione iniziale del raggio in object space:

$$r'(r, u), \quad u'(r, u).$$

Con questo metodo generale è possibile determinare la posizione e la direzione di uscita di un raggio per qualsiasi posizione e direzione di ingresso date: ad esempio, dato un raggio d'ingresso (con posizione e direzione iniziali di r_0 e u_0 , rispettivamente), il raggio può essere tracciato attraverso il sistema per determinare la sua posizione e direzione d'uscita (r'_0 e u'_0).

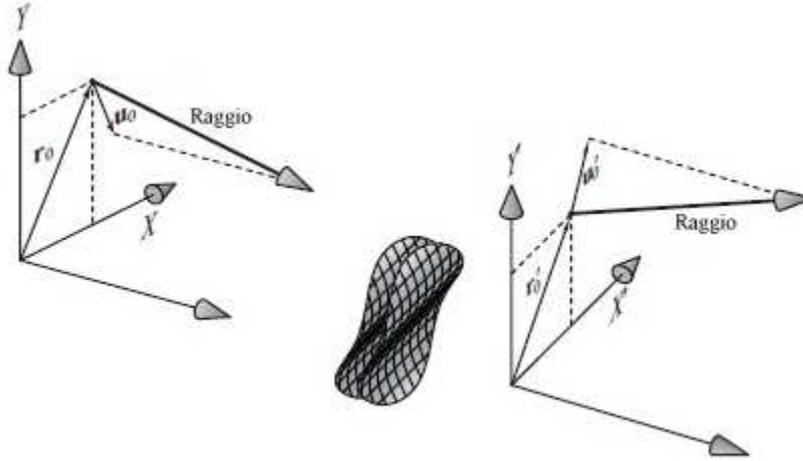


Figura 4.1: Un raggio in object space ed in image space.

Mentre questo procedimento discreto è valido per una varietà di sistemi ottici generici, in altre circostanze risulta conveniente introdurre il calcolo differenziale applicato ad un dato raggio.

Per comprendere il tipo di informazioni ricavate da un raggio differenziale, si consideri nuovamente il raggio mostrato in Figura 4.1. Le informazioni relative al raggio differenziale consentono di determinare (approssimativamente), la configurazione del raggio limitrofo al raggio considerato (da ora denominato *raggio base*), come illustrato in Figura 4.2.

In questa figura viene mostrato un raggio limitrofo al raggio base, detto *raggio adiacente*. In object space, il cambiamento di posizione e direzione del raggio adiacente rispetto al raggio base è indicato, rispettivamente, da δr e δu e, in image space, da $\delta r'$ e $\delta u'$.

Includendo le informazioni del raggio differenziale per il dato raggio base, la configurazione del raggio adiacente sono derivate per mezzo di un'approssimazione lineare:

$$r'(r, u) \approx r'_0 + A\delta r + B\delta u \Rightarrow \delta r' \approx A\delta r + B\delta u,$$

$$u'(r, u) \approx u'_0 + C\delta r + D\delta u \Rightarrow \delta u' \approx C\delta r + D\delta u.$$

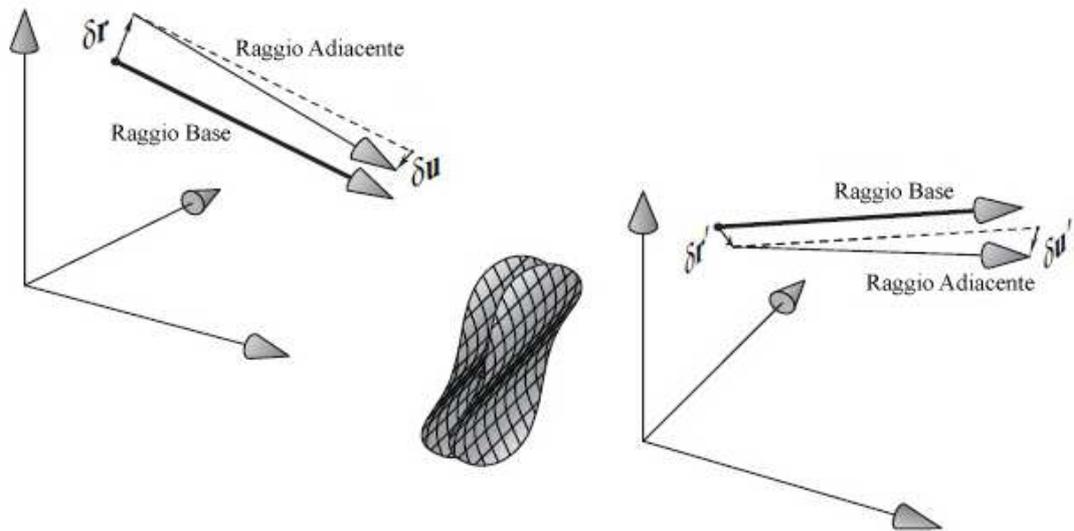


Figura 4.2: Un raggio adiacente al raggio base.

I coefficienti A , B , C e D rappresentano le informazioni del raggio differenziale relative al raggio base. Tali informazioni determinano una regione associata al raggio base in cui il comportamento dei raggi è ben definito (senza la necessità di tracciare altri raggi). Questo è illustrato schematicamente in Figura 4.3.

4.2.2 Ray differentials tracing

Nel ray tracing classico, ogni raggio è tracciato a partire dalla telecamera virtuale. Il percorso del raggio all'interno della scena dipende soltanto dalle coordinate del piano di vista; ogni altro vertice od ogni altra direzione si ricava direttamente tramite le operazioni di trasferimento, riflessione perfettamente speculare, o rifrazione.

Dato un raggio, r , definito dalla sua posizione nello spazio, \vec{p} , e dalla sua direzione, \vec{d} , la ricorsione può essere modellata attraverso una serie di funzioni su un dato insieme iniziale di parametri: il percorso del raggio attraverso la

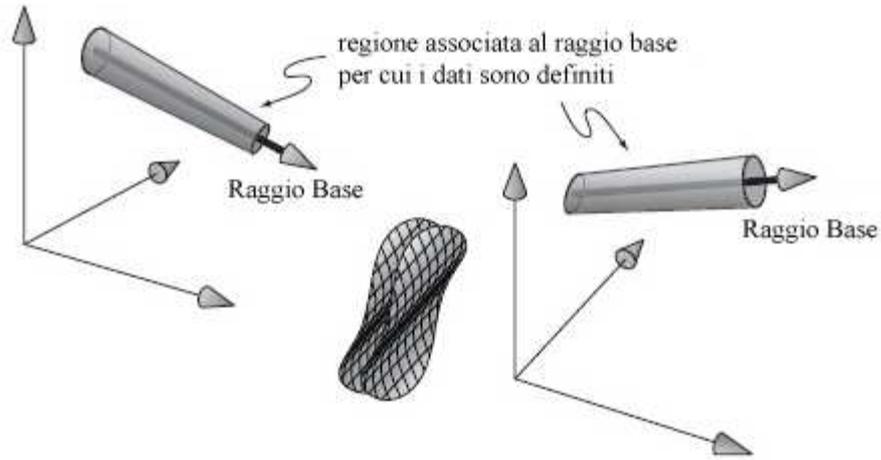


Figura 4.3: Regione associata al raggio base.

scena è tracciato applicando la serie di funzioni f_i rispetto alle coordinate (u, v) del piano di vista.

Assumendo che 'g' ed 'h' siano le corrispondenti funzioni per generare, rispettivamente, un vertice nel percorso e le direzioni nel percorso,

$$\vec{p} = g_n(g_{n-1}(\dots g_2(g_1(u, v)))),$$

$$\vec{d} = h_n(h_{n-1}(\dots h_2(h_1(u, v)))).$$

Nella tecnica del *ray differentials* introdotta da [16], le derivate parziali della posizione e della direzione calcolate rispetto alle coordinate del piano di vista definiscono la coppia di raggi differenziali:

$$\frac{\delta r}{\delta u} = \begin{pmatrix} \delta \vec{p} & \delta \vec{d} \end{pmatrix},$$

$$\frac{\delta r}{\delta v} = \begin{pmatrix} \delta \vec{p} & \delta \vec{d} \end{pmatrix}.$$

Le derivate parziali calcolate rispetto ad una coordinata del piano di vista (si assume u) sono aggiornate applicando la *regola della catena*:

$$\begin{aligned}\frac{\delta \vec{p}}{\delta u} &= \frac{\delta g_n}{\delta g_{n-1}} \cdots \frac{\delta g_2}{\delta g_1} \frac{\delta g_1}{\delta u}, \\ \frac{\delta \vec{d}}{\delta u} &= \frac{\delta h_n}{\delta h_{n-1}} \cdots \frac{\delta h_2}{\delta h_1} \frac{\delta h_1}{\delta u}.\end{aligned}$$

La differenziabilità delle operazioni impiegate per tracciare il raggio attraverso la scena consente di derivare le espressioni per la propagazione dei raggi differenziali direttamente dalle procedure originali. Per tali procedure - descritte da [16] - la derivazione risulta semplice ed immediata.

Nella terminologia usata da Suykens *et al.* [22], le derivate parziali moltiplicate ad una distanza finita Δ determinano i *vettori differenziali*. I vettori differenziali (da cui si ricava il footprint del raggio) possono essere stimati tramite l'approssimazione di Taylor del primo ordine:

$$\begin{aligned}[r(u + \Delta u, v) - r(u, v)] &\approx \Delta u \frac{\delta r}{\delta u}, \\ [r(u, v + \Delta v) - r(u, v)] &\approx \Delta v \frac{\delta r}{\delta v}.\end{aligned}$$

La scelta dell'intervallo Δ è fondamentale nel calcolo del footprint. Gli intervalli dovrebbero essere abbastanza piccoli per assicurare un contributo dell'illuminazione coerente sul footprint e, allo stesso tempo, abbastanza grandi per garantire l'efficacia delle tecniche di anti-aliasing o di riduzione della varianza.

4.2.3 Footprint per la riduzione della varianza

Il campionamento di punti in uno spazio multi-dimensionale consente di tracciare un raggio a partire dalla telecamera virtuale o dalla sorgente di luce;

ogni nuovo vertice o nuova direzione dipende dai precedenti campionamenti lungo il percorso del raggio e, possibilmente, dall'introduzione di nuove variabili (come nella generazione di una direzione riflessa a seguito dell'interazione con una superficie con proprietà diffuse).

Il calcolo di un percorso tiene conto delle sole informazioni disponibili in corrispondenza dei vertici del raggio. Il calcolo di una regione di influenza relativa ad un vertice in un percorso richiede più informazioni (come la distanza rispetto ai raggi vicini). Tale regione di influenza definisce informalmente il footprint di un raggio in un dato vertice.

La Figura 4.4 illustra il campionamento di un breve percorso di luce. Il primo vertice nel percorso, x_1 , è un punto scelto su una sorgente di luce. Per una sorgente di luce estesa, il vertice è generato attraverso un campionamento 2D usando le variabili u_1, u_2 .

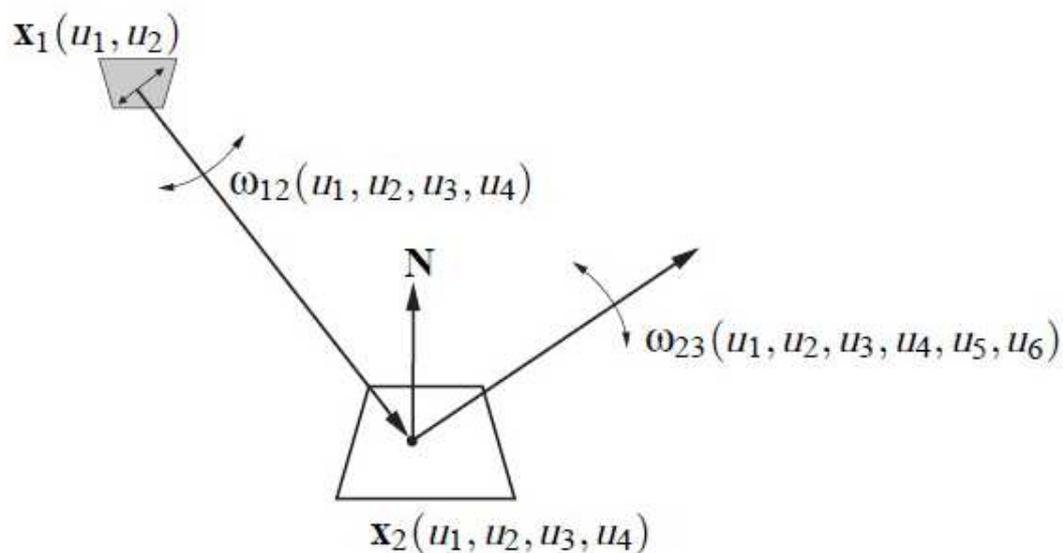


Figura 4.4: Generazione di nuove variabili nel percorso del raggio di luce.

La nuova direzione ω_{12} è campionata (ancora in 2D) seguendo una qualche funzione di distribuzione angolare. Tale distribuzione introduce le variabili u_3, u_4 per determinare, ad esempio, azimuth ed angolo di inclinazione per la nuova

direzione. Il vertice x_2 si ricava tracciando un raggio da x_1 nella direzione ω_{12} . Quest'ultima operazione non introduce nuove variabili, dato che il nuovo vertice è definito univocamente da x_2 e ω_{12} . La nuova direzione ω_{23} introduce le due nuove variabili u_5, u_6 prima di proseguire con la propagazione.

Per l'approssimazione di Taylor del primo ordine, le derivate parziali dei vertici (e delle direzioni) sono calcolate per ogni variabile del percorso. Le derivate parziali esprimono la sensitività di un vertice (o di una direzione), in termini delle variabili. Moltiplicando le derivate parziali con i corrispondenti intervalli di perturbazione si ricavano i vettori differenziali. Tali vettori differenziali permettono di stimare il footprint dei vertici.

La scelta degli intervalli di perturbazione è molto importante. Scegliendo intervalli grandi si ottengono footprint grandi; questo corrisponde ad una riduzione del rumore ma anche ad un aumento del bias. Un footprint troppo piccolo riduce in modo significativo l'effetto blurring ma non permette una gestione efficace del rumore.

L'algoritmo riportato in Tabella 4.2 fornisce una panoramica sul calcolo del footprint durante la propagazione dei percorsi.

Algoritmo
1. Tracciare il percorso \bar{x}
2. per ogni vertice x_i nel percorso (dipendente da N variabili u_k):
(a) Calcolare tutte le derivate parziali ($\frac{\delta x_i}{\delta u_k}$)
(b) Determinare l'intervallo di perturbazione appropriato Δu_k
(c) Calcolare i vettori differenziali $\frac{\delta x_i}{\delta u_k} \Delta u_k$
(d) Stimare il footprint usando i vettori differenziali
3. Valutare il contributo del footprint nella riduzione del rumore

Tabella 4.2: Stima del footprint per la riduzione del rumore.

4.3 Photon differentials

Il *photon differentials* è stato sviluppato da Schjøth *et al.* [2] come approccio alternativo per ricostruire l'illuminazione indiretta attraverso l'algoritmo del photon mapping introdotto da Jensen [1].

Nel photon mapping, la propagazione e la distribuzione della luce sono approssimate emettendo e tracciando un numero finito di fotoni da ogni sorgente di luce. Ognuno di questi fotoni può essere interpretato come il centro di un fascio la cui dimensione e la cui forma cambiano durante la propagazione attraverso la scena. In questo modo è possibile approssimare la struttura e la forma della luce che raggiunge una data superficie senza dover incrementare il numero di fotoni ma, più semplicemente, soltanto con l'ausilio del fascio circostante un dato fotone.

Il concetto di photon differentials è un esempio di applicazione dei raggi differenziali al photon tracing. A seguito dell'interazione di un fotone con una superficie, il fascio circostante il fotone è proiettato sul piano tangente alla superficie nel punto di intersezione; il fascio proiettato sulla superficie definisce un footprint in grado di approssimare la struttura e la forma dell'illuminazione.

Schjøth *et al.* [2] descrivono una variante dell'algoritmo del photon mapping standard, in cui ad ogni fotone è associata una coppia di raggi differenziali per descrivere e preservare il fascio circostante durante la propagazione attraverso la scena. Per denotare i nuovi elementi discreti rappresentanti un fascio di luce, Schjøth *et al.* [2] adottano il termine photon differentials riferendosi, appunto, alla combinazione tra i fotoni ed i raggi differenziali.

Un photon differential rappresenta un fascio di luce, la cui dimensione e forma cambia durante la propagazione attraverso la scena. Un photon differential è definito da una potenza radiante, Φ_{pd} , e da una posizione \vec{p} ed una direzione \vec{d} che descrivono, in modo corrispondente, posizione e direzione del centro del fascio. Gli altri parametri che definiscono un photon differential sono gli offset posizionali e direzionali dei due raggi virtuali adiacenti che approssimano la dimensione e la forma del fascio. Un photon differential è quindi

memorizzato insieme alle informazioni sui vettori posizionali differenziali.

I photon differentials sono tracciati attraverso la scena in modo analogo ai fotoni; essi sono emessi stocasticamente dalle sorgenti luminose e tracciati per essere memorizzati nella caustic photon map.

Sebbene nella variante dell'algoritmo del photon mapping proposta da Schjøth *et al.* [2], i photon differentials sostituiscano completamente i fotoni, i due algoritmi non presentano differenze sostanziali, se non quella dell'inizializzazione e della propagazione dei differenziali $\delta\vec{p}$ e $\delta\vec{d}$, e quella del calcolo della stima della radianza che deve considerare (nell'algoritmo del photon differentials) anche la dimensione e la forma del footprint.

4.4 Riepilogo

In questo capitolo si investiga l'impiego del ray differentials [16] nel contesto del particle tracing per ricostruire l'illuminazione mediante l'uso di un nuovo metodo denominato photon differentials [2].

La prima parte del capitolo descrive brevemente la tecnica del ray differentials nelle sue generalità; si introducono i principi su cui si basa il funzionamento dei raggi differenziali per poi descrivere al meglio gli aspetti più rilevanti riguardo il loro impiego nell'algoritmo del photon mapping.

La generazione del footprint è la conseguenza della propagazione dei raggi differenziali all'interno della scena tridimensionale. Partendo da questo presupposto, la seconda parte del capitolo introduce l'algoritmo del photon differentials nel contesto della stima della densità.

Nel photon differentials, i differenziali associati a ciascun fotone devono essere tracciati insieme al fotone stesso per modellare la propagazione del footprint attraverso la scena. Per questa ragione, i vettori differenziali posizionali e quelli direzionali dei photon differentials sono mantenuti ed aggiornati durante il photon tracing usando le equazioni fornite da Igehy [16].

L'algoritmo del photon differentials differisce inoltre dal photon mapping per l'uso di un kernel anisotropico; l'accuratezza della stima della radianza è

direttamente proporzionale all'accuratezza con la quale viene pesato il contributo dei fotoni (o dei photon differentials) coinvolti nella stima. In questo capitolo viene anche fornito un gruppo di funzioni kernel che, a fronte della particolare funzione scelta, possono apportare un guadagno differente (in termini di efficienza) nel miglioramento del trade-off tra varianza e bias.

Capitolo 5

Implementazione in XCRayt

Questo capitolo descrive un'implementazione dell'algoritmo del photon differentials esteso alla gestione delle riflessioni diffuse, come integrazione al photon mapping per XCMoel.

L'approccio utilizzato da entrambi gli algoritmi prevede l'uso della stima della radianza per approssimare la distribuzione dell'illuminazione indiretta all'interno della scena. La stima della densità classica viene eseguita su punti finiti - fotoni - emessi stocasticamente all'interno della scena tramite la tecnica del ray tracing Monte Carlo.

Nel photon mapping, l'illuminazione è ricostruita, per qualsiasi punto nella scena, mediante la stima della densità k nearest neighbor, dove il kernel bandwidth è adattato alla stima della densità locale ricercando la minima distanza tra i k fotoni più vicini ed il punto considerato.

Nel metodo presentato in questa tesi, invece di ricostruire l'illuminazione usando la stima della densità classica su punti finiti, si ricorre alla tecnica del ray differentials per generare dei footprint durante il photon tracing; tale procedura consente di ricostruire l'illuminazione tramite l'impiego della stima della densità con kernel bandwidth variabile eliminando il bisogno di localizzare i k fotoni più vicini.

L'esecuzione dell'algoritmo di resa segue le linee guida espresse nei capitoli precedenti, tuttavia se la versione estesa del photon differentials non è stata

attivata attraverso i parametri di configurazione, il sistema esegue una resa secondo il proprio algoritmo di photon mapping standard (previa attivazione di quest'ultimo) o tramite il ray tracing classico.

Qualora il nuovo algoritmo risultasse attivo, nella fase del photon tracing vengono generati i footprint tramite le nuove procedure del ray differentials implementate nel sistema, in grado di preservare, per tutta la lunghezza del percorso, gli adattamenti dei footprint generati a seguito delle riflessioni speculari e delle rifrazioni.

L'immagine finale viene realizzata combinando le procedure del ray tracing standard già presenti nel sistema e quelle implementate allo scopo di ricostruire l'illuminazione indiretta tramite le nuove informazioni contenute nella global photon map.

Il metodo proposto preserva la struttura dell'illuminazione migliorando il trade-off tra varianza e bias in modo considerevole se paragonato con gli algoritmi standard basati sul concetto di particle tracing.

5.1 Implementazione

Il pacchetto *XCRayt* integrato in XCMoDel, implementa una versione base dell'algoritmo del photon mapping. Dal momento che questa versione si limita a spiegare le idee di fondo dell'approccio standard traducendole dalla teoria alla pratica, essa non include una serie di importanti ottimizzazioni che consentono all'algoritmo di produrre i migliori risultati possibili nel modo più efficiente possibile.

Molti di questi miglioramenti sono stati introdotti nei capitoli precedenti e sono ulteriormente approfonditi in questa sezione, dove si descrive dettagliatamente l'implementazione di un'estensione che perfeziona l'algoritmo originale del photon mapping estendendo il concetto di photon differentials anche alla gestione efficiente delle riflessioni diffuse.

5.1.1 XCRayt

XCModel è stato pensato in modo procedurale per la compatibilità con il C. Tuttavia, il mio primo obiettivo è stato quello di mantenere la struttura modulare del codice originale del photon mapping e di estenderla con l'introduzione di una nuova classe e di nuovi membri. L'organizzazione del codice, con l'aiuto di un approccio orientato agli oggetti ha rappresentato una fase essenziale per il raggiungimento di questo obiettivo.

Per incorporare al meglio le nuove funzionalità all'interno di XCRayt e poter usufruire dei benefici forniti da un approccio orientato agli oggetti ho utilizzato il linguaggio C++. Questa scelta si è rivelata vantaggiosa in termini di flessibilità e leggibilità del codice realizzato.

Il codice è stato suddiviso in più file sorgenti per fornire un maggior grado di modularità:

- il modulo “*photon_differentials.cc*” incapsula le funzioni per la propagazione dei footprint all'interno della scena; processa gli eventi cooperando con il photon tracing.
- I nuovi metodi (otto) dichiarati all'interno del modulo “*photon_map.cc*” forniscono il codice principale dell'applicazione. Questi metodi estendono la classe `Photon_map` con le funzionalità per il calcolo dell'illuminazione indiretta tramite il nuovo algoritmo.

5.1.2 La classe `Photon_differentials`

L'approccio adottato nel `photon_differentials` utilizza il concetto di ray differentials. Ogni fotone è dotato di vettori differenziali $\delta_x \vec{p}$ e $\delta_x \vec{d}$, che denotano le derivate parziali della posizione \vec{p} e della direzione \vec{d} del fotone rispetto ad ogni parametro x che influenza il suo percorso attraverso la scena.

Ad ogni interazione fotone-superficie, l'espansione di Taylor del primo ordine approssima il footprint attraverso cui viene definita una regione kernel anisotropica di supporto per la stima della densità locale: i vettori differenziali posizionali, rispetto a due parametri, sono proiettati sul piano tangente alla

superficie nel punto di intersezione, dove sono interpretati come il semiasse minore ed il semiasse maggiore di un footprint ellittico asimmetrico (Figura 5.1). L'asimmetria deriva dal fatto che i due vettori differenziali possono non essere ortogonali tra loro.

Le equazioni per inizializzare ed aggiornare i differenziali dopo la trasmissione attraverso la scena e la riflessione perfettamente speculare sono fornite da [16]. Tali procedure assumono che ciascun percorso è una funzione di esattamente due parametri campionati in fase di emissione.

Rispetto a questo tipo di interazioni, in cui è sufficiente mantenere soltanto i differenziali rispetto a due parametri iniziali, le interazioni con altri tipi di superfici e la tecnica della Roulette russa richiedono il campionamento di parametri aggiuntivi durante il photon tracing. L'introduzione di ulteriori parametri comporta, da un lato, la possibilità di simulare ogni possibile percorso della luce, e dall'altro, un maggior spreco di memoria con un aumento della complessità di calcolo.

In questa sezione viene descritta l'implementazione del photon differentials in grado di gestire le riflessioni diffuse mantenendo il numero di parametri sempre costante per ogni fotone.

Il codice risulta facilmente incorporabile in XCRayt. L'algoritmo del photon mapping è esteso con le nuove funzionalità messe a disposizione dalla nuova classe `Photon_differentials`:

`Photon_differentials(void)` costruttore di classe richiamato dal programma principale per istanziare dinamicamente l'oggetto `photon_diff_ext`. I vettori differenziali sono allocati come membri privati della classe ed inizializzati a zero all'interno del costruttore.

`~Photon_differentials(void)` distruttore di classe invocato dalla `delete` al termine del programma principale.

`explicit_sampling(delta, rnd1, rnd2)` emette stocasticamente fotoni da una sorgente di luce puntiforme isotropica. Il campionamento uniforme

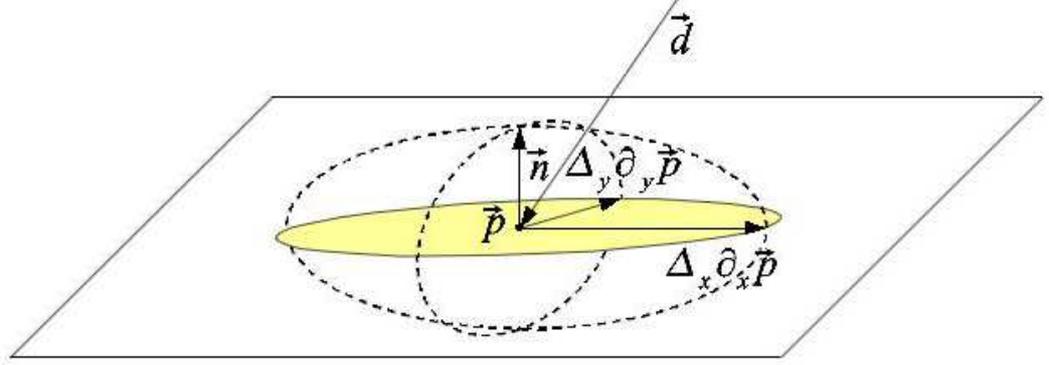


Figura 5.1: Un fotone attraversa la scena in direzione \vec{d} e colpisce una superficie nel punto \vec{p} . I vettori differenziali posizionali $\delta_x \vec{p}$ e $\delta_y \vec{p}$ moltiplicati con le corrispondenti distanze Δ_x e Δ_y definiscono un footprint ellittico asimmetrico. Insieme ad un terzo vettore normale alla superficie nel punto d'intersezione (della stessa lunghezza del semiasse minore) è possibile determinare un footprint ellissoidale asimmetrico.

di $\cos \theta \in [-1, 1]$ e $\varphi \in [0, 2\pi]$ consente di scegliere la direzione di emissione per ciascun fotone:

$$\vec{d} = \begin{pmatrix} \cos \varphi \cot \theta \\ \sin \varphi \cot \theta \\ \cos \theta \end{pmatrix}. \quad (5.1)$$

`emission(rnd1, rnd2)` determina i vettori differenziali direzionali calcolando le derivate parziali di \vec{d} , mentre i vettori differenziali posizionali sono posti a zero. I vettori differenziali direzionali risultano ortogonali tra loro, e rispetto a \vec{d} :

$$\delta_{\cos \theta} \vec{d} = \begin{pmatrix} -\cos \varphi \cot \theta \\ -\sin \varphi \cot \theta \\ 1 \end{pmatrix} \text{ e } \delta_{\varphi} \vec{d} = \begin{pmatrix} -\sin \varphi \sin \theta \\ \cos \varphi \sin \theta \\ 0 \end{pmatrix} \quad (5.2)$$

Per generare il footprint si richiede il calcolo delle distanze $\Delta_{\cos \theta}$ e Δ_{φ} rispetto a ciascun fotone. Usando l'approssimazione di Taylor del primo ordine, le direzioni di emissione sono calcolate da:

$$\vec{d} + \Delta_{\cos \theta} \delta_{\cos \theta} \vec{d} \text{ e } \vec{d} + \Delta_{\varphi} \delta_{\varphi} \vec{d}.$$

Considerando le caratteristiche di emissione della sorgente di luce isotropica, in cui la radianza è emessa uniformemente all'interno della sfera unitaria, la lunghezza delle distanze ortogonali tra \vec{d} e le direzioni di emissioni è uguale:

$$\left\| \Delta_{\cos \theta} \delta_{\cos \theta} \vec{d} \right\| = \left\| \Delta_{\varphi} \delta_{\varphi} \vec{d} \right\|. \quad (5.3)$$

Emanando un totale di n fotoni all'interno dell'angolo solido di 4π , le distanze attese sono quantificate come:

$$\left\| \Delta_{\cos \theta} \delta_{\cos \theta} \vec{d} \right\| = \left\| \Delta_{\varphi} \delta_{\varphi} \vec{d} \right\| = \sqrt{\frac{4\pi}{n}}. \quad (5.4)$$

I vettori differenziali inizializzati nell'Equazione 5.2 hanno lunghezza:

$$\left\| \delta_{\cos \theta} \vec{d} \right\| = \frac{1}{\sin \theta} \text{ e } \left\| \delta_{\varphi} \vec{d} \right\| = \sin \theta. \quad (5.5)$$

Le distanze desiderate si ricavano combinando l'Equazione 5.4 e l'Equazione 5.5:

$$\Delta_{\cos \theta} = 2\sqrt{\pi/n} \sin \theta \text{ e } \Delta_{\varphi} = 2\sqrt{\pi/n} \frac{1}{\sin \theta}. \quad (5.6)$$

Per ogni fotone bisogna tenere traccia di quattro vettori differenziali e due distanze calcolate in fase di emissione. Dato che quando un fotone

attraversa la scena i suoi differenziali cambiano ma la distanza rimane costante, una possibile ottimizzazione consiste nel ridurre la quantità di informazioni da mantenere durante la propagazione dei photon differentials.

I vettori differenziali direzionali sono inizializzati scalando $\delta_{\cos\theta}\vec{d}$ per il fattore $\sin\theta$ e $\delta_\varphi\vec{d}$ per il fattore $1/\sin\theta$, direttamente in fase di emissione. L'Equazione 5.2 diventa:

$$\delta_{\cos\theta}\vec{d} = \begin{pmatrix} -\cos\varphi\cos\theta \\ -\sin\varphi\cos\theta \\ \sin\theta \end{pmatrix} \text{ e } \delta_\varphi\vec{d} = \begin{pmatrix} -\sin\varphi \\ \cos\varphi \\ 0 \end{pmatrix} \quad (5.7)$$

Le corrispondenti distanze sono scalate per l'inverso di questi fattori generando la distanza globale $\Delta = 2\sqrt{\pi/n}$. Questa procedura consente di tenere traccia dei soli vettori differenziali usando una singola distanza globale per generare il footprint dei fotoni.

`diffuse_reflection(rnd1, rnd2)` interpreta la riflessione diffusiva come un *assorbimento* ed una *riemissione* dei fotoni. Per evitare di tracciare più differenziali a causa dell'introduzione dei nuovi parametri, $\cos\theta' \in [0, 1]$ e $\varphi' \in [0, 2\pi]$, il fotone è considerato come originato dalla superficie diffusiva con i soli parametri $\cos\theta'$ e φ' ed il numero di differenziali è mantenuto costante.

L'assorbimento e la riemissione dal punto di intersezione con i nuovi parametri comporta la perdita di qualsiasi adattamento del footprint sulla nuova origine. La soluzione a questo problema consiste nel calcolare un offset per spostare virtualmente l'origine del fotone lungo l'inverso della nuova direzione di uscita, allo scopo di preservare le informazioni sulle condizioni dell'illuminazione (Figura 5.2).

La direzione di riflessione del fotone è calcolata come nell'Equazione 5.1 utilizzando i nuovi parametri $\cos\theta'$ e φ' . I vettori differenziali posizionali sono posti a zero, mentre l'inizializzazione dei vettori differenziali

direzionali si basa sulle considerazioni precedenti sul calcolo delle distanze $\Delta_{\cos\theta'}$ e $\Delta_{\varphi'}$. In questo modo è possibile inizializzare i vettori differenziali direzionali come nell'Equazione 5.7 mantenendo un singolo Δ per il calcolo del footprint.

A seguito dell'interazione di un fotone con una superficie diffusiva, il footprint ha semiassi $\Delta\delta_{\cos\theta}\vec{p}$ e $\Delta\delta_{\varphi}\vec{p}$. L'area del footprint ellittico asimmetrico è approssimata tramite la seguente equazione:

$$A = \pi \|\Delta\delta_{\cos\theta}\vec{p} \times \Delta\delta_{\varphi}\vec{p}\| \quad (5.8)$$

Lo spostamento della nuova origine in \vec{p}' comporta l'attraversamento, da parte del fotone, di una distanza v (Figura 5.2) prima di raggiungere nuovamente il punto \vec{p} . Il tragitto compiuto permette di espandere i vettori differenziali posizionali posti inizialmente a zero:

$$\delta_{\cos\theta}\vec{p} = v\delta_{\cos\theta'}\vec{d}' \text{ e } \delta_{\varphi}\vec{p} = v\delta_{\varphi'}\vec{d}'. \quad (5.9)$$

La nuova area di footprint del fotone risulta essere:

$$A' = \pi \left\| \Delta_{\cos\theta'} v \delta_{\cos\theta'} \vec{d}' \times \Delta_{\varphi'} v \delta_{\varphi'} \vec{d}' \right\|. \quad (5.10)$$

La distanza di offset cercata si ricava rapportando A ed A' . Il metodo privato `compute_offset(void)` implementa l'equazione per il calcolo di v :

$$v = \sqrt{\frac{\|\Delta\delta_{\cos\theta}\vec{p} \times \Delta\delta_{\varphi}\vec{p}\|}{\left\| \Delta_{\cos\theta'} \delta_{\cos\theta'} \vec{d}' \times \Delta_{\varphi'} \delta_{\varphi'} \vec{d}' \right\|}} \quad (5.11)$$

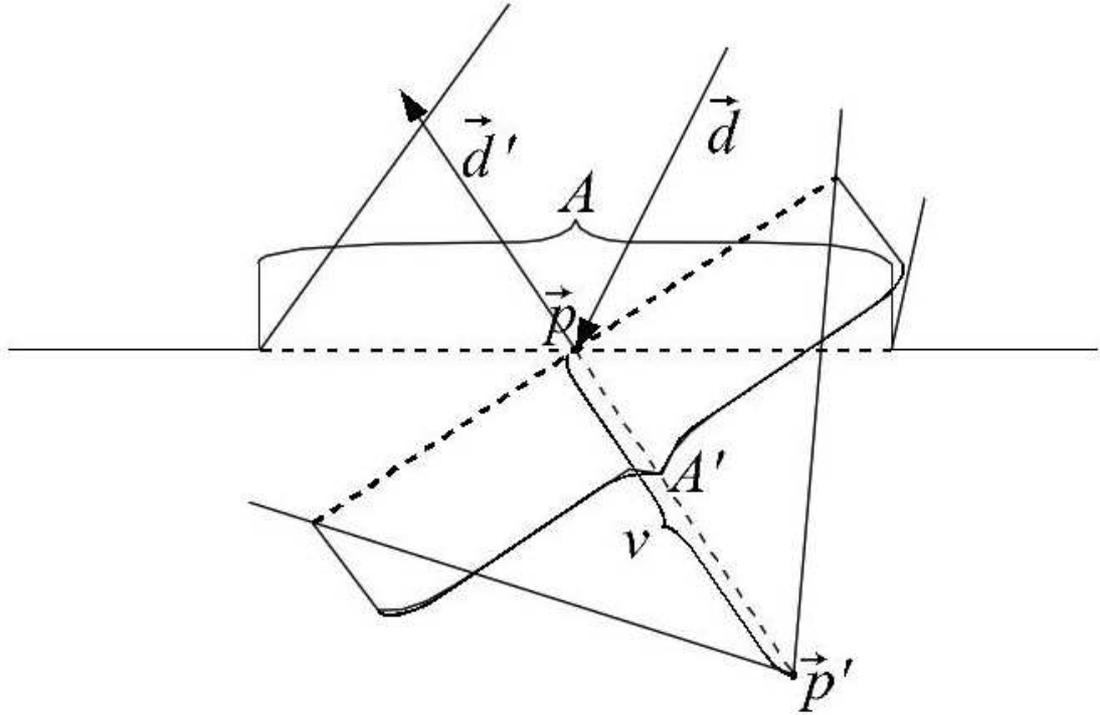


Figura 5.2: Un fotone attraversa la scena in direzione \vec{d} e colpisce una superficie diffusiva in \vec{p} . In fase di riemissione in direzione \vec{d}' , l'origine del fotone è spostata simmetricamente indietro verso \vec{p}' , in modo che le aree A ed A' dei footprint corrispondano.

`store_footprint_spanning_vectors(Mpd)` memorizza le informazioni per il calcolo dei footprint in una zona di memoria allocata allo scopo di preservare i dati durante la procedura ricorsiva del photon tracing.

Igehy [16] descrive le operazioni necessarie per il trasferimento, la riflessione speculare e la rifrazione adottate nella tecnica del photon differentials originale [2] omettendo però alcuni dettagli sulla derivazione di tali operazioni. Sporring *et al.* [23] entra nel merito dell'argomento e mostra in dettaglio le equazioni per derivare le operazioni di trasferimento, riflessione e rifrazione.

Per l'implementazione di tali funzionalità ho fatto riferimento alla notazione di Sporring *et al.* [23], mentre ho seguito le assunzioni di Igehy [16] per

la semplificazione di alcuni termini.

`transfer(delta, normal, s)` trasferisce la posizione dei differenziali nel punto di intersezione con una superficie nella scena.

`specular_reflection(N)` determina la direzione di riflessione speculare dei differenziali. La valutazione della direzione riflessa richiede in alcuni casi la derivata parziale della normale alla superficie. Questa derivazione non è stata indagata ulteriormente perchè non applicabile al caso in esame.

`refraction(ratio, cosine, N)` determina la direzione di trasmissione dei differenziali.

5.1.3 La struttura Photon estesa

La struttura `Photon` rappresenta l'unità fondamentale di memorizzazione interna e contiene le informazione relative a posizione, direzione incidente e potenza radiante; inoltre, essendo un nodo di un kd-tree, mantiene anche un riferimento al piano di suddivisione associato. L'implementazione del nuovo algoritmo estende la struttura con 52 byte aggiuntivi per fotone; la matrice di trasformazione M_{pd} è una matrice 4x3 allocata attraverso la libreria *Matrix* fornita nel pacchetto XCMoel. L'inclusione di questa libreria si è rivelata vantaggiosa per il supporto fornito nella manipolazione delle matrici ma ha comportato l'utilizzo di 12 byte extra, in aggiunta ai 40 byte normalmente richiesti nell'implementazione standard per allocare la matrice 3x3 (36 byte) e l'area di footprint (4 byte). Nell'implementazione suggerita da Schjøth *et al.* [2], la matrice di trasformazione è costruita a run-time anzichè essere precalcolata; in questo approccio, a fronte di un aumento dei tempi di elaborazione, la richiesta di memoria aggiuntiva per fotone diminuisce a 28 byte totali, sufficienti a memorizzare i vettori differenziali posizionali e l'area di footprint.

```
1 | typedef struct Photon {
```

```

2   float   pos[3];
3   short   plane;
4   unsigned char  theta, phi;
5   float   power[3];
6   /** PHOTON DIFFERENTIALS EXTENSION (40 byte) **/
7   float   Apd;
8   VEC_matrix_4x3_t  Mpd; //TO-FIX: 12-byte over
9   } Photon;

```

Listato 5.1: Struttura Photon estesa.

5.1.4 La classe Photon_map estesa

La classe `Photon_map` memorizza i photon differentials propagati all'interno della scena. La global photon map colleziona le informazioni a seguito della interazione con le superfici diffuse a prescindere dal percorso seguito dai fotoni. In questo modo, il metodo proposto estende le funzionalità del photon differentials originale [2] alla simulazione di ogni possibile fenomeno luminoso. I metodi pubblici preceduti dal suffisso `PD_` forniscono le procedure principali richieste dal nuovo algoritmo e rappresentano un'astrazione della struttura che prescinde dall'effettiva implementazione interna:

`PD_store(Mpd)` estende la funzione `store(power, position, direction)` consentendo la memorizzazione di un photon differentials all'interno della global photon map.

`PD_scale_spanning_vectors(smooth_factor)` imposta il fattore lineare di `smooth` che controlla la dimensione iniziale del footprint dei photon differentials. La dimensione del footprint corrisponde alla bandwidth nella stima della radianza. Un footprint iniziale di elevate dimensioni riduce il rumore favorendo il bias, mentre un footprint di dimensioni ridotte provoca l'effetto opposto. Nell'implementazione suggerita da Schjøth *et al.* [2], la dimensione iniziale del footprint può essere impostata manualmente o in funzione del numero di fotoni. In questa implementazio-

ne si intende fornire uno strumento che dia all'utente la possibilità di controllare manualmente il trade-off tra varianza e bias.

`PD_dampening_adaptation(void)` applica una scala non lineare ai footprint.

I migliori risultati si riscontrano con footprint sparsi sulle superfici della scena; per questa ragione è preferibile che questi abbiano una dimensione che tenda a far aumentare la varianza rispetto al bias. Al termine della fase del photon tracing, prima della costruzione della regione kernel di supporto, una scala non lineare viene applicata a tutti i footprint nella scena. Ogni semiasse dei footprint è scalato di un fattore $\|\delta_x \vec{p}\|^{-\frac{3}{4}}$ con x rappresentante il parametro del più lungo tra i due vettori differenziali posizionali,

$$\delta_x \vec{p}' = \|\delta_x \vec{p}\|^{-\frac{3}{4}} \widehat{\delta_x \vec{p}}. \quad (5.12)$$

Il vettore differenziale relativo all'altro parametro y è scalato per lo stesso fattore,

$$\delta_y \vec{p}' = \frac{\|\delta_y \vec{p}\|}{\|\delta_x \vec{p}\|} \widehat{\delta_y \vec{p}}. \quad (5.13)$$

Lo stesso fattore di scala applicato ad entrambi i differenziali permette di preservare la forma del footprint e, quindi, anche quella della regione kernel di supporto generata in seguito; in questo modo l'operazione di scala ha il solo effetto di comprimere le dimensioni dei semiassi.

`PD_compute_footprints(global_spacing)` stima l'area dell'ellisse contenuta all'interno del footprint. Schjøth *et al.* [2] ridefiniscono l'area A_{pd} del footprint, come l'area del parallelogramma costruito dai vettori differenziali posizionali, $\Delta \delta_{\cos \theta} \vec{p}$ e $\Delta \delta_{\varphi} \vec{p}$, approssimati con la formula di Taylor del primo ordine, dove \vec{p} è il centro del photon differential sulla superficie. Diversamente dall'implementazione originale di Schjøth *et*

al. [2], dove la sezione di area, A_{pd} , dell'ellissoide è un'ellisse contenuta idealmente all'interno di tale parallelogramma, nel metodo presentato in questa tesi, si assume che il parallelogramma sia esteso idealmente da $2\Delta\delta_{\cos\theta}\vec{p}$ e $2\Delta\delta_{\varphi}\vec{p}$ ed il footprint sia poi spostato, sempre idealmente, con un offset pari a $[-\Delta\delta_{\cos\theta}\vec{p}, -\Delta\delta_{\varphi}\vec{p}]^T$. I vettori differenziali posizionali sono interpretati come i semiassi principali dell'ellissoide (Figura 5.3(a)) e l'area A_{pd} è racchiusa all'interno del footprint espanso. In entrambi i casi, l'area dell'ellisse può essere approssimata da quella contenuta nel footprint del photon differential; nel primo caso tramite la formula $\frac{1}{4}\pi |\Delta\delta_{\cos\theta}\vec{p} \times \Delta\delta_{\varphi}\vec{p}|$ e, nel secondo caso (considerato in questa tesi), tramite la formula:

$$\pi |\Delta\delta_{\cos\theta}\vec{p} \times \Delta\delta_{\varphi}\vec{p}|. \quad (5.14)$$

`PD_compute_matrix(void)` calcola la matrice M_{pd} che trasforma l'ellissoide asimmetrico associato ad ogni photon differential in una sfera unitaria centrata all'origine. Nell'implementazione suggerita da Schjøth *et al.* [2], la dimensione e la forma delle regioni kernel di supporto sono definite dalla normale della superficie e dai vettori differenziali posizionali dei photon differentials. Nel metodo presentato, la regione kernel anisotropica allineata con i vettori differenziali posizionali è costruita a partire dai semiassi principali \vec{M} ed $\vec{m}1$ così definiti:

$$\begin{aligned} \vec{M} &= \max(\|s\Delta\delta_{\cos\theta}\vec{p}\|, \|s\Delta\delta_{\varphi}\vec{p}\|), \\ \vec{m}1 &= \min(\|s\Delta\delta_{\cos\theta}\vec{p}\|, \|s\Delta\delta_{\varphi}\vec{p}\|). \end{aligned}$$

Dove s è il fattore di scala manuale per lo smoothing. L'ellissoide asimmetrico centrato nel punto di intersezione \vec{z}_j è definito aggiungendo un altro semiasse principale, $\vec{m}2$, della stessa lunghezza di $\vec{m}1$ ed allineato alla normale \vec{n}_j della superficie (Figura 5.3(a)).

La regione kernel anisotropica di supporto centrata intorno a \vec{z}_j consente di valutare il contributo di illuminazione in qualsiasi punto \vec{x} trasformando la sezione contenente \vec{x} e \vec{z}_j , dal sistema di coordinate globale (o geometrico), al sistema di coordinate locale del kernel.

Nell'algoritmo originale del photon differentials [2], M_{pd} è la matrice di trasformazione che effettua il cambio di coordinate dal sistema geometrico a quello della regione kernel anisotropica definita usando la normale della superficie e la metà della lunghezza dei vettori differenziali posizionali. Nel metodo proposto in questa tesi, M_{pd} è la matrice di trasformazione dal sistema di coordinate geometriche al sistema di riferimento locale dell'ellissoide in cui \vec{M} , $\vec{m}1$ ed $\vec{m}2$ formano la base ortogonale locale.

Trasformando l'ellissoide asimmetrico in una sfera unitaria centrata all'origine degli assi, ogni punto \vec{x} può essere mappato in \vec{d} (Figura 5.3(b)). Assumendo $\vec{\delta} = \vec{x} - \vec{z}_j$, la trasformazione desiderata è $\vec{d} = M_{pd}\vec{\delta}$. La matrice inversa mappa i tre assi ortonormali nei tre semiassi principali dell'ellissoide asimmetrico, tale che $M_{pd}^{-1} = (\vec{M}, \vec{m}1, \vec{m}2)$; quindi, la matrice di trasformazione cercata è $M_{pd} = (\vec{M}, \vec{m}1, \vec{m}2)^{-1}$.

`PD_local_irradiance_estimate(void)` calcola la potenza radiante di un photon differential incidente nel punto \vec{x} . Al termine del photon tracing, quando tutti i photon differentials sono stati tracciati nella scena e proiettati sulle superfici, l'irradianza di ciascun fotone può essere precalcolata:

$$E_{pd} = \Phi_{pd}/A'_{pd}. \quad (5.15)$$

L'area iniziale cambia in relazione al percorso seguito dal fotone e viene proiettata di conseguenza sulla superficie dove il fotone incide. A'_{pd} indica il cambiamento dell'area del parallelogramma espanso dai vettori differenziali posizionali durante la propagazione nella scena, ovvero, il footprint adattato alle condizioni di illuminazione locali, $A_{pd} \rightarrow A'_{pd}$. Ad

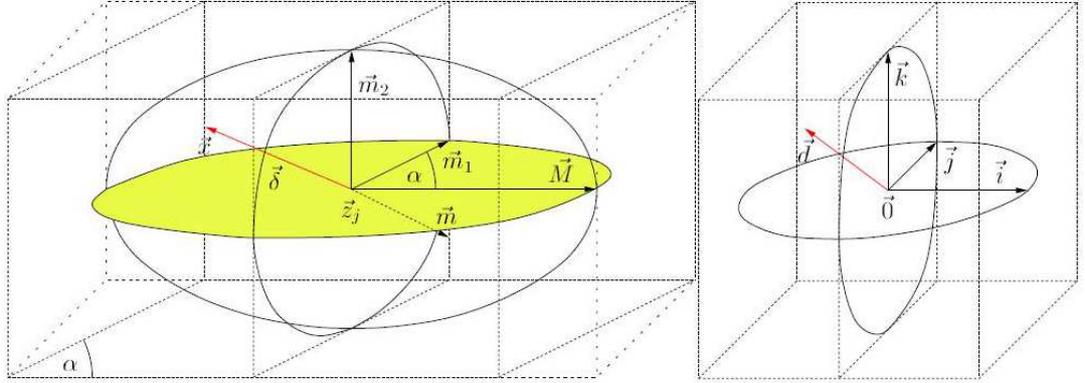


Figura 5.3: (a) Sezione di regione kernel approssimata dall'area del footprint del fotone (in giallo). (b) Regione di supporto dopo la trasformazione in una sfera unitaria.

esempio, la trasmissione attraverso lo spazio tende ad espandere il footprint, mentre, la riflessione speculare può comportare la compressione del footprint durante l'interazione con superfici convesse. L'irradianza incidente su una superficie diffusiva è calcolata come nell'Equazione 5.15.

`PD_irradiance_estimate(irrad[3], pos[3], normal[3], max_dist, filter)`

approssima il termine dell'equazione della radianza riflessa, L_r , all'interno dell'equazione di rendering (Equazione 1.12) aggiungendo, per ogni photon differential coinvolto nella stima, la potenza radiante incidente nel punto \vec{x} , precedentemente calcolata. Schjøth *et al.* [2] ridefiniscono la stima della radianza in modo che essa consideri le dimensioni e la forma dei footprint associati ai photon differentials considerati nella stima. La dimensione del footprint è usata per approssimare l'irradianza nel punto \vec{x} considerato; la forma del footprint è usata per definire una regione kernel anisotropica centrata intorno al punto \vec{x}_{pd} corrispondente alla posizione del photon differential memorizzato nella photon map. Isolando il secondo termine dell'Equazione 1.12, la radianza riflessa è data da:

$$L_r(\vec{x}, \vec{\omega}) = \int_{\Omega_x} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) \frac{d^2\Phi_i(\vec{x}, \vec{\omega}_i)}{dA_i}. \quad (5.16)$$

Nella stima della radianza standard (basata su fotoni regolari, a cui non è associata alcuna dimensione o forma) si richiede l'approssimazione dell'area proiettata dA_i attraverso l'area proiettata della sfera che comprende i k fotoni più vicini a \vec{x} . Nel photon differentials, l'area A_{pd} del footprint di un dato photon differential è l'approssimazione diretta dell'area proiettata influenzata dalla sua potenza radiante, Φ_{pd} .

Per queste ragioni sono state suggerite diverse soluzioni per l'Equazione 5.16. Jensen [1] definisce l'irradianza in un punto \vec{x} , come la potenza radiante incidente Φ_i per unità di area proiettata dA_i :

$$E(\vec{x}) = \frac{d\Phi_i(\vec{x})}{dA_i}. \quad (5.17)$$

Schjøth *et al.* [2] osservano che aggiungendo alla potenza radiante incidente Φ_i , una dipendenza dalla direzione $\vec{\omega}_i$, il risultato ottenuto è l'irradianza in \vec{x} , che associa alla potenza radiante incidente quella particolare direzione:

$$E(\vec{x}, \vec{\omega}_i) = \frac{d\Phi_i(\vec{x}, \vec{\omega}_i)}{dA_i}. \quad (5.18)$$

Questa irradianza è esattamente il risultato raggiunto tramite la tecnica del photon differentials, dove per $\vec{x} = \vec{x}_{pd}$, $\Phi_{pd} \approx d\Phi(\vec{x}_{pd}, \vec{\omega}_i)$ e $A_{pd} \approx dA_i$. Usando l'Equazione 5.15 è possibile approssimare il termine della radianza riflessa all'interno dell'equazione di rendering, con la seguente formula:

$$L_r(\vec{x}, \vec{\omega}) = \int_{\Omega_x} f_r(\vec{x}, \vec{\omega}_i, \vec{\omega}) dE(\vec{x}, \vec{\omega}_i). \quad (5.19)$$

L'integrale continuo nell'Equazione 5.19 può essere approssimato dalla seguente sommatoria:

$$\begin{aligned}
L_r(\vec{x}, \vec{\omega}) &= \sum_{pd=1}^n f_r(\vec{x}, \vec{\omega}_{pd}, \vec{\omega}) \Delta E(\vec{x}_{pd}, \vec{\omega}_{pd}) \\
&= \sum_{pd=1}^n f_r(\vec{x}, \vec{\omega}_{pd}, \vec{\omega}) \frac{\Phi_{pd}}{A_{pd}}, \tag{5.20}
\end{aligned}$$

dove n è il numero di photon differentials il cui footprint si sovrappone a \vec{x} . Ogni footprint che soddisfa il criterio di sovrapposizione col punto di intersezione è centrato idealmente intorno al punto di intersezione, con lo stesso effetto ottenibile se si assume che $\vec{x} = \vec{x}_{pd}$.

L'approssimazione dell'Equazione 5.20 diventa meno accurata quando \vec{x} ed \vec{x}_{pd} divergono, ovvero quando $\vec{x} \neq \vec{x}_{pd}$, ed in particolare all'aumentare della distanza tra i due punti. Analogamente a quanto osservato per la stima della radianza generale (Equazione 3.2) proposta da Jensen [1] nell'algoritmo del photon mapping standard, Schjøth *et al.* [2] pesano il contributo di ogni photon differential considerando la distanza di questi dal punto di intersezione \vec{x} . Tuttavia, invece di avvalersi solo della distanza $\|\vec{x}_{pd} - \vec{x}\|$, il calcolo del contributo di un photon differential si basa anche sulla forma del suo footprint.

Dato che i vettori differenziali posizionali del footprint rappresentano i due semiassi principali di un'ellissoide asimmetrico, essi definiscono un sistema di coordinate locale, detto *filter space*, rispetto alle coordinate globali (world space).

Applicando la matrice M_{pd} al vettore $\vec{\delta} = \vec{x} - \vec{x}_{pd}$, il risultato è un vettore definito in anisotropic filter space (regione anisotropica locale), $\vec{d} = M_{pd}\vec{\delta}$ (Figura 5.3(b)), la cui lunghezza è usata come input per una delle funzioni kernel riportate in Tabella 4.1. Per fornire il supporto al filtro anisotropico, l'Equazione 5.20 può essere riformulata nel seguente modo:

$$L_r(\vec{x}, \vec{\omega}) = \sum_{pd=1}^n f_r(\vec{x}, \vec{\omega}_{pd}, \vec{\omega}) K(\|\vec{d}\|) \frac{\Phi_{pd}}{A_{pd}}, \tag{5.21}$$

dove K è la funzione che utilizza il filtro kernel di Epanechnikov.

`PD_locate_photon_differentials(np, index, filter)` colleziona i photon differentials vicini al punto \vec{x} scartando i photon differentials che non influenzano la stima. Nell'algoritmo del photon mapping originale, tutti i fotoni all'interno di un raggio massimo predefinito devono essere presi inizialmente in considerazione; solo quando la ricerca dei primi k fotoni è giunta al termine, il raggio di ricerca può essere aggiornato e ridotto. Questo richiede la creazione ed il continuo aggiornamento di una struttura max-heap per ricercare, ed eventualmente sostituire, i fotoni al suo interno. Nell'algoritmo del photon differentials, la ricostruzione dell'illuminazione indiretta richiede che i photon differentials il cui footprint non si sovrappone al punto \vec{x} , e quelli che risultano al di fuori di un raggio fisso prescelto, non vengano accumulati. Nell'implementazione di Schjøth *et al.* [2] vengono presi in considerazione solo i photon differentials all'interno di un raggio fisso; tale raggio è limitato alla metà della lunghezza del vettore differenziale posizionale più lungo all'interno della photon map. Per garantire che non vengano generate regioni kernel di supporto troppo grandi è inoltre possibile limitare la lunghezza dei semiassi principali dell'ellissoide asimmetrico associato ai photon differentials, ad un valore h_{max} . Combinando queste due tecniche e limitando ad $h_{max} = 1$ la lunghezza massima dei vettori differenziali posizionali, il raggio fisso potrebbe essere impostato ad un valore medio di 0.5. Nel metodo proposto in questa tesi è stata adottata questa politica, senza però porre vincoli definitivi sulla scelta del raggio fisso utilizzabile nella stima della radianza.

5.2 Integrazione in XCRayt

La creazione ed il bilanciamento della mappa avvengono all'interno del programma principale, prima richiamando la procedura `photon_trace()` che emette i fotoni nella scena producendo la mappa relativa, e poi utilizzando il

metodo `balance()` della classe `Photon_map` che esegue il bilanciamento della mappa stessa.

All'interno di `photon_trace()` è la funzione `emit_photon` che traccia, a partire dalla sorgente luminosa, tanti raggi quanti sono i fotoni richiesti. Per decidere la direzione di emissione dei photon differentials si utilizza una tecnica denominata *explicit_sampling* (descritta nella sezione 5.1.2) che permette una distribuzione uniforme in tutte le direzioni.

Il percorso dei photon differentials viene poi seguito tramite la procedura `compute_photon()` che determina il comportamento ad ogni intersezione e, nel caso di trasmissione, riflessione (sia speculare che diffusiva) o rifrazione modella ricorsivamente il percorso dei nuovi raggi differenziali.

Nel determinare il percorso dei nuovi raggi differenziali `compute_photon()` segue un algoritmo modificato rispetto a quello del photon differentials originale [2]. Il metodo presentato è infatti in grado di calcolare i photon differentials a seguito dell'interazione con superfici diffuse. Nel raggiungere questo obiettivo si è esteso il funzionamento del photon differentials standard apportando un ulteriore contributo alla sintesi di immagini realistiche tramite XCRayt.

Tale estensione si è resa necessaria in quanto si vogliono considerare tutti i possibili effetti: rifrazione, riflessione speculare e riflessione diffusiva. Nell'ultimo caso (riflessione diffusiva) si vuole anche mantenere costante il numero di differenziali preservando l'adattamento delle condizioni di illuminazione nel punto di riemissione sulla superficie e durante la propagazione dei footprint attraverso la scena.

Una volta che la photon map è stata popolata e bilanciata, prima di passare alla fase di resa vera e propria, il controllo torna alla funzione `emit_photon` che esegue parte delle nuove procedure sui photon differentials memorizzati nella global photon map. In questo modo, parte delle informazioni necessarie nella fase di rendering risultano precalcolate e non necessitano di essere eseguite a run-time. Tale ottimizzazione consente di ridurre il costo computazionale e di migliorare le prestazioni nella delicata fase di rendering.

Un'altra modifica introdotta riguarda il calcolo del nuovo termine di il-

luminazione dovuto al photon differentials, termine che viene poi aggiunto a quelli già calcolati dall'algoritmo standard del ray tracing già implementato nel sistema. Questo avviene all'interno del metodo `compute_ray()` che ha il compito di restituire i valori di illuminazione del pixel attraverso cui passa il raggio tracciato.

All'interno di questa procedura, dopo aver sommato i contributi dovuti ad ambiente, riflessione speculare ed eventuale trasmissione, si calcola la stima della radianza per il punto considerato usando la photon map e si aggiunge anche tale termine all'illuminazione del pixel corrispondente. Il calcolo del nuovo contributo viene eseguito dal metodo `PD_irradiance_estimate()` della classe `Photon_map` estesa. Tale calcolo segue quanto espresso dalla formula dell'Equazione 5.20 o, se il filtro anisotropico risulta abilitato, dell'Equazione 5.21. Infine l'algoritmo procede come nel caso standard con eventuali chiamate ricorsive.

5.2.1 Parametri di configurazione

L'implementazione del nuovo algoritmo all'interno di XCRayt ha comportato l'introduzione di tre nuovi parametri di configurazione. Per il numero di photon differentials da emettere per ogni sorgente luminosa, così come per il numero massimo di ricorsioni e per il numero massimo di memorizzazioni (questi ultimi rappresentanti, rispettivamente, la quantità di percorsi che è possibile effettuare nel seguire i photon differentials e la quantità di photon differentials memorizzabili nella mappa) si è usufruito dei valori associati alle impostazioni del photon mapping standard. Per il parametro relativo al raggio massimo dell'area di stima, ora interpretato come raggio fisso, il valore deve essere ridefinito allo scopo di impostare la distanza massima di ricerca dei photon differentials all'interno della photon map. Il parametro relativo al fattore lineare per lo smoothing ed al controllo della dimensione iniziale dei footprint è stato invece introdotto appositamente per selezionare la bandwidth da usare con il nuovo algoritmo. Oltre a questi parametri è indispensabile fornire al sistema un'indicazione riguardo la volontà di usufruire del photon map-

ping nella resa dell'immagine; questo deve precedere l'attivazione del photon differentials ed, eventualmente, quella del filtro anisotropico. Questi parametri devono essere definiti all'interno del file di configurazione `< scene > .arg` secondo la seguente sintassi:

- `--d` per indicare l'utilizzo del photon differentials esteso alle riflessioni diffuse;
- `--k` per indicare l'utilizzo del filtro kernel anisotropico;
- `--f<num smooth factor>` per controllare il trade-off tra varianza e bias.

Per poter inserire i nuovi parametri nel sistema si è intervenuto sulla routine `load_param()` che effettua la lettura e la memorizzazione di tali parametri all'interno di XCRayt. I parametri aggiuntivi estendono una sottostruttura di tipo `Photon_par_t`, chiamata `photon_par`, che rappresenta un campo della struttura globale di tipo `Param_t`, la quale mantiene nel sistema tutti i parametri di configurazione.

5.2.2 Statistiche

La routine `print_photon_mapping_stat(fp)` nel modulo "*statistics.c*" è stata estesa per restituire le statistiche di esecuzione del nuovo algoritmo. Nel caso di resa tramite il photon differentials esteso alle riflessioni diffuse, i valori aggiuntivi sono scritti nel file `< scene > .sta`, all'interno di una nuova sezione creata ad-hoc. Le informazioni rilevanti per avere un riscontro quantitativo dello sforzo computazionale richiesto dal nuovo algoritmo riguardano:

- il numero di photon differentials emessi per sorgente di luce;
- il numero totale di photon differentials emessi all'interno della scena;
- il numero totale di photon differentials memorizzati nella photon map;
- il numero di photon differentials utilizzati nella stima della radianza;

- il coefficiente lineare per il controllo del trade off tra varianza e bias;
- il raggio fisso dell'area utilizzata per la ricerca dei photon differentials nella photon map;
- il tempo occorso per terminare la propagazione dei photon differentials nella scena.

5.3 Riepilogo

Questo capitolo descrive l'implementazione e illustra, contestualmente, le modifiche apportate all'algoritmo del photon mapping standard di XCRayt introdotte allo scopo di migliorare, tramite una variante dell'algoritmo originale del photon differentials [2], la simulazione dei possibili fenomeni luminosi, come le caustiche e l'illuminazione indiretta.

Il metodo proposto permette di estendere il suo funzionamento anche alla gestione delle superfici diffusive. L'idea di base consiste nel calcolare, ad ogni interazione con una superficie diffusiva, un offset v tramite il prodotto vettoriale dei differenziali posizionali associati al photon differential. Il photon differential è assorbito e riemesso con una nuova direzione scelta utilizzando l'Equazione 5.1 per il campionamento dei nuovi parametri. In questa procedura, il numero dei differenziali è mantenuto costante (così come ogni scala ed adattamento precedenti) e le condizioni di illuminazione nel punto di riemissione sono preservate. A seguito dello spostamento del punto di riemissione, l'area di footprint continua ad essere adattata nel corso della propagazione.

Nell'attuazione pratica di questa e di altre procedure relative al nuovo algoritmo di illuminazione globale si è cercato di realizzare modifiche che risultassero il meno invasive possibili sul codice del sistema già esistente. Alcune parti di questa estensione sono analoghe al photon mapping; in questo capitolo si omette il codice simile nelle due implementazioni e si concentra la descrizione sulle differenze rilevanti tra il metodo presentato e l'algoritmo originale.

Capitolo 6

Risultati sperimentali

Lo scopo di questo capitolo è quello di comparare il contributo del photon differentials e quello del photon mapping standard nella ricostruzione dell'illuminazione indiretta. Il numero di comparazioni è stato determinato gradualmente, procedendo per stadi, fino ad arrivare alle condizioni limite di utilizzo della stima della radianza standard in una scena di riferimento.

La verifica del lavoro svolto richiede la definizione delle misure in grado di identificare, in termini non soggettivi, ciò che costituisce un più accurato - *migliore* - risultato. L'accuratezza della stima della radianza standard aumenta con il numero di fotoni coinvolti per ogni stima; con questo presupposto, il miglior risultato può essere definito come un'immagine che sia l'approssimazione diretta dell'immagine prodotta attraverso la stima della radianza standard effettuata con un elevato numero di fotoni per stima. Questo approccio richiede scene di riferimento che mostrino con estrema chiarezza la discrepanza tra le immagini prodotte usando la stima della radianza standard con pochi fotoni, e quelle prodotte aumentando il numero di fotoni, rispettivamente per ogni stima.

Diversamente dal metodo di Schjøth *et al.* [2], dove i photon differentials sono memorizzati in due distinte photon maps, l'estensione realizzata include la possibilità di gestire le riflessioni diffuse; questo permette di applicare la stima della radianza del nuovo algoritmo e del photon mapping standard,

con la stessa quantità di photon differentials e fotoni per ogni stima e con gli stessi parametri per la photon map. Invece di memorizzare nella caustic photon map solo i photon differentials che sono stati riflessi specularmente o trasmessi, ed infine assorbiti da una superficie diffusiva, il metodo proposto in questa tesi non pone alcun vincolo sulle interazioni di un photon differential durante la fase del photon tracing permettendo la memorizzazione di questo nella global photon map. Il risultato è un'immagine in cui l'illuminazione appare omogenea su tutte le superfici della scena.

Tuttavia, dal momento che le caustiche tendono a delineare contorni netti e molto ben definiti, solo i photon differentials coinvolti nella loro generazione contribuiscono ad aumentare il contrasto tra le superfici illuminate e quelle non illuminate nell'immagine finale. Questo comportamento accentua le differenze tra i due metodi ed è funzionale per i test. Per queste ragioni si è scelto di basare la sperimentazione su scene test caratterizzate dalle sole caustiche.

Le prestazioni dei due metodi sono confrontate paragonando i tempi di resa di alcune semplici scene di test. Tali scene sono renderizzate con una risoluzione di 400 x 400 pixel usando il ray tracing per l'illuminazione diretta ed il photon mapping (in configurazione standard ed estesa) per l'illuminazione indiretta.

6.1 Test della scena

Nel ricostruire l'illuminazione reale, f , tramite la stima delle densità kernel, \hat{f} , il photon mapping standard introduce un errore casuale; la stima della radianza converge a f , solo quando la varianza decresce ed il valore della bandwidth tende a zero. L'aumento del numero di fotoni ed il calcolo della bandwidth più appropriata costituiscono i parametri su cui poter intervenire per controllare il trade-off tra varianza e bias.

Nella stima della densità basata sulla ricerca k nearest neighbor, la bandwidth è variabile; essa è determinata automaticamente in base al numero di fotoni usati per ogni stima della radianza. L'aumento del numero totale di

fotoni comporta una diminuzione della bandwidth ed un progressivo miglioramento nell'illuminazione della scena. Analogamente, la stima della radianza del nuovo algoritmo converge alla funzione reale (f) se all'aumento del numero totale di fotoni corrisponde una diminuzione della lunghezza iniziale dei vettori differenziali. Un fattore di smooth s risulta essenziale per bilanciare al meglio varianza e bias.

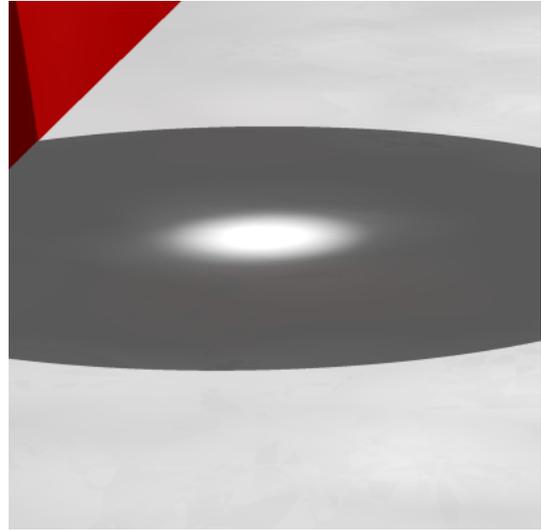
Nelle seguenti comparazioni si evidenzia come il metodo proposto converga a f molto più rapidamente rispetto al photon mapping standard. La stessa scena di riferimento è stata resa usando la stima della densità k nearest neighbor tradizionale, e la stima della densità locale con il supporto della regione kernel anisotropica illustrata in Figura 5.3. Le immagini nella prima colonna sono prodotte col photon mapping, mentre la seconda colonna include le immagini realizzate tramite l'estensione del photon differentials integrata nel nuovo algoritmo. Quest'ultima impiega anche un filtro per pesare il contributo dei photon differentials mediante la funzione di Epanechnikov. Ogni riga è basata su un differente numero di fotoni coinvolti nella stima della radianza. Tutte le rese sono state effettuate aumentando gradualmente il numero di fotoni e di photon differentials per ogni stima, fino ad arrivare a definire la bandwidth appropriata, in modo da ridurre il rumore senza introdurre un effetto blurring non necessario.

La caustica generata dalla rifrazione della luce attraverso una sfera trasparente consente di esplorare le capacità del nuovo algoritmo nel riprodurre fedelmente l'illuminazione indiretta, nella simulazione avanzata delle caustiche. Questo rende tale scena particolarmente indicata come modello di riferimento per l'esecuzione dei test. Una serie di rese della scena di riferimento è mostrata in Figura 6.1, dove i fotoni che attraversano la sfera sono concentrati sull'area della superficie piana sottostante. La telecamera virtuale è stata posizionata in modo da isolare e catturare la sola caustica; i photon differentials sono emessi da una sorgente di luce isotropica supportando un massimo di *tre* interazioni per ogni percorso. Tutti i campionamenti sono controllati da numeri casuali pregenerati. Per evitare un decadimento generale delle prestazioni, il

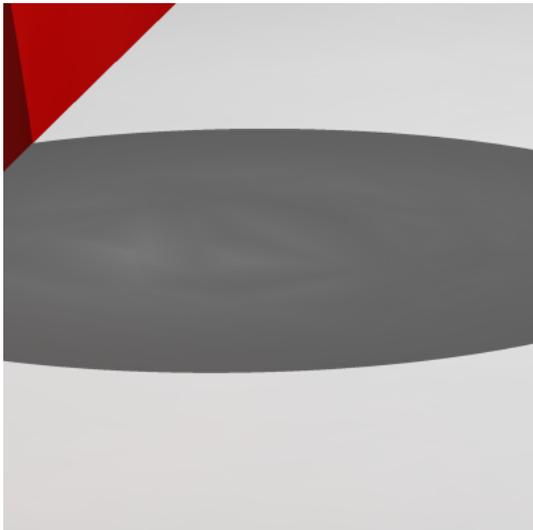
raggio per la ricerca k nearest neighbor è limitato a 1.0.



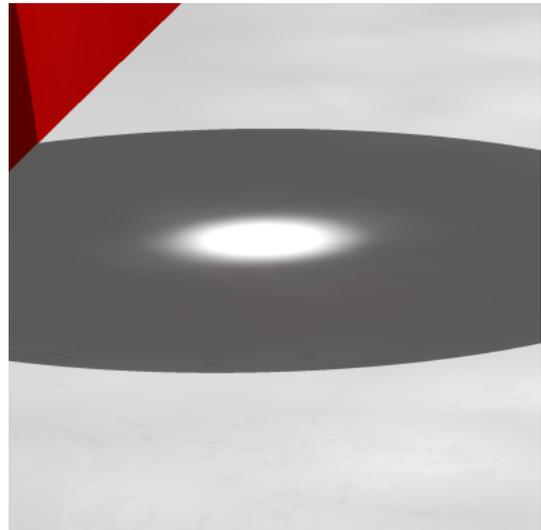
(a) 30 fotoni per stima.



(b) 30 photon differentials per stima e fattore di smooth 1.



(c) 40 fotoni per stima.



(d) 40 photon differentials per stima e fattore di smooth 1.

Figura 6.1: Scena test resa con 5000 fotoni.

Nella Figura 6.1, tutte le immagini sono state prodotte emanando 5000 fotoni totali; la prima riga include le immagini prodotte usando 30 fotoni per stima, mentre la seconda riga si riferisce ad una stima eseguita con 40

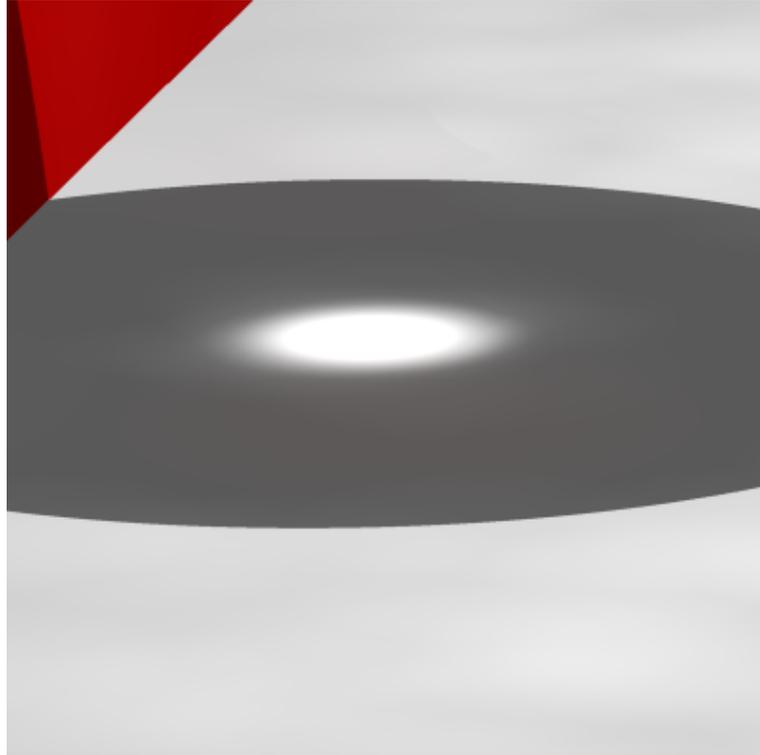
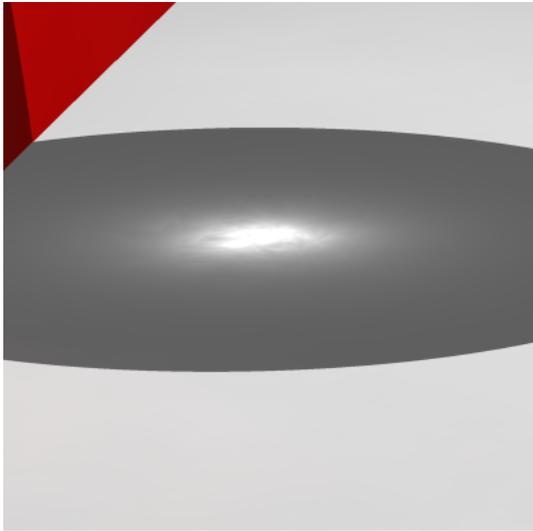


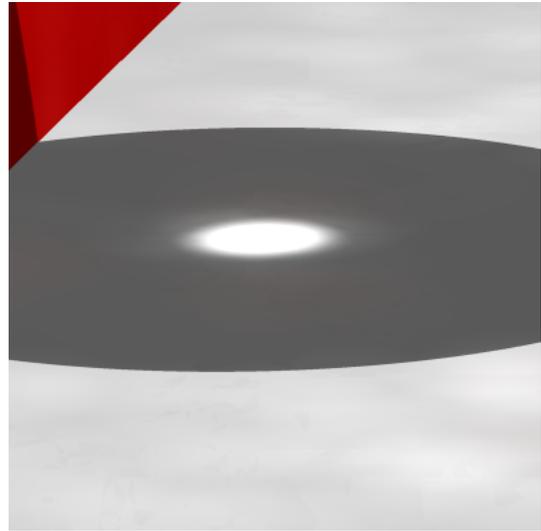
Figura 6.2: Scena test resa con 5000 fotoni, 80 photon differentials per stima e fattore di smooth 1.

fotoni. Nella Figura 6.2, per meglio evidenziare il contributo dato dal photon differentials è stata effettuata un'ulteriore resa della scena utilizzando 80 fotoni nella stima della radianza. Nella Figura 6.1(a), a causa del basso numero di fotoni e del particolare algoritmo adottato (il photon mapping standard), i contorni della caustica appaiono particolarmente soggetti alla produzione di artefatti, fino ad essere pesantemente sottostimati dalla stima della densità k nearest neighbor (Figura 6.1(c)). Nelle Figure 6.1(b) e 6.1(d), il contributo aggiuntivo dei photon differentials dovuto alla rifrazione dei raggi attraverso la sfera si delinea sempre meglio verso un unico fuoco, fino a configurarsi come una concentrazione di luce in cui l'intensità luminosa risulta ben definita.

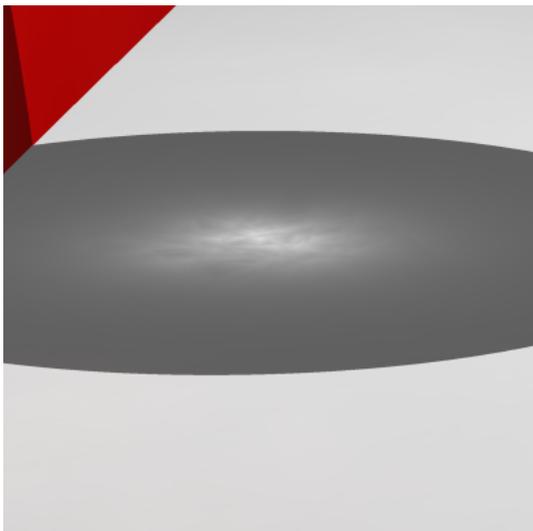
La scena della Figura 6.3 è stata resa con l'algoritmo del photon differentials usando 10000 fotoni. Nelle Figure 6.3(b) e 6.3(d), relative alla stima della radianza effettuata, rispettivamente, con 40 e 50 photon differentials,



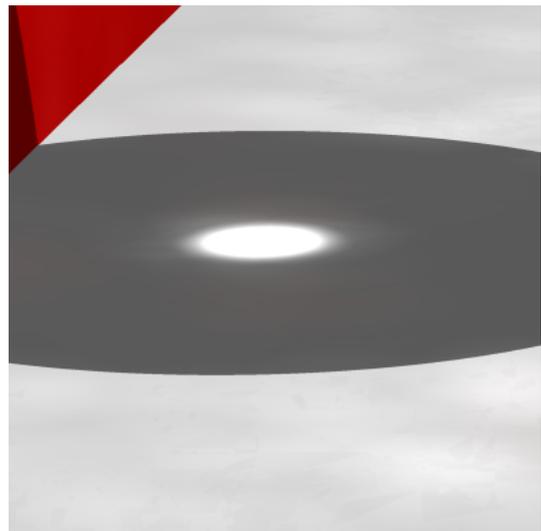
(a) 40 fotoni per stima.



(b) 40 photon differentials per stima e fattore di smooth 1.



(c) 50 fotoni per stima.



(d) 50 photon differentials per stima e fattore di smooth 1.

Figura 6.3: Scena test resa con 10000 fotoni.

la caustica appare evidente e con contorni ben definiti. Dalle corrispondenti rese effettuate con l'algoritmo originale del photon mapping (Figure 6.3(a) e 6.3(c)), i problemi riscontrati sono analoghi a quanto osservato precedentemente, con l'eccezione che, nella prima immagine, gli artefatti sono meno

evidenti.

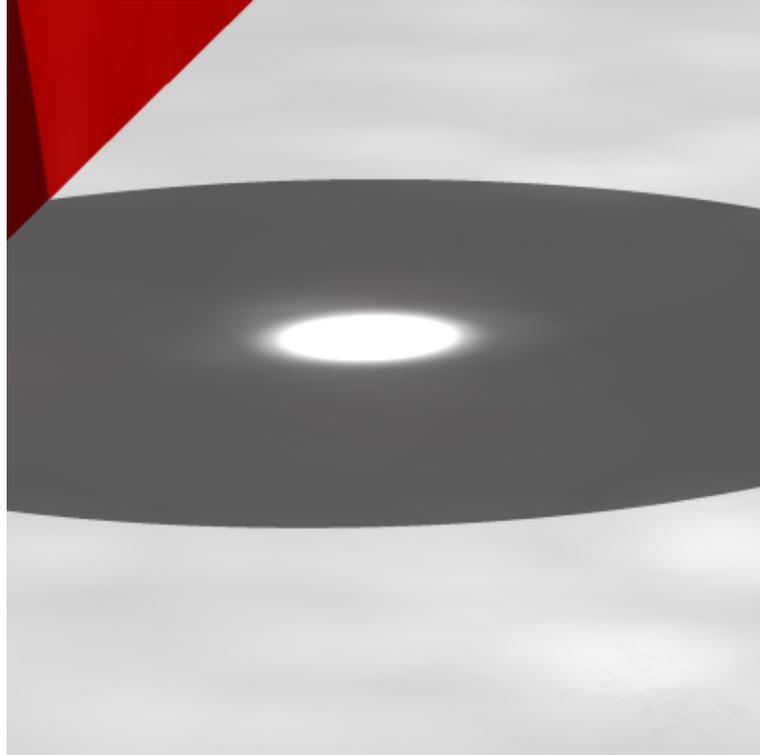


Figura 6.4: Scena test resa con 10000 fotoni, 100 photon differentials per stima e fattore di smooth 1.

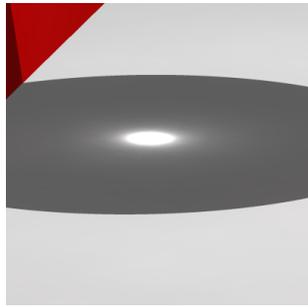
Diversamente da quanto si evince dalle sequenze di Figura 6.1 e di Figura 6.3, dove la differenza tra i due algoritmi è notevole, nella sequenza di immagini di Figura 6.5 appare evidente come la qualità migliori sensibilmente (per entrambi i metodi) aumentando il numero di fotoni inclusi nella photon map. Le caustiche risultano maggiormente definite, con contorni più nitidi e meno mossi, in configurazioni con 100, 200 e 400 fotoni per stima. In contrasto con il nuovo algoritmo, il photon mapping standard tende a produrre un effetto di allungamento dei contorni; il photon differentials riproduce distintamente la caustica ma l'immagine risultante presenta una varianza indotta dall'elevato numero di photon differentials da considerare in ogni punto di intersezione (Figura 6.5(b)). Il rumore nelle immagini è ridotto controllando manualmente la bandwidth. Nel photon differentials, lo smoothing consiste nel regolare

la dimensione iniziale del footprint. Il fattore lineare scala le dimensioni dei semiassi principali dell'ellissoide, per limitare il rumore introducendo uno sfocamento nell'illuminazione. Come illustrato nelle Figure 6.5(d) e 6.5(f), il rumore è stato attenuato sensibilmente tramite una minima variazione del fattore di smooth da 1 a 2.

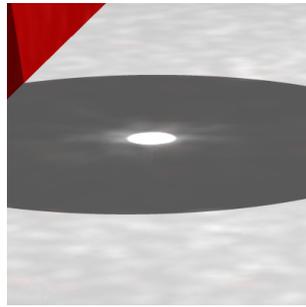
La Figura 6.6 mostra una serie di immagini rese con il nuovo algoritmo, in cui il fattore di smooth è applicato per ridurre la varianza indotta da un'alta concentrazione di fotoni; piccole variazioni di scala sono in grado di eliminare in modo apprezzabile la percezione del rumore. A partire dall'immagine di Figura 6.6(a), la varianza decresce man mano che si aumentano le dimensioni iniziali del footprint; questo corrisponde ad una espansione della regione kernel anisotropica di supporto. A dispetto di una graduale introduzione, seppur minima, dell'errore di bias (sottoforma di sfocamento), la qualità dell'immagine prodotta con il photon differentials (Figura 6.7) supera quella delle immagini prodotte con il photon mapping standard, anche con 800000 fotoni nella photon map e ben 400 fotoni coinvolti nella stima della radianza.

Figura 6.2 evidenzia come la struttura della caustica sia chiaramente identificabile anche quando il numero di photon differentials nella photon map è ridotto al minimo e la stima della radianza è effettuata con pochi campioni. L'immagine prodotta con il metodo standard del photon mapping (Figure 6.1(a) e 6.3(c)) mostrano evidenti artefatti. Nella prima immagine in Figura 6.3 si osserva una deformazione dei contorni della caustica; questo comportamento sottolinea la necessità (da parte del photon mapping) di ricorrere ad un più alto numero di fotoni totali, o di considerare metodi alternativi, come il photon differentials.

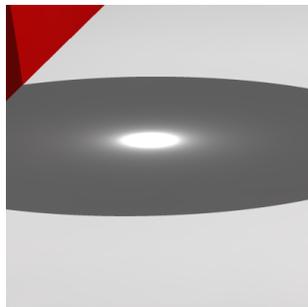
Nella comparazione delle immagini prodotte usando il nuovo algoritmo appare evidente come esso produca contorni smussati ed un minor numero di artefatti confermando, inoltre, l'efficacia della scena di riferimento impiegata per svolgere l'analisi comparativa. Mettendo a confronto le immagini prodotte usando l'estensione del photon mapping, con le immagini prodotte in configurazione standard, queste ultime sono caratterizzate da una minore de-



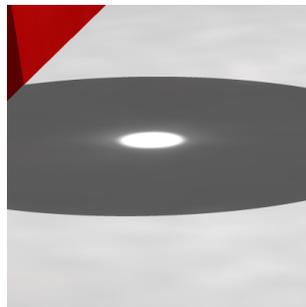
(a) 100 fotoni per stima.



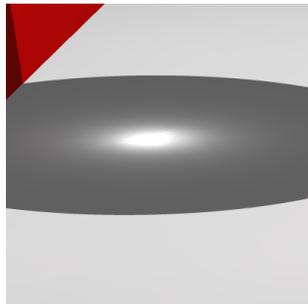
(b) 100 photon differentials per stima e fattore di smooth 1.



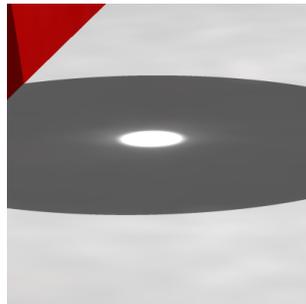
(c) 200 fotoni per stima.



(d) 200 photon differentials per stima e fattore di smooth 2.



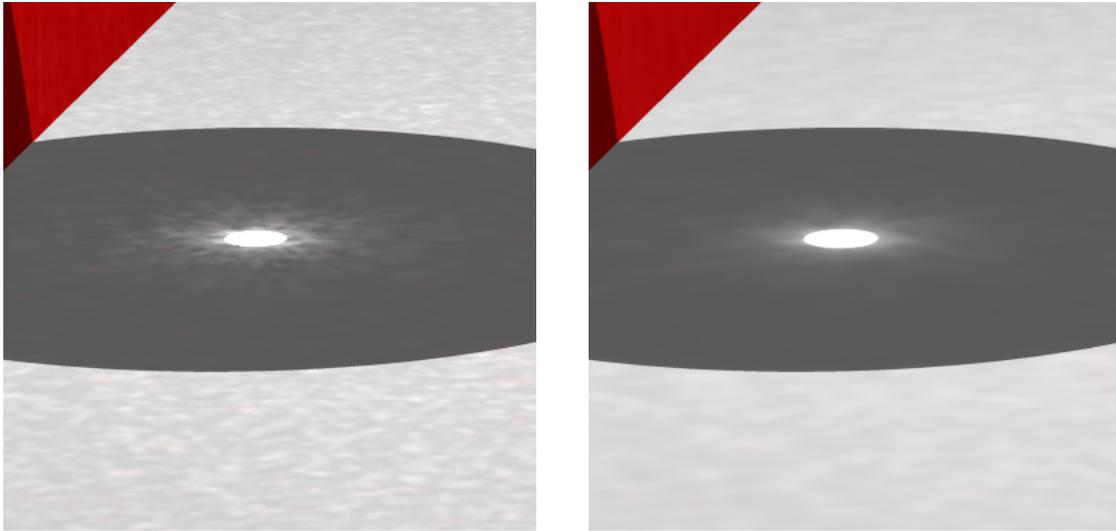
(e) 400 fotoni per stima.



(f) 400 photon differentials per stima e fattore di smooth 2.

Figura 6.5: Scena test resa con 100000 fotoni.

finizione dei contorni. La costruzione di una regione kernel anisotropica aiuta a delineare meglio i confini della caustica, d'altro canto, la stima della densità



(a) 400 photon differentials per stima e fattore di smooth 1.

(b) 400 photon differentials per stima e fattore di smooth 2.

Figura 6.6: Scena test resa con 800000 fotoni e 400 photon differentials per stima.

k nearest neighbor riduce il rischio di ottenere valori anomali (eccessivamente grandi o piccoli) associati al kernel bandwidth, tuttavia le regioni kernel di supporto sono sferiche e quindi non adatte a definire la forma ed i contorni dell'illuminazione. Il photon differentials permette un adattamento efficace della stima della densità locale; effetti come le caustiche sono riprodotti con un'illuminazione diffusa in modo più regolare, in particolare lungo i bordi. L'impatto visivo e la qualità finale dell'immagine migliorano ulteriormente con l'uso del filtro kernel di Epanechnikov per lo smoothing, al posto del semplice kernel uniforme. Il photon differentials apporta un miglioramento nella struttura dell'illuminazione che implica, in termini di resa delle caustiche, un *migliore* risultato rispetto al photon mapping. In altre parole, l'estensione realizzata supera abbondantemente tutti i test effettuati in questa sperimentazione.

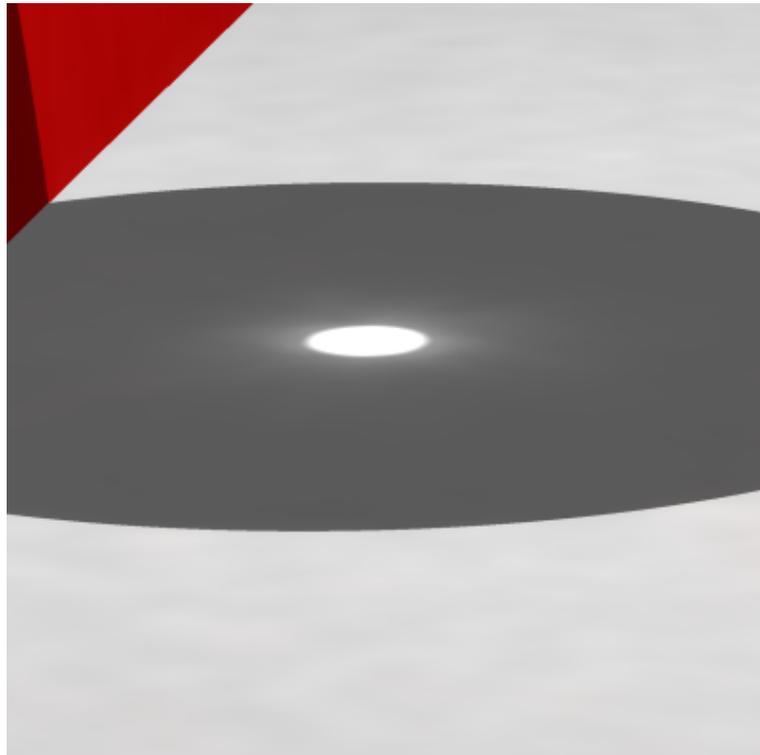


Figura 6.7: Scena test resa con 800000 fotoni, 400 photon differentials per stima e fattore di smooth 3.

Conclusioni

In questa tesi è stata realizzata un'estensione dell'algoritmo del photon mapping, in grado di calcolare l'illuminazione indiretta diffusa e speculare. Ritenendo essenziale introdurre i principi fisico-matematici alla base del trasporto della luce, nel capitolo introduttivo si è investigato il campionamento alla base di alcune semplici sorgenti luminose. Nel secondo capitolo, la descrizione delle tecniche e degli algoritmi di illuminazione globale si concentra sul particle tracing, con particolare riferimento al photon mapping, di cui si sono illustrate entrambe le fasi del processo di resa. Il terzo capitolo descrive i concetti e la teoria della stima della densità nell'ambito della computer grafica. Nel quarto capitolo si spiega la tecnica del ray differentials esplorando i differenti approcci per incorporare il footprint nella stima della radianza definita da Schjøth *et al.* [2] nel photon differentials. Nel quinto capitolo si è discussa l'implementazione all'interno del framework di XCRayt, il quale incorpora l'algoritmo del photon mapping originale. Utilizzando XCRayt, nel sesto capitolo si sono applicati i due metodi su una scena di riferimento comune e poi comparati i risultati.

Da questa disamina si rileva come l'utilizzo di una mappa di photon differentials risulti particolarmente conveniente nella simulazione realistica delle caustiche e, grazie all'estensione apportata nell'implementazione dell'algoritmo originale del photon differentials, in scene dove gli effetti di illuminazione indiretta dovuti alle riflessioni diffuse risultano più importanti. Una delle modifiche più significative che ha permesso il raggiungimento di questo obiettivo è stata quella dell'utilizzo di un differente tipo di kernel nel processo della

stima della densità. Diversamente dal photon mapping standard, che impiega un kernel costante in cui assegna a tutti i fotoni un egual peso (in termini di contributo di potenza irradiante), il nuovo algoritmo si avvale di un kernel variabile, che conferisce, rispetto al punto osservato, un peso maggiore ai fotoni vicini, ed un peso minore ai fotoni più lontani.

Il metodo proposto in questa tesi può essere ulteriormente esteso ed ottimizzato. Tra i possibili miglioramenti vi è senz'altro quello della possibile implementazione di una struttura dati gerarchica. Questa struttura potrebbe essere impiegata con successo allo scopo di accelerare la ricerca dei fotoni rilevanti durante le stime della radianza dell'algoritmo del photon differentials. Dato che a ciascun photon differential è associato un footprint ellissoidale, il cui volume costituisce la primitiva fondamentale per la costruzione di un BVH (acronimo di Bounding Volume Hierarchy), la scelta più logica e naturale per ottimizzare le richieste alla photon map sarebbe quella di sostituire il kd-tree con tale struttura.

Bibliografia

- [1] H.W. Jensen, *Realistic Image Synthesis Using Photon Mapping*. AK Peters, 2001.
- [2] L. Schjøth, J.R. Frisvad, K. Erleben, e J. Sporring, *Photon differentials*. In GRAPHITE 2007, pp. 179-186, 2007.
- [3] M. Born e E. Wolf, *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, settima edizione, 1999.
- [4] P. Dutré, K. Bala, e P. Bekaert, *Advanced Global Illumination*. AK Peters, seconda edizione, 2006.
- [5] W.R. McCluney, *Introduction to Radiometry and Photometry*. Artech House, 1994.
- [6] J.T. Kajiya, *The rendering equation*. ACM SIGGRAPH 1986: Proceedings of the 13th annual conference on Computer graphics and interactive techniques, pp. 143-150, ACM Press, (New York, NY, USA), 1986.
- [7] F.E. Nicodemus, *Directional reflectance and emissivity of an opaque surface*. Applied Optics 4(7), pp. 767-775, 1965.
- [8] H. von Helmholtz, *Handbuch der Physiologischen Optik*. Allgemeine Encyklopädie der Physik (9), Leopold Voss, 1867.
- [9] E. Veach, *Non-symmetric scattering in light transport algorithms*. EGWR 1996, pp. 82-91, 1996.

- [10] C.M. Goral, K.E. Torrance, D. P. Greenberg, e B. Battaile, *Modeling the interaction of light between diffuse surfaces*. SIGGRAPH 1984: Proceedings of the 11th annual conference on Computer graphics and interactive techniques, pp. 213-222, ACM Press, (New York, NY, USA), 1984.
- [11] D. S. Kay e D. Greenberg, *Transparency for computer synthesized images*. SIGGRAPH 1979: Proceedings of the 6th annual conference on Computer graphics and interactive techniques, pp. 158-164, ACM Press, (New York, NY, USA), 1979.
- [12] T. Whitted, *An improved illumination model for shaded display*. Commun. ACM 23(6), pp. 343-349, 1980.
- [13] E.P. Lafortune e Y.D. Willems, *Bi-directional Path Tracing*. Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Compugraphics 1993), H. P. Santo, ed., pp. 145-153, (Alvor, Portugal), 1993.
- [14] A. Appel, *Some techniques for shading machine renderings of solids*. AFIPS Spring Joint Computer Conference 1968, pp. 37-45, 1968.
- [15] P. S. Heckbert e P. Hanrahan, *Beam tracing polygonal objects*. 1984.
- [16] H. Igehy, *Tracing ray differentials*. ACM SIGGRAPH 1999, pp. 179-186, 1999.
- [17] L. Neumann, *Monte carlo radiosity*. Computing 55(1), pp. 23-42, 1995.
- [18] R.L. Cook, T. Porter e L. Carpenter, *Distributed ray tracing*. In ACM SIGGRAPH 1984, pp. 137-145, 1984.
- [19] J. Arvo, *Backward ray tracing*. ACM SIGGRAPH 1986, Courses, pp. 259-263, 1986.
- [20] B. Silverman, *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability, Chapman and Hall, London-New York, 1986.

- [21] L. Schjøth, O.F. Olsen e J. Sporring, *Diffusion based photon mapping*. International conference on Computer Graphics - Theory and Applications, INSTICC Press, Setùbal, Portugal, pp. 168-175, 2006.
- [22] F. Suykens e Y.D. Willems, *Path differentials and applications*. Proceedings of the 12th Eurographics Workshop on Rendering, Springer-Verlag, London, UK, pp. 257-268, 2001.
- [23] J. Sporring, L. Schjøth e K. Erleben, *Spatial and temporal ray differentials*. Technical report, Dept. of Computer Science, University of Copenhagen, 2009.